UNIVAC®

1108

MULTI-PROCESSOR SYSTEM

OPERATING SYSTEM

EXEC 8

PROGRAMMERS REFERENCE

A

UP-4144 Rev. 1

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

PRINTED IN U.S.A.

TABLE OF CONTENTS

INTRODUCTION

THIS DOCUMENT CONTAINS A DESCRIPTION OF THE EXECUTIVE COMPONENT OF THE OPERATING SYSTEM FOR THE UNIVAC 1108 COMPUTER AND TAKES THE FORM OF A PROGRAMMERS REFERENCE MANUAL (PRM). THE EXECUTIVE IS DESIGNED TO OPERATE AS A MASTER CONTROL PROGRAM WHICH ESTABLISHES THE EFFICIENT MULTI-PROGRAMMING ENVIRONMENT NEEDED FOR UTILIZING THE FULL CAPABILITIES OF THE UNIVAC 1108 MULTI-PROCESSOR SYSTEM. THIS MANUAL INCLUDES DETAILS OF UTILIZATION PROCEDURES AND FUNCTIONAL CAPABILITIES BUT DOES NOT IN ALL CASES PRESENT THE DETAILED PROGRAMMING LOGIC WHICH MAKES POSSIBLE THOSE PROCEDURES AND CAPABILITIES.

THE VERSION OF THE EXECUTIVE SYSTEM DESCRIBED IS FOR THE UNIT PROCESSOR CONFIGURATION. THE INTERFACE OF THE USER TO THE EXEC IS THAT SHOWN. THE SAME INTERFACE WILL BE USED IN THE MULTI-PROCESSING VERSION WITH ADDITIONS MADE FOR CONTROL AND USE OF NEW FEATURES AVAILABLE ONLY IN THE MULTI-PROCESSOR CONFIGURATION. ALL SYSTEM SHIPMENTS WILL INCLUDE A 'SYSTEM MEMORANDUM' WHICH DETAILS THE CHARACTERISTICS OF THE CURRENT SYSTEM. THE MEMOS WILL POINT OUT THE VARIOUS ADDITIONS OR ENHANCEMENTS MADE AS THE SYSTEM IS UPDATED.

INCLUDED IN THIS MANUAL ARE SECTIONS ESTABLISHING THE OPERATIONAL REQUIREMENT FOR THE EXECUTIVE SYSTEM, EXPLAINING CERTAIN BASIC CONCEPTS NECESSARY FOR EXECUTIVE SYSTEM OPERATIONS--IN THE REAL-TIME, BATCH AND DEMAND (OR 'CONVERSATIONAL') MODES, AND OUTLINING THE ORGANIZATION OF THE SYSTEM. THESE GENERALIZED INTRODUCTORY SECTIONS ARE FOLLOWED BY MORE DETAILED EXPLANATIONS OF THE EXECUTIVE CONTROL LANGUAGE, BATCH-PROCESSING OPERATIONS, DEMAND-PROCESSING OPERATIONS, REAL-TIME PROCESSING OPERATIONS, AND THE PROCESSING SUPERVISOR. AFTER THIS CLARIFICATION OF THE MAJOR SYSTEM COMPONENTS AND OPERATIONS, SECTIONS ARE PRESENTED ON OTHER COMPONENTS OF THE 1108 EXECUTIVE SYSTEM.

1. EXECUTIVE SYSTEM DESIGN CRITERIA

1.1. OPERATIONAL CAPABILITIES

TO TAKE MAXIMUM ADVANTAGE OF THE SPEED AND HARDWARE CAPABILITIES OF THE UNIVAC 1108 COMPUTER AND TO MAKE EFFECTIVE USE OF A GIVEN HARDWARE CONFIGURATION, A COMPLEX INTERNAL OPERATING ENVIRONMENT HAS BEEN CREATED.

THIS ENVIRONMENT ALLOWS FOR THE CONCURRENT OPERATION OF MANY PROGRAMS; IT ALLOWS THE SYSTEM TO REACT IMMEDIATELY TO THE INQUIRIES, REQUESTS, AND DEMANDS OF MANY DIFFERENT USERS AT LOCAL AND REMOTE STATIONS; IT ALLOWS FOR THE STRINGENT DEMANDS OF REAL-TIME APPLICATIONS; IT IS ABLE TO STORE, FILE, RETRIEVE AND PROTECT LARGE BLOCKS OF DATA; AND IT MAKES OPTIMUM USE OF ALL AVAILABLE HARDWARE FACILITIES, WHILE MINIMIZING JOB TURNAROUND TIME.

ONLY THROUGH CENTRAL CONTROL OF ALL ACTIVITIES OF THE UNIVAC 1108 CAN THIS ENVIRONMENT OF THE COMBINED HARDWARE AND SOFTWARE SYSTEMS BE FULLY ESTABLISHED AND MAINTAINED TO SATISFY THE REQUIREMENTS OF ALL APPLICATIONS. THIS RESPONSIBILITY FOR EFFICIENT, FLEXIBLE, CENTRALIZED CONTROL IS BORNE BY THE EXEC. THE EXEC CONTROLS AND COORDINATES THE FUNCTIONS OF THIS COMPLEX INTERNAL ENVIRONMENT AND, BY PRESENTING A RELATIVELY SIMPLE INTERFACE TO THE PROGRAMMER, ALLOWS HIM TO USE THE SYSTEM EASILY WHILE RELIEVING HIM OF CONCERN FOR THE INTERNAL INTERACTION BETWEEN HIS PROGRAM AND OTHER CO-EXISTENT PROGRAMS.

1.2. EXEC RELATION TO OTHER SYSTEM COMPONENTS

THE UNIVAC 1108 EXECUTIVE SYSTEM INCLUDES A COMPLETE SET OF SOURCE-LANGUAGE PROCESSORS INCLUDING FORTRAN V, COBOL, ALGOL AND THE ASSEMBLER. THE OPERATION OF ALL OF THESE PROCESSORS IS CONTROLLED BY THE EXEC FOR THE USER OF THE SYSTEM. BY THE EXECUTIVE'S ASSUMPTION OF THE RESPONSIBILITY FOR: 1) CALLING IN PROCESSORS AS REQUIRED, 2) PROVIDING INPUTS TO THE PROCESSORS, 3) STORAGE AND MAINTENANCE OF THE OUTPUTS OF THE PROCESSORS, AND 4) THE INTEGRATION OF ACTIVITIES INVOLVING SEQUENCES OF PROCESSOR CALLS, A PROCESSOR'S OPERATION CAN BE CONFINED TO THE ACTUAL PROCESSING INVOLVED IN A SINGLE ACTIVITY. THE EXECUTIVE SYSTEM WILL TAKE CARE OF ALL OTHER FUNCTIONS.

OTHER COMPONENTS OF THE 1108 SOFTWARE SYSTEM SUCH AS SORT/MERGE, APT, PERT/COST, AND LP (LINEAR PROGRAMMING) INTERFACE WITH THE EXECUTIVE SYSTEM IN A SIMILAR MANNER.

1.3. FUNCTIONAL OBJECTIVES

THE PRIMARY OBJECTIVES IN THE DESIGN OF THE 1108 EXECUTIVE SYSTEM ARE AS FOLLOWS:

1. TO OPTIMIZE MACHINE FACILITIES USAGE, AND AT THE SAME TIME OPTIMIZE INTERACTION FOR ALL USERS BY THE USE OF MULTI-PROGRAMMING/MULTI-PROCESSING TECHNIQUES.
2. TO MAKE AVAILABLE TO REMOTE USERS THE COMPLETE FACILITIES OF THE 1108 SYSTEM.
3. TO PROVIDE AN EXECUTIVE CONTROL LANGUAGE WHOSE STRUCTURE WILL ALLOW SIMPLE PROGRAMS TO HAVE A SIMPLE MEANS OF EXPRESSING THEIR REQUIREMENTS.
4. TO PROVIDE THE FLEXIBILITY TO EXPRESS A COMPLEX ENVIRONMENT FOR COMPLEX PROGRAMS.
5. TO PROVIDE A BROAD AND EASILY-USED SPECTRUM OF PROGRAM CONSTRUCTION, MANIPULATION, AND CHECKOUT AIDS, INCLUDING THE PERMANENT STORAGE OF PROGRAM ELEMENTS ON RANDOM-ACCESS DEVICES.
6. TO PROVIDE FOR TASKS TO BE EXECUTED IN EITHER BATCH, DEMAND,

OR REAL-TIME MODE.

7.TO PROVIDE A SIMPLE AND FLEXIBLE MEANS OF COMPLETE SOFTWARE SYSTEM GENERATION AND MAINTENANCE AT THE INDIVIDUAL INSTALLATION.

8.TO PROVIDE SYSTEM INVULNERABILITY TO PROGRAMMING ERROR AND, AS FAR AS IS REASONABLE, HARDWARE ERRORS.

9.TO PROVIDE THE SIMPLEST POSSIBLE OPERATIONAL CHARACTERISTICS CONSISTENT WITH FULL UTILIZATION OF THE CAPABILITES OF THE SYSTEM.

1.4. RANGE OF EXECUTIVE SYSTEM CAPABILITIES

THE TECHNICAL CAPABILITIES OF THE UNIVAC 1108 EXECUTIVE SYSTEM SPAN A BROAD SPECTRUM OF DATA PROCESSING ACTIVITIES. ITS DESIGN IS SUCH THAT NO PENALTIES OF INEFFICIENCY ARE IMPOSED UPON ONE OF THESE ACTIVITIES BY THE SUPPORT PROVIDED FOR THE OTHER ACTIVITIES. AN INSTALLATION NOT INTERESTED IN UTILIZATION OF THE FULL SPECTRUM MAY SPECIFY CAPABILITIES TO BE ELIMINATED AT SYSTEM GENERATION TIME.

1.4.1. BATCH PROCESSING

FOREMOST AMONG SYSTEM CAPABILITIES IS THE SUPPORT PROVIDED FOR BATCH PROCESSING. DESIGN EMPHASIS HAS BEEN PLACED UPON THE ACHIEVEMENT OF EASE OF RUN PREPARATION AND SUBMISSION, MINIMIZATION OF JOB TURN-AROUND TIME, AND MINIMIZATION OF OPERATOR INTERVENTION AND DECISION REQUIREMENTS. RUN SUBMISSION MAY COME FROM MANY SOURCES, REMOTE AND CENTRAL. THESE VARIOUS INPUTS THROUGH THE EXEC'S USE AND CONTROL OF EFFICIENT MULTI-PROGRAMMING TECHNIQUES, MAY UNDERGO WHAT IS ESSENTIALLY SIMULTANEOUS INPUT, PROCESSING, AND OUTPUT. THUS, IN A DEMANDING ENVIRONMENT, THE FULL CAPABILITIES OF THE 1108 CAN BE UTILIZED EFFICIENTLY.

1.4.2. DEMAND PROCESSING

THE EXEC PROVIDES SIMULTANEOUS USE OF THE 1108 BY MANY USERS AT REMOTE CONSOLES TO OPTIMIZE THE USER/SYSTEM INTERACTION RATE. EACH USER SHARES CONTROL OF THE COMPUTATIONAL FACILITIES AND HAS THE FULL CAPABILITY OF THE 1108 CONFIGURATION AT HIS DISPOSAL.

THE DEMAND MODE OF PROCESSING IS INITIATED AND CONTROLLED BY THE EXECUTIVE CONTROL LANGUAGE. COMMANDS ARE INPUT VIA THE USER'S REMOTE CONSOLE ON A CONVERSATIONAL BASIS, THAT IS, AN IMMEDIATE SYSTEM RESPONSE WILL BE APPARENT.

PROVISIONS ARE MADE FOR: 1) DIALED COMMUNICATION CONNECTION IN ADDITION TO LEASED LINES AND REMOTE CONSOLES ON SITE; 2) PAPER TAPE INPUT ALLOWING PRE-TYPED COMMAND PROGRAMS WITH DATA FOR HIGH EFFICIENCY COMMUNICATION TRANSMISSION; 3) USER COMMUNICATION WITH THE COMPUTER CENTER, OTHER CONSOLES AND THE EXEC ITSELF.

1.4.3. REAL-TIME PROCESSING

A BASIC RESPONSIBILITY OF THE EXEC IS TO ASSIST REAL-TIME COMMUNICATIONS(RT/C) PROGRAMS WITH EXEC FUNCTIONS PROVIDED TO ALLOW RT/C PROGRAMS TO APPROPRIATELY INFLUENCE THE EXEC AND THE MULTI-PROGRAM BACKGROUND. NO ATTEMPT IS MADE TO GENERALIZE THE CONTROL REQUIRED IN EACH RT/C PROGRAM IN RECOGNITION OF THE SPECIFIC TAILORING OF A RT/C PROGRAM TO BOTH THE HARDWARE CONFIGURATION AND THE PROCESS CONTROLLED.

EXEC IS SENSITIVE TO THE NATURE OF RT/C PROCESSING AND PROVIDES APPROPRIATE MECHANISMS FOR: LOCKOUT PROTECTION FROM SIMULTANEOUS RECORD ACCESS DURING PROGRAM EXECUTION; PRIORITY SENSITIVITY; PROTECTION TO RT/C PROGRAMS FROM INTERFERENCE BECAUSE OF PERIPHERAL ACCESS OF BACKGROUND PROGRAMS (SEARCH FUNCTIONS,ETC.).

INTERFACE WITH NON-STANDARD PERIPHERALS CAN BE AT THE HARDWARE LEVEL (I/O COMMANDS AND INTERRUPTS). EXEC AWARENESS OF INDIVIDUAL TRANSMISSION LINES PROVIDES FOR

ADEQUATE RESPONSE AND FLEXIBILITY.

1.5. PROGRAM PROTECTION

THE MULTIPROGRAMMING CAPABILITIES OF THE EXECUTIVE SYSTEM IMPLIES THAT MANY UNRELATED PROGRAMS MAY BE RESIDING IN MAIN STORAGE AT THE SAME TIME. SUCH PROGRAMS MAY BE REAL-TIME RUNS, PRODUCTION RUNS, CLASSIFIED RUNS, OR SIMPLE DEBUGGING RUNS. INFRINGEMENT OF PRIVACY IN SUCH A MIXTURE IS HIGHLY PROBABLE ESPECIALLY IN CASES WHERE DEBUGGING RUNS ARE EXECUTING. THE KNOWLEDGE OR IGNORANCE OF AN INVASION MAY RANGE FROM LITTLE OR NO CONCERN FOR SOME RUNS TO GREAT CONCERN FOR CLASSIFIED OR REALTIME RUNS.

TO COMBAT THIS INVASION, INTENTIONAL OR UNINTENTIONAL, THE EXECUTIVE SYSTEM HAS UNIQUE FEATURES THAT AUTOMATICALLY GUARANTEE ABSOLUTE PROTECTION FOR EACH PROGRAM. THE PROTECTION GUARDS AGAINST TWO FORMS OF INVASION, DIRECT AND INDIRECT.

DIRECT PROTECTION SAFEGUARDS ALL PROGRAMS IN MAIN STORAGE FROM AN ACTIVE PROGRAM THAT MAY ATTEMPT TO READ, WRITE, OR JUMP INTO ANOTHER PROGRAM AREA. THIS SAFEGUARD IS EFFECTED BY 'LOCKING OUT' ANY AREA OF MAIN STORAGE THAT IS NOT ASSIGNED TO THE PRESENTLY ACTIVE PROGRAM OR, IN EFFECT, 'LOCKING IN' THE ACTIVE PROGRAM. ANY ATTEMPT TO PERFORM ANY OF THE ABOVE FUNCTIONS IS IMMEDIATELY REPORTED TO THE EXECUTIVE SYSTEM.

INDIRECT PROTECTION IS REALIZED BY RESERVING CERTAIN CONTROL FUNCTIONS FOR THE EXCLUSIVE USE OF THE EXECUTIVE SYSTEM. THESE FUNCTIONS ARE OF THE TYPE THAT COULD CAUSE A SYSTEM MALFUNCTION AND, IN TURN, A PROGRAM MALFUNCTION IF ERRONEOUSLY USED. THE EXECUTIVE SYSTEM WILL PROHIBIT THE USE OF THESE FUNCTIONS.

IN BOTH FORMS OF PROTECTION, THE EXECUTIVE SYSTEM IS, IN REALITY, GUARANTEEING ITS OWN SAFETY FROM ABUSES THAT MAY PROVE CATASTROPHIC TO THE SYSTEM.

1.6. MASS STORAGE UTILIZATION TECHNIQUES

THE UNIVAC 1108 EXECUTIVE SYSTEM IS DESIGNED TO PROVIDE INSTALLATIONS WITH AN EFFECTIVE AND EFFICIENT UTILIZATION OF THE MASS STORAGE DEVICES AVAILABLE WITH THE 1108. THE RESULT IS AN UNPRECEDENTED ABILITY TO RELIEVE OPERATORS AND PROGRAMMERS OF RESPONSIBILITIES IN MAINTAINING AND PHYSICALLY HANDLING CARDS, MAGNETIC TAPES, ETC., THUS ELIMINATING MANY OF THE ERRORS WHICH PREVIOUSLY ACCOMPANIED THE USE OF LARGE-SCALE SOFTWARE SYSTEMS. AT THE SAME TIME, THE OVERALL EFFICIENCY OF OPERATION IS CONSIDERABLY IMPROVED.

PROVISIONS ARE MADE FOR THE MAINTENANCE OF PERMANENT DATA FILES AND PROGRAM FILES ON THE MASS STORAGE DEVICES, WITH FULL FACILITIES FOR MODIFICATION AND MANIPULATION OF THESE FILES. SECURITY MEASURES ARE INVOKED BY THE EXECUTIVE SYSTEM TO INSURE THAT FILES ARE NOT SUBJECTED TO UNAUTHORIZED USE. AS UNUSED MASS STORAGE SPACE APPROACHES EXHAUSTION, PROVISIONS ARE ALSO MADE WITHIN THE EXECUTIVE SYSTEM FOR AUTOMATIC RELOCATION OF FILES OF LOW USAGE-FREQUENCY TO MAGNETIC TAPE. WHEN THE USE OF FILES RELOCATED IN SUCH A MANNER IS REQUESTED, THEY ARE RETRIEVED AND RESTORED, UNDER CONTROL OF THE EXECUTIVE SYSTEM, WITH NO INCONVENIENCE TO THE USER. FOR THE MOST PART, DYNAMIC ASSIGNMENT OF MASS STORAGE SPACE IS AVAILABLE TO THE USER VIA THE EXECUTIVE SYSTEM. TO FACILITATE EFFICIENT UTILIZATION OF AVAILABLE FACILITIES, THE USER IS ALSO ABLE TO RETURN PORTIONS OF MASS STORAGE TO GENERAL USE AS HE FINISHES WITH THEM.

1.7. PROGRAM FILES

1.7.1. BASIC CONCEPT

THE CONCEPT OF A PROGRAM FILE IS FUNDAMENTAL TO AN UNDERSTANDING OF THE 1108 SOFTWARE SYSTEM. A PROGRAM FILE IS ESSENTIALLY A NAMED SET OF ELEMENTS. THE FILE NAME IS THE PRIME IDENTIFIER FOR THE SET OF ELEMENTS. TO IDENTIFY AND LOCATE THE ELEMENTS WITHIN A PROGRAM FILE, A TABLE OF CONTENTS IS CREATED, AND MAINTAINED WITHIN THE PROGRAM FILE BY THE SYSTEM.

1.7.2. PROGRAM FILE ELEMENTS

WITHIN THE TABLE OF CONTENTS, EACH ELEMENT WITHIN THE PROGRAM FILE IS UNIQUELY IDENTIFIED BY THE FOLLOWING FOUR PARAMETERS:

1. ELEMENT TYPE
2. ELEMENT NAME
3. ELEMENT VERSION
4. ELEMENT CYCLE

ALSO INCLUDED ARE VARIOUS OTHER PARAMETERS SUCH AS THE DATE OF ELEMENT CREATION AND THE CURRENT RELATIVE LOCATION OF THE ELEMENT ON MASS STORAGE.

THE ELEMENTS CONTAINED WITHIN A PROGRAM FILE ARE OF THE FOLLOWING THREE TYPES:

1. SOURCE LANGUAGE, OR MORE GENERALLY, VARIABLE LENGTH DATA IMAGES
2. RELOCATABLE BINARY
3. ABSOLUTE BINARY

TYPICAL SOURCE-LANGUAGE ELEMENTS ARE THE FOLLOWING:

1. FORTRAN SOURCE PROGRAM
2. COBOL SOURCE PROGRAM
3. ASSEMBLER SOURCE PROGRAM
4. COLLECTOR SOURCE ELEMENT

ANY OF THESE ELEMENTS MAY BE INTRODUCED INTO A PROGRAM FILE OR MANIPULATED WITH A FILE BY THE USE OF THE APPROPRIATE PROCESSOR (FORTRAN, COBOL, ETC.) OR BY CERTAIN UTILITY ROUTINES.

THE FOLLOWING ELEMENTS MAY BE THOUGHT OF AS BEING SPECIAL-CASE SOURCE-LANGUAGE ELEMENTS:

1. ASSEMBLER PROCEDURE ELEMENTS
2. COBOL PROCEDURE ELEMENTS
3. FORTRAN PROCEDURE ELEMENTS

THESE ELEMENTS ARE AVAILABLE TO THE LANGUAGE PROCESSORS ESSENTIALLY AS SOURCE-LANGUAGE LIBRARY ELEMENTS. SPECIAL ELEMENTS ARE REQUIRED BY THE SYSTEM TO FACILITATE THE RETRIEVAL OF SOURCE LANGUAGE LIBRARY ELEMENTS AT COMPILATION OR ASSEMBLY TIME. HOWEVER, THESE ELEMENTS ARE CREATED AND MAINTAINED BY THE SYSTEM AND REQUIRE NO CONCERN ON THE PART OF THE USER.

IN ADDITION TO THE ABOVE SOURCE ELEMENTS, SETS OF EXECUTIVE CONTROL STATEMENTS MAY BE ENTERED AS SOURCE ELEMENTS. THESE ELEMENTS MAY BE CALLED BY THE @START OR @ADD STATEMENTS.

RELOCATABLE ELEMENTS ARE THE BINARY OUTPUT OF THE PROCESSORS SUCH AS FORTRAN, COBOL, THE ASSEMBLER, AND ONE SPECIAL USE OF THE COLLECTOR. ABSOLUTE ELEMENTS ARE PLACED IN A PROGRAM FILE BY THE COLLECTOR.

1.7.3. ELEMENT NAME AND VERSION

EACH ELEMENT WITHIN A PROGRAM FILE IS GIVEN A NAME SPECIFIED BY THE USER. THIS NAME IS REFERRED TO SIMPLY AS THE ELEMENT NAME. TO DISTINGUISH BETWEEN ELEMENTS OF

THE SAME NAME AND TYPE, A USER MAY SPECIFY A SUBNAME FOR AN ELEMENT, AND THIS SUBNAME IS CALLED THE ELEMENT VERSION.

BOTH AN ELEMENT NAME AND AN ELEMENT VERSION MAY BE FROM ONE TO TWELVE CHARACTERS IN LENGTH, AND THESE TWO PARAMETERS TOGETHER MUST UNIQUELY IDENTIFY ONE ELEMENT AMONG ALL ELEMENTS OF ANY PARTICULAR TYPE. ELEMENTS OF DIFFERENT TYPES (E.G., SOURCE LANGUAGE VS. RELOCATABLE BINARY)MAY, HOWEVER, HAVE THE SAME NAME AND VERSION. AN ELEMENT NAME IS REQUIRED FOR ALL ELEMENTS WITHIN A PROGRAM FILE(A NAME IS SUPPLIED AUTOMATICALLY BY THE EXEC IN MANY CASES); HOWEVER, THE SPECIFICATION OF AN ELEMENT VERSION IS NOT REQUIRED.

1.7.4. ELEMENT VERSIONS

RELOCATABLE ELEMENTS MAY BE FURTHER CLASSIFIED BY SPECIFYING A CLASS DESIGNATION WHICH IS APPLIED TO THE VERSION NAME. THE PURPOSE OF THIS CLASSIFICATION IS THE SELECTION OF ELEMENTS BASED ON PARAMETERS SUITED FOR THE PARTICULAR ALLOCATION TO BE MADE. LETTERS WITHIN THE VERSION NAMES OF ELEMENTS ARE GIVEN MEANING BY THE PROGRAMMER WHICH CAN THEN BE USED TO SELECT A PROPER CLASS OR CLASSES ACCORDING TO THE NEED. EACH REQUIRED ELEMENT NEED NOT BE NAMED, BUT THE PROPER ELEMENT WILL BE SELECTED BY ELIMINATION.

1.7.5. 'CYCLE' PARAMETER

FOR DIFFERENTIATION AMONG SYMBOLIC ELEMENTS, AN INTEGER PARAMETER CALLED 'CYCLE' IS ASSOCIATED WITH EACH ELEMENT. THIS ALLOWS SEVERAL 'COPIES' OF THE SAME VERSION OF AN ELEMENT TO BE RETAINED WITHIN A PROGRAM-FILE. EACH ITEM(IMAGE) OF A SYMBOLIC ELEMENT HAS A CYCLE NUMBER INDICATING TO WHICH CYCLE IT BELONGS, AND, IF DELETED, A DELETE-CYCLE NUMBER TO INDICATE IN WHICH CYCLE THIS ITEM WAS DELETED. WHEN A SYMBOLIC ELEMENT IS UPDATED, THE UPDATE ITEMS ARE INSERTED WHERE THEY BELONG IN THE ELEMENT AND GIVEN A CYCLE NUMBER ONE GREATER THAN THE LAST CYCLE OF THE ELEMENT. ANY PREVIOUS CYCLE ITEMS THAT HAVE BEEN DELETED BY THIS UPDATE ARE MARKED SO. THE USER MAY MAKE REFERENCES BY CYCLE NUMBER. THIS GIVES THE SAME EFFECT AS THOUGH SEVERAL DIFFERENT COPIES OF THE ELEMENT WERE MAINTAINED. THE USER MAY SET THE NUMBER OF UPDATE CYCLES TO BE RETAINED AT ANY LEVEL HE DESIRES; HOWEVER, HE NEED SET THAT NUMBER ONLY IF HE DESIRES TO CHANGE IT FROM THE STANDARD SYSTEM ASSUMPTION. THIS STANDARD VALUE MAY BE ALTERED AT SYSTEM GENERATION TIME.

IN SPECIFYING A SYMBOLIC ELEMENT FOR COMPILATION OR ASSEMBLY, THE USER MAY REFERENCE A SPECIFIC UPDATE FROM A SEQUENCE OF RETAINED UPDATES BY SPECIFYING THE PROPER UPDATE CYCLE NUMBER AS PART OF THE EXECUTIVE CONTROL STATEMENT CALLING FOR THE COMPILER OR ASSEMBLER. IN COMPILATION, THE UPDATE ENTRY WILL BE COMBINED WITH THE ELEMENT IN ITS COMPLETE STATE THEREBY CREATING A COMPLETE ELEMENT AS OF THAT CYCLE.

AS SOON AS THE NUMBER OF UPDATES RETAINED FOR AN ELEMENT EXCEEDS THE SPECIFIED MAXIMUM, THE UPDATE OF THE LOWEST CYCLE NUMBER (THE ORIGINAL, COMPLETE ELEMENT) IS COMBINED WITH THE UPDATE NEXT LOWEST IN CYCLE NUMBER;IN EFFECT, THE OLDEST ENTRY IS DISCARDED, AND THE NEXT-OLDEST, IN ITS COMPLETED FORM, BECOMES THE OLDEST TO MAKE ROOM FOR THE LATEST CYCLE ENTRY. THESE CORRECTIONS THUS BECOME INCORPORATED PERMANENTLY INTO THE BASIC ELEMENTS AND CAN ONLY BE REMOVED BY ENTERING NEW CORRECTION STATEMENTS.

THIS TECHNIQUE OF HANDLING SYMBOLIC ELEMENTS OFFERS TWO DISTINCT ADVANTAGES;

(1)THE USER IS ALLOWED TO KEEP MANY DIFFERING COPIES OF THE SAME ELEMENT IN A PROGRAM FILE WHILE REQUIRING LITTLE ADDITIONAL STORAGE OVER THAT NEEDED FOR A SINGLE COPY.

(2)THE USER IS ABLE TO REFER EASILY TO EARLIER COPIES OF A SPECIFIC ELEMENT WITHOUT HAVING TO PREPARE CORRECTIONS DELETING PREVIOUSLY INPUT CORRECTIONS. HOWEVER, IF A SET OF CORRECTIONS ARE APPLIED TO ANY CYCLE EXCEPT THE LATEST AND THE UPDATED CYCLE IS TO BE RETAINED, ALL CYCLES THAT PREVIOUSLY FOLLOWED THE CYCLE TO BE UPDATED WILL BE DELETED. THE NEW CYCLE NUMBER WILL BE THE UPDATED CYCLE NUMBER PLUS ONE.

2. HARDWARE CONFIGURATIONS

THE EXECUTIVE SYSTEM INCLUDES FACILITIES FOR THE SUPPORT AND USE OF MANY DIFFERENT CONFIGURATIONS AND EQUIPMENTS. THE DETAILED DESCRIPTIONS OF HOW ALL DEVICES ARE INCLUDED IN THE CONFIGURATION AS EACH EXECUTIVE SYSTEM IS TAILORED TO A PARTICULAR SITE ARE GIVEN IN THE SECTION ON SYSTEM SETUP (SECTION 18). IN SECTION 2.1 IS A LIST OF THE DEVICES SUPPORTED AND INFORMATION ON THE CALCULATION OF THE REQUIRED EQUIPMENT FOR PARTICULAR INSTALLATIONS.

2.1. MINIMUM CONFIGURATION

A MINIMUM CONFIGURATION IS GIVEN HERE ONLY TO INDICATE THE BASIC EQUIPMENT REQUIRED AND ASSUMED IN THE DESIGN OF THE EXECUTIVE SYSTEM. IN SECTION 3 FURTHER DETAIL IS GIVEN ON STORAGE REQUIREMENTS THAT EACH USER WILL FIND NECESSARY TO PROPERLY OPERATE UNDER THE EXPECTED WORK LOAD.

MINIMUM CONFIGURATION

PROCESSOR WITH CONSOLE
131K MAIN STORAGE
ONE DRUM SUBSYSTEM WITH 1-FH880, OR 3-FH432 DRUMS
ONE ON-LINE 1004-II (READ-PRINT-PUNCH) OR EQUIVALENT
ONE TAPE SUBSYSTEM WITH TWO VIC-VIIIC TAPE UNITS
ONE FASTRAND SUBSYSTEM WITH ONE UNIT

THE FASTRAND SUBSYSTEM FORMS A PART OF THE MINIMUM CONFIGURATION TO PROVIDE SUFFICIENT SPACE FOR EFFECTIVE USE OF MANY EXECUTIVE FEATURES. THE TABLE BELOW IS PROVIDED AS AN AID IN DETERMINING THE SPACE DESIRED WHEN USING THE MASS STORAGE MANAGEMENT FACILITIES OF THE EXECUTIVE. ONLY THE FASTRAND SUBSYSTEM CLEARLY MEETS THE TYPICAL NEEDS WHEN USING THE SYSTEM FULLY, AND, IT IS FOR THIS REASON THAT THE DEFINITION ABOVE IS MADE. HOWEVER, COMPLETE SUBSTITUTION BY HIGH SPEED DRUMS IS POSSIBLE WHEN A SUFFICIENT QUANTITY IS MADE AVAILABLE.

THIS CONFIGURATION IS REQUIRED FOR MAINTENANCE OF THE SYSTEM ITSELF. THE SYSTEM IS LOADED FROM A SINGLE TAPE UNIT AND STORED ON DRUM. ONCE LOADED THE SYSTEM DOES NOT REQUIRE THIS TAPE UNIT FOR OPERATION, AND IT IS FREE FOR USE IN OTHER APPLICATIONS. THE PARTICULAR SITE WILL EXPAND THE SYSTEM TO REFLECT THE CONFIGURATION. THE METHOD OF DESCRIBING A SYSTEM IS DETAILED IN THE SECTION ON SYSTEM GENERATION (SECTION 18).

2.2. STORAGE REQUIREMENTS

THE EXECUTIVE SYSTEM REQUIRES TWO FH-432 DRUMS FOR PERMANENT STORAGE OF THE SYSTEM, PROCESSORS, AND LIBRARIES. ALL OTHER MASS STORAGE IS AVAILABLE TO THE USER, OR IS REQUIRED IN THE PERFORMANCE OF EXECUTIVE SYSTEM FUNCTIONS AT THE DIRECTION OF THE USER.

AN APPROXIMATION OF THE REQUIRED MASS STORAGE SPACE NEEDED CAN BY FOUND BY ESTIMATING THE QUANTITIES BELOW.

WORDS (DECIMAL)	REQUIREMENT
440000	PERMANENT STORAGE FOR EXECUTIVE, PROCESSORS, AND SYSTEM LIBRARY.
LENGTH OF ALL ACTIVE PROGRAMS	SWAP STORAGE AREA ON HIGH SPEED DRUM.

LENGTH + 100 FOR EACH ACTIVE PROGRAM	STORAGE FOR EACH PROGRAM IN ABSOLUTE FORM.
100000 OR USER SCRATCH WHICHEVER IS GREATER	SCRATCH SPACE FOR OPERATING PROCESSORS OR USER PROGRAMS.
LENGTH OF USER DATA	SPACE ALLOCATED FOR STORAGE OF PERMANENT USER DATA FILES.
4000	EACH PROGRAM FILE
20 PER EACH SYMBOLIC LINE OF CODE STORED	FOR STORAGE OF SYMBOLIC AND RELOCATABLE PROGRAM ELEMENTS OR COMMAND STREAMS STORED FOR LATER REFERENCE.
60 PER ACCOUNT NUMBER	PERMANENT ACCOUNT FILE MAINTAINED BY EXECUTIVE.
20000	TEMPORARY LOG FILE.
250000 PER PRINTER	DATA REQUIRED TO PROVIDE 20 MINUTES OF BACKUP FOR ONE PRINTER.
250000 PER CARD READER	20 MINUTE READER BACKUP.
125000 PER CARD PUNCH	20 MINUTE PUNCH BACKUP.
750000 PER REMOTE 1004-DLT1 TERMINAL	CONSIDERABLE BUFFER SPACE IS REQUIRED FOR REMOTE BATCH TERMINALS BECAUSE OF LINE SPEEDS.
LENGTH OF ALL DUMPS + 10%	SCRATCH STORAGE OF ALL DIAGNOSTIC SNAPSHOTS AND DUMPS THAT HAVE NOT BEEN EDITED FOR OUTPUT.
LENGTH + 500 FOR PROGRAM CHECKPOINTED.	SPACE FOR EACH MASS STORED CHECKPOINT BEING HELD BY THE SYSTEM.
10% OF ALL ABOVE REQUIREMENTS	DIRECTORY, EXECUTIVE OVERHEAD, ETC.

2.3. EXECUTIVE MAIN STORAGE REQUIREMENTS

THE EXECUTIVE SYSTEM REQUIRES A VARIABLE AMOUNT OF MAIN STORAGE. THE STORAGE USED AT A GIVEN INSTALLATION IS DIVIDED BETWEEN THE LOWER PART OF THE FIRST MODULE AND THE UPPER PART OF THE LAST MODULE. THIS INCLUDES:

1. RESIDENT ROUTINES.
2. PERMANENT DATA STORAGE.
3. SEGMENT OVERLAY STORAGE.
4. VARIABLE DATA SPACE.

AS THE EXECUTIVE NEEDS ADDITIONAL STORAGE, BECAUSE OF THE NATURE OF THE CURRENT WORK LOAD, THE UPPER END OF THE I-AREA AND/OR THE LOWER END OF THE D-AREA WILL BE EXPANDED TO MEET THE NEW STORAGE REQUIREMENTS.

2.4. TAPE USAGE IMPLIED BY EXECUTIVE OPERATION

IN THE DETERMINATION OF THE APPROPRIATE NUMBER OF TAPE UNITS IN ANY CONFIGURATION, IT IS NECESSARY TO INCLUDE A SUFFICIENT NUMBER OF UNITS FOR USER REQUESTED BUT SYSTEM CONTROLLED USAGE. EXECUTIVE FEATURES WHICH REQUIRE THE ASSIGNMENT OF TAPE UNITS ARE:

1. MASS STORAGE FILE ROLLOUT AND ROLLBACK
   WHEN THE QUANTITY OF MASS STORAGE AVAILABLE FOR ASSIGNMENT HAS DECREASED BY THE ASSIGNMENT OF PERMANENT FILE STORAGE SO THAT MORE SPACE MUST BE SECURED, A TAPE UNIT WILL BE USED BY EXEC TO RELIEVE THIS OVERLOAD BY TEMPORARY MOVEMENT OF FILES TO TAPE.

2. CHECKPOINT
   AT THE USERS REQUEST OR BY OPERATOR ACTION, A TAPE UNIT MAY BE CALLED UPON FOR THE CHECKPOINT TAKEN OF THE RUN BY THE EXECUTIVE SYSTEM.

3. SYMBIONT INPUT/OUTPUT
   CERTAIN DATA TAPES CREATED OR USED BY THE SYMBIONT COMPLEX MUST BE ASSIGNED. OPERATIONS SUCH AS CARD-TO-TAPE AND TAPE-TO-PRINTER ARE AVAILABLE WHERE NO DIRECT ASSOCIATION WITH A USER PROGRAM EXISTS.

4. LOGGING OUTPUT
   AN OPTIONAL OUTPUT OF THE EXECUTIVE SYSTEM IS THE LOG OF ALL SYSTEM ACTIVITY. THIS MAY BE SENT TO A TAPE WHICH WOULD BE ASSIGNED ALMOST CONTINUALLY FOR THIS PURPOSE.

2.5. UTILIZATION OF ADDITIONAL HARDWARE

THE EXECUTIVE SYSTEM IS DESIGNED TO FULLY UTILIZE ALL HARDWARE ATTACHED TO THE 1108. IN SO FAR AS UTILIZATION OF THE CAPABILITIES OF THE 1108 IS CONCERNED, ADDITIONAL HARDWARE, AFFORDS AN INSTALLATION A MORE EFFICIENT OPERATION. DELAYS RESULTING FROM SETUP TIME REQUIREMENTS ARE MINIMIZED WITH ADDITIONAL EQUIPMENT, AND MAIN STORAGE SPACE AND CPU TIME CAN BE MORE EFFICIENTLY UTILIZED IF ADDITIONAL PERIPHERAL EQUIPMENT IS AVAILABLE. BECAUSE OF THE FEATURES OF THE EXECUTIVE SYSTEM WHICH PROVIDE FOR PERMANENT RESIDENCE OF BOTH PROGRAM AND DATA FILES ON MASS STORAGE DEVICES, MOST INSTALLATIONS CONCERNED WITH BATCH PROCESSING BENEFIT CONSIDERABLY BY ADDITIONAL MASS STORAGE. PHYSICAL HANDLING OF TAPES AND CARDS AS WELL AS THE TOTAL AMOUNT OF ACTIVITY CONCERNED ONLY WITH INPUT CAN BE GREATLY REDUCED BY INCREASING MASS STORAGE.

ADDITIONAL MAIN STORAGE EXPANDS THE CAPABILITIES OF THE SYSTEM CONSIDERABLY AS A RESULT OF THE MULTI-PROGRAMMING TECHNIQUES AND CAPABILITIES DESIGNED INTO THE EXECUTIVE SYSTEM. IT IS TRUE, OF COURSE, THAT A VARIETY OF PERIPHERAL EQUIPMENT SUITED TO THE GENERAL NATURE OF THE ACTIVITY AT A SPECIFIC INSTALLATION MUST BE AVAILABLE BEFORE A HIGH DEGREE OF UTILIZATION OF ADDITIONAL MAIN STORAGE CAN BE SUSTAINED. THIS IS TRUE BECAUSE A RUN CANNOT BE PROCESSED UNTIL ALL OF ITS FACILITY REQUIREMENTS ARE MET, REGARDLESS OF THE AMOUNT OF MAIN STORAGE SPACE AVAILABLE.

2.6. I/O DEVICE-SUPPORT CAPABILITIES

THE FOLLOWING INPUT/OUTPUT DEVICES ARE SUPPORTED BY THE 1108 EXECUTIVE SYSTEM:

EQUIPMENT TYPE	MODEL	OPTIONS
MAGNETIC TAPES	IIIA	DUAL CHANNEL, 100 KC FORMAT.
	IVC	800 PPI, TRANSLATE FEATURE.
	VIC	TRANSLATE FEATURE.
	VIIIC	TRANSLATE FEATURE-- DUAL CHANNEL
	IIA	
MAGNETIC DRUMS	FH880	
	FH432	
	FASTRAND II	DUAL CHANNEL, FASTBAND, WRITE LOCKOUT.
PRINTERS	755	
CARD READ-PUNCH	READER-80 COL. 900 CPM	
	PUNCH-80 COL. 300 CPM	
1004 ON-SITE	I-02,04,06,07	80 COLUMN PUNCH
	II-02,04,06,07	PAPER TAPE READ
	III-02,04,06,07	PAPER TAPE PUNCH
	WORD INTERFACE, EXTERNAL INTERRUPT, AND ILLEGAL CARD CODE DETECTOR FEATURES ARE MANDATORY	CODE IMAGE READ CODE IMAGE PUNCH

THE 1004 MODELS I-02 OR 04, II-02 OR 04, III-02 OR 04 WILL BE CAPABLE OF LIMITED OPERATIONS ONLY. THE MODEL 02 1004'S WILL HANDLE 80 COLUMN READ, PUNCH AND PRINT ONLY. THE MODEL 04 1004'S WILL HANDLE SOME ADDITIONAL FUNCTIONS. FOR FULL USE OF ALL THE OPTIONS SUPPORTED, A MODEL 06 OR 07 1004 MUST BE USED.

1004 REMOTE VOICE-GRADE LINE	DLT-1	80 COLUMN PUNCH
	I-06,07	
	II-06,07	
	III-06,07	
	DLT-1B (SLOW-)	80 COLUMN PUNCH
	LINE OPTION)	CODE IMAGE READ
	I-06,07	CODE IMAGE PUNCH
	II-06,07	PAPER TAPE READ
	III-06,07	PAPER TAPE PUNCH
TELPAK-A*	DLT-1B	80 COLUMN PUNCH
	I-06,07	CODE IMAGE READ
	II-06,07	CODE IMAGE PUNCH
	III-06,07	PAPER TAPE READ
		PAPER TAPE PUNCH

*TRADEMARK OF AMERICAN TELEPHONE AND TELEGRAPH COMPANY

THE 1108 EXECUTIVE SYSTEM SUPPORTS COMMUNICATION SUBSYSTEMS WHICH ALLOW A NUMBER OF REMOTE STATIONS TO SIMULTANEOUSLY EXCHANGE DATA WITH THE 1108 AS WELL AS SUBSYSTEMS WHICH ALLOW ONLY ONE REMOTE STATION TO EXCHANGE DATA AT ONE TIME. UNATTENDED ANSWERING AND AUTOMATIC DIALING ARE AVAILABLE TO ALL COMMUNICATIONS SUBSYSTEMS. EACH SUBSYSTEM IS LISTED BELOW WITH ITS MODES OF OPERATION, CORRESPONDING DATA SET, AND REMOTE DEVICES.

COMMUNICATION SUBSYSTEM	MODE OF OPERATION	REMOTE DEVICE
COMMUNICATIONS TERMINAL SYNCHRONOUS (CTS).	SWITCHED NETWORK (AT + T 201A DATA SET OR EQUIVALENT (2000 BITS/SEC.))	1004 II, 1004 III, *READ/PUNCH/ PRINT UNIT,
	LEASED LINE (AT + T 201B DATA SET OR EQUIVALENT (2400 BITS/SEC.))	SAME AS ABOVE.
	BROADBAND(TELPAK) (AT + T 301B DATA SET OR EQUIVALENT (40800 BITS/SEC.))	SAME AS ABOVE.
COMMUNICATIONS TERMINAL MODULE CONTROLLER	ANY SWITCHED NETWORK OR LEASED LINE EXCEPT BROADBAND CONFORMING TO EIA STANDARDS FOR USE WITH RELAY INTERFACE OR APPROPRIATE DATA SET (MODEM)	USER CONTROLLED DEVICES SUCH AS TELETYPES, KEYBOARDS AND PRINTERS, CRT DISPLAYS, PAPER TAPE READERS AND REPERFORATORS VIA SIMPLEX, HALF DUPLEX OR FULL DUPLEX CIRCUITS
WORD TERMINAL SYNCHRONOUS(WTS).	SAME AS CTS.	REMOTE 1004 II AND 1004 III

FOLLOWING ARE THREE CONFIGURATORS WHICH ILLUSTRATE THE HARDWARE SUPPORTED BY THE THREE 1004 SYSTEMS. FIGURE 2.1 SHOWS SHOWS THE CONFIGURATION FOR THE ON-SITE 1004 (BOARD #1); FIGURE 2.2 SHOWS THE COMPRESSED REMOTE 1004 (BOARD #2); FIGURE 2.3 SHOWS THE NON-COMPRESSED REMOTE 1004 (BOARD #3).

```
      1004 CONFIGURATION
      BOARD #1 (ON-SITE)

ON-LINE, ON-SITE, 80 COLUMN ONLY


      1108 CHANNEL                  IOC CHANNEL (MP ONLY)
          :                                   :
          :                                   :
          :---------------:-------------------:
                          :
                          :
                          :
                          :
          :---------------:-------------------:
          :                                   :
          :                                   :
          :           :-  REQUIRED  -:        :
     1004 I-02   -:   : EXTERNAL     :   :-1004 I-04,-06,-07
     1004 II-02   :---: INTERRUPT    :---: 1004 II-04,-06,-07
     1004 III-02 -:   : FEATURE,AND  :   :-1004 III-04,-06,-07
          :           : ILLEGAL CARD :        :
          :           : CODE DETECT  :        :
     -OPTIONS-        : FEATURE      :     -OPTIONS-
     80COLUMN PUNCH   :-            -:     80 COLUMN PUNCH
                                           PAPER TAPE READ
                                           PAPER TAPE PUNCH
                                           CODE IMAGE READ
                                           CODE IMAGE PUNCH


   ON-SITE FUNCTIONS
      READ 80 COLUMN CARDS. . . . . . . . . ALL MODELS SHOWN
      PUNCH 80 COLUMN CARDS . . . . . . . . ALL MODELS SHOWN
      PRINT . . . . . . . . . . . . . . . . ALL MODELS SHOWN
      READ CODE IMAGE CARDS . . . . . . . . -04,-06,-07 ONLY
      PUNCH CODE IMAGE CARDS. . . . . . . . -04,-06,-07 ONLY
      READ PAPER TAPE (4 LEVELS). . . . . . -04,-06,-07 ONLY
      PUNCH PAPER TAPE (4 LEVELS) . . . . . -04,-06,-07 ONLY


FIGURE 2.1
```

1004 CONFIGURATION
BOARD #2 (REMOTE COMPRESSED)

REMOTE, VOICE-GRADE LINES, DLT-1, 80 COLUMN ONLY

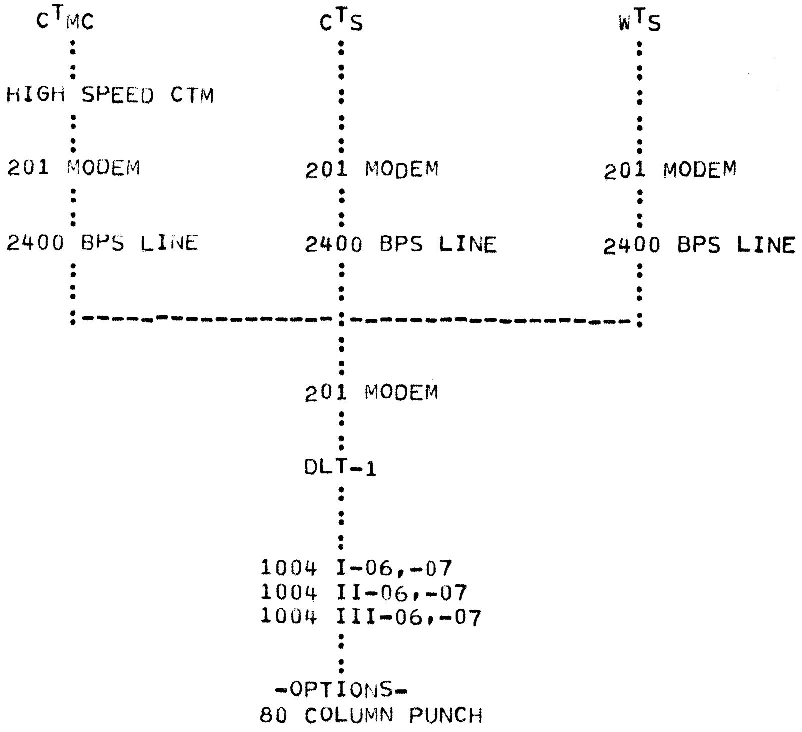


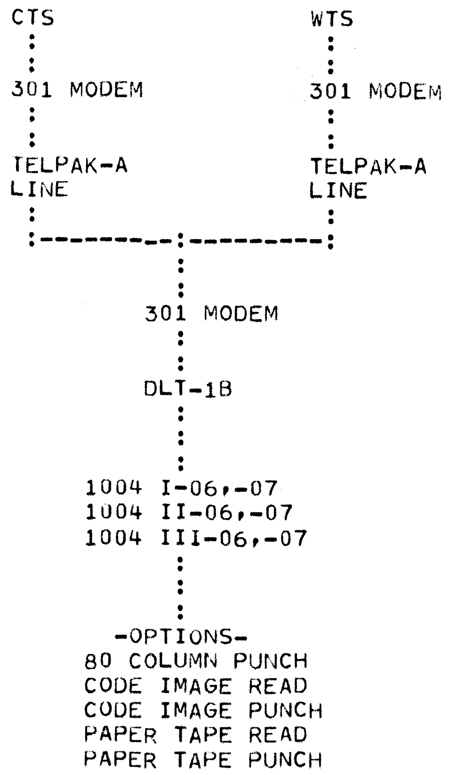
FIGURE 2.2

```
    REMOTE FUNCTIONS (BOARD #2)
       READ 80 COLUMN CARDS. . . . . . . . . . ALL MODELS SHOWN
       PUNCH 80 COLUMN CARDS . . . . . . . . . ALL MODELS SHOWN
       PRINT . . . . . . . . . . . . . . . . . ALL MODELS SHOWN

*NOTE* THIS BOARD READS 80 COLUMN CARDS, REMOVES BLANKS FROM
       THE IMAGE THROUGH COMPRESSION, BUILDS A 330 CHARACTER
       BUFFER AND TRANSMITS IT.  THE BOARD ALSO ACCEPTS 330
       CHARACTER COMPRESSED OUTPUT BUFFERS, DECOMPRESSES AND
       PRINTS OR PUNCHES THE DATA.

*NOTE* WITH MODIFICATION BOARD #2 COULD BE USED WITH TELPAK-A
       LINES AND A DLT-1B.  HOWEVER, THIS SET-UP WOULD NOT
       PROVIDE FULL 1004 SPEEDS.
```

FIGURE 2.2 CONTINUED

1004 CONFIGURATION
BOARD #3 (REMOTE NON-COMPRESSED)

REMOTE, TELPAK-A LINES, DLT-1B



```
CTS                 WTS
 :                   :
 :                   :
301 MODEM           301 MODEM
 :                   :
 :                   :
TELPAK-A            TELPAK-A
LINE                LINE
 :                   :
 :---------:---------:
           :
           :
         301 MODEM
           :
           :
         DLT-1B
           :
           :
           :
     1004 I-06,-07
     1004 II-06,-07
     1004 III-06,-07
           :
           :
           :
       -OPTIONS-
     80 COLUMN PUNCH
     CODE IMAGE READ
     CODE IMAGE PUNCH
     PAPER TAPE READ
     PAPER TAPE PUNCH
```

FIGURE 2.3

```
    REMOTE FUNCTIONS (BOARD #3)
       READ 80 COLUMN CARDS. . . . . . . . . . ALL MODELS SHOWN
       PUNCH 80 COLUMN CARDS . . . . . . . . . ALL MODELS SHOWN
       PRINT . . . . . . . . . . . . . . . . . ALL MODELS SHOWN
       READ CODE IMAGE CARDS . . . . . . . . . -06,-07 ONLY
       PUNCH CODE IMAGE CARDS. . . . . . . . . -06,-07 ONLY
       READ PAPER TAPE (4 LEVELS). . . . . . . -06,-07 ONLY
       PUNCH PAPER TAPE (4 LEVELS) . . . . . . -06,-07 ONLY

*NOTE* WITH SLIGHT MODIFICATION BOARD #3 COULD BE USED WITH
       THE SLOW-SPEED OPTION ON DLT-1B AND VOICE-GRADE LINES.
       THE SPEED OF THE 1004 WOULD BE GREATLY REDUCED.


FIGURE 2.3 CONTINUED
```

3. BASIC CONCEPTS OF THE UNIVAC 1108 EXECUTIVE SYSTEM

3.1. DEFINITIONS

CERTAIN TERMS ARE REFERRED TO IN THIS MANUAL WITH THE ASSUMPTION THAT THE READER IS ACQUAINTED WITH THEIR MEANING. THE FOLLOWING DEFINITIONS ARE FOR THE CONVENIENCE OF THE READER.

3.1.1. ACTIVITY

A DIVISION OF A PROGRAM WHICH MAY BE EXECUTED INDEPENDENT OF OTHER PORTIONS OF THE PROGRAM. IT IS USUALLY CONSIDERED PART OF A TASK.

3.1.2. ACTIVITY REGISTRATION

THE ACT OF REGISTERING WITH THE EXECUTIVE SYSTEM AN ACTIVITY WHICH CAN BE EXECUTED ASYNCHRONOUSLY WITH OTHER PARTS OF A PROGRAM. FORKING.

3.1.3. BATCH PROCESSING

A MODE OF OPERATION WHERE SEVERAL RUNS ARE GROUPED PRIOR TO PROCESSING. TRANSITION FROM RUN TO RUN IS EFFECTED BY THE EXECUTIVE SYSTEM.

3.1.4. BREAKPOINT

THE DIVISION OF SYMBIONT DEFINED FILES. ALLOWS THOSE PORTIONS OF THE FILE TO BE QUEUED INDEPENDENTLY OF RUN COMPLETION. MAXIMUM USE OF AVAILABLE PRINTERS AND PUNCHES IS ACHIEVED IN THIS MANNER.

3.1.5. CENTRAL SITE

THE 1108 COMPUTER AND ITS ATTACHED PERIPHERAL EQUIPMENT.

3.1.6. COLLECTION

THE PROCESS BY WHICH ELEMENTS OF A PROGRAM ARE COLLECTED BY SATISFYING THE EXTERNAL SYMBOLS OF THE INITIAL ELEMENT AND ALL REFERENCED ELEMENTS. THE RESULTING STRUCTURE DEFINES A PROGRAM TO BE ALLOCATED AND EXECUTED.

3.1.7. COMMUNICATION DEVICE

AN INPUT OR OUTPUT DEVICE WHICH OPERATES IN A REAL-TIME MODE. THE CENTRAL PROCESSING UNIT MUST BE PREPARED TO RECEIVE INPUT AT ANY TIME OR THE INFORMATION MAYBE LOST.

3.1.8. DEMAND PROCESSING

THE MANNER OF PROCESSING IN WHICH THE EXECUTIVE SYSTEM OR A PROCESSOR SPONTANEOUSLY REACTS TO THE INPUTS FROM A REMOTE INQUIRY TERMINAL WHICH IS SENDING MESSAGES AS REQUIRED. THIS IS ESSENTIALLY A DEMAND AND RESPONSE TYPE OF ACTIVITY.

3.1.9. ELEMENT

THE BASIC COMPONENT OF A PROGRAM FILE USUALLY DEFINED AND MANIPULATED AS A UNIT. THE FORM OF AN ELEMENT IS DEPENDENT UPON THE PROGRAM USING IT.

3.1.10. EXECUTIVE CONTROL LANGUAGE

SPECIFICALLY FORMATTED INPUT INFORMATION WHICH IS USED TO DIRECT THE ACTIVITY OF

THE EXECUTIVE SYSTEM.

3.1.11. FACILITIES

THE PERIPHERAL UNITS, MAIN STORAGE, TAPE DRIVES, DRUM STORAGE, ETC.

3.1.12. FILE

AN ORGANIZED COLLECTION OF DATA STORED IN SUCH A MANNER SO AS TO FACILITATE THE RETRIEVAL OF EACH INDIVIDUAL DATUM.

3.1.13. GRANULE

THE INCREMENTAL SIZE IN WHICH A STORAGE UNIT IS ASSIGNABLE.

3.1.14. MULTI-PROGRAMMING

THE CONCURRENT EXECUTION OF SEVERAL PROGRAMS WHICH OCCUPY MAIN STORAGE. THIS IS ACCOMPLISHED BY SHARING THE ATTENTIONS OF THE CENTRAL PROCESSOR.

3.1.15. PACKET

A CONTIGUOUS SET OF WORDS WHICH CONTAIN INFORMATION DESCRIBING AN INPUT/OUTPUT OPERATION TO BE PERFORMED

3.1.16. PROCESSOR CALL STATEMENTS

SPECIFICALLY FORMATTED INPUT INFORMATION WHICH IS USED TO DIRECT THE ACTIVITY OF A SYSTEM PROCESSOR. A SUBSET OF THE EXECUTIVE CONTROL LANGUAGE.

3.1.17. PROGRAM

A COLLECTION OF INSTRUCTIONS, EXECUTION OF WHICH RESULTS IN PERFORMANCE OF ONE OR MORE LOGICAL FUNCTIONS. A PROGRAM IS THE SUB-DIVISION OF THE EXECUTABLE ASPECTS OF A RUN.

3.1.18. PROGRAM FILE

A FILE IN WHICH THE DATA ARE THE CONSTITUENTS OF A PROGRAM OR OF SEVERAL PROGRAMS. THIS DATA MAY CONSIST OF PROGRAM ELEMENTS IN SYMBOLIC, RELOCATABLE BINARY, OR ABSOLUTE BINARY FORM. SPECIAL INFORMATION IN THE PROGRAM FILE IS USED TO AID THE SYSTEM IN THE MANIPULATION OF THE PROGRAM CONSTITUENTS.

3.1.19. REAL-TIME PROCESSING

AN OPERATING ENVIRONMENT IN WHICH THE RESPONSE TO AN EXTERNAL STIMULI IS SUFFICIENTLY FAST TO ACHIEVE A DESIRED OBJECTIVE. DEPENDING UPON, THE APPLICATION THE RESPONSE TIME MAY VARY FROM SECONDS TO MICROSECONDS. GENERALLY, REAL-TIME PROCESSING IS UNDER THE INFLUENCE OF ASYNCHRONOUS INPUTS FROM ONE OR MORE DEVICES.

3.1.20. RE-ENTRANT CODING

A SET OF INSTRUCTIONS CODED IN SUCH A MANNER THAT THEY MAY LOGICALLY PERFORM THE SAME TASK ON DIFFERENT DATA SETS SIMULTANEOUSLY.

3.1.21. REMOTE SITE

A COMMUNICATIONS TERMINAL WHICH IS CAPABLE OF SENDING INFORMATION TO AND RECEIVING INFORMATION FROM THE CENTRAL PROCESSOR VIA SOME COMMON CARRIER OR TRANSMISSION SCHEME.

3.1.22. RUN

A RUN IS THE STANDARD UNIT IN WHICH WORK IS ENTERED INTO THE OPERATING SYSTEM.

THIS CONSISTS OF A RUN COMMAND FOLLOWED BY ONE OR MORE CONTROL COMMANDS WHICH CAUSES THE ORDERED EXECUTION OF PROCESSORS AND/OR WORKER PROGRAMS.

3.1.23. SIMULATED FASTRAND

DRUM SIMULATION OF FASTRAND WHICH ALLOWS EXECUTION OF A PROGRAM WITH FILES DESIGNED FOR FASTRAND ALLOCATION ALLOCATED TO THE SECTION OF THE 'FLYING HEAD' DRUM STORAGE DESIGNATED AS SIMULATED FASTRAND.

3.1.24. SWAPPING

THE PROCESS OF STORING LOW PRIORITY OR SUSPENDED PROGRAMS ON SECONDARY STORAGE IN ORDER TO ALLOW SPACE TO RETRIEVE ANOTHER PROGRAM INTO PRIMARY STORAGE FOR EXECUTION.-

3.1.25. SYSTEMS PROCESSOR

A PROGRAM WHICH PERFORMS SPECIALIZED FUNCTIONS UNDER THE CONTROL OF THE EXECUTIVE SYSTEM.

3.1.26. TASK

A LOGICAL STEP IN THE PROCESSING OF A RUN. FOR EXAMPLE, EXECUTION OF A SYSTEM PROCESSOR OR A USER PROGRAM.

3.2. SYSTEM CONVENTIONS

3.2.1. PRIVILEGED INSTRUCTIONS

SEVERAL INSTRUCTIONS ARE RESERVED FOR THE EXECUTIVE SYSTEM USAGE.ONLY. IF ANY OF THE PRIVILEGED INSTRUCTIONS ARE EXECUTED BY A USER PROGRAM A GUARD MODE INTERRUPT WILL OCCUR. THE HANDLING OF THE GUARD MODE INTERRUPT IS DESCRIBED IN DETAIL IN THE SUPERVISOR SECTION (SECTION 7). THE PRIVILEGED INSTRUCTIONS ARE:

72 - 15	LOAD PROCESSOR STATE REGISTER
72 - 16	LOAD STORAGE LIMITS REGISTER
73 - 14	INITIATE INTER-PROCESSOR INTERRUPT
73 - 15	SELECT INTERRUPT LOCATION
73 - 16	LOAD CHANNEL SELECT REGISTER
75 (0-15)	ALL I/O INSTRUCTIONS

THE INSTRUCTION, 'PREVENT ALL I-O INTERRUPTS AND JUMP'(72 13), IS NOT A PRIVILEGED INSTRUCTION FOR THE UNIT PROCESSOR, BUT CAUTION SHOULD BE EXERCISED WHEN USING IT. THE HARDWARE ALLOWS INTERRUPTS TO BE LOCKED OUT FOR ONLY 100 MICROSECONDS WHILE IN GUARD MODE. DATA TRANSFERS ASSUME TIME PRIORITY OVER INSTRUCTION EXECUTION, THEREFORE, A GUARANTEE CAN NOT BE MADE ON THE NUMBER OF INSTRUCTIONS EXECUTED BEFORE INTERRUPTS ARE ENABLED.

3.2.2. SYMBOLISM

1. WHEN IT IS NECESSARY TO INDICATE PARTICULAR BITS IN A WORD, THEY ARE NUMBERED FROM RIGHT TO LEFT.

```
 35                                                    0
--------------------------------------------------------
:                                                      :
--------------------------------------------------------
```

2. WHEN PARTS OF WORDS ARE REFERENCED THE FOLLOWING SYMBOLS ARE USED:

```
 35     30 29     24 23     18 17     12 11      6 5        0
--------------------------------------------------------------
:    S1   :    S2   :    S3    :   S4    :    S5   :    S6   :
--------------------------------------------------------------

 35              24 23              12 11                  0
--------------------------------------------------------------
:         T1        :         T2         :        T3         :
--------------------------------------------------------------

 35                        18 17                           0
--------------------------------------------------------------
:              H1              :              H2             :
--------------------------------------------------------------
```

3. CONTROL REGISTERS ARE INDICATED BY THE FOLLOWING:

   A0, A1, ETC. - ACCUMULATORS
   X0, X1, ETC. - INDEX REGISTERS
   R0, R1, ETC. - R REGISTERS

4. WHEN REFERENCING AN EXTERNALLY DEFINED EXECUTIVE SYSTEM SYMBOL THE LAST CHARACTER IS ALWAYS THE $. PROCEDURE NAMES USE THE $ AS THEIR SECOND CHARACTER. THEREFORE, IT IS RECOMMENDED THAT THE USER NOT USE THE $ IN HIS SYMBOLS.

5. 1108 ASSEMBLER MNEMONICS ARE USED WHENEVER REFERENCES ARE MADE TO MACHINE INSTRUCTIONS.

6. USER PROGRAMS ARE NORMALLY PROVIDED THE OPTION OF USING ONE OF TWO SETS OF CONTROL REGISTERS. THE MINOR SET CONSISTS OF X11, A0 THRU A5, AND R1 THRU R3, AND THE MAJOR SET CONSISTS OF ALL A, X AND R REGISTERS WITH THE EXCEPTION OF R0. INTERRUPT ACTIVITIES ARE ALWAYS ASSIGNED THE MINOR SET.

7. 1108 ASSEMBLER SYNTACTIC RULES ARE USED FOR NUMBER REPRESENTATION (AN OCTAL VALUE HAS A PRECEDING ZERO).

4. COMPONENTS OF THE EXECUTIVE SYSTEM

THE UNIVAC 1108 EXECUTIVE SYSTEM IS COMPOSED OF MANY DIFFERENT ROUTINES, EACH OF WHICH PERFORM SPECIFIC FUNCTIONS. THESE ROUTINES ARE ORGANIZED INTO SEVERAL SEPARATE GROUPS WHICH ARE THE BASIS OF DISCUSSION IN SUBSEQUENT SECTIONS OF THIS MANUAL. FOR INTRODUCTORY PURPOSES, A BRIEF DESCRIPTION OF EACH COMPONENT GROUP FOLLOWS.

4.1. SUPERVISOR

THE SUPERVISOR CONTROLS THE SEQUENCING, SETUP, AND EXECUTION OF ALL RUNS. AMONG THOSE ROUTINES INCLUDED WITHIN THE SUPERVISOR ARE THE SCHEDULING ROUTINES, INTERRUPT PROCESSING ROUTINES, TIMING ROUTINES, AND ACCOUNTING ROUTINES.

4.2. EXECUTIVE REQUESTS

EXECUTIVE REQUESTS ARE ENTRANCES INTO THE EXECUTIVE SYSTEM WHICH PROVIDE FUNCTIONS FOR A USER PROGRAM. DEPENDING ON THE FUNCTION, IT MAY BE PERFORMED ASYNCHRONOUSLY, SYNCHRONOUSLY, OR IMMEDIATELY. IF IT IS NOT AN IMMEDIATE REQUEST, A QUEUE IS MAINTAINED.

4.3. SYMBIONTS

SYMBIONTS PROVIDE THE INTERFACE BETWEEN THE PRIMARY UNIT RECORD EQUIPMENT AND THE USER PROGRAM. THESE ROUTINES ARE REFERENCED BY USING EXECUTIVE REQUESTS FOR INPUT AND OUTPUT. INPUT AND OUTPUT DATA ARE BUFFERED ON THE MASS STORAGE DEVICES.

4.4. INPUT-OUTPUT DEVICE HANDLERS

THE INPUT/OUTPUT HANDLERS ARE RESPONSIBLE FOR CONTROLLING THE ACTIVITIES OF ALL I/O CHANNELS AND PERIPHERAL EQUIPMENT ATTACHED TO THE UNIVAC 1108. THESE DEVICE HANDLERS PROVIDE THE USER WITH A FULL CAPABILITY OF PERIPHERAL DEVICE OPERATIONS.

4.5. OPERATOR COMMUNICATIONS

THE COMMUNICATIONS SECTION OF THE EXECUTIVE SYSTEM HANDLES ALL COMMUNICATIONS BETWEEN THE OPERATOR AND THE OPERATING PROGRAMS. THIS COMMUNICATION TAKES PLACE VIA THE COMPUTER KEYBOARD AND ON-LINE PRINTER ON THE CONSOLE CHANNEL. NEITHER THE KEYBOARD NOR THE CONSOLE PRINTER CAN BE ASSIGNED TO OPERATING PROGRAMS.

4.6. FILE CONTROL SYSTEM

THE FILE SUPERVISOR CONTROLS THE CREATION AND MAINTENANCE OF ALL PROGRAM AND DATA FILES. IT ALSO MAINTAINS AN UP-TO-DATE MASTER DIRECTORY OF ALL FILES CATALOGUED IN THE SYSTEM AND THE AVAILABILITY OF ALL MASS STORAGE.

4.7. DATA HANDLING

THE DATA HANDLING ROUTINES ARE DESIGNED TO PROCESS A WIDE VARIETY OF FILE FORMATS

USING A GENERAL TECHNIQUE. FEW RESTRICTIONS ARE PLACED ON THE FORMATS ACCEPTABLE TO THE SYSTEM.

FILES MAY BE PROCESSED AT THE ITEM OR BLOCK LEVELS WITH GENERAL DISREGARD FOR THE PHYSICAL CHARACTERISTICS OF THE I/O DEVICE ASSIGNED. DATA IS PRESENTED OR ACCEPTED, RANDOMLY OR SEQUENTIALLY, ON REQUEST OF THE USER THEREBY PROVIDING COMPLETE OPERATIONAL FLEXIBILITY FOR EFFICIENT FILE MANIPULATION.

4.8. FILE UTILITY ROUTINES

TO AID THE USER IN THE MANIPULATION OF PROGRAM AND DATA FILES, A SET OF FILE UTILITY ROUTINES IS PROVIDED BY THE EXECUTIVE SYSTEM. THESE ROUTINES PERFORM A VARIETY OF FUNCTIONS FOR SYSTEM AND USER DATA FILE MAINTENANCE.

4.9. AUXILIARY PROCESSORS

A SET OF AUXILIARY PROCESSORS IS INCLUDED IN THE EXECUTIVE SYSTEM. THESE PROCESSORS COMPLEMENT THE SOURCE LANGUAGE PROCESSORS SUCH AS FORTRAN. THIS SET OF PROCESSORS INCLUDES THE COLLECTOR FOR LINKING RELOCATABLE SUBPROGRAMS, AND THE PROCEDURE DEFINITION PROCESSOR FOR INSERTING AND MODIFYING ASSEMBLER, COBOL, OR FORTRAN PROCEDURE DEFINITIONS IN A PROGRAM-FILE.

4.10. PROCESSOR INTERFACE ROUTINES

THE PROCESSOR INTERFACE ROUTINES PROVIDE A SIMPLE, STANDARD INTERFACE FOR ALL PROCESSORS WITHIN THE SYSTEM. COMPLETE FACILITIES ARE PROVIDED FOR THE INPUT OF SOURCE-LANGUAGE STATEMENTS AND THE OUTPUT OF THE RESULTING RELOCATABLE BINARY CODE.

4.11. THE DIAGNOSTIC SYSTEM

A COMPREHENSIVE DIAGNOSTIC SYSTEM IS AVAILABLE WITHIN THE 1108 EXECUTIVE SYSTEM TO AID THE CHECKOUT OF USER PROGRAMS. COMMANDS ARE AVAILABLE WHICH CAN TRIGGER SNAPSHOT DUMPS AT THE TIME OF COMPILATION OR COLLECTION OF A USER ROUTINE. POST-MORTEM DUMPS ARE ALSO AVAILABLE THROUGH AN EXECUTIVE CONTROL STATEMENT.

4.12. SYSTEM SETUP

THE SYSTEM GENERATION ROUTINE PROVIDES THE MEANS OF GENERATING AND MAINTAINING A SYSTEM TAILORED TO THE PARTICULAR NEEDS OF EACH INSTALLATION.

4.13. UTILITY ROUTINES

INCLUDED WITHIN THE UTILITIES SECTION OF THE EXECUTIVE SYSTEM ARE DIAGNOSTIC ROUTINES, FILE CONVERSION ROUTINES, AND OTHER PROGRAMMING AIDS.

5. EXECUTIVE CONTROL LANGUAGE

5.1. PURPOSE

CONTROL OF THE OPERATING ENVIRONMENT ON THE UNIVAC 1108 IS ACCOMPLISHED THROUGH A SET OF CONTROL STATEMENTS. THESE STATEMENTS DIRECT THE EXECUTIVE IN SCHEDULING, ASSIGNMENT OF FACILITIES, AND IN THE DISPOSITION OF PROGRAM AND DATA FILES. THE LANGUAGE IS DESIGNED IN A COMPACT AND DESCRIPTIVE MANNER TO FACILITATE USE AND YET PROVIDE ALL OF THE FEATURES AND FUNCTIONS OF A MODERN EXECUTIVE SYSTEM. STATEMENTS MAY BE EASILY ADDED, MODIFIED, OR DELETED IN THE SYSTEM DEFINED SET.

5.2. STATEMENTS

5.2.1. GENERAL CONTENT

THE BASIC FORMAT OF THE EXECUTIVE CONTROL STATEMENTS IS QUITE SIMPLE AND IS AMENABLE TO A LARGE NUMBER OF INPUT DEVICES. STATEMENTS ARE NOT RESTRICTED TO A CARD-IMAGE FORMAT; HENCE, THEY MAY BE OF VARIABLE LENGTHS. EACH STATEMENT CONSISTS OF A RECOGNITION CHARACTER IN COLUMN ONE, FOLLOWED BY A COMMAND WHICH CATEGORIZES THE STATEMENT, FOLLOWED BY A VARIABLE NUMBER OF SPECIFICATIONS FIELDS, AND CONCLUDED BY A COMMENTS FIELD. THE RECOGNITION CHARACTER IS A MASTER SPACE(@), WHICH IS A MULTIPLE (7-8) CARD PUNCH OR ITS EQUIVALENT FOR OTHER TYPES OF INPUT DEVICES (SUCH AS THE TTY35 WHERE THE POUND-SIGN (#) IS USED). THE END OF A STATEMENT IS SIGNIFIED BY THE END OF A CARD FOR CARD-IMAGE INPUT, OR BY A CARRIAGE RETURN OR ITS EQUIVALENT FOR OTHER TYPES OF INPUT DEVICES.

EXECUTIVE CONTROL STATEMENTS ARE ALWAYS LOGGED IN A BATCH RUN'S PRINT FILE. IF A CONTROL STATEMENT IS IN ERROR, THE DIAGNOSTIC IS PRINTED IMMEDIATELY FOLLOWING THE STATEMENT.

5.2.2. STATEMENT FORMAT

THE GENERAL FORMAT OF AN EXECUTIVE CONTROL STATEMENT IS:

```
@LABEL:COMMAND,OPTIONS  SPEC1,SPEC2,...,SPECN     COMMENT
```

THE FOLLOWING GIVES A DESCRIPTION OF EACH OF THE 'FIELDS' OF THE EXECUTIVE CONTROL STATEMENT AS WELL AS FORMAT AND CONTINUATION RULES.

5.2.2.1. LABEL FIELD

THE LABEL FIELD NEED NOT APPEAR BUT MAY BE USED TO NAME A CONTROL STATEMENT. THE LABEL IS LIMITED TO SIX CHARACTERS FROM THE ALPHANUMERIC SET(A...Z,0...9), THE FIRST OF WHICH MUST BE AN ALPHABETIC. IF A LABEL IS SPECIFIED, IT MUST BE IMMEDIATELY FOLLOWED BY THE COLON(:). A LABEL IS USED ONLY WHEN DYNAMIC ADJUSTMENT OF THE CONTROL STREAM IS REQUIRED. THE DISCUSSION OF THEIR USE IS DEFERRED TO THE SECTION ENTITLED 'CONDITIONAL STATEMENTS'.

5.2.2.2. COMMAND FIELD

THE COMMAND FIELD MUST ALWAYS BE SPECIFIED AS IT DETERMINES THE STATEMENT'S BASIC OPERATION. THE COMMAND IS LIMITED TO SIX CHARACTERS FROM THE ALPHANUMERIC SET (A...Z,0...9), THE FIRST OF WHICH MUST BE ALPHABETIC. FOR CERTAIN CONTROL STATEMENTS, THE OPTIONS FIELD, WHICH IS AN APPENDAGE TO THE COMMAND FIELD, IS RECOGNIZED. WHEN THE OPTIONS FIELD IS SPECIFIED, THE COMMAND FIELD TERMINATOR IS THE COMMA(,). HOWEVER, IF AN OPTIONS FIELD IS NOT SPECIFIED BLANK( ) IS THE COMMAND

FIELD TERMINATOR.

5.2.2.3. OPTIONS FIELD

THE OPTIONS FIELD PROVIDES THE USER WITH THE ABILITY TO SPECIFY CERTAIN OPTIONS, IN THE FORM OF UNSEQUENCED ALPHABETIC CHARACTERS, TO THE PARTICULAR PROCESSOR ADDRESSED IN THE COMMAND FIELD OR TO A SPECIFIC PROGRAM AS IT IS EXECUTED. ON SOME CONTROL STATEMENTS THE OPTIONS FIELD CAN BE BROKEN INTO SUB-FIELDS, EACH OF WHICH IS SEPARATED BY A SLASH(/). A BLANK CHARACTER OR A SERIES OF BLANK CHARACTERS SEPARATES THE COMMAND OR OPTIONS FIELD FROM THE SPECIFICATIONS FIELDS.

5.2.2.4. SPECIFICATIONS FIELDS

THE SPECIFICATIONS FIELDS OF AN EXECUTIVE CONTROL STATEMENT ARE SEPARATED BY COMMAS AND ARE SPECIFIED BY THE USER AS DICTATED BY HIS REQUIREMENTS. THE CONTENT OF EACH SPECIFICATION FIELD, THE NUMBER OF SPECIFICATION FIELDS, AND WHETHER EACH IS REQUIRED OR OPTIONAL,VARIES WITH THE COMMAND SELECTED. SPECIFICATION FIELDS, IN TURN, MAY CONTAIN SUBFIELDS THAT ARE SEPARATED BY A SLASH (/). FOR THE MOST PART, THESE SUB-FIELDS ARE OPTIONAL WITHIN A FIELD. THUS, IT IS POSSIBLE TO SPECIFY PARTS OF A FIELD WITHOUT SPECIFYING THE ENTIRE FIELD.

5.2.2.5. LEADING BLANKS

LEADING BLANKS ARE ALLOWABLE FOLLOWING THE RECOGNITION CHARACTER (@), THE COLON (:) IF A LABEL IS SPECIFIED, THE FIELD SEPARATOR (,), AND THE SUB-FIELD SEPARATOR (/). A BLANK IN ANY OTHER POSITION ACTS AS THE SEPARATOR SIGNIFYING THE START OF THE SPECIFICATIONS FIELDS OR COMMENTS FIELD. AN EMPTY FIELD OR SUB-FIELD IS ONE THAT CONTAINS NO CHARACTERS OR ONE OR MORE BLANK CHARACTERS. WHEN ALL REMAINING FIELDS OR SUB-FIELDS ARE EMPTY, THEY MAY BE OMITTED.

5.2.2.6. COMMENTS FIELD

AT LEAST ONE BLANK CHARACTER MUST PRECEDE THE COMMENT FIELD. THE COMMENT ITSELF MAY CONTAIN ANY CHARACTER EXCEPT THE SEMICOLON (;), THE CONTINUATION CHARACTER. THE COMMENT FIELD IS ENDED BY END-OF-CARD OR ITS EQUIVALENT FOR OTHER INPUT DEVICES. THE COMMENT FIELD IS NEVER REQUIRED. IF SPECIFICATIONS FIELDS ARE OMITTED, THE COMMENT FIELD MUST BEGIN WITH A PERIOD (.) FOLLOWED BY A BLANK. THIS IS ALSO TRUE WHEN THE CONTENT OF A SPECIFICATIONS FIELD IS UNRESTRICTED AND VARIABLE IN LENGTH (AS WITH THE @LOG STATEMENT AND @MSG STATEMENT). THE @XQT STATEMENT IS AN EXAMPLE OF A STATEMENT WHERE SPECIFICATIONS ARE POSSIBLE BUT MAY BE OMITTED.

5.2.3. CONTINUATION RULES

IN CERTAIN SITUATIONS, A STATEMENT MAY REQUIRE MORE THAN ONE LINE OR CARD. IN SUCH CASES, CODING OF A SEMICOLON (;) INDICATES CONTINUATION ON THE NEXT CARD OR LINE. A STATEMENT MAY BE SPLIT AT ANY POINT, AFTER THE OPTIONS FIELD, WHERE A LEADING SPACE IS ALLOWABLE OR WITHIN THE COMMENT FIELD. IT IS TREATED LOGICALLY AS A SPACE. CONTINUATION ON THE NEXT LINE CAN BEGIN IN ANY COLUMN, WITH ONE EXCEPTION: A MASTER SPACE CHARACTER (@) SHOULD NOT BE PLACED IN COLUMN ONE ON A CONTINUATION LINE.

5.3. STATEMENT TYPES

5.3.1. GENERAL

THE 1108 EXECUTIVE SYSTEM RECOGNIZES FIVE TYPES OF CONTROL STATEMENTS:

1. ORGANIZATIONAL STATEMENTS;
2. INPUT/OUTPUT SPECIFICATION STATEMENTS;
3. PROCESSOR CALL STATEMENTS;
4. PROGRAM EXECUTION STATEMENTS; AND
5. CONDITIONAL STATEMENTS

EACH STATEMENT IS DISCUSSED INDIVIDUALLY IN SUCCEEDING PARAGRAPHS. THE ORDER OF PRESENTATION IS AS SHOWN IN THE TABLE BELOW.

IN ADDITION, A SUMMARY SHOWING THE RESPECTIVE FORMATS, IS PRESENTED IN THE LAST SECTION OF THIS CHAPTER (SECTION 5.9).

SUMMARY OF EXECUTIVE CONTROL STATEMENTS

STATEMENT TYPE	COMMAND	GENERAL USAGE
--------------	-------	-------------
ORGANIZATIONAL STATEMENTS	@RUN	APPEARS AT THE BEGINNING OF EACH RUN. PROVIDES ACCOUNTING , SCHEDULING, AND ID. INFORMATION.
	@FIN	APPEARS AT THE END OF EACH RUN.
	@LOG	PLACES USER SPECIFIED INFORMATION IN THE SYSTEM LOG.
	@MSG	PLACES A MESSAGE ON THE CENTRAL-SITE CONSOLE TYPEWRITER.
	@HDG	USED TO PLACE A HEADING LINE ON PRINT OUTPUT.
	@ADD	USED TO DYNAMICALLY EXPAND THE RUN STREAM.
	@START	USED TO SCHEDULE THE EXECUTION OF AN INDEPENDENT RUN.
	@SYM	USED TO SCHEDULE NON-STANDARD SYMBIONT ACTION.
	@COL*	USED TO SPECIFY VARIOUS FORMS OF INPUT.
	@CKPT	USED TO ESTABLISH A CHECKPOINT DUMP THAT MAY BE USED FOR RESTART AT SOME FUTURE TIME.
	@RSTRT	USED TO RESTART A RUN AT SOME PREVIOUSLY TAKEN CHECKPOINT.
INPUT/OUTPUT SPECIFICATION STATEMENTS	@ASG	USED TO ASSIGN A PARTICULAR INPUT/OUTPUT DEVICE OR MASS STORAGE FILE TO A RUN. THERE ARE FOUR TYPES OF @ASG STATEMENTS: FASTRAND TAPE DRUM ARBITRARY DEVICE ALSO USED TO CATALOGUE FILES.
	@MODE	USED TO CHANGE THE MODE SETTINGS (DENSITY,PARITY,ETC.) OF A TAPE FILE.
	@CAT	CATALOGUES FASTRAND FORMATTED OR EXISTING TAPE FILES.
	@FREE	USED TO DEASSIGN A FILE AND ITS INPUT/OUTPUT DEVICE OR MASS STORAGE AREA.
	@USE	USED TO SET UP A CORRESPONDENCE BETWEEN INTERNAL AND EXTERNAL FILE NAMES.
	@ELT	INSERTS OR UPDATES A PROGRAM-FILE ELEMENT FROM THE CONTROL STREAM.
	@DATA	USED TO INTRODUCE OR UPDATE A DATA FILE FROM THE CONTROL STREAM.
	@END*	USED TO TERMINATE A DATA FILE.
	@FILE*	USED TO CAUSE THE DIRECT CREATION OF A FILE CONTAINING DATA TAKEN FROM THE CONTROL STREAM.
	@ENDF*	USED TO TERMINATE THE DATA THAT FOLLOWS THE @FILE STATEMENT.
	@QUAL	USED TO DEFINE A STANDARD FILE NAME QUALIFIER

PROCESSOR CALL STATEMENTS	@PROCESSOR	USED TO EXECUTE A PROCESSOR (@COB FOR COBOL COMPILER, @FOR FOR FORTRAN, @ASM FOR ASSEMBLER,ETC.)
PROGRAM EXECUTION STATEMENTS	@MAP	USED TO CALL THE COLLECTOR AND PREPARE AN ABSOLUTE ELEMENT.
	@XQT	USED TO INITIATE THE EXECUTION OF A PROGRAM.
	@EOF*	USED TO SEPARATE DATA WITHIN THE CONTROL STREAM
	@PMD	USED TO TAKE EDITED POST-MORTEM DUMPS OF THE PROGRAM JUST EXECUTED.
CONDITIONAL STATEMENTS	@LABEL:	USED TO ATTACH A LABEL TO AN EXISTING CONTROL STATEMENT.
	@SETC	PLACES A VALUE IN THE 'CONDITION' WORD.
	@JUMP	USED TO BRANCH CONTROL WITHIN THE CONTROL STREAM.
	@TEST	USED TO TEST THE 'CONDITION' WORD IN THE COURSE OF DECIDING THE EFFECTIVE CONTROL STREAM.

* THESE CONTROL STATEMENTS CANNOT HAVE LABELS. THE ASTERISK IS NOT PART OF THE STATEMENT.

5.4. ORGANIZATIONAL STATEMENTS

5.4.1. THE @RUN STATEMENT

THE @RUN STATEMENT MUST BE THE FIRST STATEMENT OF EACH RUN. ITS PURPOSE IS TO IDENTIFY THE RUN AND TO FURNISH CERTAIN PARAMETERS NECESSARY FOR SCHEDULING AND ACCOUNTING PURPOSES. THE FORMAT OF THE @RUN STATEMENT IS:

```
@RUN,PRIORITY/RUN-OPTIONS  RUN-ID,ACCOUNTING,PROJECT,:
     RUNNING-TIME/DEADLINE,PAGES/CARDS,START-TIME
```

ON THE @RUN STATEMENT THE NORMAL OPTIONS FIELD IS DIVIDED INTO TWO SUB-FIELDS SEPARATED BY A SLASH (/). THE FIRST SUB-FIELD SPECIFIES THE 'PRIORITY' OF THE RUN AND THE SECOND SPECIFIES THE 'RUN-OPTIONS'.

THE 'RUN-ID' AND THE 'ACCOUNTING' FIELDS ARE THE ONLY SPECIFICATION FIELDS THAT NEED TO BE SPECIFIED BY THE USER. FOR DEMAND RUNS, THE RUNNING-TIME/DEADLINE AND START-TIME FIELDS ARE NOT HONORED.

5.4.1.1. PRIORITY SUB-FIELD

THE PRIORITY SUB-FIELD CONTAINS AN ALPHABETIC CHARACTER. THE NEARER THE CHARACTER TO THE HEAD OF THE ALPHABET, THE HIGHER THE PRIORITY OF THE RUN.

AT SYSTEM GENERATION TIME, THE FOLLOWING INFORMATION MAY BE SPECIFIED FOR EACH ACCOUNT NUMBER:

1. THE HIGHEST PRIORITY LETTER ALLOWED FOR THIS ACCOUNT.
2. PRIORITY TO USE IF NONE IS SPECIFIED ON THE @RUN STATEMENT.
3. WHETHER OR NOT A 'DEADLINE' SPECIFICATION IS TO BE ALLOWED FOR THIS ACCOUNT. (DEADLINE RUNS MAY FORCE OTHER PROGRAMS TO BE SUSPENDED FROM CORE. IT MAY BE DESIRABLE TO CONTROL THIS INTERRUPTION OF NORMAL SCHEDULING.)

THE ABOVE FEATURES ALLOW THE PARTICULAR INSTALLATION TO EXERCISE CONTROL OVER THE USE OF PRIORITIES AND DEADLINES. IN THE ABSENCE OF THESE SPECIFICATIONS AT GENERATION TIME, BOTH THE 'A' PRIORITY AND DEADLINE ARE ALLOWED. IF THE PRIORITY ON THE @RUN STATEMENT IS HIGHER THAN ALLOWED, IT IS ADJUSTED TO THE 'ALLOWED' LEVEL. IF THE DEADLINE SPECIFICATION IS NOT TO BE ALLOWED FOR A PARTICULAR ACCOUNT IT IS SIMPLY IGNORED IF ENCOUNTERED (SEE SYSTEM GENERATION STATEMENTS - ACCOUNT PRIORITY, SECTION 18.1.3).

IF THE PRIORITY SUB-FIELD IS LEFT BLANK, A PRIORITY CHARACTER IS CHOSEN AS SPECIFIED BY THE PARTICULAR INSTALLATION AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - RUN PRIORITY, AND INITIAL SYSTEM).

THE ALGORITHM USED BY THE EXECUTIVE SYSTEM IN SCHEDULING BATCH RUNS IS AS FOLLOWS:

> ALL RUNS IN THE GROUP HAVING THE HIGHEST PRIORITY LETTER MUST BE INITIATED BEFORE RUNS FROM A LOWER PRIORITY GROUP WILL BE SELECTED. WITHIN THE HIGHEST PRIORITY GROUP, WHERE SELECTION IS TAKING PLACE, THE EXECUTIVE IS FREE TO CHOOSE THE ORDER OF INITIATION, BASED FIRST ON THE FACILITIES AVAILABLE AT THE MOMENT, AND SECOND ON THE ORDER OF RUN SUBMISSION. RUNS WITH A 'DEADLINE' MAY BE SELECTED AT ANY TIME, REGARDLESS OF THEIR PRIORITY.

IT SHOULD BE NOTED THAT THE OPERATOR CAN, VIA AN

UNSOLICITED KEYIN, CHANGE THE PRIORITY, START TIME AND/OR DEADLINE TIME OF ANY RUN; SELECTIVELY HOLD THE SCHEDULING OF RUNS, OBTAIN A SUMMARY OF RUNS THAT HAVE NOT BEEN OPENED; AND REMOVE A RUN FROM THE RUN QUEUE. THESE FEATURES CAN BE USED TO DYNAMICALLY CHANGE THE SCHEDULING PROCESS (SEE THE OPERATOR COMMUNICATIONS SECTION FOR A COMPLETE EXPLANATION OF UNSOLICITED MESSAGES).

THIS ALGORITHM PLACES PART OF THE RESPONSIBILITY FOR PARALLEL RUN PROCESSING IN THE HANDS OF THE PERSON(S) SETTING UP THE PRIORITIES. IT ALLOWS PARTICULAR RUNS TO BE GROUPED UNDER PRIORITY LETTERS SO THAT MINIMUM DELAYS ARE ENCOUNTERED AND BETTER EXTERNAL FACILITY ALLOCATION IS ACHIEVED. A MORE DETAILED EXPLANATION OF THE SCHEDULING MECHANISM IS FOUND WITH THE DISCUSSION OF THE 'COARSE SCHEDULER', SECTION 7.3.4.

FOR DEMAND MODE RUNS THE PRIORITY LETTER IS USED TO RESOLVE CONFLICTS WITH OTHER RUNS IN ACQUIRING FACILITIES. MOST REQUESTS WHILE IN THE DEMAND MODE ARE SATISFIED IMMEDIATELY. A DETAILED EXPLANATION OF 'DEMAND' RUN SCHEDULING IS FOUND WITH THE DISCUSSION OF THE 'DYNAMIC ALLOCATOR' AND 'COARSE SCHEDULER'.

5.4.1.2. RUN-OPTIONS SUB-FIELD

THE RUN-OPTIONS SUB-FIELD MAY BE USED TO PLACE CERTAIN CONSTRAINTS ON THE RUN. THIS FIELD IS NEVER REQUIRED AND WHEN LEFT BLANK NORMAL SYSTEM ACTION OCCURS. THE OPTIONS WHICH CAN BE SPECIFIED ARE AS FOLLOWS:

T TERMINATE THE RUN IF THE ESTIMATED RUNNING TIME IS EXCEEDED. IF A RUNNING TIME IS NOT SPECIFIED ON THE @RUN STATEMENT, THE RUNNING TIME SPECIFICATION AS DEFINED FOR THE INSTALLATION IS USED AS GROUNDS FOR TERMINATION. THIS RUNNING TIME MAY BE ALTERED AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS -CONTINUE OR TERMINATE RUN - STANDARD RUN TIME, AND INITIAL SYSTEM).

P TERMINATE THE RUN IF THE 'PAGES' ESTIMATE IS EXCEEDED FOR PRINTED OUTPUT. THE INSTALLATION DEFINED MAXIMUM IS USED IN THE ABSENCE OF A PAGES ESTIMATE ON THE @RUN STATEMENT (SEE SYSTEM GENERATION STATEMENTS - CONTINUE OR TERMINATE RUN - STANDARD PAGE AND CARD LIMIT, AND INITIAL SYSTEM).

C SAME AS FOR 'PAGES' EXCEPT THAT THE RUN IS TERMINATED IF THE 'CARDS' TO BE PUNCHED ESTIMATE IS EXCEEDED.

S THE RUN IS TO BE PROCESSED IN SEQUENCE WITH THE PREVIOUS RUN INPUT FROM THE SAME DEVICE. A SEQUENCED RUN IS NOT CONSIDERED FOR EXECUTION UNTIL THE PREVIOUS RUN HAS TERMINATED. AT THAT TIME ITS PRIORITY, DEADLINE, AND START-TIME ARE TAKEN INTO CONSIDERATION.

B THIS OPTION SPECIFIES THAT A BATCH RUN IS BEING SUBMITTED FROM A TERMINAL WHERE 'DEMAND' RUNS ARE THE NORMAL CASE. THE SYSTEM IS MADE AWARE AT SYSTEM GENERATION TIME OF THOSE TERMINALS FROM WHICH A DEMAND MODE RUN MAY BE SUBMITTED. FROM THESE TERMINALS, A RUN IS ASSUMED TO BE 'DEMAND' UNLESS DESIGNATED AS 'BATCH' BY THIS OPTION.

D THIS OPTION INDICATES THAT A 'DEMAND' RUN IS BEING SUBMITTED FROM A TERMINAL WHERE 'BATCH' RUNS ARE THE NORMAL CASE. THE SYSTEM IS MADE AWARE AT SYSTEM GENERATION TIME OF THOSE TERMINALS FROM WHICH A

BATCH MODE RUN MAY BE SUBMITTED. FROM THESE TERMINALS, A RUN IS ASSUMED TO BE 'BATCH' UNLESS DESIGNATED AS 'DEMAND' BY THIS OPTION.

5.4.1.3. RUN-ID FIELD

THE RUN-ID (IDENTIFICATION) FIELD MUST BE SPECIFIED TO UNIQUELY IDENTIFY THE RUN TO THE SYSTEM. THIS FIELD IS LIMITED TO A MAXIMUM OF SIX CHARACTERS FROM THE ALPHANUMERIC SET (A...Z,0...9).

IF THE SYSTEM FINDS THAT A RUN BEING SUBMITTED HAS THE SAME RUN-ID AS A PREVIOUS RUN THAT HAS NOT FINISHED EXECUTION, THE EXECUTIVE WILL ASSIGN A UNIQUE RUN-ID TO THE RUN, NOTIFY THE OPERATOR OF THE CHANGE, AND CONTINUE PROCESSING THE RUN. THE NEW ID IS USED FOR ALL OPERATOR-EXECUTIVE COMMUNICATIONS CONCERNING THE RUN. NORMALLY, THE NEW ID IS ESTABLISHED BY ADDING AN ALPHABETIC CHARACTER IF THE SUBMITTED ID IS LESS THAN SIX CHARACTERS. OR, IF THE SUBMITTED ID IS SIX CHARACTERS, THE LEFT MOST CHARACTER IS REPLACED. THE EXECUTIVE RESERVES THE RIGHT TO MAKE MORE DRASTIC CHANGES TO THE SUBMITTED ID IN ORDER TO ESTABLISH UNIQUENESS. BOTH THE ORIGINAL RUN-ID AND THE ASSIGNED ID ARE OUTPUT ONCE ON ALL OUTPUT FILES SUCH AS THE SYSTEM LOG AND THE PRINTER LISTING WHICH ARE GENERATED BY THE SYSTEM.

5.4.1.4. ACCOUNTING FIELD

THE ACCOUNTING FIELD IS USED TO SPECIFY ACCOUNTING CODES, AND IT MUST BE FILLED. THE FIELD CONTAINS FROM ONE TO TWELVE CHARACTERS FROM THE SET A...Z, 0...9, ., AND -. THE SYSTEM DESIGN IS FLEXIBLE ENOUGH THAT SPECIFIC INSTALLATIONS CAN EASILY ADD ADDITIONAL FIELDS (OR SUB-FIELDS) FOR IDENTIFICATION OR ACCOUNTING PURPOSES; ALTERNATIVELY, THEY MAY ELECT TO USE THE PROJECT FIELD FOR THE ACCOUNTING INFORMATION.

A SET OF ALLOWABLE ACCOUNT NUMBERS IS SUPPLIED BY THE USER INSTALLATION AND INCORPORATED INTO THE SYSTEM AT GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - ACCOUNT PRIORITY AND DEADLINE). A RUN IS ACCEPTED IMMEDIATELY IF ITS ACCOUNT NUMBER IS KNOWN TO THE SYSTEM. IF NOT, THE OPERATOR IS NOTIFIED OF THE RUN ID AND THE GIVEN ACCOUNT NUMBER, IF ANY, AND GIVEN THE CHOICE OF ABORTING THE RUN OR ACCEPTING THE ACCOUNT NUMBER. IF THE NEW NUMBER IS ACCEPTED, IT IS ADDED TO THE PERMANENT SET.

5.4.1.5. PROJECT FIELD

THE PROJECT FIELD CLASSIFIES THE RUN FOR ACCOUNTING PURPOSES AND PERMITS INSERTION OF THE IMPLIED QUALIFICATION OF FILE NAMES WHEN NO SPECIFIC QUALIFICATION IS GIVEN (SEE @QUAL CONTROL STATEMENT, SECTION 5.5.11.). THIS FIELD IS LIMITED TO 12 CHARACTERS FROM THE SET A...Z, 0...9, -, AND $. THE PROJECT SPECIFICATION IS OPTIONAL IN THE SENSE THAT AN EMPTY FIELD IS TREATED AS A PROJECT-ID CONSISTING OF 12 SPACE CHARACTERS. FOR A MORE DETAILED EXPLANATION OF USE OF THIS FIELD AS A FILE QUALIFIER, SEE THE 'INPUT/OUTPUT SPECIFICATION STATEMENTS', SECTION 5.5.

5.4.1.6. RUNNING TIME/DEADLINE FIELD

USE OF THE RUNNING TIME/DEADLINE FIELD IS OPTIONAL; IT PROVIDES ADDITIONAL SCHEDULING INFORMATION. THE RUNNING TIME SUB-FIELD SPECIFIES THE PROGRAMMER ESTIMATED NUMBER OF MINUTES OF CENTRAL PROCESSOR UNIT (CPU) TIME REQUIRED FOR THE RUN. IF THIS TIME IS EXCEEDED, AS MEASURED BY THE TIME THAT THE RUN HAS CONTROL OF THE CPU, THE OPERATOR IS NOTIFIED AND MAY OPTIONALLY TERMINATE THE RUN, AS GUIDED BY THE INSTALLATION'S OPERATING PHILOSOPHY OR THE PROGRAMMER'S INSTRUCTIONS. SOME INSTALLATIONS MAY REQUIRE THAT A RUN BE IMMEDIATELY TERMINATED IF THE ESTIMATED RUNNING TIME IS EXCEEDED. THIS FEATURE IS PROVIDED AS AN OPTION AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - CONTINUE OR TERMINATE RUN, AND INITIAL SYSTEM).

THE DEADLINE SUB-FIELD IS USED BY THE PROGRAMMER TO SPECIFY THE TIME OF DAY OR AN ELAPSED TIME, FROM TIME OF RUN SUBMISSION, BY WHICH HIS RUN MUST BE COMPLETED. THE DEADLINE IS BASED ON A 24 HOUR CLOCK. IF A 'D' PRECEDES THE TIME SPECIFICATION, IT

IS TAKEN AS THE TIME OF DAY; OTHERWISE, IT IS TAKEN AS THE ELAPSED TIME FROM RUN SUBMISSION. THE TIME IS GIVEN IN HOURS AND MINUTES AND CANNOT EXCEED 2400 (24 HOURS, 0 MINUTES). FOR EXAMPLE, A SPECIFICATION OF 'D910' WOULD BE TAKEN AS 9:10 A.M.

THE ENTIRE RUNNING TIME/DEADLINE FIELD MAY BE OMITTED; OR JUST THE DEADLINE SUB-FIELD MAY BE OMITTED. IF OMITTED, THE RUNNING TIME SPECIFICATION SUPPLIED BY THE INSTALLATION AT SYSTEM GENERATION TIME IS USED AS THE PROGRAMMER'S ESTIMATE, AND IT IS ASSUMED THAT THE RUN HAS NO SPECIFIC DEADLINE (SEE SYSTEM GENERATION STATEMENTS - STANDARD RUN TIME, AND INITIAL SYSTEM). THE DEADLINE SPECIFICATION WILL NOT BE HONORED UNLESS A RUNNING TIME ESTIMATE IS PRESENT ON THE @RUN STATEMENT.

THE DEADLINE SPECIFICATION SHOULD NOT BE USED IN AN ARBITRARY MANNER. AS POINTED OUT EARLIER, ITS USE MAY BE ALLOWED ONLY FOR PARTICULAR ACCOUNTS.

IF A DEADLINE CANNOT BE MET VIA NORMAL SCHEDULING, THE SYSTEM WILL TAKE THE NECESSARY ACTION TO INSURE THE REQUIRED COMPLETION TIME, IF POSSIBLE. THIS ACTION MAY DEGRADE THE GENERAL OPERATION OF THE SYSTEM AS FAR AS MULTI-PROGRAMMING AND SYSTEM-OVERHEAD ARE CONCERNED. ON THE OTHER HAND, IF DEADLINE SPECIFICATIONS ARE SUCH THAT IN GENERAL THEY DO NOT BECOME CRITICAL, THE SYSTEM WILL SIMPLY ORDER THE RUNS IN A MANNER TO BEST MEET THE DEADLINES AND EXECUTE THEM IN THE NORMAL MANNER. A DEADLINE RUN IS LEFT AT ITS GIVEN PRIORITY LEVEL UNTIL SUCH TIME AS IT BECOMES NECESSARY TO CONSIDER IT FOR HIGHER PRIORITY EXECUTION, IF AT ALL. IF THE DEADLINE SPECIFICATION ALLOWS THE TIME, THE RUN IS ELEVATED TO A HIGHER PRIORITY IN TIME TO ALLOW IT TO FINISH ON TIME WITH NO SPECIAL ACTION IN THE WAY OF SUSPENDING OTHER RUNS, BARRING OTHER DEADLINES (UNFORESEEN OR OTHERWISE). IN SUMMARY, THE DEADLINE FACILITY CAN BE USED SIMPLY TO INSURE A MAXIMUM TURN-AROUND FOR A RUN, OR IT CAN BE USED TO INSURE COMPLETION OF A 'RUSH' JOB. THE SECTION ON THE 'COARSE SCHEDULER' CAN BE SEEN FOR ADDITIONAL INFORMATION ON THE SCHEDULING OF DEADLINE JOBS.

5.4.1.7. PAGES SUB-FIELD

USE OF THE PAGES SUB-FIELD IS OPTIONAL; IT PROVIDES THE SYSTEM WITH A PAGE-NUMBER ESTIMATE OF PRINTED OUTPUT THAT THE PROGRAMMER IS EXPECTING. IF THIS FIELD IS OMITTED, THE NUMBER SET BY THE INSTALLATION AT SYSTEM GENERATION TIME IS ASSUMED (SEE SYSTEM GENERATION STATEMENTS - STANDARD PAGE AND CARD LIMIT, AND INITIAL SYSTEM). IF THE PAGE ESTIMATE IS EXCEEDED, THE OPERATOR IS NOTIFIED. A SYSTEM GENERATION PARAMETER CAN SPECIFY THAT THE RUN BE AUTOMATICALLY TERMINATED WHEN THE ESTIMATE IS EXCEEDED (SEE SYSTEM GENERATION STATEMENTS - CONTINUE OR TERMINATE RUN, AND INITIAL SYSTEM).

5.4.1.8. CARDS SUB-FIELD

THE USE OF THE CARDS SUB-FIELD IS IDENTICAL TO THE PAGES SUB-FIELD EXCEPT THAT IT APPLIES TO THE NUMBER OF PUNCHED CARDS EXPECTED DURING THE RUN, RATHER THAN THE NUMBER OF PRINTED PAGES.

5.4.1.9. START-TIME FIELD

THE START-TIME FIELD IS USED TO SPECIFY (DELAY) THE TIME AT WHICH THE RUN WILL BE CONSIDERED FOR EXECUTION. IN THE ABSENCE OF A START-TIME SPECIFICATION, WHICH IS THE NORMAL CASE, THE RUN IS CONSIDERED FOR EXECUTION IMMEDIATELY AND EXECUTED ACCORDING TO ITS PRIORITY. WHEN A START-TIME IS SPECIFIED, THE RUN IS NOT INCLUDED IN THOSE AVAILABLE FOR EXECUTION UNTIL THE START-TIME HAS ARRIVED. AT THAT TIME IT IS CONSIDERED FOR EXECUTION ACCORDING TO THE GIVEN PRIORITY. THE START-TIME IS SPECIFIED IN EXACTLY THE SAME MANNER AS THE DEADLINE (SEE DESCRIPTION OF DEADLINE SPECIFICATION, 5.4.1.6.).

THE START-TIME FIELD ALLOWS A RUN TO BE SUBMITTED WITH THE ASSURANCE THAT IT WILL NOT BE EXECUTED PRIOR TO THE GIVEN TIME. THIS FEATURE IS DESIRABLE WHEN INPUT DATA IS NOT YET READY BUT WILL BE BY START-TIME. IT IS ALSO DESIRABLE TO HAVE CERTAIN TYPES OF RUNS (PRODUCTION,UTILITY,ETC.), EXECUTED DURING CERTAIN PERIODS OF THE DAY.

IF A DEADLINE SPECIFICATION IS ALSO GIVEN ON THE @RUN STATEMENT, IT IS NOT INTERPRETED UNTIL THE START-TIME HAS BEEN REACHED. IN OTHER WORDS, THE START-TIME IS

TAKEN AS THE TIME OF RUN SUBMISSION IN CONSIDERING THE DEADLINE AND PRIORITY.

5.4.1.10. RUN RESTRICTIONS

THE FOLLOWING IS A SUMMARY OF EXECUTIVE ACTION CONCERNING RUN RESTRICTIONS:

AT SYSTEM GENERATION TIME THE INSTALLATION HAS THE CAPABILITY TO SPECIFY MAXIMUM RUNNING TIME, PAGE COUNT AND/OR PUNCH COUNT FOR ALL RUNS ENTERING THE SYSTEM. IT CAN ALSO BE SPECIFIED WHETHER OR NOT THE RUNS SHOULD BE TERMINATED IF THE SPECIFICATION IS EXCEEDED. (IF ANY OF THE ABOVE SYSTEM GENERATION PARAMETERS ARE NOT SPECIFIED, A CORRESPONDING SYSTEM STANDARD WILL BE USED.)

AT RUN SUBMISSION TIME THE USER HAS THE CAPABILITY TO SPECIFY ESTIMATED RUNNING TIME, PAGE COUNT AND/OR PUNCH COUNT FOR THAT PARTICULAR RUN. THIS SPECIFICATION(S) TAKES PRECEDENCE OVER THE MAXIMUM(S) SET AT GENERATION TIME. THE USER CAN FURTHER SPECIFY, AS OPTIONS FOR EACH OF THE ABOVE, WHETHER OR NOT THAT PARTICULAR RUN SHOULD BE TERMINATED IF THE SPECIFICATION(S) IS EXCEEDED. THESE TERMINATION OPTIONS ARE MEANINGFUL ONLY IF AUTOMATIC TERMINATION ON REACHING A MAXIMUM WAS NOT SPECIFIED AT SYSTEM GENERATION TIME BY THE INSTALLATION MANAGER. (THE TERMINATION SPECIFIED AT GENERATION TIME IMPLIES THE CORRESPONDING OPTION BE SET ON ALL RUNS.)

IN THE ABSENCE OF A TERMINATION INDICATOR (BY INSTALLATION MANAGER OR OPTION ON @RUN STATEMENT), THE OPERATOR IS SIMPLY NOTIFIED.

5.4.1.11. @RUN STATEMENT EXAMPLES

CONSIDER THE FOLLOWING @RUN STATEMENT EXAMPLES:

```
(1) @RUN  R231,03412,CAPER,10/100     J. JONES
```

THE OPTIONS FIELD IS NOT USED, MEANING THAT THE PRIORITY AS SET BY THE EXECUTIVE WILL BE SATISFACTORY AND THAT RUN-OPTIONS ARE NOT REQUIRED. THIS IS RUN R231 OF PROJECT CAPER. EXPENSES INCURRED BY THIS RUN ARE TO BE CHARGED AGAINST CODE 03412. THE ESTIMATED RUNNING TIME IS TEN MINUTES AND RESULTS ARE EXPECTED WITHIN ONE HOUR AFTER RUN SUBMISSION. THE ANTICIPATED NUMBER OF PAGES AND CARDS IS SET BY THE SYSTEM. THE COMMENT FIELD CONTAINS INFORMATION 'J. JONES'.

```
(2) @RUN,C/P R231, 03412, CAPER,, 300
```

THE PRIORITY CODE IS 'C'. A RUN-OPTION OF 'P' SPECIFIES THAT THE RUN SHOULD BE TERMINATED IF MORE THAN 300 PAGES ARE PRODUCED. THE RUNNING TIME IS NOT SPECIFIED.

```
(3) @RUN,A 201,90431010,EXODUS1,10,/50,D830
```

HIGHEST PRIORITY IS TO BE USED FOR RUN 201 ON THE EXODUS1 PROJECT. THE RUNNING TIME IS ESTIMATED AT TEN MINUTES. NO MORE THAN 50 CARDS ARE EXPECTED. THE RUN WILL NOT BE CONSIDERED FOR EXECUTION UNTIL 8:30AM.

```
(4) @RUN,E/TCS Z,A-1396,SUPER, 20/230,/80
```

RUN Z OF PROJECT SUPER IS TO BE PROCESSED AFTER THE COMPLETION OF THE PREVIOUS RUN (INPUT ON SAME DEVICE) AND HAS A PRIORITY OF 'E'. THE RUN MUST BE COMPLETED WITHIN 2.5 HOURS AFTER COMPLETION OF THE PREVIOUS RUN. THE RUN WILL BE TERMINATED ('T' RUN-OPTION) IF MORE THAN 20 MINUTES OF CPU TIME ARE NEEDED. IF

AN ATTEMPT IS MADE TO PUNCH MORE THAN 80 CARDS, THE RUN WILL BE TERMINATED ('C' RUN-OPTION).

5.4.2. THE @FIN STATEMENT

THE @FIN STATEMENT IS USED TO SIGNAL THAT THE END-OF-RUN HAS BEEN REACHED. IT IS REQUIRED WITH ALL RUNS AND MUST APPEAR AS THE LAST STATEMENT. IT IS NEVER PASSED AS A DATA IMAGE FOR @ELT, @DATA, OR @FILE. THIS STATEMENT CANNOT BE CONTINUED ON A SECOND CARD OR LINE.

THE @FIN STATEMENT'S FORMAT IS:

```
    @FIN                          COMMENT
```

WHEN THE @FIN STATEMENT IS ENCOUNTERED BY THE COARSE SCHEDULER, THE ACCOUNTING ROUTINES ARE ENTERED AND ALL REMAINING FACILITIES, TEMPORARY FILES, AND CORE SPACE ARE RELEASED.

5.4.3. THE @LOG STATEMENT

THE @LOG STATEMENT PROVIDES THE USER WITH A MEANS OF ENTERING INFORMATION INTO THE SYSTEM LOG. THE @LOG STATEMENT'S FORMAT IS:

```
    @LOG    INFORMATION              . COMMENT
```

THE INFORMATION FIELD IS VARIABLE IN LENGTH WITH A MAXIMUM OF 132 CHARACTERS ALLOWED. THE FIRST NON-BLANK CHARACTER IS THE BEGINNING OF INFORMATION AND THE END-OF-INFORMATION IS THE LAST CHARACTER PRIOR TO THE END-OF-LINE, THE COMMENT FIELD, OR THE 132 CHARACTER MAXIMUM, WHICHEVER OCCURS FIRST.

WHEN A @LOG STATEMENT IS ENCOUNTERED BY THE EXECUTIVE SYSTEM, IT EXTRACTS THE INFORMATION, PREFIXES IT WITH PROGRAM IDENTIFICATION, DATE AND TIME, AND OUTPUTS THIS INFORMATION TO THE TEMPORARY MASTER RUN LOG.

THE SEMICOLON (;) IS USED AS A CONTINUATION CHARACTER; THEREFORE, IT CANNOT BE PART OF THE INFORMATION. THE CHARACTER SEQUENCE SPACE-PERIOD-SPACE (.) IS NOT ALLOWED AS PART OF THE INFORMATION BECAUSE THIS SEQUENCE DENOTES THE START OF THE COMMENT FIELD.

CONSIDER THE FOLLOWING @LOG STATEMENT EXAMPLE:

```
    @LOG TRANSPORT PROB. NO. 128     . REVISED MAY 1
```

THE INFORMATION FIELD IS 28 CHARACTERS IN LENGTH AND THE COMMENT IS 'REVISED MAY 1'.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @LOG FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN SECTION 8, 'EXECUTIVE REQUEST FUNCTIONS'.

5.4.4. THE @MSG STATEMENT

THE @MSG CONTROL STATEMENT IS USED TO TYPE A MESSAGE ON THE CENTRAL SITE CONSOLE TYPEWRITER. IT HAS THE FORM:

```
    @MSG,OPTIONS    MESSAGE     . COMMENT
```

THE MESSAGE HAS A MAXIMUM LENGTH OF 50 CHARACTERS AND THE FORMAT RULES AND RESTRICTIONS ARE IDENTICAL TO THOSE OF THE @LOG STATEMENT (SEE 5.4.3). THE @MSG STATEMENT CAN BE USED TO DIRECT THE OPERATOR IN SUCH AREAS AS DISPOSAL OF OUTPUT, ABNORMAL OR UNDOCUMENTED PROCEDURES, ETC.

THE MESSAGE IS PREFACED BY THE RUN-ID WHEN TYPED. THE USER IS FREE TO USE THE CARRIAGE RETURN CHARACTER IN FORMATING THE MESSAGE TO HIS CHOOSING. THE CARRIAGE

RETURN CHARACTER WILL CAUSE BOTH A CARRIAGE RETURN AND A LINE FEED.

THE @MSG STATEMENT MAY CONTAIN THE FOLLOWING OPTIONS:

W CAUSES THE RUN TO BE HELD UNTIL THE OPERATOR RESPONDS TO THE MESSAGE. THE USER MESSAGE IS FOLLOWED BY THE ADDITIONAL MESSAGE 'WAIT'. 'WAIT' INDICATES TO THE OPERATOR THAT THE MESSAGE MUST BE ANSWERED. THE OPERATOR ANSWERS THE MESSAGE WITH 'GO' WHEN HE HAS COMPLIED WITH THE REQUEST. IF THE OPERATOR CANNOT COMPLY, HE MAY ABORT THE RUN VIA THE KEYIN 'X'.

N CAUSES SUPPRESSION OF THE TYPING OF THE MESSAGE ON THE CONSOLE TYPEWRITER. IN THIS CASE THE STATEMENT IS LISTED ON THE PRINTER ONLY. WHEN THE N OPTION IS PRESENT, THE W OPTION IS NOT EFFECTIVE.

THE W OPTION CAN BE USED TO DIRECT THE OPERATOR IN THE LOADING AND GENERAL MANAGEMENT OF PERIPHERAL DEVICES (IN THOSE CASES NOT AUTOMATICALLY TAKEN CARE OF BY THE EXECUTIVE).

THE N OPTION CAN BE USED TO SIMPLY PLACE A MESSAGE ON THE PRINTER OR AS A WAY TO SUPPRESS CONSOLE ACTION WITHOUT REMOVING THE @MSG STATEMENT.

AN EXAMPLE OF THE @MSG CONTROL STATEMENT IS:

```
@MSG   EXPECT 2 REELS OF OUTPUT FOR FILE XYZ
```

ANOTHER EXAMPLE, WHERE THE OPERATOR MUST RESPOND, COULD BE:

```
@MSG,W    IS REMOTE HOOKUP READY?
```

5.4.5. THE @HDG STATEMENT

THIS CONTROL STATEMENT PROVIDES THE USER WITH AN AUTOMATIC MEANS OF PRINTING A HEADING ON EACH SUCCEEDING PAGE OF THE PRINT FILE. THE FORMAT OF THIS STATEMENT IS:

```
@HDG,OPTIONS  HEADING TEXT
```

THE ALLOWABLE OPTIONS ARE:

N- TURN OFF PRINTING OF THE HEADING.
P- BEGIN PAGE NUMBER WITH 'PAGE 1'.
X- DO NOT PRINT DATE OR PAGE COUNT.

THE 'HEADING TEXT' FIELD IS VARIABLE IN LENGTH WITH A MAXIMUM OF 96 CHARACTERS ALLOWED. THIS FIELD IS SEPARATED FROM THE CONTROL FIELD WITH A SPACE, THEREBY ALLOWING LEADING SPACES IN THE TEXT. THE END OF THE TEXT IS DENOTED BY THE LAST CHARACTER PRIOR TO THE END-OF LINE, OR THE COMMENT FIELD, OR THE 96 CHARACTER MAXIMUM, WHICHEVER OCCURS FIRST.

THE HEADING IS PRINTED ON THE SECOND LINE ABOVE LOGICAL PRINT LINE1. IF THIS UPPER MARGIN IS ONE LINE OR NON-EXISTENT, THE HEADING WILL NOT BE PRINTED. THE DATE AND PAGE NUMBER WILL APPEAR TO THE RIGHT OF THE HEADING TEXT. A PAGE COUNT FOR EACH PRINT FILE IS MAINTAINED BY THE PROCESSING SYMBIONT. WHEN HEADING IS SPECIFIED WITHOUT THE 'P' OPTION, THE PAGE COUNT CURRENT TO THE FILE IS USED TO BEGIN PAGE NUMBERING. ANY NUMBER OF @HDG STATEMENTS MAY APPEAR IN THE CONTROL STREAM.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @HDG FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN THE CHAPTER ENTITLED 'SYMBIONTS'

5.4.6. THE @ADD STATEMENT

THE @ADD CONTROL STATEMENT PROVIDES A MEANS OF INSERTING IMAGES INTO THE CONTROL STREAM FROM ANY FILE IN THE SYSTEM DATA FORMAT. THESE FILES MAY CONTAIN DATA OR ANY CONTROL CARDS ALLOWED IN A RUN STREAM. THE FILE BEING ADDED MAY HAVE BEEN CREATED BY THE @DATA STATEMENT, THE @FILE STATEMENT, THE @ELT STATEMENT OR A USER PROGRAM. THE IMAGES IN THE FILE BEING ADDED NEED NOT EXIST UNTIL THE @ADD COMMAND IS EXECUTED. THIS MEANS THAT THE USER IS FREE TO HAVE WORKER PROGRAMS IN THE FIRST PART OF A RUN GENERATE FILES TO BE ADDED LATER IN THE RUN.

THE FORMAT OF THE @ADD CONTROL STATEMENT IS

```
    @ADD   FILENAME    . COMMENTS
```

WHERE 'FILENAME' MAY BE THE EXTERNAL NAME OF THE FILE IF AN ENTIRE FILE IS TO BE ADDED, OR IT MAY BE REPLACED BY THE STANDARD REFERENCE TO AN ELEMENT 'PROGRAM FILE. ELEMENT/VERSION(CYCLE)'.

WHEN THE @ADD CONTROL STATEMENT IS ENCOUNTERED IN A CONTROL STREAM, THE FIRST IMAGE OF THE ADDED FILE REPLACES THE @ADD CONTROL IMAGE. ALL SUBSEQUENT CONTROL STREAM IMAGES WILL BE TAKEN FROM THE ADDED FILE UNTIL THE END OF FILE OR IF AN ELEMENT IS BEING ADDED, UNTIL THE END OF THE ELEMENT IS ENCOUNTERED. FOLLOWING THE END OF THE ADDED FILE, THE CONTROL STREAM IS AUTOMATICALLY RESUMED AT THE IMAGE FOLLOWING THE @ADD STATEMENT.

@ADD STATEMENTS MAY BE NESTED TO ANY LEVEL PROVIDED THERE IS NO ATTEMPT TO ADD A GIVEN FILE (OR ELEMENT) TWICE IN THE SAME NEST. WHEN THIS OCCURS, OR WHEN A NON-EXISTENT FILE IS SPECIFIED, THE RUN IS PLACED IN THE ERROR MODE AND PROCESSING CONTINUES.

THE @ADD FEATURE IS OF PARTICULAR VALUE TO THE REMOTE USER (BATCH OR DEMAND) IN THAT CONTROL STATEMENTS AND/OR DATA CAN BE SUBMITTED ONLY ONCE BUT USED IN MANY SUBSEQUENT RUNS. THE PRESTORED, PARTIAL CONTROL STREAMS CAN BE CORRECTED PRIOR TO THEIR ADDITION BY PLACING CORRECTION LINES FOLLOWING THE @DATA STATEMENT.

THE FOLLOWING LIST OF CONTROL STATEMENTS ARE CONSIDERED ILLEGAL WITHIN AN @ADD FILE:

```
    @RUN
    @COL
    @FILE
```

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @ADD FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN THE CHAPTER ENTITLED 'SYMBIONTS'.

5.4.7. THE @START STATEMENT

THE @START STATEMENT AFFORDS THE USER A MEANS OF SCHEDULING ONE OR MORE RUNS FROM WITHIN A RUN CONTROL STREAM. RUNS TO BE SCHEDULED IN THIS MANNER MUST BE CATALOGUED DATA FILES CREATED BY THE @DATA OR @FILE STATEMENT OR BY A USER PROGRAM. THEY MAY ALSO BE ELEMENTS OF A CATALOGUED PROGRAM-FILE CREATED BY THE @ELT,D STATEMENT. THE RUN FILE AND THE RUN ELEMENT ARE IN SYSTEM DATA FILE (SDF) FORM. FOR THE FORMAT DESCRIPTION SEE 'DATA HANDLING', SECTION 13.

THE @START FEATURE CAN BE USED WHEN ONE RUN MUST GENERATE A DATA FILE FOR INPUT BY ANOTHER. IN FACT, THE GENERATING RUN MAY ELECT TO BUILD A CATALOGUED FILE CONTAINING AN ENTIRE RUN CONTROL STREAM AND THEN CALL FOR IT TO BE SCHEDULED. NOTICE THAT THE @START STATEMENT CAN BE USED TO ALLOW THE PARALLEL PROCESSING OF CERTAIN OPERATIONS, SINCE TASKS FROM DIFFERENT RUNS CAN BE EXECUTED CONCURRENTLY.

IT MAY ALSO BE EMPLOYED BY TYPEWRITER TERMINALS AS A MEANS OF INITIATING A BATCH RUN WHOSE CONTROL STREAM HAS BEEN PREVIOUSLY ENTERED INTO THE SYSTEM AS A DATA FILE, THUS ELIMINATING THE NECESSITY OF RE-TYPING THE REQUIRED CONTROL STATEMENTS. THE @START IS OF PARTICULAR BENEFIT AT THE CENTRAL SITE IN INITIATING PRESTORED UTILITY ROUTINES AND STANDARD PRODUCTION RUNS.

IN ITS SIMPLEST FORM, THE @START STATEMENT'S FORMAT IS:

```
@START  NAME,SET
```

THE 'NAME' FIELD MUST BE EITHER A DATA FILE NAME OR AN ELEMENT NAME IN THE STANDARD FORMAT FOR SYMBOLIC ELEMENT DESCRIPTION (PROGRAM-FILENAME.ELEMENTNAME/VERSION(CYCLE)). THE 'SET' FIELD CAN CONTAIN AN OCTAL NUMBER TO BE 'SET' IN THE CONDITION WORD OF THE RUN BEING SCHEDULED IN ORDER TO DETERMINE THE EFFECTIVE CONTROL STREAM (SEE SECTION ON CONDITIONAL STATEMENTS). THE SET SPECIFICATION IS NEVER REQUIRED. THE REFERENCED STREAM MUST BEGIN WITH A @RUN STATEMENT FOR THIS NEW INDEPENDENT, ASYNCHRONOUS RUN. THE END OF THE FILE OR ELEMENT DENOTES AN IMPLIED @FIN. WHEN SCHEDULING SUCH A RUN, IT IS SOMETIMES DESIRABLE TO BE ABLE TO CHANGE SOME OF THE PARAMETERS ON THE @RUN STATEMENT THAT HEADS A PRESTORED CONTROL STREAM. THE USER MAY WANT TO SUPPLY PARAMETERS SUCH AS THE ACCOUNT NUMBER AND PRIORITY, OR TO SUBSTITUTE AN ENTIRELY NEW @RUN STATEMENT. IT IS QUITE POSSIBLE THAT THE USER WILL WANT TO SUBSTITUTE HIS OWN RUN-ID.

A SUBSTITUTION CAN BE MADE FOR ALL OR ANY PART OF A PRESTORED @RUN STATEMENT BY THE USE OF A MORE COMPLEX @START STATEMENT OF THE FORM:

```
@START,PRIORITY/RUN-OPTIONS  NAME,SET,RUN-ID,ACCOUNTING,;
 PROJECT,RUNNING-TIME/DEADLINE,PAGES/CARDS,START-TIME
```

NOTE THAT THE STATEMENT HAS THE SAME FORMAT AS THE @RUN STATEMENT EXCEPT THAT THE 'NAME' FIELD AND THE 'SET' FIELD PRECEDE THE RUN-ID. ANOTHER NOTABLE DIFFERENCE FROM THE @RUN STATEMENT IS THAT ALL FIELDS ARE OPTIONAL EXCEPT THE NAME FIELD. ALL NON-BLANK FIELDS WILL BE SUBSTITUTED IN PLACE OF THOSE ON THE PRESTORED @RUN STATEMENT.

A SUBSTITUTION IS ALWAYS MADE TO REPLACE THE ACCOUNT NUMBER ON THE PRESTORED @RUN STATEMENT. THE ACCOUNT NUMBER IS TAKEN FROM THE @START STATEMENT IF PRESENT THERE; OTHERWISE, IT IS TAKEN FROM THE @RUN STATEMENT OF THE INITIATING RUN.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @START FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN THE CHAPTER ENTITLED 'EXECUTIVE REQUEST FUNCTIONS'.

THE @START FUNCTION PROVIDES THE MEANS FOR AUTOMATICALLY SCHEDULING RUNS THAT ARE TO BE EXECUTED DAILY. A WORKER PROGRAM LINKING TO THE EXECUTIVE VIA THE START EXECUTIVE REQUEST FUNCTION, CAN SCHEDULE RUNS OVER A TWENTY-FOUR HOUR PERIOD. THE WORKER PROGRAM MUST ALSO SCHEDULE ITSELF TO BE EXECUTED AGAIN IN THE PERIOD COVERED. A SYSTEM GENERATION PARAMETER IS PROVIDED TO DIRECT THE EXECUTIVE TO SCHEDULE THE CONTROL PROGRAM WHENEVER THE SYSTEM IS BOOTSTRAPPED.

THE @START FUNCTION IS ALSO AVAILABLE VIA THE UNSOLICITED 'ST' KEYIN (SEE THE CHAPTER ENTITLED 'OPERATOR COMMUNICATIONS'). THE FORMAT OF THE KEYIN IS IDENTICAL THE THE @START STATEMENT EXCEPT THAT 'ST ' REPLACES THE '@START'. THE SPACE FOLLOWING ST IS NECESSARY TO IDENTIFY THE REQUEST TO THE EXECUTIVE. THE ACCOUNTING FIELD IS REQUIRED SINCE NO RUN INITIATES THE REQUEST.

5.4.8. THE @BRKPT STATEMENT

THE @BRKPT STATEMENT IS USED TO CLOSE OUT ONE PORTION OF THE PRINT$ OR PUNCH$ FILE AND START A NEW PART. IF THE PART OF THE FILE BEING CLOSED IS SYSTEM DEFINED, IT WILL BE AUTOMATICALLY QUEUED FOR THE PROPER SYMBIONT.

THE FORMAT OF THE @BRKPT STATEMENT IS:

```
@BRKPT  FILENAME1/FILENAME2
```

THE FIELDS 'FILENAME1' AND 'FILENAME2' MAY CONTAIN THE WORDS 'PRINT$', 'PUNCH$' OR AN INTERNAL REFRENCE TO A PREVIOUSLY DEFINED FILE. THE FIRST NAME IS FOR THE PART TO BE CLOSED AND THE SECOND IS FOR THE NEW FILE TO BE OPENED. IF IT IS DESIRED TO BREAKPOINT THE PRINT$ OR PUNCH$ FILE WITHOUT SWITCHING TO A USER DEFINED FILE, IT IS ONLY NECESSARY TO USE 'PRINT$' OR 'PUNCH$' AS 'FILENAME1' AND OMIT 'FILENAME2'.

5.4.9. THE @SYM STATEMENT

THE @SYM STATEMENT PROVIDES THE USER WITH THE CAPABILITY OF SELECTING A SYMBIONT, OR CLASS OF SYMBIONTS, TO PRINT OR PUNCH SELECTED FILES. A STANDARD SYSTEM PROCEDURE EXISTS FOR PRINTING AND PUNCHING THOSE FILES PRODUCED WITH THE INTERFACE ROUTINES PRINT$,PRNTA$,PUNCH$ AND/OR PNCHA$ DURING THE COURSE OF A RUN IF THEY RESIDE ON MASS STORAGE. AS THESE FILES ARE COMPLETED, THEY ARE ENTERED INTO THE APPROPRIATE PRINT OR PUNCH QUEUE DETERMINED BY THE RUN'S ASSOCIATED INPUT SOURCE. WHEN A @SYM STATEMENT IS ENCOUNTERED THE SPECIFIED FILE IS ENTERED INTO THE SPECIFIED SYMBIONT QUEUE. ANY TAPE FILE WHICH IS TO BE PRINTED OR PUNCHED, WHETHER OR NOT IT IS PRODUCED WITH ONE OF THE INTERFACE ROUTINES, MUST BE SCHEDULED FOR PROCESSING WITH THE @SYM STATEMENT.

THE FORMAT OF THE @SYM STATEMENT IS

```
@SYM,OPTIONS FILENAME,TYPE,SYMBIONT,FILE-LABELS
```

THE 'OPTIONS' FIELD IS USED TO SPECIFY THE 'MODE OPTIONS' AVAILABLE WITH THE MAGNETIC TAPE HANDLER. THE OPTIONS ARE DEFINED AS:

L LOW DENSITY
M MEDIUM DENSITY
H HIGH DENSITY
E EVEN PARITY
O ODD PARITY
I DECIMAL (TRANSLATE)
HARDWARE IS USED IF AVAILABLE, OTHERWISE THE STANDARD SOFTWARE CONVERSION ROUTINE IS USED (BCD TO FIELDATA ON INPUT AND THE OPPOSITE ON OUTPUT). THE 'E' OPTION IS ASSUMED ONLY IF SOFTWARE IS USED.
B BINARY (NO TRANSLATE)

S INDICATES THAT A SPECIFIC SYMBIONT MUST BE USED TO PRINT OR PUNCH THE FILE. CAN BE USED WITH EITHER TAPE OR DRUM FILES. THE SPECIFIC SYMBIONT IS NAMED IN THE SYMBIONT FIELD.

EXAMPLE

```
@SYM,S   PRINT$,,PR1
```

IN THIS EXAMPLE THE PRINT OUTPUT PRODUCED WILL BE QUEUED TO AND PRINTED BY THE PRINT SYMBIONT 'PR1'.
C USED WITH A PUNCH FILE WHEN THE SYMBIONT FIELD INDICATES A REMOTE SITE.

THE 'FILENAME' FIELD IS USED TO SPECIFY THE FILE TO BE PROCESSED AND TO DIRECT TAPE MOUNTING IF REQUIRED.

THE 'TYPE' FIELD SPECIFIES THE TAPE TYPE OF THE INPUT FILE. THIS FIELD IS OMITTED FOR ANY SYSTEM CATALOGUED FILES. THE FIELD VALUES ARE:

8C	UNISERVO VIIIC
6C	UNISERVO VIC
4C	UNISERVO IVC
3A	UNISERVO IIIA
2A	UNISERVO IIA

THE 'SYMBIONT' FIELD IS THE NAME OF A SYMBIONT, OR SYMBIONT CLASS, WHICH IS TO

OUTPUT THE FILE. IF OMITTED, THE PRINT SYMBIONT ASSOCIATED WITH THE RUN INITIATION DEVICE IS ASSUMED. TO TRANSMIT A PRINT FILE TO A REMOTE SITE, THE SITE ID MUST REPLACE THE SYMBIONT NAME AND THE 'S' OPTION IS USED.

```
EXAMPLE
      @SYM,S  FILEA,T,SP01
```

IN THE EXAMPLE, THE TAPE FILE 'FILEA' WILL BE PRINTED AT REMOTE SITE 'SP01'. TO TRANSMIT A PUNCH FILE TO A REMOTE SITE, BOTH THE 'S' AND 'C' OPTIONS ARE USED AND THE SITE ID MUST REPLACE THE SYMBIONT NAME.

```
EXAMPLE
      @SYM,SC  FILEB,,SP02
```

IN THE EXAMPLE, THE FILE 'FILEB' WILL BE PUNCHED AT REMOTE SITE 'SP02'.

THE 'FILE-LABELS' FIELD IS USED FOR SELECTIVE PRINTING OR PUNCHING OF FILES RESIDING ON 'FILENAME', WHERE THE DEVICE DEFINED BY 'NAME' IS MAGNETIC TAPE. FILE LABELS ARE SEPARATED WITH A SLASH (/).

MULTIPLE FILE PRINTINGS FROM MASS-STORAGE WILL BE EXECUTED CONCURRENTLY FOR EACH @SYM STATEMENT ENCOUNTERED. TAPE FILES HOWEVER, ARE SERIAL. THE FILE-LABELS CAN BE REPEATED FOR MULTIPLE OUTPUT OPERATIONS. EACH ENTRY IN THE FILE-LABELS FIELD WILL PRODUCE ONE OUTPUT OPERATION.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @SYM FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM VIA CSF$. THE LINKAGE DESCRIPTION IS FOUND IN THE CHAPTER ENTITLED 'SYMBIONTS'.

5.4.9.1. USE OF @SYM WITH PRINT$ AND PUNCH$

EACH RUN ENTERED INTO THE SYSTEM HAS SYMBIONTS DEFINED FOR PROCESSING THE SYSTEM INITIATED PRINT (PRINT$) FILE AND PUNCH (PUNCH$) FILE. THESE OUTPUT SYMBIONTS ARE CLASSIFIED FOR EACH RUN AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - ASSOCIATED SYMBIONTS). HOWEVER IT MAY BECOME NECESSARY TO REDEFINE EITHER, OR BOTH, OUTPUT SYMBIONTS FOR A PARTICULAR RUN TO PROCESS ALL, OR PORTIONS, OF THE OUTPUT FILE. THE 'FILENAME' FIELD IS USED TO DENOTE THE PRINT OR PUNCH FILE WITH EITHER PRINT$ OR PUNCH$ RESPECTIVELY. THE SYMBIONT FIELD IS USED AS DEFINED ABOVE. THE REMAINING TWO FIELDS, 'TYPE' AND 'FILELABELS', DO NOT APPLY TO PRINT$/PUNCH$ USAGE.

EXAMPLE

```
RUN   ---
  -
  -
  -
@SYM  PRINT$,,RPN4/STP
@SYM,S  PRINT$,,STP
@XQT  PROG1
  -
  -
  -
```

IN THIS EXAMPLE, THE PRINT OUTPUT PRODUCED BY PRINT$ IS TO BE SENT TO THE REMOTE SITE 'STP' VIA THE REMOTE 1004 PRINT SYMBIONT

5.4.10. THE @COL STATEMENT

EACH 1004 OF A SYSTEM HAS A DEFINED STANDARD CARD COLUMN MODE FOR READING AND FOR PUNCHING WHICH IS ESTABLISHED AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - CONNECT AND DISCONNECT CHANNEL). THE 80 COLUMN MODE IS ASSUMED STANDARD UNLESS OTHERWISE SPECIFIED. WHEN OPERATING IN THE STANDARD MODE IT MAY BECOME

NECESSARY TO READ CARDS IN A DIFFERENT COLUMN LENGTH. E.G. 80 COLUMN STANDARD MODE IS DEFINED AND COLUMN BINARY CARDS ARE TO BE READ AS PART OF THE INPUT STREAM. THE ABILITY TO SWITCH CARD COLUMN MODE IS ACHIEVED WITH THE CONTROL STATEMENT

```
@COL,OPTIONS     XX              COMMENT
```

NOTE: THE @COL STATEMENT IS FIXED IN FORMAT THRU THE FIRST FOUR CHARACTERS. E.G. A @COL CARD IS WRITTEN:

```
COLUMN......123456789
CHARACTER...@COL
```

WHERE 'XX' SPECIFIES THE INPUT MODE TO BE USED TO READ THE FOLLOWING DATA OF THE CONTROL STREAM. THE INPUT MODE IS SWITCHED TO FACILITATE THE READING OF THE FOLLOWING CARDS AND IS MAINTAINED UNTIL THE TERMINATION SENTINEL CARD IS ENCOUNTERED OR UNTIL THE END OF THE INPUT STREAM IS DETECTED. THE LATTER TWO CASES RESTORE THE STANDARD MODE.

XX	DEFINITION
CB	SWITCH TO COLUMN BINARY INPUT MODE-900 CPM READER OR 1004
PT5	SWITCH TO 5 LEVEL PAPER TAPE INPUT MODE-1004
PT6	SWITCH TO 6 LEVEL PAPER TAPE INPUT MODE-1004
PT7	SWITCH TO 7 LEVEL PAPER TAPE INPUT MODE-1004
PT8	SWITCH TO 8 LEVEL PAPER TAPE INPUT MODE-1004

THE OPTIONS FIELD MAY HAVE THE VALUE:

T- TRANSLATE PAPER TAPE INPUT TO FIELD DATA

PAPER TAPE INPUT IS TRANSLATED TO FIELD DATA FROM A STANDARD CODE WHEN REQUESTED ON THE @COL CONTROL CARD.

EACH BINARY MODE IS TERMINATED WITH A PARTICULAR SENTINEL IMAGE WHICH IS USED TO NOTIFY THE INPUT SYMBIONT OF END OF BINARY DATA. THIS IMAGE HAS THE FIELDATA INTERPRETATION '@ENDCL'. THIS SENTINEL MUST APPEAR IN COLUMNS 1-6 OF THE TERMINATION IMAGE.

A BLANK CARD IS REQUIRED FOLLOWING THE @COL STATEMENT AND THE END SENTINEL IMAGE. THIS IS NECESSARY TO PROPERLY CONDITION THE INPUT MEDIA TO HANDLE THE CHANGE IN THE INPUT OPERATION. THE DATA CARDS READ IN THE NEW MODE MUST IMMEDIATELY FOLLOW THE BLANK CARD.

EXAMPLE

```
@COL CB
IGNORED CARD
DATA CARDS
@ENDCL
IGNORED CARD
CONTINUATION OF RUN STREAM
```

THE BINARY MODE TERMINATION SENTINEL CAN BE REDEFINED WHEN SPECIFYING A BINARY MODE OF INPUT WITH THE @COL CARD.

```
@COL,OPTIONS  XX,SENT
```

'SENT' IS A FIVE CHARACTER FIELD REDEFINING THE BINARY TERMINATION SENTINEL. THE FIRST FIVE CHARACTERS IMMEDIATELY FOLLOWING THE COMMA ARE CONVERTED TO THE BINARY

IMAGE IN WHICH THE TERMINATION IMAGE WILL BE READ. THE 'SENT' FIELD HAS AN IMPLIED '@' PRECEEDING THE FIVE CHARACTER STRING, AND MUST APPEAR ON TERMINATION STATEMENT AS '@SENT'. E.G.

```
      @COL CB,NEWSN
      BLANK CARD (DISREGARDED BY SYSTEM)
      BINARY DATA
      @NEWSN
      BLANK CARD
```

THE INTERPRETATION OF THE @COL AND BINARY MODE TERMINATION CARDS IS DONE DURING THE READING OF THE INPUT STREAM, AND THEY ARE NOT PLACED IN THE RUN FILE. NEITHER CARD CAN HAVE A MEANINGFUL LABEL. THE IGNORED CARDS ARE ALSO ELIMIMATED FROM THE RUN FILE.

THE 'SENT' FIELD IS ALSO USED FOR TERMINATION OF PAPER TAPE INPUT STREAMS. SENT IS THEN AN OCTAL VALUE, EACH COLUMN REPRESENTING THREE BITS, WHICH IS THE END OF INPUT SENTINEL. IF NO SENTINEL IS SPECIFIED ON THE @COL CONTROL CARD, STANDARD SYSTEM SENTINELS ARE ASSUMED AS FOLLOWS.

PR5-	03324
PR6-	043
PR7-	041
PT8-	041

5.4.11. THE @CKPT STATEMENT

THE @CKPT STATEMENT IS USED TO ESTABLISH A CHECKPOINT DUMP AT SOME POINT IN THE CONTROL STREAM TO BE USED AS A RESTART POINT IN THE EVENT SOME CONTINGENCY FORCED THE RUN TO ABORT. CHECKPOINTS ARE RECORDED ON OUTPUT FILES ONLY AND THERE MAY BE AS MANY STATEMENTS AS THE USER DEEMS NECESSARY FOR THE RUN. THE FORMAT OF THE @CKPT CONTROL STATEMENT IS:

```
@CKPT,OPTIONS  FILENAME
```

THE OPTIONS FIELD CAN INCLUDE TWO CHARACTERS. THESE ARE:

P THIS SPECIFIES THAT THE DUMP COMPLETION MESSAGE AND ALL ERROR MESSAGES ARE TO BE WRITTEN ON THE CONSOLE. IF NOT PRESENT NONE OF THESE MESSAGES WILL BE WRITTEN.

T THIS SPECIFIES THAT THE RUN IS TO BE TERMINATED AFTER THE CHECKPOINT IS TAKEN.

THE FIELD 'FILENAME' CONTAINS THE NAME OF A FILE PRESENTLY ASSIGNED TO THE RUN INTO WHICH THE CHECKPOINT INFORMATION WILL BE RECORDED. IT MAY BE AN OUTPUT DATA FILE OR AN INDEPENDENT FILE DEFINED BY THE USER TO CONTAIN CHECKPOINT INFORMATION ONLY. IF THE DUMP IS TO RESIDE ON FASTRAND, IT MUST BE ON A SEQUENTIAL FILE; THE CHECKPOINT CAN NOT BE TAKEN ON A RANDOM FILE. THE FILENAME MAY ALSO BE THE NAME OF A FILE ASSIGNED TO THE SYSTEM BY THE OPERATOR TO CONTAIN CHECKPOINT INFORMATION FROM SEVERAL RUNS. SEE A LATER SECTION OF THIS MANUAL FOR A DETAILED DESCRIPTION OF THE CHECKPOINT FACILITIES.

5.4.12. THE @RSTRT STATEMENT

THE @RSTRT STATEMENT IS USED TO RE-ESTABLISH (RESTART) A RUN AT SOME DESIGNATED CHECKPOINT PREVIOUSLY TAKEN. THE FILE ON WHICH THE DUMP IS LOCATED MUST BE CATALOGUED AT THE TIME OF @RSTRT. ITS EFFECT WITHIN THE CONTROL STREAM MAY BE SIMILAR IN FUNCTION TO THAT OF THE @START CONTROL STATEMENT IN THAT SETUP AND EXECUTION OF THE RUN TO BE RESTARTED IS AN INDEPENDENT OPERATION, OR IT MAY BE A RESTART OF THE RUN REQUESTING RESTART. IN THE LATTER CASE THE RUN IS TERMINATED AND RESTARTED AT THE NAMED CHECKPOINT. THE FORMAT OF THE @RSTRT STATEMENT IS:

```
@RSTRT,PRIORITY/OPTION RUNID,ACCOUNTING,FILENAME,CKPT#,REEL#
```

THE PRIORITY FIELD SPECIFIES THE NEW PRIORITY UNDER WHICH TO REESTABLISH AND RERUN THE RUN. IF NOT SPECIFIED THE STANDARD SYSTEM PRIORITY IS USED.

THE OPTIONS FIELD MAY CONTAIN A 'P' TO SPECIFY THAT THE RESTART ERROR MESSAGES ARE TO BE PRINTED ON THE CONSOLE PRINTER AS WELL AS THE PROGRAM LOG.

THE RUN-ID FIELD IS THE ID OF THE RUN TO BE RESTARTED. IT WILL BE USED TO LOCATE THE CORRECT CHECKPOINT DUMP.

THE ACCOUNTING FIELD SPECIFIES THE ACCOUNT TO WHICH THE CHARGE FOR RELOADING THE RUN IS TO BE ATTRIBUTED. THE ORIGINAL ACCOUNT WILL BE CHARGED WHEN THE RE-ESTABLISHED RUN RECEIVES CONTROL. BOTH THE PRIORITY AND ACCOUNTING SPECIFICATIONS MAY BE OMITTED. IF OMITTED THOSE OF THE INITIATING RUN ARE USED.

THE CKPT# FIELD SPECIFIES THE PARTICULAR CHECKPOINT DUMP OF THIS RUN-ID THAT IS TO BE RESTARTED.

THE FILENAME FIELD SPECIFIES THE NAME OF THE CATALOGUED FILE THAT CONTAINS THE CHECKPOINT DUMP. THE REEL FIELD CONTAINS THE NUMBER OF THE REEL OF THE FILE THAT ACTUALLY CONTAINS THE DUMP TO RESTART. IF A REEL NUMBER IS NOT GIVEN, THE SEARCH WILL BEGIN WITH THE FIRST REEL OF THE FILE. ALL OF THE INFORMATION NECESSARY TO READ THE CHECKPOINT FILE WILL BE TAKEN FROM THE DIRECTORY.

5.5. INPUT/OUTPUT SPECIFICATION STATEMENTS

5.5.1. THE @ASG STATEMENT

THE @ASG (ASSIGN) CONTROL STATEMENT IS USED TO NAME AN EXTERNAL FILE, STATE ITS I/O FACILITY REQUIREMENTS, AND CAUSE THEIR ASSIGNMENT TO THE REQUESTING RUN, UNDER THE GIVEN EXTERNAL FILE NAME. IF THE FILE IS CATALOGUED THE FACILITY REQUIREMENTS ARE KNOWN AND NEED NOT BE SPECIFIED IN ASSIGNING THE FILE AS INPUT. THE VARIETY OF INPUT/OUTPUT DEVICES AVAILABLE MAKES SEVERAL FORMATS NECESSARY FOR THIS STATEMENT. THERE ARE FOUR BASIC FORMATS, AS FOLLOWS:

1. THE FASTRAND @ASG STATEMENT
2. THE MAGNETIC TAPE @ASG STATEMENT
3. THE DRUM @ASG STATEMENT
4. THE ARBITRARY DEVICE @ASG STATEMENT

EACH SPECIFIC FORMAT IS DISCUSSED INDIVIDUALLY IN SUCCEEDING PARAGRAPHS.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @ASG FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN 'EXECUTIVE REQUEST FUNCTIONS', SECTION 8.

ALL USER FILES MUST BE ASSIGNED PRIOR TO BEING REFERENCED FOR I/O OPERATIONS. THE ASSIGNMENTS MAY OCCUR IN ONE OF THREE WAYS:

1. VIA AN @ASG CONTROL STATEMENT.
2. VIA AN EXECUTIVE REQUEST FROM WITHIN A USER PROGRAM.
3. VIA AN EXECUTIVE REQUEST FROM WITHIN A PART OF THE SYSTEM ITSELF, SUCH AS A SYSTEM PROCESSOR OR THE FILE UTILITY ROUTINE (FUR)

THE ONLY CASE WHERE A FILE CAN BE REFERENCED WITHOUT AN ASSIGNMENT SPECIFICATION (@ASG STATEMENT) ON THE PART OF THE USER IS WHEN A CATALOGUED FASTRAND FILE IS BEING NAMED ON A CONTROL STATEMENT OR BEING NAMED IN THE SOURCE LANGUAGE TO A SYSTEM PROCESSOR, SUCH AS THE COLLECTOR. AS IMPLIED IN CASE 3 ABOVE, THE ACTUAL ASSIGNMENT IS MADE BY THE PARTICULAR PART OF THE SYSTEM HANDLING THE CONTROL STATEMENT AND ALL INFORMATION CONCERNING THE ASSIGNMENT IS TAKEN FROM THE DIRECTORY. IF ANY INFORMATION IS NEEDED ABOUT THE FILE OTHER THAN ITS NAME, SUCH AS AN OPTION LETTER, AN @ASG STATEMENT MUST BE USED TO ASSIGN THE FILE. THE USER IS ALWAYS FREE TO ASSIGN A FILE PRIOR TO REFERENCING IT ON A CONTROL STATEMENT. IN THIS CASE THE PART OF THE SYSTEM HANDLING THE CONTROL STATEMENT WILL DETECT THAT THE ASSIGNMENT HAS ALREADY BEEN MADE. THIS ASSIGNMENT MUST BE PERFORMED IF THE FILE IS NOT A CATALOGUED FASTRAND FILE.

5.5.1.1. NOTATION FOR FILE NAMES

THE @ASG STATEMENTS, AS WELL AS OTHER CONTROL STATEMENTS, REQUIRE THAT AN 'EXTERNAL' FILE NAME BE SPECIFIED. PRIOR TO DEALING WITH THE INDIVIDUAL @ASG STATEMENTS, AN EXPLANATION OF THE NOTATION USED FOR FILE NAMES IS IN ORDER.

IN THE FORMAT DESCRIPTION OF THE VARIOUS CONTROL STATEMENTS, THE 'EXTERNAL' FILE NAME IS INDICATED BY 'FILENAME' OR SIMPLY BY 'NAME'. IT SHOULD BE NOTED THAT ALTHOUGH SOME OF THE CONTROL STATEMENTS JUST SPECIFY THE 'EXTERNAL' FILE NAME SPECIFICATION FIELD, THE 'READ' AND 'WRITE' KEYS SUB-FIELDS ARE ALWAYS IMPLIED. (THE KEYS ARE NOT PART OF THE FILE NAME BUT ARE ALWAYS ASSOCIATED WITH IT.) NORMALLY, WHEN A FILE IS REFERENCED IN THE CONTROL STREAM, IT IS THE 'EXTERNAL' NAME THAT IS INTENDED, ALTHOUGH IT CAN ALWAYS BE AN INTERNAL NAME.

AN 'EXTERNAL' FILE NAME HAS THE FORMAT

```
QUALIFIER*FILE(F-CYCLE)
```

WHERE THE 'QUALIFIER', THE '*', AND THE '(F-CYCLE)' ARE ALL OPTIONAL AND BOTH THE 'QUALIFIER' AND THE 'FILE' ARE LIMITED TO 12 CHARACTERS EACH FROM THE SET A...Z, 0...9, -, AND $. THE OMISSION OF THE 'QUALIFIER' WITH THE '*' PRESENT CAUSES THE @QUAL STATEMENT TO SUPPLY THE QUALIFIER USED. IF THE @QUAL STATEMENT HAS NOT OCCURRED, THE 'PROJECT' FIELD FROM THE @RUN STATEMENT IS USED AS THE QUALIFIER. THE OMISSION OF BOTH THE 'QUALIFIER' AND THE '*' ALSO CAUSES THE 'PROJECT' FIELD FROM THE @RUN STATEMENT TO BE USED AS THE QUALIFIER (PROVIDED THE 'FILE' IS NOT AN 'ATTACHED' NAME WHICH POINTS TO A PARTICULAR EXTERNAL NAME). THE PURPOSE OF THE QUALIFIER (PROJECT) IS TO ALLOW 24 CHARACTER UNIQUENESS IN THE CATALOGUE DIRECTORY - BUT MORE IMPORTANT - IF A PARTICULAR USER IS GIVEN A PRIVATE PROJECT NUMBER(S), IT INSURES THE ABILITY TO CATALOGUE THE FILE WITHOUT NAME CONFLICTION, AS LONG AS THE USER INSURES THAT THE 'FILE' DESIGNATORS ARE UNIQUE WITHIN THE PROJECT. THE 'F-CYCLE' NUMBER SERVES TO MAINTAIN SUCCESSIVE VERSIONS OF THE SAME FILE (SAME 'QUALIFIER' AND 'FILE'). OMISSION OF THE F-CYCLE IMPLIES THAT THE MOST RECENTLY CONSTRUCTED FILE IS INTENDED.

THE USE OF THE F-CYCLE FOR FILE CATALOGUING IS SIMILAR TO THE USE OF THE 'CYCLE' FIELD IN PRODUCING UPDATED PROGRAM ELEMENTS.
THE DISCUSSION TO FOLLOW ASSUMES THAT THE READER HAS SEEN THE SECTION ON 'PROCESSOR CALL STATEMENTS' AND IS FAMILIAR WITH HOW THE 'CYCLE' IS USED FOR ELEMENTS WITHIN A FILE.

A FILE WITH A PARTICULAR F-CYCLE NUMBER CAN BE REFERENCED BY THE ABSOLUTE F-CYCLE NUMBER (IDENTICAL TO THAT DESCRIBED BY CYCLE) OR BY A RELATIVE F-CYCLE NUMBER. WITH THE RELATIVE NUMBER, THE LAST FILE TO BE PRODUCED AND CATALOGUED IS REFERENCED BY '+0' OR A VOID; THE ONE BEING PRODUCED AND TO BE CATALOGUED BY A '+1'; AND THE BACKUP FILES BY '-1', '-2', ETC. AS AN EXAMPLE, IF THE LAST FILE TO BE CATALOGUED HAD AN ABSOLUTE F-CYCLE NUMBER OF 28, IT COULD BE REFERENCED BY 28, +0, OR A VOID WITH THE NEW FILE TO BE CATALOGUED CALLED +1. PRIOR TO THE CATALOGUING OF 29, IF A BACKUP EXISTS FOR 28, IT IS NECESSARILY CALLED -1. WHEN THE NEW FILE(29) IS CATALOGUED, IT BECOMES '+0' WITH 28 MOVED TO -1 AND 27 MOVED TO -2. A PLUS SIGN IS ILLEGAL WHEN THE INTENTION IS FOR ABSOLUTE NOTATION. ABSOLUTE F-CYCLE NUMBERS BEGIN WITH 1 AND CONTINUE TO 999 AT WHICH POINT NUMBERING BEGINS WITH 1 AGAIN. THE RELATIVE F-CYCLE NUMBER ALLOWS THE USER TO ACCESS A PARTICULAR RELATIVE BACKUP, FOR EXAMPLE NUMBER 2, WITH A '-2' AT ALL TIMES, WHEREAS WITH THE ABSOLUTE F-CYCLE THE NUMBER CHANGES WITH SUCCESSIVE RUNS.

(PROGRAM-FILENAME.ELEMENTNAME/VERSION(CYCLE)). WHEN UPDATING IS CALLED FOR, THE F-CYCLE NUMBER FOR THE NEW FIELD MUST BE SPECIFIED AS +1. IF AN EXISTING F-CYCLE

IS BEING RENAMED TO SOMETHING OUTSIDE THE SET ALL OLDER RELATIVE F-CYCLES OF THE FILE ARE INCREASED BY ONE . THE ABSOLUTE F-CYCLE DESIGNATOR WILL NOT CHANGE. IF IT IS RENAMED TO RESIDE WITHIN THE SET IT MUST BE GIVEN AN F-CYCLE NUMBER OF +1 TO CONFORM TO THE F-CYCLE RULES. THE 'OLD NAME' FILE IS TREATED AS THOUGH IT WERE BEING MOVED OUTSIDE THE SET, BUT IS PLACED IN THE SAME SET WITH THE 'NEW NAME'. ACTUALLY A DIFFERENT F-CYCLE , IT IS TREATED AS A CATALOGUE ACTION. NUMBER. IF FILE IS BEING RENAMED TO SOMETHING OUTSIDE THE SET ALL OLDER RELATIVE F-CYCLES OF THE FILE ARE INCREASED BY ONE. IF IT IS RENAMED TO RESIDE WITHIN THE SET, IT MUST BE GIVEN AN F-CYCLE NUMBER OF +1 TO CONFORM TO THE F-CYCLE RULES. THE 'OLD NAME' FILE IS TREATED AS THOUGH IT WERE BEING MOVED OUTSIDE THE SET, BUT WHEN PLACED BACK IN THE SAME SET WITH THE 'NEW NAME', ACTUALLY A DIFFERENT F-CYCLE, IT IS TREATED AS A CATALOGUE ACTION.

THE MAXIMUM RANGE OF ABSOLUTE F-CYCLES MAINTAINED FOR A PARTICULAR FILE IS DETERMINED EITHER AS THE SYSTEM STANDARD FOR ALL FILES SUPPLIED AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - RETAIN CYCLES AND F-CYCLES, AND INITIAL SYSTEM) OR AS THE NUMBER SPECIFIED BY THE USER VIA THE FILE UTILITY ROUTINE FOR THIS FILE. AUTOMATIC DELETION OF THE DIRECTORY ENTRY FOR THE OLDEST FILE FILE ITSELF IS ON FASTRAND, IT TOO IS DELETED. IF THE FILE IS ON MAGNETIC TAPE, THE OPERATOR IS NOTIFIED OF THE DIRECTORY DELETION.

WITHIN EACH RUN, THE 'EXTERNAL' NAMES OF THE CURRENT FILE ASSIGNMENTS MUST BE UNIQUE. CURRENT ASSIGNMENTS ARE THOSE ASSIGNED VIA THE @ASG REQUEST BUT NOT YET RELEASED VIA THE @FREE STATEMENT. IN MAINTAINING UNIQUENESS, ANY TWO OF THE FILES ARE UNIQUE IN ONE OF THREE WAYS:

1. UNIQUE BY BOTH 'QUALIFIER' AND 'FILE'
2. UNIQUE BY THE F-CYCLE NUMBER ONLY

IF NONE OF THESE CONDITIONS IS MET, THE ASSIGNMENT IS REJECTED BY THE SYSTEM AND THE RUN IS TERMINATED AND PLACED IN THE ERROR MODE. CATALOGUED FILES ARE NECESSARILY UNIQUE WITHIN THEMSELVES; HOWEVER, THE USER MAY DEFINE ADDITIONAL FILES THAT CAUSE A CONFLICT. IN THE NORMAL CASE, THE USER MUST GUARD ONLY AGAINST A CONFLICT AMONG THE 'FILE' SPECIFICATIONS WITHIN HIS GIVEN QUALIFIER (PROJECT). TWO CASES ARISE HOWEVER WHERE THERE IS A CONFLICT AMONG THE 'FILE' PORTIONS OF EXTERNAL NAMES. THIS OCCURS WHEN THE CYCLING FEATURE IS BEING USED AND WHEN THE NAMES ARE UNIQUE ONLY BY QUALIFIER BECAUSE THE USER MUST USE A FILE OUTSIDE OF HIS PROJECT BUT WITH THE SAME 'FILE' PORTION AS ONE OF HIS OWN FILES. (THE SIGNIFICANCE OF THESE TWO POINTS WILL BE BROUGHT OUT SUBSEQUENTLY.)

THE 'INTERNAL' FILE NAME IS USED BY THE WORKER PROGRAM ON AN I/O REFERENCE TO THE EXECUTIVE AND SPECIFIES THE FILE TO BE USED FOR THE I/O OPERATION. IT IS LIMITED TO A MAXIMUM OF 12 CHARACTERS FROM THE SET A...Z, 0...9, -, AND $. THE 'INTERNAL' NAME MUST POINT TO SOME 'EXTERNAL' NAME BEFORE THE I/O REFERENCE CAN BE HONORED. THIS CONNECTION IS AUTOMATIC BY HAVING THE 'INTERNAL' NAME THE SAME AS THE 'FILE' PORTION OF THE 'EXTERNAL' NAME (ALSO 12 CHARACTERS). AS AN EXAMPLE, IF THE 'EXTERNAL' NAME ON THE @ASG STATEMENT IS 'BLACK*CAT', THEN AN 'INTERNAL' NAME OF 'CAT' WILL POINT TO THE FILE.

THE 'INTERNAL' NAME NEED NOT BE THE SAME AS THE 'FILE' PORTION OF THE 'EXTERNAL' NAME. FOR EXAMPLE, AN 'INTERNAL' NAME OF 'CAT' CAN BE MADE TO POINT TO THE FILE 'BLACK*DOG' BY THE FOLLOWING CONTROL STATEMENT:

```
@USE  CAT,BLACK*DOG
```

THIS STATEMENT CAUSES THE NAME 'CAT' TO BE 'ATTACHED' TO THE 'EXTERNAL' NAME. THIS FEATURE ALLOWS 'INTERNAL' NAMES TO BE FIXED AND SUBSEQUENTLY CONNECTED TO ANY 'EXTERNAL' FILE, DEPENDING ON THE PARTICULAR RUN. FOR A COMPLETE EXPLANATION OF THIS STATEMENT, THE READER SHOULD SEE SECTION 5. ON THE '@USE STATEMENT'.

IN THE TWO CASES MENTIONED ABOVE WHERE THE 'FILE' PORTIONS OF 'EXTERNAL' NAMES ARE NOT UNIQUE, A @USE STATEMENT IS REQUIRED IN ORDER TO CONNECT AN 'INTERNAL' NAME TO A PARTICULAR FILE INVOLVED IN THE CONFLICT. WHEN AN ASSIGNMENT IS MADE, IF THE 'FILE' PORTION OF THE 'EXTERNAL' NAME IS THE SAME AS THAT OF A PREVIOUS ASSIGNMENT, THE FILE

BEING ASSIGNED IS MARKED AS NOT-AVAILABLE FOR I/O REFERENCE EXCEPT VIA AN 'ATTACHED' NAME SUPPLIED BY A @USE STATEMENT TO RESOLVE THE CONFLICT. IN OTHER WORDS, OF ALL FILES THAT HAVE THE SAME 'FILE' PORTION IN THEIR 'EXTERNAL' NAMES, ONLY THE FIRST CAN BE REFERENCED BY THE 'FILE' PORTION ON AN I/O REFERENCE. ALL OTHERS MUST HAVE 'ATTACHED' NAMES. AS STATED EARLIER, THIS SITUATION ARISES ONLY WHEN THE F-CYCLE FEATURE IS BEING USED AND WHEN THE 'FILE' PORTIONS ARE NOT UNIQUE BECAUSE OF THE NECESSITY TO USE FILES FROM DIFFERENT PROJECTS BUT WITH THE SAME 'FILE' IDENTIFIERS.

AN 'ATTACHED' NAME MAY BE USED FROM WITHIN THE CONTROL STREAM IF SPECIFIED BY A @USE STATEMENT. THE EXTERNAL NAME (WHERE THE PROJECT OR QUALIFIER IS USED) WILL ALWAYS SUFFICE, HOWEVER IT MAY BE DESIRABLE TO USE A SHORTER NAME OR TO USE AN 'ATTACHED' NAME THAT HAD TO BE SPECIFIED FOR SOME OTHER REASON. THE SYSTEM TREATS FILE SPECIFICATIONS IN THE CONTROL STREAM AS FOLLOWS. IF ANY PART OF THE 'EXTERNAL' NAME OTHER THAN THE 'FILE' PORTION IS GIVEN, THE NAME IS ALWAYS TREATED AS 'EXTERNAL'. IF ONLY THE 'FILE' PORTION IS SPECIFIED AND THE 'FILE' CANNOT BE FOUND IN THE 'ATTACHED' LIST, THE 'EXTERNAL' LIST IS SEARCHED FOR THE 'PROJECT*FILE'. IF A MATCH IS NOT FOUND, IT IS THEN ASSUMED THAT THE FILE IS A CATALOGUED FILE YET TO BE ASSIGNED. IF AN 'ATTACHED' NAME IS NOT INTENDED, BUT SUCH AN 'ATTACHED' NAME EXISTS, THE '*' MUST BE SPECIFIED EVEN IF THE 'QUALIFIER' IS ACTUALLY THE PROJECT NUMBER. THIS IS THE ABNORMAL CASE AND OCCURS ONLY WHEN THERE IS A CONFLICT BETWEEN 'ATTACHED' NAMES AND THE 'FILE' PORTION OF AN 'EXTERNAL' NAME.

5.5.1.2. EXCLUSIVE USE FILE AND FACILITY HANDLING

THE EXEC PROVIDES FOR THE PLACEMENT OF @ASG AND @FREE STATEMENTS ANYWHERE WITHIN THE CONTROL STREAM. DYNAMIC ASSIGN AND FREE REQUESTS MAY APPEAR WITHIN THE PROGRAMS. THESE FEATURES ALLOW THE USER TO ASSIGN AND FREE FILES AS REQUIRED, WITHOUT 'TYING-UP' THE FILES AND/OR FACILITIES FROM THE BEGINNING OF THE RUN UNTIL ITS COMPLETION. HOWEVER, THE USER MIGHT BE FORCED TO WAIT UNTIL THE FACILITY OR FILE IS MADE AVAILABLE WHEN THE REQUEST IS FOR ONE OF THE FOLLOWING:

1. A MAGNETIC TAPE UNIT THAT IS BEING USED BY ANOTHER RUN.
2. AN ARBITRARY DEVICE THAT IS BEING USED BY ANOTHER RUN.
3. A MAGNETIC TAPE FILE THAT IS BEING USED BY ANOTHER RUN.
4. EXCLUSIVE USE OF A FASTRAND FILE BY ANOTHER RUN OR THIS RUN.

TO PREVENT THE POSSIBLE PROLONGED WAIT OF A RUN WHEN REQUESTING AN EXCLUSIVE USE FACILITY AND YET NOT FORCE A RUN TO SPECIFY ALL REQUIREMENTS BEFORE THE FIRST PROGRAM (TASK) OF THE CONTROL STREAM, THE EXECUTIVE:

1. WILL NOT OPEN A RUN FOR EXECUTION UNTIL ALL THE @ASG STATEMENTS LOCATED BEFORE THE FIRST TASK IN THE CONTROL STREAM HAVE BEEN SATISFIED.
2. WILL NOT START THE EXECUTION OF A PROGRAM UNTIL ALL THE @ASG STATEMENTS LOCATED BEFORE THE PROGRAM IN THE CONTROL STREAM HAVE BEEN SATISFIED.
3. WILL MAINTAIN FOR EACH RUN (ONCE IT HAS BEEN OPENED) A PRIVATE FACILITY POOL.

ON THE MAGNETIC TAPE @ASG STATEMENTS, THERE IS AN OPTION WHICH RELEASES JUST THE FILE AND NOT THE PHYSICAL FACILITY AT THE OCCURRENCE OF A @FREE STATEMENT. THE 'SAVED' PHYSICAL FACILITY IS PLACED IN THE FACILITY POOL OF THE RUN AND IS AVAILABLE FOR RE-ASSIGNMENT AT ANY POINT WITHIN THIS RUN. THE FACILITY WILL NOT BE RETURNED TO THE EXECUTIVE'S FACILITY POOL, AVAILABLE TO ALL RUNS, UNTIL IT IS RE-ASSIGNED AND COMPLETELY RELEASED OR UNTIL RUN TERMINATION. THE USER RE-ASSIGNS FACILITIES THROUGH NORMAL MEANS, CONFIDENT THAT THE REQUEST CAN BE IMMEDIATELY HONORED, SINCE THE RUN'S FACILITY POOL IS ALWAYS REFERENCED BEFORE THE EXECUTIVE'S FACILITY POOL. BY USING THE ABOVE MENTIONED OPTION (SEE DISCUSSIONS OF @ASG STATEMENTS) AND SPECIFYING BEFORE THE FIRST PROGRAM OF THE RUN, THE MAXIMUM AMOUNT OF EACH TYPE OF MAGNETIC TAPE OR ARBITRARY DEVICE THE RUN WILL REQUIRE AT ANY ONE GIVEN TIME, THE USER HAS THE ABILITY TO:

PLACE @ASG STATEMENTS AND/OR DYNAMIC ASSIGN REQUESTS

ANYWHERE IN THE CONTROL STREAM AND/OR PROGRAMS AND STILL BE ASSURED THAT THE RUN AND/OR PROGRAMS WILL ALWAYS IMMEDIATELY RECEIVE THE FACILITY REQUESTED.

5.5.1.3. THE FASTRAND @ASG STATEMENT

THE FASTRAND @ASG CONTROL STATEMENT IS USED TO ASSIGN:

FASTRAND	(MODEL II)
FASTBAND	
SIMULATED FASTRAND	(DRUM IS USED AS THE RECORDING DEVICE BUT THE FILES ARE CALLED AS IF THEY WERE FOR FASTRAND ALLOCATION)

THE GENERAL FORM OF THE FASTRAND @ASG STATEMENT IS:

```
@ASG,OPTIONS  NAME/KEY1/KEY2,TYPE/RESERVE/GRANULE/MAXIMUM
```

THE FIELDS OF THE STATEMENT ARE EXPLAINED IN SUCCEEDING PARAGRAPHS AND IN THE ORDER OF APPEARANCE ON THE STATEMENT.

THE OPTIONS SUB-FIELD IS USED TO CAUSE A FILE TO BE CATALOGUED (OR DECATALOGUED) AND TO PLACE OR REMOVE CONSTRAINTS ON THE USE OF THE FILE. IT SHOULD BE NOTED THAT WHEN AN ERROR CONDITION OCCURS WHICH WOULD CAUSE A BATCH RUN TO BE TERMINATED, THE DEMAND USER RECEIVES AN ERROR MESSAGE AND IS ALLOWED TO SUBMIT A NEW STATEMENT.

CATALOGUING OPTIONS ARE AS FOLLOWS :

C SPECIFIES THAT THE FILE IS TO BE CATALOGUED IF THE RUN TERMINATES NORMALLY. IF A @FREE COMMAND (CONTROL STATEMENT OR EXECUTIVE REQUEST) IS ENCOUNTERED FOR THE FILE PRIOR TO TERMINATION, THE FILE IS CATALOGUED AT THAT TIME (SEE 'THE @FREE STATEMENT'). IF A FILE BY THIS 'NAME' ALREADY EXISTS IN THE MASTER DIRECTORY, THE RUN IS PLACED IN THE ERROR MODE.

U SAME AS 'C' OPTION EXCEPT THAT THE FILE IS TO BE CATALOGUED AT RUN TERMINATION REGARDLESS OF THE MANNER OF TERMINATION (BEYOND THIS STATEMENT). THE @FREE COMMAND MAY CAUSE CATALOGUING PRIOR TO THE TERMINATION.

R SPECIFIES THAT THE FILE IS TO BE PLACED IN THE ' READ-ONLY' STATE WHEN IT IS CATALOGUED. THIS OPTION IS MEANINGFUL ONLY WHEN THE 'C' OR 'U' OPTION IS ALSO PRESENT. A FILE CATALOGUED WITH THE 'R' OPTION PRESENT CANNOT BE OVER-WRITTEN. THE FILE CAN ONLY BE READ OR DECATALOGUED. ANY ACTIVITY REQUESTING TO WRITE IN THE FILE WILL BE PLACED IN THE ERROR MODE.

P SPECIFIES THAT THE FILE IS TO BE CATALOGUED AS A 'PUBLIC' FILE RATHER THAN A 'PRIVATE' FILE. THE DISTINCTION BETWEEN THEM IS THAT ONLY THE RUNS WHICH HAVE THE SAME PROJECT ID AS THE RUN WHICH CREATED THE FILE CAN ACCESS A 'PRIVATE' FILE WHILE ANY RUN CAN ACCESS A 'PUBLIC' FILE. (FOR PRIVACY IN 'PRIVATE' FILES SEE THE DISCUSSION CONCERNING THE TWO 'KEY' SUB-FIELDS).

W SPECIFIES THAT THE FILE IS TO BE CATALOGUED AS A WRITE ONLY FILE. THE FILE CAN ONLY BE WRITTEN INTO, AND IN THE PROCESS EXTENDED.

THE ABOVE OPTIONS ARE FOR USE ONLY WITH FILES THAT ARE NOT PRESENTLY CATALOGUED. IF NEITHER OF THE CATALOGUING OPTIONS ('C' AND 'U') APPEAR, THE FILE, UNLESS CURRENTLY CATALOGUED, IS TREATED AS TEMPORARY AND RELEASED AT RUN TERMINATION. IT WILL BE RELEASED PRIOR TO RUN TERMINATION IF A @FREE STATEMENT IS ENCOUNTERED. IN THE ABSENCE OF THE 'P' OPTION, A FILE IS ALWAYS CATALOGUED AS 'PRIVATE'.

OPTIONS TO BE USED WHEN THE @ASG STATEMENT NAMES A FILE THAT IS PRESENTLY CATALOGUED ARE AS FOLLOWS:

D SPECIFIES THAT THE CATALOGUED FILE IS TO BE DELETED FROM THE DIRECTORY(DE-CATALOGUED) IF THE RUN TERMINATES NORMALLY OR WHEN A @FREE COMMAND IS ENCOUNTERED PRIOR TO TERMINATION. THE EXECUTIVE WILL INSURE THE FILE IS ASSIGNED ONLY TO THIS RUN AT THE TIME OF RELEASE.

K SAME AS 'D' OPTION EXCEPT THAT THE FILE IS TO BE DELETED AT RUN TERMINATION REGARDLESS OF THE MANNER OF TERMINATION. THE @FREE COMMAND MAY CAUSE THE FILE TO BE DE-CATALOGUED PRIOR TO TERMINATION.

X SPECIFIES THAT THIS RUN IS TO HAVE 'EXCLUSIVE USE' OF THE FILE UNTIL THE RUN HAS TERMINATED OR THE FILE IS RELEASED VIA THE @FREE COMMAND. NO OTHER RUN CAN BE USING THE FILE. (IF THE FILE IS NOT CURRENTLY CATALOGUED, THE 'X' OPTION IS NOT NEEDED BECAUSE THE RUN NECESSARILY HAS 'EXCLUSIVE USE').

A SPECIFIES THAT THE FILE IS CURRENTLY CATALOGUED AND INSURES THAT THE EXECUTIVE WILL NOT TREAT THE FILE AS TEMPORARY IF THE NAME CANNOT BE FOUND. THE RUN WILL BE TERMINATED IF THE NAME CANNOT BE FOUND IN THE DIRECTORY.

THE ABOVE OPTIONS ARE TO BE USED ONLY WITH FILES THAT ARE CURRENTLY CATALOGUED. IF NEITHER OF THE DE-CATALOGUING OPTIONS ('D' OR 'K') APPEAR, THE CATALOGUED FILE IS LEFT INTACT AT RUN TERMINATION. IF EITHER THE 'D' OR 'K' OPTIONS APPEAR AND THE FILE HAS EITHER OR BOTH KEYS, THE KEY(S) MUST BE SPECIFIED. FAILURE TO DO SO CAUSES THE RUN TO BE PLACED IN THE ERROR MODE.

AN OPTION TO BE USED FOR A TEMPORARY FILE (NOT CATALOGUED AND NOT TO BE CATALOGUED) IS:

T SPECIFIES THAT THE FILE IS TEMPORARY AND ALLOWS IT TO HAVE A NAME THE SAME AS THAT OF A CATALOGUED FILE. NO THOUGHT NEED BE GIVEN AS TO WHETHER A FILE BY THIS NAME IS CURRENTLY CATALOGUED. IF THIS OPTION IS --NOT-- PRESENT.FOR TEMPORARY FILES, THE SYSTEM WILL ATTEMPT TO FIND THE FILE IN THE DIRECTORY. IF A FIND IS MADE, THE ASSIGNMENT WILL BE MADE FROM THE DIRECTORY.

THE FOLLOWING OPTIONS CONTROL THE DUMPING OF CATALOGUED FASTRAND FILES AT A CHECKPOINT, AND SUBSEQUENT SYSTEM ACTION ON RESTARTING:

F DUMP THE FILE AS A PART OF ANY CHECKPOINT.

G RELOAD THIS FILE IF ANY OTHER RUN HAS REFERENCED THE FILE SINCE THE CHECKPOINT.

H RELOAD THIS FILE ONLY IF NO OTHER RUN HAS

REFERENCED THE FILE SINCE CHECKPOINT.

M IF A CATALOGUED FILE BY THIS NAME EXISTS WHEN RELOADING, MAKE THE RELOADED FILE AVAILABLE TO THIS RUN AS A TEMPORARY FILE.

N RENAME THIS FILE UPON RELOADING IF A CATALOGUED FILE WITH THIS NAME EXISTS.

OPTION F FORCES THE FILE TO BE DUMPED ON A CHECK POINT. WITHOUT ONE OF THE OPTIONS, G OR H, THE FILE IS ALWAYS RELOADED ON RESTART. OPTIONS M AND N CONTROL THE MANNER OF RELOAD.

THE FIELD 'NAME' ON THE @ASG STATEMENT IS USED TO SPECIFY THE 'EXTERNAL' NAME OF THE FILE. THE NAME MUST BE PRESENT AND IS SPECIFIED IN THE NORMAL MANNER:

QUALIFIER*FILE

WHERE THE 'QUALIFIER' AND '*' ARE OPTIONAL AND NEITHER THE 'QUALIFIER' NOR THE 'FILE' MAY EXCEED 12 CHARACTERS. ALTHOUGH NOT SHOWN ABOVE AN 'F-CYCLE' NUMBER MAY ALSO NEED TO BE SPECIFIED FOR CATALOGED FILES (SEE SECTION 5.6.1. FOR A MORE DETAILED DESCRIPTION OF THE 'EXTERNAL' NAME).

WHEN CATALOGUING, THE SUB-FIELDS 'KEY1' AND 'KEY2' LOCK A FILE AGAINST INDISCRIMINATE READING AND WRITING, RESPECTIVELY, BY OTHER USERS. THEY MAY CONTAIN UP TO SIX CHARACTERS AND ALL CHARACTERS ARE LEGAL EXCEPT THE BLANK, THE SLASH (/), THE COMMA (,), AND THE SEMICOLON(;). A FILE IS CATALOGUED WITH 'READ' AND/OR 'WRITE' LOCK BY SPECIFYING THE KEY1 AND/OR KEY2 SUB-FIELDS ALONG WITH THE 'C' OR 'U' OPTION. TO GAIN READ AND/OR WRITE ACCESS TO SUCH A FILE, THE APPROPRIATE KEY(S) MUST BE SPECIFIED AT ASSIGN TIME OR THE REQUEST(S) WILL NOT BE HONORED. (ONCE THE ASSIGNMENT HAS BEEN MADE, WITH THE APPROPRIATE KEY(S) MADE AVAILABLE THROUGH THE @ASG OR @USE STATEMENT, THE KEY(S) NEED NOT BE SPECIFIED IN FURTHER REFERENCES.) IF THE KEY(S) ARE KNOWN, A 'LOCKED' FILE CAN BE PARTIALLY OR COMPLETELY 'UNLOCKED' OR HAVE ITS KEY(S) CHANGED BY USING THE FILE UTILITY ROUTINE STATEMENT CREATED FOR THIS PURPOSE (SECTION 14, 'FILE UTILITY ROUTINES').

A COMBINATION OF THE TWO KEYS IS USED FOR CATALOGUING. THE FOLLOWING TABLE SHOWS THE ACTION ALLOWED ACCORDING TO THE KEY(S) GIVEN AT CATALOGUING TIME AND THE KEY(S) GIVEN AT ASSIGN TIME. WHERE 'MESSAGE' APPEARS AS AN ACTION, A 'FAC WARNING DDDDDDDDDDDD' MESSAGE WILL BE PRINTED.

IF A KEY IS FURNISHED AND IT DOES
NOT MATCH THE CATALOGUED KEY, THE RUN IS ABORTED, AND THE MESSAGE WILL BE 'FAC REJECTED DDDDDDDDDDDD'.

KEY(S) SPECIFIED AT CATALOGUING TIME	KEY(S) SPECIFIED AT ASSIGN TIME. READ	WRITE	BOTH	NEITHER
READ	READ WRITE	ABORT	ABORT	WRITE
WRITE	ABORT	READ WRITE	ABORT	READ
BOTH	READ MESSAGE	WRITE MESSAGE	READ WRITE	MESSAGE
NEITHER	ABORT	ABORT	ABORT	READ WRITE

TABLE 5.1 ALLOWABLE ACTION ACCORDING TO KEYS

ON ALL @ASG STATEMENTS (FASTRAND, MAGNETIC TAPE, ETC.), THE FIELD THAT FOLLOWS THE NAME FIELD IS CALLED THE 'FACILITIES' FIELD. AS SHOWN PREVIOUSLY, THE FACILITIES FIELD FOR THE FASTRAND @ASG STATEMENT IS 'TYPE/RESERVE/GRANULE/MAXIMUM'. IN GENERAL, IF THE FILE IS CATALOGUED AND TO BE READ, THE ENTIRE FACILITIES FIELD NEED NOT BE SPECIFIED.

THE SUB-FIELD 'TYPE' SPECIFIES THAT THE STATEMENT APPLIES TO FASTRAND FORMAT AND, IN ADDITION; POINTS OUT THE TYPE OF EQUIPMENT TO BE USED. IF EQUIPMENT TYPE IS SPECIFIED FOR CATALOGUED FILES, IT IS CHECKED FOR COMPATIBILITY. IF NOT COMPATIBLE, THE STATEMENT IS REJECTED. FOR CATALOGUED FASTRAND FORMAT FILES THE FURNISHED TYPE MUST AGREE WITH CATALOGUED TYPE. THE ALLOWABLE TYPES FOR THE FASTRAND @ASG STATEMENT AND ASSOCIATED EQUIPMENT FOR EACH TYPE ARE:

F4	FASTRAND SIMULATED ON FH 432
F17	FASTRAND SIMULATED ON FH 1782 (DELAYED IMPLEMENTATION)
F8	FASTRAND SIMULATED ON FH 880 IMPLEMENTATION)
F2	FASTRAND MODEL II
FB	FASTBAND
F	FASTRAND, TYPE INDEPENDENT

A FILE PLACED ON DRUM SIMULATED FASTRAND HAS ALL THE CHARACTERISTICS OF A FASTRAND FILE EXCEPT FOR SECTOR PADDING ON WRITE FUNCTIONS. WHEN TYPE F4 OR F8 IS SPECIFIED AND DRUM SPACE IS NOT CURRENTLY AVAILABLE, THE SYSTEM WILL SUBSTITUTE F8 OR F2 FOR F4, OR F2 FOR F8. NO SUBSTITUTION IS MADE FOR FASTBAND REQUESTS. CATALOGUING IS PERMISSIBLE ON ANY HARDWARE TYPE.

RULES FOR FASTRAND FORMAT ALLOCATION REQUESTS

REQUEST	ORDER OF SATISFYING REQUEST
F4	F4, F17, F8, F2
F17	F17, F8, F2
F8	F8,F2
F2	F2
FB	FB
F	F4, F17, F8, F2

THE SUB-FIELD 'RESERVE' IS USED TO SPECIFY THE APPROXIMATE NUMBER OF GRANULES TO BE USED BY THE FILE. THE SUB-FIELD 'GRANULE' IS USED TO SPECIFY THE GRANULE SIZE. IN CERTAIN CASES, EITHER OR BOTH SUB-FIELDS MAY BE OMITTED. IF THE GRANULE SUB-FIELD IS SPECIFIED IT MUST CONTAIN EITHER 'TRK' OR 'POS'.

TRK -- SPECIFIES A GRANULE OF ONE TRACK (64 SECTORS)
POS -- SPECIFIES A GRANULE OF ONE POSITION (64 TRACKS)

IF THE GRANULE SPECIFICATION IS OMITTED, THE GRANULE IS ASSUMED TO BE 'TRK'. THE GRANULE SUB-FIELD IS IGNORED IF THE FILE IS CURRENTLY CATALOGUED.

THE RESERVE SUB-FIELD IS IGNORED AND NEED NOT BE SPECIFIED WHEN THE FILE IS CATALOGUED AND IS TO BE READ ONLY. IF THE FILE IS TO BE CREATED OR UPDATED, THE RESERVE MAY CONTAIN AN INTEGER SPECIFYING THE NUMBER OF GRANULES TO RESERVE FOR THE FILE (ON AN UPDATE THE RESERVE SPECIFICATION INCLUDES THAT PORTION OF THE FILE THAT ALREADY EXISTS). IF THE RESERVE SPECIFICATION IS OMITTED, NO GRANULES (OR ADDITIONAL GRANULES) ARE INITIALLY ASSIGNED, THEY ARE ASSIGNED DYNAMICALLY AS NEEDED. WHEN THE RESERVE IS SUPPLIED BUT EXCEEDED, ADDITIONAL GRANULES ARE ALSO ASSIGNED DYNAMICALLY AS NEEDED.

NOTE: WHEN CREATING A FILE, THE RESERVE SUB-FIELD SHOULD CONTAIN A REASONABLE ESTIMATE OF THE NUMBER OF GRANULES NEEDED. IF A FILE CAN BE CONTAINED WITHIN THE LIMITS OF THE RESERVE THE RUN IS ASSURED OF BEING ABLE TO CREATE THE FILE WITHOUT DELAY (WITH DYNAMIC EXPANSION, THE REQUESTING PROGRAM MAY BE REMOVED FROM CORE WHILE FASTRAND IS BEING MADE AVAILABLE). IN ADDITION, THE SPECIFICATION OF A RESERVE AIDS THE EXECUTIVE IN ALLOCATING FASTRAND AREA EFFICIENTLY (IF A RESERVE IS USED THE TRACK OR POSITIONS WILL BE ADJACENT, IF POSSIBLE).
FOR MOST EFFICIENT USE OF MASS STORAGE ALL FILES THAT ARE TO BE PROGRAM FILES SHOULD BE ALLOCATED UNDER TRACK GRANULES ('TRK') . A SPECIFICATION OF 'POS' CREATES UNUSED SPACE IN THE PROGRAM FILE IN THAT 64 CONTIGUOUS TRACKS WILL BE ASSIGNED .

IF THE FILE TAKES FEWER GRANULES THAN RESERVED, THE EMPTY GRANULES ARE RETURNED TO THE AVAILABLE STATUS WHEN THE FILE IS CATALOGUED. FURTHERMORE, IF 'POS' IS SPECIFIED IN THE GRANULE SUB-FIELD AND CONSIDERABLE SPACE IN THE HIGHEST REFERENCED GRANULE HAS NOT BEEN REFERENCED, THIS NONREFERENCED SPACE IS PUT BACK INTO THE AVAILABLE POOL AT 'FREE' TIME. THE RESERVE VALUE IS PLACED IN THE DIRECTORY AND WILL BE USED ON FUTURE UPDATES UNLESS A RESERVE IS SUPPLIED ON THE UPDATE @ASG STATEMENT. IN THAT CASE IT IS USED AND REPLACES THE PREVIOUS VALUE IN THE DIRECTORY.

THE SUB-FIELD 'MAXIMUM' IS USED TO INDICATE THAT THE RUN IS TO BE TERMINATED IF THE LENGTH OF THE FILE BEING CREATED OR UPDATED EXCEEDS THE NUMBER OF GRANULES SPECIFIED. THIS FIELD IS USED PRIMARILY TO INSURE THAT A RUN-AWAY-FILE SITUATION DOES NOT OCCUR DURING DEBUGGING. HOWEVER, IT MAY ALSO BE USED TO OVER-RIDE THE 'SYSTEM-MAXIMUM' FOR ALL FILES, AS SPECIFIED BY THE PARTICULAR INSTALLATION AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - MAXIMUM FILE SIZE, AND INITIAL SYSTEM). THE MAXIMUM SUB-FIELD IS NEVER A REQUIRED SPECIFICATION. IF THE FILE IS BEING CREATED OR UPDATED AND A MAXIMUM IS GIVEN, ITS VALUE IS PLACED IN THE DIRECTORY ALONG WITH THE NAME, TYPE, RESERVE, AND GRANULE SIZE.

IF A MAXIMUM WAS SUPPLIED WHEN THE FILE WAS CATALOGUED, ITS VALUE IS RETAINED AND USED WHEN AN UPDATE OCCURS. IF A MAXIMUM IS SUPPLIED ON THE UPDATING @ASG STATEMENT, IT IS USED. IT IS ALSO PLACED IN THE DIRECTORY, THEREBY REPLACING THE PREVIOUS MAXIMUM.

CONSIDER THE FOLLOWING EXAMPLES OF @ASG STATEMENTS FOR FASTRAND:

```
@ASG,CR  FILEX,F/5
```

IF THE RUN TERMINATES NORMALLY OR A @FREE STATEMENT FOR FILEX IS PROCESSED, FILEX WILL BE CATALOGUED IN THE 'READ-ONLY' MODE. FIVE TRACKS ARE ASSIGNED INITIALLY AND THE 'SYSTEM-MAXIMUM' SIZE IS ASSUMED AS NO MAXIMUM WAS SPECIFIED.

```
@ASG,D  FILEX/A2294B
```

FILEX IS CURRENTLY CATALOGUED AND IS TO BE DE-CATALOGUED IF THE RUN TERMINATES NORMALLY, THE KEY A2294B IS REQUIRED TO READ THE FILE.

```
@ASG,T FILEX,F/4/POS/5
```

FILEX IS A TEMPORARY FILE REQUIRING 4 FASTRAND POSITIONS TO BE RESERVED INITIALLY. TERMINATION IS TO OCCUR IF MORE THAN 5 POSITIONS ARE REQUIRED.

```
@ASG,X  FILEX,/6//8
```

FILEX IS CURRENTLY CATALOGUED AND THIS RUN IS TO HAVE 'EXCLUSIVE USE' OF THE FILE FOR UPDATING. A RESERVE OF 6 TRACKS IS SPECIFIED AND THE RUN IS TO BE TERMINATED IF MORE THAN 8 TRACKS ARE USED.

DIAGNOSTIC MESSAGES

A GENERALIZED FORMAT IS CURRENTLY USED IN THE PRINT FILE ASSIGNED TO EACH RUN FOR THE ASG, MODE, CAT, FREE, AND USE STATEMENTS. THE FORMAT IS AS FOLLOWS:

1. (STATEMENT IMAGE)

   FAC REJECTED DDDDDDDDDDDD

2. (STATEMENT IMAGE)

   FAC WARNING DDDDDDDDDDDD

THE FIRST MESSAGE WILL APPEAR FOR A RUN THAT ABORTED DUE TO A STATEMENT THAT CANNOT BE HONORED BY THE SYSTEM. THE SECOND MESSAGE IS A WARNING THAT THE STATEMENT COULD CAUSE A PROBLEM. IN EITHER CASE THE REASON FOR REJECTION OR WARNING IS DETERMINED BY EXAMINING THE BITS SET IN THE OCTAL WORD 'DDDDDDDDDDDD'. (THIS STATUS WORD IS SET IN A0 IF THE STATEMENT WAS PROCESSED BY CSF$.) THE FOLLOWING TABLE DEFINES THE MEANING OF THE BITS IF SET (1=SET). BITS ARE NUMBERED 35-0 READING LEFT TO RIGHT.

BIT	SOURCE		DESCRIPTION OF MEANING IF BIT SET
	RUN	CSF	
35	K		REQUEST NOT ACCEPTED - EXAMINE REST OF BITS AS TO WHY.
34	K		FIELD ERROR IN STATEMENT OTHER THAN SYNTAX. ALSO OPTION CONFLICT 'MLH','OE', 'IB'
33	K		FILE NAME HAS ALREADY BEEN ASSIGNED TO THIS RUN.
32			(RESERVED INTERNAL USE BY F.I.) SET ON FIND IN DIRECTORY.
31	K		EQUIP TYPE ON ASG STATEMENT IS NOT COMPATABLE

			WITH CATALOGUED EQUIP TYPE.
30			(RESERVED INTERNAL USE BY F.I.), NAME FOUND IN ATTACHED NAMES LIST IN PCT
29	W		12 CHARACTER NAME IS NOT UNIQUE (THAT PORTION OF NAME USED AS INTERNAL NAME FOR I/O PACKETS).
28			(RESERVED INTERNAL USE BY F.I.), X OPTION SPECIFIED ON FILE ALREADY ASG THIS RUN
27	K	K	READ KEY INCORRECT FOR CATALOGUED FILE
26	K	K	WRITE KEY INCORRECT FOR CATALOGUED FILE.
25	W		WRITE KEY EXISTS IN DIRECTORY, NOT SPECIFIED ON ASG STATEMENT (ASSIGNED READ MODE ONLY).
24	W		READ KEY EXISTS IN DIRECTORY, NOT SPECIFIED ON ASG STATEMENT (ASSIGNED WRITE MODE ONLY).
23	K	K	READ KEY FURNISHED ON ASG STATEMENT NONE EXISTS IN DIRECTORY.
22	K	K	WRITE KEY FURNISHED ON ASG STATEMENT NONE EXISTS IN DIRECTORY.
21	K		'A' OPTION SPECIFIED ON ASG STATEMENT AND FILE NAME WAS NOT FOUND IN DIRECTORY.
20	K		INVALID REEL NUMBER ON ASG STATEMENT FOR CATALOGUED TAPE FILE.
19			(RESERVED INTERNAL USE BY F.I.) MASS STORAGE FILE HAS BEEN ROLLED OUT.
18			(RESERVED INTERNAL USE BY F.I.) REQUEST ON WAIT STATUS FOR FACILITIES.
17	K		OPTION CONFLICT FOR CATALOGUED FILE, BOTH 'D'&'K' OR 'CUPRW' WHICH ARE OPTIONS FOR NEW FILES
16			(RESERVED INTERNAL USE BY F.I.) -FILE WAS BEEN ASSIGNED EXCLUSIVELY SOME OTHER RUN
15			(RESERVED INTERNAL USE BY F.I.) FIND MADE IN DIRECTORY AND ALREADY ASSIGNED
14			(RESERVED INTERNAL USE BY F.I.) FIND MADE IN DIRECTORY AND NOT ASSIGNED
13	K		PROJECT I.D. INCORRECT FOR CATALOGUED PRIVATE FILE
12			(RESERVED INTERNAL USE BY F.I.) EQUIP TYPE IS TAPE; IS SET ON FIND OF NAME IN DIRECTORY
11			(RESERVED INTERNAL USE BY F.I.) READ ONLY FILE CATALOGUED WITH R OPTION
10			(RESERVED INTERNAL USE BY F.I.) WRITE ONLY FILE CATALOGUED WITH W OPTION

9			EQUIPMENT IS DOWN
8			UNUSED
.			
0			

NOTE: K IN RUN OR CSF$ SOURCE
MEANS THAT THE RUN WILL BE TERMINATED
W IN RUN STREAM MEANS THAT A WARNING MESSAGE WILL
BE PRINTED.

5.5.1.4. THE MAGNETIC TAPE @ASG STATEMENT

FOR MAGNETIC TAPE THE FORMAT OF THE @ASG STATEMENT IS:

```
@ASG,OPTIONS  NAME/KEY1/KEY2,TYPE/UNITS/LOG/NOISE,;
 REEL1/REEL2/.../REELN
```

THE 'NAME' FIELD AND THE 'KEY1' AND 'KEY2' SUB-FIELDS ARE THE SAME AS FOR THE FASTRAND @ASG STATEMENT. THE NAME MUST ALWAYS APPEAR.

OPTIONS TO BE USED FOR CATALOGUING ARE:

C	SAME AS FOR FASTRAND
U	SAME AS FOR FASTRAND
P	SAME AS FOR FASTRAND
R	SAME AS FOR FASTRAND

OPTIONS TO BE USED WHEN THE FILE IS PRESENTLY CATALOGUED ARE:

D	SAME AS FOR FASTRAND
K	SAME AS FOR FASTRAND
A	SAME AS FOR FASTRAND

THE OPTION TO SPECIFY A FILE AS TEMPORARY IS:

T	SAME AS FOR FASTRAND

THE FOLLOWING OPTIONS 'POOL' FACILITIES WITHIN A RUN WHEN A DYNAMIC FREE OR A @FREE STATEMENT IS ENCOUNTERED:

S	HOLD THE PHYSICAL ASSIGNMENT FOR THIS FILE NAME, I.E., FREE ONLY THE FILE AND SAVE THE UNIT.

THE FOLLOWING OPTIONS, CALLED THE 'MODE OPTIONS', CORRESPOND TO THE 'MODES' AVAILABLE WITH THE 'SET MODE' FUNCTION OF THE MAGNETIC TAPE HANDLER:

L	LOW DENSITY
M	MEDIUM DENSITY
H	HIGH DENSITY
E	EVEN PARITY
O	ODD PARITY
B	BINARY (NO TRANSLATE)
I	DECIMAL (TRANSLATE) HARDWARE IS USED IF AVAILABLE, OTHERWISE THE STANDARD SOFTWARE CONVERSION ROUTINE IS USED(BCD TO FIELDATA ON INPUT AND THE OPPOSITE ON OUTPUT). THE 'E' OPTION IS ASSUMED ONLY IF SOFTWARE IS USED.

IF THE EQUIPMENT 'TYPE' IS NINE-CHANNEL PER FRAME(SEE BELOW), THE DENSITY IS FIXED AT HIGH, THE PARITY IS FIXED AT ODD, AND ANY ATTEMPT TO CHANGE THESE SETTINGS IS ILLEGAL. HARDWARE TRANSLATION IS NOT AVAILABLE ON NINE-CHANNEL PER FRAME UNITS.

WHEN A FILE IS TO BE CATALOGUED, THE OPTIONS PLACED IN THE DIRECTORY ARE THOSE WHICH WERE IN EFFECT AT THE TIME OF FIRST I/O REFERENCE BY A PROGRAM WITHIN THE RUN (OTHER THAN THE 'SET MODE' REFERENCE).

THE FIELD 'TYPE/UNITS/LOG/NOISE' IS CALLED THE 'FACILITIES' FIELD AND NORMALLY DOES NOT HAVE TO BE SPECIFIED IF THE FILE IS CURRENTLY CATALOGUED.

THE SUB-FIELD 'TYPE' IS USED TO SHOW THAT THE @ASG STATEMENT IS FOR MAGNETIC TAPE AND CONTAINS A SYMBOL DENOTING THE EXACT TYPE OF TAPE UNITS REQUIRED. THIS SPECIFICATION IS REQUIRED IF THE FILE IS NOT PRESENTLY CATALOGED. IT MAY BE SPECIFIED, BUT IS IGNORED IF CURRENTLY CATALOGUED. ALLOWABLE TYPES AND THEIR MEANINGS ARE AS FOLLOWS:

T	TAPE, TYPE INDEPENDENT
C	UNISERVOS VIIIC, VIC, AND IVC
U	UNISERVOS VIIIC AND VIC
8C	UNISERVO VIIIC
6C	UNISERVO VIC
4C	UNISERVO IVC
3A	UNISERVO IIIA
2A	UNISERVO IIA

THE USE OF TYPE 'T' OR 'C' IS ENCOURAGED AS IT GIVES THE SYSTEM MORE FREEDOM IN ASSIGNING UNITS. WHEN USING TYPE 'T', ONLY THOSE FUNCTIONS AND OPTIONS COMPATIBLE WITH ALL TYPES OF UNITS MAY BE SPECIFIED.

SOME INSTALLATIONS MAY NOT HAVE '9-CHANNEL/FRAME' CAPABILITIES ON ALL UNITS WHERE POSSIBLE. IN ADDITION THERE MAY NOT BE 'TRANSLATION' HARDWARE ON ALL TAPE CHANNELS WHERE POSSIBLE. IN ORDER TO SELECT THIS EQUIPMENT, THE CHARACTER '9' OR THE CHARACTER 'B' MAY BE ADDED TO THE 'TYPE' SYMBOLS TO INDICATE '9-CHANNEL/FRAME' UNIT OR 'TRANSLATE' CHANNEL, RESPECTIVELY. AS AN EXAMPLE, IF AN VIIIC UNIT WITH '9-CHANNEL' CAPABILITIES IS NEEDED (BUT NOT AVAILABLE ON ALL 8C UNITS), THE 'TYPE' SUBFIELD WOULD CONTAIN '8C9'. THE SYMBOL '6CB' WOULD CALL FOR A UNISERVO 6C CHANNEL WITH THE HARDWARE TRANSLATION FEATURE. THE COMBINATION OF TB AND T9 IS NOT ACCEPTABLE.

THE SYSTEM DOES NOT ASSUME THE TRANSLATE OPTION (I) IF A CHANNEL WITH TRANSLATION EQUIPMENT WAS REQUESTED. THIS ACTION MUST BE CALLED FOR EITHER ON THE @ASG STATEMENT OR BY USE OF THE 'SET MODE' FUNCTION OF THE MAGNETIC TAPE HANDLER.

THE SUB-FIELD 'UNITS' IS AN INTEGER (1 OR 2) SPECIFYING THE NUMBER OF UNITS REQUIRED. IF OMITTED OR AN INTEGER OTHER THAN 1 OR 2 THE NUMBER OF UNITS IS ASSUMED TO BE ONE. A MAXIMUM OF 2 UNITS PER FILE IS ALLOWED. THE VALUE SUPPLIED IN THE @ASG STATEMENT IS ALWAYS USED FOR BOTH CATALOGUED AND TEMPORARY TAPE FILES. WHEN 2 UNITS ARE REQUESTED THEY WILL BE ON A COMMON CHANNEL. THESE UNITS WILL BE THE SAME TYPE.

THE SUB-FIELD 'LOG', IF SPECIFIED, MUST BE A SINGLE LETTER TO INDICATE A 'LOGICAL CHANNEL'. THE SYSTEM WILL ATTEMPT TO ASSIGN ALL FILES WITH THE SAME LETTER TO THE SAME PHYSICAL CHANNEL. THE LETTER IS NOT PLACED IN THE DIRECTORY ON CATALOGUING. ON REFERENCING A CATALOGUED FILE, THE LETTER CAN BE SPECIFIED AND WILL BE HONORED IF POSSIBLE. SPECIFICATION OF THE 'LOG' SUB-FIELD IS NEVER REQUIRED.

THE 'NOISE' SUBFIELD IS AN INTEGER IN THE RANGE 1-99 THAT WILL OVERRIDE THE SYSTEM STANDARD IF PRESENT. THE SYSTEM STANDARD NOISE CONSTANT IS ASSUMED IF THE 'NOISE' SUBFIELD IS NOT SPECFIED.

THE FIELD 'REEL1/REEL2/.../REELN' MAY BE USED TO LIST THE SPECIFIC REELS TO BE USED AND IS CALLED THE 'REEL' FIELD. THE SUB-FIELDS (REEL1/REEL2/ETC.) CONTAIN REEL IDENTIFIERS, HEREAFTER CALLED NUMBERS. EACH SUB-FIELD IS LIMITED TO A MAXIMUM OF SIX ALPHANUMERIC CHARACTERS. THE REEL SPECIFICATIONS TAKE ON A PARTICULAR MEANING DEPENDING ON THE STATUS OF THE FILE:

FILES BEING CATALOGUED (C OR U OPTION):

THE FIELD MAY BE EMPTY THEREBY SPECIFYING THAT THE EXECUTIVE REQUESTS THE OPERATOR TO MOUNT A BLANK REEL(S) AND SUPPLY THE REEL NUMBER(S) FOR CATALOGUING

THE FIELD MAY CONTAIN THE SPECIFIC REEL NUMBERS IN WHICH CASE THEY ARE USED AND CATALOGUED IN THE ORDER GIVEN. ALL GIVEN REEL NUMBERS ARE PLACED IN THE DIRECTORY, EVEN IF SOME OR ALL REELS WERE NOT USED. IF ADDITIONAL REELS ARE REQUESTED (VIA A REEL SWAP AFTER LAST GIVEN REEL IS USED), THE SYSTEM WILL THEN ASK FOR BLANK REELS AND INCLUDE THEIR NUMBERS IN THE DIRECTORY ENTRY.

FILES CURRENTLY CATALOGUED:

NORMALLY THE FIELD WILL BE VOID THUS INDICATING THAT THE REELS LISTED IN THE DIRECTORY ARE TO BE USED IN THE ORDER IN WHICH THEY WERE CREATED.

IF REEL NUMBERS ARE SUPPLIED, THEY MUST BE OF THE SET LISTED IN THE DIRECTORY, BUT MAY BE A SUBSET AND/OR IN ANY ORDER. THIS FEATURE ALLOWS THE USER TO OMIT AND/OR ACCESS THEM IN ANY ORDER. IF AN INVALID REEL NUMBER IS SUPPLIED, THE @ASG STATEMENT IS NOT HONORED.

IN EITHER OF THE ABOVE CASES (REEL NUMBERS TAKEN IN ORDER FROM THE DIRECTORY OR TAKEN FROM A SUBSET) WHEN THE KNOWN REELS ARE EXHAUSTED AND ADDITIONAL REELS ARE REQUESTED, BLANK REELS WILL BE USED AND THEIR NUMBERS ADDED TO THE DIRECTORY. (THIS FEATURE IS NOT ALLOWED IF THE FILE IS CATALOGUED IN THE 'READ-ONLY' STATE).

FILES NOT CATALOGUED AND NOT TO BE CATALOGUED (TEMPORARY):

IF THE REEL FIELD IS NOT SPECIFIED, THE OPERATOR WILL BE REQUESTED TO MOUNT BLANK REELS BUT THE REEL NUMBERS WILL NOT BE REQUIRED FROM THE OPERATOR.

IF REEL NUMBERS ARE GIVEN ON THE @ASG STATEMENT, THEY WILL BE USED IN THAT ORDER. WHEN THE GIVEN REELS ARE EXHAUSTED AND ADDITIONAL REELS ARE REQUESTED, BLANKS WILL BE USED BUT REEL NUMBERS ARE NOT REQUESTED.

WHEN A TAPE ASSIGNMENT IS MADE, THE FOLLOWING INFORMATION IS MADE AVAILABLE TO THE ON-SITE OPERATOR:

1. THE RUN-ID.
2. THE 'FILE' PORTION OF THE 'EXTERNAL' NAME.
3. THE UNIT(S) ASSIGNED.
4. THE INITIAL REEL(S) TO BE MOUNTED.

AS ADDITIONAL REELS ARE NEEDED, ONLY THE UNIT AND REEL NUMBER ARE GIVEN. THE FORMAT AND INTERPRETATION OF THE MESSAGES INVOLVED WITH TAPE MOUNTING IS DEFINED IN THE FILE CONTROL SYSTEM (SECTION 12).

THE FOLLOWING ARE EXAMPLES OF THE USE OF THE @ASG CONTROL STATEMENT FOR TAPE FILES.

```
@ASG   FILEY
```

FILEY IS CATALOGUED AND ALL NECESSARY OPTIONS, FACILITY REQUIREMENTS, AND REEL NUMBERS ARE TAKEN FROM THE DIRECTORY IF A FIND IS MADE IN THE DIRECTORY. NOTE: THE PROJECT ID OF THE

```
               CURRENT RUN IS USED AS A QUALIFIER.  IF THE NAME
               DOES NOT EXIST IN THE DIRECTORY A TEMPORARY
               FASTRAND FORMAT FILE IS ASSUMED. (@ASG,T FILEY,F)

        @ASG,T   FILEY,T///36

               FILEY IS A TEMPORARY FILE REQUIRING ONE UNIT
               OF THE SYSTEMS CHOOSING, AND ONE OR MORE SCRATCH
               REELS WILL BE USED.  THE NOISE CONSTANT IS TO
               BE SET TO 36 CHARACTERS.

        @ASG,TEL FILEY,6C/2,N432

               FILEY IS A TEMPORARY FILE THAT REQUESTS 2 VIC
               UNITS, IT IS TO BE RECORDED IN EVEN PARITY AND
               LOW DENSITY. REEL NUMBER N432 IS TO BE USED.

        @ASG,CR FILEY,8C9

               FILEY IS TO BE CATALOGUED IF THE RUN TERMINATES
               NORMALLY.  ONE VIIIC UNIT WITH 9-CHANNEL/FRAME
               CAPABILITIES IS REQUIRED.  THE FILE WILL BE
               CATALOGUED IN THE 'READ-ONLY' STATE.

        @ASG,D   FILEY/4AB96,8C//A,N212

               FILEY IS CURRENTLY CATALOGUED BUT IS TO BE
               RELEASED WHEN THE RUN TERMINATES NORMALLY.  A
               KEY OF 4AB96 IS REQUIRED TO READ THIS FILE.
               THE 8C UNIT IS TO BE ON LOGICAL 'A' AND REEL
               N212 IS TO BE USED.

        @ASG,U   FILEY/492671/RA1234,8C/2,707/708/709/710

               FILEY IS TO BE CATALOGUED, REQUIRES TWO VIIIC UNITS
               ON ANY CHANNEL.  REELSS 707, THRU 710
               ARE TO BE USED.  THE FILE WILL BE LOCKED WITH
               THE GIVEN 'READ' AND 'WRITE' KEYS.

        @ASG,T   FILEY,T,SCRTCH

               FILEY IS A TEMPORARY FILE AND THE SYMBOL
               'SCRTCH' IS USED AS A REEL NUMBER, WHICH MAY
               INDICATE THAT THE OPERATOR SHOULD MOUNT A
               SCRATCH REEL (DEPENDING ON THE INSTALLATION'S
               OPERATING CONVENTIONS).
```

MAGNETIC TAPE UNITS MAY BE ASSIGNED FOR A PARTICULAR CHANNEL (WITH THE EXECUTIVE CHOOSING THE UNIT) OR A SPECIFIC UNIT CAN BE ASSIGNED. (SPECIFIC ASSIGNMENT IS NOT RECOMMENDED EXCEPT IN THESE CASES SUCH AS MAINTENANCE ROUTINES, REAL-TIME PROGRAMS, OR SPECIAL HARDWARE WHICH DICTATES ABSOLUTE ASSIGNMENTS).

A UNIT THAT IS IN A RESERVE STATUS VIA AN UNSOLICITED KEY-IN 'RV' (SEE SECTION 11.3.1.1) IS ONLY ASSIGNABLE BY SPECIFYING THE ABSOLUTE CHANNEL AND UNIT ON THE ASG STATEMENT.

FOR THE CASE WHERE A PARTICULAR CHANNEL IS SPECIFIED AND THE EXECUTIVE CHOOSES THE UNIT(S), THE FORMAT OF THE FACILITIES FIELD OF THE TAPE @ASG STATEMENT IS:

```
      TYPE/UNITS
```

WHERE 'TYPE' SPECIFIES THE CHANNEL BY THE SYMBOL

```
      CXX
```

WHERE 'XX' IS THE CHANNEL NUMBER (0 TO 15) AND 'UNITS' IS THE SAME AS DESCRIBED EARLIER. CATALOGUING IS THE SAME AS INDICATED EARLIER.

FOR THE CASE WHERE A PARTICULAR UNIT (OR 2 UNITS) IS REQUIRED THE FORMAT IS:

```
CXX/UYY/UZZ
```

WHERE 'YY' AND 'ZZ' ARE THE UNIT NUMBERS.

CONSIDER THE FOLLOWING EXAMPLES:

```
@ASG   FILEY,C12/2
```

FILEY IS ASSIGNED TWO UNITS OF THE SYSTEMS CHOOSING FROM CHANNEL TWELVE.

```
@ASG   FILEY,C12/U6,29416
```

FILEY IS ASSIGNED UNIT 6 ON CHANNEL 12, REEL 29416 IS TO BE USED.

```
@ASG   FILEY,C12/U6/U4
```

UNITS 6 AND 4 FROM CHANNEL 12 ARE ASSIGNED TO THE FILE.

5.5.1.5. THE DRUM @ASG STATEMENT

THE DRUM @ASG STATEMENT IS USED ONLY WHEN A FILE 'MUST' RESIDE ON A RANDOM ACCESS DEVICE AND WILL BE BUILT AND READ BY THE USERS DIRECT REFERENCE TO THE MAGNETIC DRUM HANDLER (SEE SECTION 10.3 ON THE 'MAGNETIC DRUM HANDLER' FOR FUNCTIONS AVAILABLE). NORMALLY, AS WHEN THE BLOCK BUFFERING PACKAGE OR ITEM HANDLER IS USED, FILES ARE PLACED ON DRUM BY USING A FASTRAND @ASG STATEMENT WITH SIMULATED FASTRAND AS THE 'TYPE'

FILES SPECIFED BY THE DRUM @ASG STATEMENT CANNOT BE CATALOGUED AND ARE TREATED AS TEMPORARY SCRATCH FILES. THEY ARE RELEASED AT RUN TERMINATION OR WHEN A @FREE COMMAND IS ENCOUNTERED FOR A PARTICULAR FILE.

THE FORMAT OF THE DRUM @ASG STATEMENT IS:

```
@ASG  NAME,TYPE/LOCATIONS
```

WHERE THE 'NAME' FIELD IS ALWAYS THE SAME AS FOR THE FASTRAND @ASG STATEMENT. 'KEY' SUB-FIELDS ARE NOT AVAILABLE SINCE DRUM FILES CAN NOT BE CATALOGUED.

THE 'TYPE' FIELD CONTAINS ONE OF THE FOLLOWING SYMBOLS:

D	DRUM, TYPE INDEPENDENT
D4	FH 432
D8	FH 880
D17	FH 1782 (DELAYED IMPLEMTATION)

TYPE 'D' IS RECOMMENDED AND SHOULD BE USED WHENEVER POSSIBLE SO THAT THE SYSTEM HAS MORE FREEDOM IN ALLOCATING FILE SPACE.

THE SUB-FIELD 'LOCATIONS' SPECIFIES THE NUMBER OF WORDS NEEDED FOR THE FILE. THE NUMBER WILL BE TREATED AS OCTAL IF THE FIRST DIGIT IS A ZERO. AN ERROR STATUS IS RETURNED TO THE PROGRAM IF IT ATTEMPTS TO REFERENCE OUTSIDE OF THE REQUESTED AREA.

AN EXAMPLE OF A DRUM @ASG STATEMENT WOULD BE:

```
@ASG   FILEX,D/2000
```

FILEX REQUIRES TWO THOUSAND LOCATIONS ON A
MAGNETIC DRUM OF THE SYSTEMS CHOOSING.

5.5.1.6. THE ARBITRARY DEVICE @ASG STATEMENT

THE ARBITRARY DEVICE @ASG STATEMENT IS USED TO ASSIGN ALL EQUIPMENT EXCEPT FASTRAND, DRUM AND MAGNETIC TAPE(SEE RESPECTIVE @ASG STATEMENT FORMATS). ITS MAIN USE IS IN ASSIGNING SPECIAL I/O DEVICES AND COMMUNICATIONS EQUIPMENT IN THAT MOST STANDARD PERIPHERALS (READERS,PUNCHES,ETC.) NEED NOT BE ASSIGNED SINCE THEY ARE USED VIA THE SYMBIONT MECHANISM.

THE FORMAT OF THE ARBITRARY DEVICE @ASG STATEMENT IS:

```
@ASG,OPTIONS  NAME,TYPE
```

THE FIELD 'NAME' GIVES THE 'EXTERNAL' IDENTIFICATION OF THE SELECTED UNIT.

THE 'TYPE' FIELD CONTAINS EITHER:

1. THE SYMBOLIC NAME OF A CLASS OF DEVICES, WHERE THE SYSTEM CHOOSES THE UNIT, IF MORE THAN ONE;
2. AN ABSOLUTE CHANNEL, WHERE THE SYSTEM CHOOSES THE UNIT; OR,
3. AN ABSOLUTE CHANNEL/UNIT.

SYMBOLIC NAMES OF STANDARD DEVICES OTHER THAN MAGNETIC TAPE AND MASS STORAGE ARE LISTED BELOW. THE USER MUST SELECT THE SYMBOLIC NAMES TO BE GIVEN TO NONSTANDARD DEVICES AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - CONNECT CHANNEL AND NAME ASSOCIATION).

CRD	CARD SUBSYSTEM
PTP	PAPER TAPE SUBSYSTEM
P	PRINTING DEVICE
HSP	HIGH SPEED PRINTER
1004	1004 PRINTING UNIT

AT SYSTEM GENERATION TIME, GROUPS OF DEVICES, CHANNELS, OR SPECIFIC UNITS MAY BE GIVEN SYMBOLIC NAMES (SEE SYSTEM GENERATION STATEMENTS - NAME AND C/U ASSOCIATION). THESE NAMES ARE USED IN THE 'TYPE' FIELD TO OBTAIN AN ASSIGNMENT. FOR EXAMPLE, COMMUNICATIONS DEVICES ARE DEFINED AT GENERATION TIME AS UNITS UNDER A GROUP (CLASS) IDENTITY, CALLED 'THE LT GROUP IDENTITY' (SEE THE SECTION ON 'COMMUNICATIONS MULTIPLEXOR HANDLER'). THE LT GROUP IDENTITY IS USED IN THE 'TYPE' FIELD AND THE EXECUTIVE CHOOSES THE UNIT.

FOR ABSOLUTE CHANNEL ASSIGNMENT, THE 'TYPE' FIELD CONTAINS THE SYMBOL:

```
CXX
```

WHERE 'XX' IS THE CHANNEL NUMBER (0 TO 15).

FOR ABSOLUTE UNIT ASSIGNMENT, THE 'TYPE' FIELD CONTAINS:

```
CXX/UYY
```

WHERE 'YY' IS THE UNIT NUMBER. THE STATEMENT:

```
@ASG  NAME,C6/U2
```

CAUSES THE ASSIGNMENT OF UNIT 2 ON CHANNEL 6.

5.5.2. THE @MODE STATEMENT

THE @MODE STATEMENT IS USED TO CHANGE THE 'MODE' SETTINGS OF A TAPE FILE. THESE MODES ARE SET INITIALLY WHEN THE @ASG STATEMENT IS PROCESSED AND MAY ALSO BE CHANGED

INTERNALLY BY USE OF THE 'SET MODE' FUNCTION OF THE MAGNETIC TAPE HANDLER. THE FORMAT OF THE @MODE STATEMENT IS:

```
@MODE,OPTIONS   NAME,NOISE
```

THE FIELD 'NAME' IS THE SAME AS FOR THE @ASG STATEMENT. THE FILE MUST BE CURRENTLY ASSIGNED TO THE RUN (AN @ASG STATEMENT WITH THIS NAME MUST PRECEDE THE @MODE STATEMENT). IF THE FILE IS NOT ASSIGNED (NEVER ASSIGNED OR RELEASED VIA A @FREE STATEMENT), THE RUN IS PLACED IN THE ERROR MODE. THE NOISE SUB-FIELD IS ALSO THE SAME AS FOR THE @ASG STATEMENT.

THE 'OPTIONS' FIELD MAY CONTAIN THE FOLLOWING OPTIONS:

L	LOW DENSITY
M	MEDIUM DENSITY
H	HIGH DENSITY
E	EVEN PARITY
O	ODD PARITY
I	DECIMAL (TRANSLATE)
B	BINARY (NO TRANSLATE)

WITH THE @MODE STATEMENT, OPTIONS (MODES) ARE NEVER ASSUMED IN THE ABSENCE OF OTHERS. THE SPECIFIED OPTIONS ARE NOT PLACED IN THE CATALOGUE SINCE THEY APPLY ONLY TO THE CURRENT ASSIGNMENT.

DIAGNOSTIC MESSAGES

THE GENERALIZED FORMAT AS DESCRIBED IN SECTION 5.5.1.4 FOR THE ASG STATEMENT IS USED. THE FOLLOWING TABLE DEFINES THE MEANING OF THE BITS (1=SET) WHEN SET. BITS ARE NUMBERED 35-0 READING LEFT TO RIGHT:

BIT	SOURCE		DESCRIPTION OF MEANING IF BIT SET
	RUN	CSF	
35	K		REQUEST NOT ACCEPTED - EXAMINE REST OF BITS AS TO WHY.
34	K		FIELD ERROR - NOISE CONSTANT
33	K		NAME NOT IN PCT FOR THIS RUN (FILE HAS NOT BEEN ASSIGNED TO THIS RUN)
32			UNUSED
31	K		EQUIPMENT TYPE NOT TAPE FOR FOUND FILE IN PCT
30 , , 18			UNUSED
17	K		OPTION CONFLICT, ONLY ONE OPTION IN THE FOLLOWING 3 GROUPS ALLOWED (1) 'MLH' (2) 'EO' (3) 'BI'
16 , , , 0			UNUSED

NOTE: K IN SOURCE FIELD MEANS THAT RUN WILL BE TERMINATED

5.5.3. THE @CAT STATEMENT

CATALOGUING IS NORMALLY DONE IN THE COURSE OF CREATING THE FILE WHERE THE @ASG STATEMENT SPECIFIES THAT THE FILE IS TO BE CATALOGUED. THE CATALOGUING IS DONE WHEN THE RUN TERMINATES OR WHEN A @FREE STATEMENT IS FOUND. THE NEED ARISES TO BE ABLE TO CATALOGUE ONE OR MORE FILES WITHOUT HAVING THEM (AND THE REQUIRED FACILITIES) ASSIGNED TO THE RUN. THIS MAY BE THE CASE WHEN BUILDING THE INITIAL MASTER DIRECTORY OR WHEN FOR ANY REASON A PRE-PREPARED TAPE FILE IS TO BE CATALOGUED. THE @CAT STATEMENT IS USED FOR THIS PURPOSE. THE FILE IS CATALOGUED BUT IS NOT ASSIGNED TO THE RUN. NO FACILITIES ARE ASSIGNED.

IN ANY CASE, USE OF THE @CAT STATEMENT IS ILLEGAL IF THE NAMED FILE IS CURRENTLY ASSIGNED TO THE RUN.

IF THE'TYPE'FIELD IS OMITTED ON THE STATEMENT, 'F' IS ASSUMED. E.G. '@CAT NAME' IS ASSUMED TO BE '@CAT NAME,F'

AS IT PERTAINS TO THE CATALOGUING OF EXISTING TAPE FILES, THE FORMAT OF THE @CAT STATEMENT IS IDENTICAL TO THAT OF THE TAPE @ASG STATEMENT, NAMELY:

```
@CAT,OPTIONS  NAME/KEY1/KEY2,TYPE/UNITS/LOG/NOISE,;
```

REEL1/REEL2/.../REELN

THE 'NAME' AND 'TYPE' SPECIFICATIONS ARE REQUIRED AND AT LEAST ONE 'REEL' NUMBER MUST BE GIVEN. KEY1 AND/OR KEY2 ARE SPECIFIED IF THE FILE IS TO BE CATALOGUED WITH 'READ' AND/OR 'WRITE' LOCK, RESPECTIVELY. ALLOWABLE 'OPTIONS' ARE:

R — PLACE IN 'READ-ONLY' STATE.

W — PLACE IN 'WRITE-ONLY' STATE.

P — CATALOGUE AS PUBLIC.

L,M,H,E,O,B,I — ANY OF THE 'MODE' OPTIONS USED ON THE @ASG STATEMENT. THE MODES AS SET AT SYSTEM GENERATION TIME ARE ASSUMED IN THE ABSENCE OF THESE SETTINGS.

AS IT PERTAINS TO FASTRAND FORMATTED FILES, THE FORMAT OF THE @CAT STATEMENT IS IDENTICAL TO THAT OF THE FASTRAND @ASG STATEMENT, NAMELY:

```
@CAT,OPTIONS  NAME/KEY1/KEY2,TYPE/RESERVE/GRANULE
```

THE SPECIFICATIONS FIELDS ARE INTERPRETED AS THEY ARE FOR THE @ASG STATEMENT, HOWEVER, THE ACTUAL 'RESERVE' IS NOT MADE. ALLOWABLE OPTIONS ARE:

R — PLACE IN 'READ-ONLY' STATE.

W — PLACE IN 'WRITE-ONLY' STATE.

P — SPECIFIES THAT THE FILE IS TO BE CATALOGUED AS A 'PUBLIC' FILE RATHER THAN A 'PRIVATE' FILE.

EXAMPLE OF THE @CAT STATEMENT:

```
@CAT  FILEY/A2962,8C,N297
```

FILEX IS TO BE CATALOGUED WITH THE READ KEY A2962. STANDARD MODE SETTINGS ARE PLACED IN THE DIRECTORY. THE FILE IS FOUND ON REEL NUMBER N297 AND A UNISERVO VIIIC DRIVE WILL BE USED ON FUTURE ASSIGNMENTS.

TO CHANGE THE NAME AND/OR KEY(S) OF CURRENTLY CATALOGUED FILES AND TO RELEASE (DECATALOGUE) FILES, THE READER IS REFERRED TO THE CHAPTER ENTITLED 'FILE UTILITY ROUTINES'.

DIAGNOSTIC MESSAGES

THE GENERALIZED FORMAT AS DESCRIBED IN SECTION 5.5.1.4 FOR THE ASG STATEMENT IS USED. THE FOLLOWING TABLE DEFINES THE MEANING OF THE BITS (1=SET) WHEN SET. BITS ARE NUMBERED 35-0 READING LEFT TO RIGHT.

BIT	SOURCE		DESCRIPTION OF MEANING IF BIT SET
	RUN	CSF	
35	K		REQUEST NOT NOT ACCEPTED - EXAMINE REST OF BITS AS TO WHY
34	K		FIELD ERROR - NOISE CONSTANT OR EQUIPMENT TYPE IS OTHER THAN TAPE OR FASTRAND FORMAT
33	K		FILE NAME FOUND IN RUN PCT (ALREADY ASSIGNED TO THIS RUN).
32	K		NAME ALREADY EXISTS IN DIRECTORY CONTROL FOR CATALOGUED FILES
31 ' ' ' 21			UNUSED
20	K		NO REEL NUMBER SPECIFIED FOR TAPE FILE - MUST BE AT LEAST ONE.
19 ' ' ' 18			UNUSED
17	K		OPTION CONFLICT - ONLY ONE OPTION ALLOWED FOLLOWING GROUPS (1) 'MLH' (2) 'EO' (3) 'BI' (4) 'RW'
16 ' ' ' 0			UNUSED

NOTE: K IN SOURCE FIELD MEANS THAT RUN WILL BE TERMINATED

5.5.4. THE @FREE STATEMENT

THE @FREE CONTROL STATEMENT MAKES PROVISION FOR THE DE-ASSIGNING OF A FILE, AND, OPTIONALLY, THE RELEASE OF ITS INPUT/OUTPUT FACILITIES. IN THE ABSENCE OF A @FREE STATEMENT, THE FILE AND ITS FACILITIES ARE HELD UNTIL END-RUN. FILES SHOULD BE DE-ASSIGNED AT THE MOMENT THEY ARE NO LONGER NEEDED SO AS TO ALLOW FACILITIES, REELS, AND 'EXCLUSIVE USE' AREAS TO BE ASSIGNED TO OTHER RUNS. THE FORMAT OF THE @FREE STATEMENT IS:

```
@FREE,OPTIONS   FILE NAME
```

WHERE 'FILE NAME' IS EITHER 'INTERNAL' OR 'EXTERNAL'. AN ERROR DIAGNOSTIC IS GIVEN IF THE FILE HAS NOT BEEN PREVIOUSLY ASSIGNED.

THE 'OPTIONS' FIELD MAY CONTAIN ANY OF THE FOLLOWING OPTIONS.

R RELEASES THE FILE ASSIGNED BUT RETAINS THE @USE NAME RELATIONSHIPS TO THE FILE NAME AND F-CYCLE. THE READ AND WRITE KEYS ARE LOST IF THEY WERE PRESENT ON THE @USE CARD.

A RELEASES ONLY THE @USE NAME RELATIONSHIP TO THE FILE NAME.

B RELEASES ONLY THE @USE NAME ASSOCIATION TO THE FILE NAME IF THE ATTACHED NAME IS NOT THE ONLY ATTACHMENT. OTHERWISE IT ACTS LIKE THE BLANK OPTION ON THE @FREE CARD FREEING THE FILE.

C RELEASES THE FILE AND ALL NAMES ASSOCIATED WITH THE FILE.

D DROPS A CATALOGUED FILE REGARDLESS OF HOW IT WAS ASSIGNED.

I INHIBITS FINAL CATALOGUING ACTION IF THE FILE WAS ASSIGNED WITH A 'C' OR 'U' OPTION.

X RELEASE THE EXCLUSIVE USE OPTION SET ON THE FILE BUT DOES NOT FREE THE FILE.

S HOLD THE PHYSICAL TAPE UNIT FOR THIS FILE.

A FILE THAT IS NAMED ON A @FREE STATEMENT CAN NO LONGER BE REFERENCED BY THE RUN; IT CAN OF COURSE BE RE-ESTABLISHED BY AN @ASG STATEMENT PROVIDED ITS FACILITY REQUIREMENTS CAN BE MET.

THE ACTIONS TAKEN BY THE SYSTEM WHEN A FILE IS NAMED ON A @FREE STATEMENT (AND THE 'S' OPTION WAS NOT SPECIFIED) ARE DISCUSSED BELOW.

FOR A TEMPORARY FILE (NOT CATALOGUED OR TO BE CATALOGUED):

FASTRAND - THE FASTRAND AREA IS MADE AVAILABLE AS FILE SPACE FOR OTHER RUNS.

DRUM - SAME AS FASTRAND. ALWAYS TEMPORARY.

TAPE - UNITS ARE RELEASED FOR USE BY OTHER RUNS. THE OPERATOR IS NOTIFIED THAT THE REELS ARE TO BE REMOVED AND THAT THE SYSTEM HAS NOT RECORDED THE REEL NUMBERS (HAS NOT CATALOGUED THE FILE).

OTHER EQUIPMENT (COMMUNICATIONS GEAR,ETC.) - THE DEVICE IS RELEASED FOR USE BY OTHER RUNS. ALWAYS TEMPORARY.

FOR A FILE BEING CATALOGUED (C OR U OPTION ON @ASG):

FASTRAND - CATALOGUE ENTRY IS MADE IN THE MASTER DIRECTORY AND FASTRAND AREA CONTAINING THE FILE IS HELD. THE FILE CAN NOW BE REFERENCED BY OTHER RUNS.

TAPE - CATALOGUE ENTRY CONTAINING REEL NUMBERS IS MADE, UNITS ARE RELEASED FOR OTHER RUNS. THE OPERATOR IS TOLD TO REMOVE THE FILE (REELS) AND THAT THE

REEL NUMBERS ARE BEING HELD BY THE SYSTEM FOR FUTURE CALLS ON THE FILE (THAT THE FILE WAS CATALOGUED).

FOR A FILE BEING DE-CATALOGUED (D OR K OPTION ON @ASG):

FASTRAND - SAME AS FOR A TEMPORARY FILE EXCEPT THAT THE FILE AREA IS NOT RELEASED UNTIL ALL RUNS CURRENTLY USING THE FILE HAVE ALSO FINISHED. IT IS NO LONGER AVAILABLE FOR ASSIGNMENT.

TAPE - SAME AS FOR A TEMPORARY FILE.

A TYPICAL @FREE STATEMENT IS SHOWN IN THE FOLLOWING EXAMPLE OF A PARTIAL CONTROL STREAM:

```
@ASG,C  FILEX,F/3
@ASG,T  FILEY,8C
.......
.......
@FREE  FILEX
@FREE  FILEY
```

FILEX IS A FASTRAND FILE TO BE CATALOGUED AND REQUIRES 3 TRACKS INITIALLY.
FILEY IS A TEMPORARY TAPE FILE REQUIRING 1 VIIIC UNIT.
WHEN THE @FREE STATEMENT IS ENCOUNTERED, FILEX IS CATALOGUED WITH THE FILE AREA HELD FOR FUTURE REFERENCE. FOR FILEY, THE VIIIC UNIT IS MADE AVAILABLE TO OTHER RUNS AND THE OPERATOR IS NOTIFIED TO REMOVE THE REELS AND FOLLOW THE USERS INSTRUCTIONS AS TO THEIR DISPOSAL.

THE SYSTEM PROVIDES A LINKAGE WHICH ALLOWS THE @FREE FUNCTION TO BE REQUESTED FROM WITHIN A USER PROGRAM. THE LINKAGE DESCRIPTION IS FOUND IN SECTION 8, 'EXECUTIVE REQUEST FUNCTIONS'. THIS FEATURE ALLOWS A FILE AND ITS FACILITIES TO BE RELEASED AS SOON AS THE FILE IS NO LONGER TO BE USED.

DIAGNOSTIC MESSAGES

THE GENERALIZED FORMAT AS DESCRIBED IN SECTION 5.5.1.4 FOR THE ASG STATEMENT IS USED. THE FOLLOWING TABLE DEFINES THE MEANING OF THE BITS (1=SET) WHEN SET. BITS ARE NUMBERED 35-0 READING LEFT TO RIGHT

BIT	SOURCE		DESCRIPTION OF MEANING IF BIT SET
	RUN	CSF	
35	K		REQUEST NOT ACCEPTED - EXAMINE REST OF BITS AS TO WHY.
34			UNUSED
33	K		FILE IS NOT CURRENTLY ASSIGNED TO THIS RUN.
32 . . . 0			UNUSED

NOTE: K IN SOURCE FIELD MEANS THAT RUN WILL BE TERMINATED

5.5.5. THE @USE STATEMENT

THE @USE CONTROL STATEMENT PROVIDES THE USER WITH THE ABILITY TO REFER TO ANY PARTICULAR FILE BY TWO OR MORE NAMES. THE NEED FOR THE ADDITIONAL NAMES ARISES FROM THREE CONDITIONS:

1) SIMPLIFY RUN CONSTRUCTION BY ALLOWING THE EQUATING OF AN 'EXTERNAL' NAME TO A SHORTER 'ATTACHED' NAME.
2) RESOLVE IDENTICAL FILE NAME PORTIONS TO 'EXTERNAL' NAMES.
3) CONNECT NAMES CODED INTO PROGRAMS TO 'EXTERNAL' OR 'ATTACHED' NAMES.

THE FORMAT OF THE @USE STATEMENT IS AS FOLLOWS:

```
@USE  INTERNAL,EXTERNAL
         OR
 @USE  INTERNAL,INTERNAL
```

WHERE 'INTERNAL' IS THE 12 CHARACTER NAME BY WHICH THE FILE IS REFERRED TO WITHIN THE RUN OR AFTER THE @USE STATEMENT IN THE CONTROL STREAM AND 'EXTERNAL' IS THE NAME UNDER WHICH THE FILE IS TO BE ASSIGNED OR IS ASSIGNED (AND POSSIBLY CATALOGUED). THE 'EXTERNAL' NAME HAS THE FORM:

```
'QUALIFIER*FILE/KEY1/KEY2
```

IF THE 'INTERNAL' NAME IS THE SAME AS THE 'FILE' PORTION OF THE 'EXTERNAL' NAME, THEN THE 'INTERNAL' NAME AUTOMATICALLY POINTS TO THAT 'EXTERNAL' FILE AND A @USE STATEMENT IS NOT NEEDED. IF THE 'INTERNAL' NAME IS NOT THE SAME AS THE 'FILE' PORTION OF SOME 'EXTERNAL' FILE, THEN A USE STATEMENT IS REQUIRED BEFORE THE 'INTERNAL' NAME WILL BE HONORED.

ASSUME THAT THE 'INTERNAL' NAME IS 'FILEA' AND THE 'EXTERNAL' NAME IS 'PROJ1*FILEA'. THE FILE 'PROJ1*FILEA' WILL AUTOMATICALLY BE USED UNLESS A @USE STATEMENT IS PRESENTED MAKING 'FILEA' POINT TO A DIFFERENT 'EXTERNAL' FILE, FOR EXAMPLE,IN THE STATEMENT:

```
@USE  FILEA,PROJ1*FILEZ
```

'PROJI*FILEZ' IS THE 'EXTERNAL' FILE.

THE @USE STATEMENT CAUSES THE 'INTERNAL' NAME TO BE ATTACHED TO THE SPECIFIED 'EXTERNAL' FILE. ALL SUCH ATTACHED NAMES ARE MAINTAINED FOR AN 'EXTERNAL' FILE. (THE 'INTERNAL' NAME NO LONGER POINTS TO ANY OTHER 'EXTERNAL' FILE AND IF THE FILE HAD A PREVIOUS ATTACHMENT, IT IS MAINTAINED RATHER THAN BEING DELETED.) UNLESS A CONFLICT OF NAMES OCCURS, EITHER THE 'FILE' PORTION OF THE EXTERNAL NAME OR THE 'ATTACHED' NAME CAN BE USED ON A REFERENCE FROM ANYWHERE WITHIN A PROGRAM OR CONTROL STREAM. NO RESTRICTION IS PLACED ON AN 'ATTACHED' NAME BEING THE SAME AS THE 'FILE' PORTION OF AN EXTERNAL NAME. THE LIST OF 'ATTACHED' NAMES IS ALWAYS SEARCHED FIRST ON AN I/O REFERENCE - WITH THE 'FILE' PORTIONS OF THE EXTERNAL NAMES USED NEXT ON A NO-FIND. IF AN 'ATTACHED' NAME IS THE SAME AS THE 'FILE' PORTION OF SOME EXTERNAL FILE, THAT 'EXTERNAL' FILE MUST HAVE A @USE STATEMENT IN EFFECT BEFORE THE FILE CAN BE USED. THIS IS ALSO TRUE FOR A RECENTLY ASSIGNED FILE (VIA @ASG STATEMENT), SINCE THE 'FILE' PORTION OF ITS EXTERNAL NAME MAY BE IN THE 'ATTACHED' LIST, POINTING TO SOME OTHER FILE. HOWEVER, IN THIS CASE THE CONFLICT CAN BE REMOVED IF THE REFERENCE IS MADE BY A CONTROL STATEMENT. THE 'QUALIFIER' OR AT LEAST THE '*' WILL SPECIFY THAT THE NAME IS NOT AN 'ATTACHED' NAME.

IT IS ASSUMED THAT THE READER IS FAMILIAR WITH THE SECTION ENTITLED 'NOTATION FOR FILE NAMES' PRESENTED EARLIER. THE SECTION CONTAINS FURTHER EXAMPLES OF THE @USE STATEMENT AS WELL AS SOME CASES WHERE THE @USE STATEMENT -MUST- BE USED.

THE @USE STATEMENT ALLOWS THE PERSON SETTING UP THE RUN TO CHOOSE 'EXTERNAL' FILE NAMES DESCRIPTIVE TO HIS RUN, OR TO A PARTICULAR CATALOGUED FILE. IT ALSO ALLOWS A PARTICULAR 'INTERNAL' NAME(IN TWO OR MORE EXECUTIONS) TO POINT TO DIFFERENT 'EXTERNAL' FILES DURING THE COURSE OF A RUN; OR FOR DIFFERENT 'INTERNAL' NAMES TO POINT TO THE SAME 'EXTERNAL' FILE.

5.5.6. THE @ELT STATEMENT

THE @ELT CONTROL STATEMENT INTRODUCES AN ELEMENT INTO A PARTICULAR PROGRAM-FILE FROM THE CONTROL STREAM. IT MAY ALSO BE USED TO MAKE CORRECTIONS TO A SOURCE ELEMENT IN A PROGRAM-FILE. THE ELEMENT OR THE CORRECTIONS FOLLOW THE @ELT STATEMENT IN THE CONTROL STREAM.

THE FORMAT OF THE @ELT STATEMENT IS:

```
@ELT,OPTIONS   ELEMENT1,ELEMENT2,SENTINEL
```

THE OPTIONS ARE:

A	ABSOLUTE ELEMENT
R	RELOCATABLE ELEMENT
S	SYMBOLIC ELEMENT
D	DATA ELEMENT
I	INSERT. INITIAL INSERTION OF AN ELEMENT INTO A PROGRAM FILE.
U	UPDATE. PRODUCE A NEW CYCLE OF SOURCE LANGUAGE.
L	PRODUCE A LISTING OF THE COMPLETE SOURCE ELEMENT.

THE OPTIONS 'A', 'R', 'S', AND 'D' IDENTIFY THE ELEMENT TYPE. TYPES 'S' AND 'D' ARE BOTH CONSIDERED SOURCE LANGUAGE ELEMENTS AND MAY BE CORRECTED IN THE SAME MANNER. (SEE SECTION OF THIS CHAPTER ON 'PROCESSOR CALL STATEMENTS'). A SOURCE LANGUAGE ELEMENT IN A PROGRAM-FILE HAS THE SAME FORMAT AS THE SYSTEM DATA FILE. THE FORMAT IS DESCRIBED IN THE CHAPTER ENTITLED 'DATA HANDLING'. WHEN AN 'A', 'R', 'S', OR 'D' OPTION IS NOT PRESENT, THE 'S' OPTION IS ASSUMED.

THE @ELT STATEMENT INITIATES THE ELEMENT PROCESSOR WHICH OPERATES IN ONE OF TWO MODES. IT INSERTS NEW ELEMENTS INTO THE PROGRAM-FILE FROM THE CONTROL STREAM OR UPDATES AN ELEMENT ALREADY IN THE PROGRAM-FILE.

THE FIELD 'ELEMENT1' IDENTIFIES THE INPUT ELEMENT BY FILE, ELEMENT NAME, VERSION, AND CYCLE (WHEN APPROPRIATE). FIELD 'ELEMENT2', IF SPECIFIED, IDENTIFIES THE NEW OUTPUT ELEMENT.

WHEN THE 'I' OPTION IS SPECIFIED, THE ELEMENT IN THE CONTROL STREAM IS GIVEN THE NAME SPECIFIED IN THE 'ELEMENT1' FIELD AND INSERTED INTO THE PROGRAM-FILE SPECIFIED IN THE 'ELEMENT1' FIELD.

WHEN THE 'U' OPTION IS SPECIFIED, THE CORRECTIONS IN THE CONTROL STREAM ARE APPLIED TO THE ELEMENT IDENTIFIED IN THE 'ELEMENT1' FIELD, AND A NEW CYCLE OF THE SOURCE LANGUAGE IS PRODUCED.

WHEN THE 'ELEMENT2' FIELD IS PRESENT AND THE 'U' OPTION IS NOT SPECIFIED, THE CORRECTIONS IN THE CONTROL STREAM ARE APPLIED TO 'ELEMENT1', AND A NEW SOURCE ELEMENT IS PRODUCED. IT WILL BE GIVEN THE NAME SPECIFIED IN THE 'ELEMENT2' FIELD AND INSERTED INTO THE PROGRAM-FILE SPECIFIED IN 'ELEMENT2' FIELD.

THE 'L' OPTION WILL PRODUCE A COMPLETE LISTING OF A SOURCE ELEMENT. THE 'L' OPTION IS NOT APPLICABLE FOR ABSOLUTE OR RELOCATABLE ELEMENTS.

WHEN THE 'U' OR 'I' OPTION IS NOT PRESENT AND THE 'ELEMENT2' FIELD IS VOID, THE 'L' OPTION IS ASSUMED AND 'ELEMENT1' WILL BE LISTED.

THE 'DATA ELEMENT' MAY CONTAIN CONTROL STATEMENTS. THEREFORE, THE DATA FOLLOWING THE @ELT,D STATEMENT MUST BE TERMINATED WITH AN @END STATEMENT WITH A SENTINEL EXACTLY THE SAME AS FOUND ON THE @ELT,D STATEMENT. THE SENTINEL FIELD NEED NOT BE CODED (BLANK SENTINELS). IT IS A SIX CHARACTER FIELD USED TO SEARCH OUT THE PROPER @END SENTINEL. ALL IMAGES WILL BE PASSED INTO THE DATA ELEMENT BEING CREATED UNTIL AN @END COMMAND IS FOUND WITH THE SAME CHARACTER STRING. THE @ELT,D STATEMENT MAY BE USED TO INSERT @RUN OR @ADD CONTROL STREAMS INTO A PROGRAM-FILE AS ELEMENTS WHICH MAY BE CALLED LATER BY THE @START OR @ADD STATEMENT.

ELEMENT TYPES 'A', 'R', AND 'S' ARE TERMINATED BY THE NEXT NON-@EOF CONTROL STATEMENT IN THE CONTROL STREAM. THEY NEED NO CORRESPONDING @END COMMAND, THEREFORE, NO SENTINEL IS NECESSARY.

WHEN AN ELEMENT IS PUNCHED BY A PROCESSOR OR BY PROGRAM UTILITY ROUTINE (PUR), IT IS ALWAYS PRECEDED BY A @ELT CONTROL STATEMENT. THE 'FILENAME' ON THE PUNCHED @ELT IS THAT OF THE FILE FROM WHICH THE ELEMENT WAS PUNCHED. SUCH DECKS CAN SIMPLY BECOME PART OF THE INPUT TO SUBSEQUENT RUNS. (THE FILE NAME MUST BE CHANGED IF THE ELEMENT IS TO BE ADDED TO A FILE DIFFERENT FROM THE ONE FROM WHICH IT WAS PUNCHED).

THE AUTOMATIC DELETION RULES APPLY TO THE INSERTION OF ELEMENTS BY AN @ELT CONTROL STATEMENT.

EXAMPLES

```
@ELT,I  PF.E                  . A NEW SYMBOLIC ELEMENT
...                           . 'E' IS INSERTED IN THE
...                           . PROGRAM-FILE 'PF'.
SOURCE IMAGES
...
...

@ELT,U  PF.E                  . THE CORRECTIONS FOLLOWING
...                           . THIS STATEMENT ARE APPLIED
...                           . TO THE ELEMENT 'E' OF
CORRECTIONS IMAGES            . PROGRAM-FILE 'PF'.
...                           . THE UPDATED ELEMENT 'E'
...                           . REPLACES THE OLD 'E' IN
...                           . THE PROGRAM-FILE.

@ELT    PF.E,PF.N             . THE CORRECTIONS FOLLOWING
...                           . THIS STATEMENT ARE
...                              . APPLIED TO THE ELEMENT
CORRECTION IMAGES                . 'E' TO PRODUCE A NEW
...                              . ELEMENT 'N'.
...

@ELT,L  PF.E                     . ELEMENT 'E' WILL BE LISTED

@ELT  PF.E                       . ELEMENT 'E' WILL BE LISTED

@ELT,L  PF.E                     . THE CORRECTIONS FOLLOWING
...                              . THIS STATEMENT ARE APPLIED
...                              . TO THE ELEMENT 'E'.  THE
CORRECTION IMAGES                . NEW ELEMENT WILL BE
...                              . LISTED, BUT A NEW ELEMENT
...                              . WILL NOT BE PRODUCED.

@ELT,IA  PF.E                    . A NEW ABSOLUTE ELEMENT
...                              . 'E' IS INSERTED IN THE
...                              . PROGRAM-FILE 'PF'.
ABSOLUTE IMAGES
...
...

@ELT,IR  PF.E                    . A NEW RELOCATABLE ELEMENT
...                              . 'E' IS INSERTED IN THE
...                              . PROGRAM-FILE 'PF'.
RELOCATABLE IMAGES
...
...
@EOF
...
...
PREAMBLE IMAGES
...
...

@ELT,DI  PF,D,,X                 . A NEW DATA ELEMENT 'D'
...                              . IS INSERTED IN THE
...                              . PROGRAM-FILE 'PF'.
DATA IMAGES
...
...
@END X
```

5.5.7. THE @DATA STATEMENT

THE @DATA STATEMENT MAY BE USED TO INTRODUCE STANDARD FORMAT DATA FILES, FOUND IN THE CONTROL STREAM, INTO THE SYSTEM FOR RESIDENCE ON A MASS STORAGE DEVICE. A PRIMARY USE FOR THIS FEATURE IS TO ALLOW THE USER TO BUILD DATA FILES WHICH ARE ACTUALLY WHOLE OR PARTS OF CONTROL STREAMS. THESE FILES CAN THEN BE CALLED ON BY THE @START STATEMENT TO START AN INDEPENDENT RUN, OR BY THE @ADD STATEMENT FOR INCLUSION INTO THE CURRENT RUN OR A SUBSEQUENT RUN. A DATA FILE CORRECTION FEATURE IS ALSO AVAILABLE VIA THE @DATA STATEMENT. THE USER CAN MAKE A CORRECTION TO AN INDEPENDENT RUN STREAM AND THEN @START IT, OR MAKE CORRECTIONS TO A PARTIAL STREAM AND THEN @ADD IT TO THE RUN. THE @DATA STATEMENT CAN OF COURSE SIMPLY BE USED AS A CONVENIENT MEANS OF GENERATING AND MAINTAINING A USER DATA FILE, RATHER THAN A CONTROL-STREAM TYPE FILE.

A NOTABLE DIFFERENCE BETWEEN THE @DATA STATEMENT AND THE @FILE STATEMENT IS THAT THE @DATA STATEMENT HANDLES DATA AS IT IS PRESENTED TO THE MAIN STREAM, USUALLY FROM FASTRAND, WHEREAS THE @FILE STATEMENT BUILDS THE FILE DIRECTLY AS THE DATA IS BEING INITIALLY INPUT BY THE SYSTEM. THE @DATA STATEMENT CAUSES THE INITIATION OF A SYSTEM PROCESSOR (THE DATA PROCESSOR), IN THE SAME MANNER FOR EXAMPLE, AS THE @COB STATEMENT CAUSES THE INITIATION OF THE COBOL PROCESSOR.

THE FORMAT OF THE FILE CREATED AS A RESULT OF THE @DATA STATEMENT IS THE SYSTEMS DATA FILE FORMAT. THE FORMAT IS DESCRIBED IN THE CHAPTER ENTITLED 'DATA HANDLING'.

THE FORMAT OF THE @DATA CONTROL STATEMENT IS AS FOLLOWS:

```
@DATA,OPTIONS  FILENAME1,FILENAME2,SENTINEL
```

THE OPTIONS FIELD MAY CONTAIN THE FOLLOWING CHARACTERS:

I	INSERT. INITIAL INSERTION OF DATA INTO THE FILE.
U	UPDATE. PRODUCE A NEW VERSION OF THE DATA.
L	PRODUCE A COMPLETE LISTING OF THE FILE.

THE 'L' OPTION WILL PRODUCE A COMPLETE LISTING OF THE FILE WHICH WILL INCLUDE SEQUENTIAL ITEM NUMBERS. THESE ITEM NUMBERS WILL BE USED WHEN MAKING CORRECTIONS TO THE FILE. CORRECTIONS TO THE FILE ARE MADE IN THE SAME MANNER AS CORRECTIONS TO A SOURCE LANGUAGE ELEMENT. (SEE SECTION OF THIS CHAPTER ON 'PROCESSOR CALL STATEMENTS'). IF 'L' AND 'FILENAME1' ARE THE ONLY INFORMATION PRESENT IN THE @DATA STATEMENT, 'FILENAME1' WILL BE LISTED.

WHEN THE 'I' OPTION IS PRESENT, THE DATA FOLLOWING THE @DATA STATEMENT IS WRITTEN TO 'FILENAME1'.

WHEN THE 'U' OPTION IS PRESENT, THE DATA FOLLOWING THE @DATA STATEMENT IS TAKEN AS CORRECTIONS TO 'FILENAME1' AND A NEW F-CYCLE OF 'FILENAME1' IS PRODUCED.

IF NEITHER THE 'U' OR 'I' OPTION IS PRESENT, THE DATA FOLLOWING THE @DATA STATEMENT IS TAKEN AS CORRECTIONS TO 'FILENAME1', AND A NEW UPDATED FILE ('FILENAME2') IS CREATED

IF NEITHER THE 'I' OR 'U' OPTION IS PRESENT AND THE 'FILENAME2' FIELD IS VOID, THE 'L' OPTION IS ASSUMED AND 'FILENAME1' IS LISTED. NO NEW FILE WILL BE GENERATED.

THE DATA FOLLOWING THE @DATA STATEMENT IS TERMINATED WITH AN @END STATEMENT WITH A MATCHING SENTINEL. AS IN THE @ELT,D STATEMENT A SEARCH IS MADE FOR THE APPROPRIATE @END WITH ALL IMAGES IN BETWEEN PLACED IN THE DATA FILE.

ANY CONTROL STATEMENTS (EXCEPT @FIN) APPEARING BETWEEN THE @DATA STATEMENT AND THE END OF DATA SENTINEL (@END) ARE TREATED AS DATA BY THE SYSTEM. THIS ALLOWS CONTROL STREAMS TO BE ENTERED AS FILES AND CALLED LATER FOR EXECUTION.

IF A FILE (FILENAME1 OR FILENAME2) IS TO BE CATALOGUED, OR IS TO BE PLACED ON OR RETRIEVED FROM A DEVICE OTHER THAN FASTRAND (WITH ONE TRACK GRANULARITY), THE FILE MUST HAVE BEEN PREVIOUSLY ASSIGNED VIA AN @ASG STATEMENT.

EXAMPLES:

```
@DATA,I  X                    . THIS STATEMENT WILL
...                           . GENERATE A NEW FILE 'X'
...                           . CONTAINING THE DATA
DATA IMAGES                   . FOLLOWING THE STATEMENT.
...
...
@END

@DATA  X,Y                    . THIS STATEMENT WILL
...                           . APPLY THE CORRECTIONS
CORRECTION IMAGES             . TO FILE 'X' AND CREATE
...                           . A NEW FILE 'Y'.
...
@END

@DATA,L  Y                    . THIS STATEMENT WILL
@END                          . LIST FILE 'Y'.
```

```
@DATA   Y                            . THIS STATEMENT WILL
@END                                 . LIST FILE 'Y'.

@DATA,L   X                          . THIS STATEMENT WILL
...                                  . APPLY THE CORRECTIONS
...                                  . FOLLOWING THE STATEMENT
CORRECTION IMAGES                    . TO FILE 'X'.  THE NEW
...                                  . FILE WILL BE LISTED, BUT
...                                  . A NEW FILE WILL NOT BE
@END                                 . GENERATED.

@DATA,IL   A                         . THIS STATEMENT WILL
...                                  . GENERATE A NEW FILE 'A'
...                                  . CONTAINING THE DATA
 DATA IMAGES                         . FOLLOWING THE STATEMENT.
...                                  . FILE 'A' WILL ALSO BE
...                                  . LISTED.
@END

@DATA,U   X                          . THIS STATEMENT WILL
...                                  . APPLY THE CORRECTIONS
...                                  . FOLLOWING THE STATEMENT
CORRECTION IMAGES                    . TO FILE 'X' AND CREATE
...                                  . A NEW F-CYCLE OF FILE 'X'.
...
@END
```

5.5.8. THE @END STATEMENT

THE @END CONTROL STATEMENT MARKS THE END OF THE DATA THAT FOLLOWS A @DATA OR @ELT,D STATEMENT. THE FORMAT OF THE @END STATEMENT IS:

```
@END    SENTINEL                    COMMENT
```

THIS STATEMENT CANNOT BE CONTINUED ON A SECOND LINE. THE SENTINEL FIELD IS OPTIONAL. IT IS CODED EXACTLY THE SAME AS THE CORRESPONDING FIELD ON A @DATA OR @ELT,D STATEMENT WHEN BEING USED TO BRACKET IMAGES OF THE DATA.

5.5.9. THE @FILE STATEMENT

THIS STATEMENT IS USED TO CREATE DATA FILES DIRECTLY THROUGH A SYMBIONT CONTROLLED DEVICE, WITH DATA FROM THE CONTROL STREAM. THIS STATEMENT IS INTERPRETED AND PERFORMED BY THE PROCESSING SYMBIONT. WHEN A @FILE STATEMENT IS ENCOUNTERED IN A RUN, ALL FOLLOWING INPUT IS SENT TO THE NAMED DEVICE OF THE @FILE STATEMENT. THE @ENDF STATEMENT IS USED TO TERMINATE THE FILE MODE. THAT PORTION OF THE INPUT STREAM PRECEEDING @FILE AND FOLLOWING @ENDF IS ENTERED INTO THE SCHEDULE QUEUE. THE FORMAT OF THE @FILE STATEMENT IS:

```
@FILE,OPTIONS  FILENAME,DEVICE-TYPE
```

THE 'DEVICE-TYPE' SPECIFIED ON THIS STATEMENT IS DIRECTLY ASSIGNED TO THE PROCESSING SYMBIONT VIA A DYNAMIC FACILITY REQUEST. READING FROM THE INPUT DEVICE IS SUSPENDED UNTIL THE SPECIFIED 'DEVICE-TYPE' IS MADE AVAILABLE. THE FOLLOWING DATA IS THEN BUFFERED DIRECTLY TO THE 'DEVICE-TYPE'.

THE 'OPTIONS' FIELD SPECIFIES CATALOGUING PROCEDURES AND MAGNETIC TAPE RECORDING OPTIONS. THE CATALOGUE OPTIONS ARE:

C SPECIFIES THIS FILE IS TO BE CATALOGUED.

R SPECIFIES THE FILE IS TO BE PLACED IN 'READ-ONLY' STATE WHEN IT IS CATALOGUED. A FILE CATALOGUED WITH THIS OPTION CANNOT BE OVERWRITTEN.

THE MAGNETIC TAPE OPTIONS ARE AS FOLLOWS.

L LOW DENSITY
M MEDIUM DENSITY
H HIGH DENSITY
E EVEN PARITY
O ODD PARITY
I DECIMAL (TRANSLATE)
HARDWARE IS USED IF AVAILABLE, OTHERWISE STANDARD SOFTWARE CONVERSION ROUTINE IS USED (FD-BCD). THE 'E' OPTION IS ASSUMED ONLY IF SOFTWARE USED. NOTE: IF EVEN PARITY MODE IS SPECIFIED WITHOUT THE FD-BCD CONVERSION, SHORT DATA BLOCKS MAY BE WRITTEN DUE TO THE FIELDATA MASTER SPACE (@) INTERNAL CODE OF 00.
B BINARY (NO TRANSLATE)

THE 'FILENAME' FIELD IS USED TO DIRECT TAPE MOUNTING IF REQUIRED, AND FOR FILE CATALOGUING.

THE 'DEVICE-TYPE' FIELD SPECIFIES THE REPOSITORY MEDIUM FOR THE FILE. THE LEGAL VALUES FOR THIS FIELD ARE:

F4	FASTRAND SIMULATED ON FH 432
F8	FASTRAND SIMULATED ON FH 880
F2	FASTRAND MODEL II
F	FASTRAND, TYPE INDEPENDENT
FB	FASTBAND
T	TAPE,TYPE INDEPENDENT
C	UNISERVOS VIIIC, VIC, AND IVC
U	UNISERVOS VIIIC AND VIC
8C	UNISERVO VIIIC
6C	UNISERVO VIC
4C	UNISERVO IVC
3A	UNISERVO IIIA
2A	UNISERVO IIA

THE 'F' VALUE IS ASSUMED IF THE DEVICE-TYPE FIELD IS VOID. IF A UNISERVO IS DEFINED AS THE @FILE REPOSITORY MEDIUM THE REEL IS REWOUND WITH INTERLOCK WHEN @ENDF IS ENCOUNTERED.

ANY CONTROL STATEMENTS (EXCEPT @FIN) APPEARING BETWEEN @FILE AND @ENDF ARE IGNORED BY THE SYSTEM.

NOTE: IF A @FILE STATEMENT IS ENCOUNTERED WHILE IN THE @FILE MODE, THE RESULTANT ACTION IS DETERMINED BY A FILENAME COMPARISON OF THE FILE BEING GENERATED AND THE JUST ENCOUNTERED @FILE STATEMENT. THE ACTION TAKEN IS ONE OF THE FOLLOWING:

FILENAMES UNEQUAL
THE CURRENT FILE IS TERMINATED AS IF AN @ENDF STATEMENT WERE ENCOUNTERED FOLLOWED BY A NEW @FILE STATEMENT.

FILENAMES EQUAL
IF THE REPOSITORY MEDIUM FOR THE FILE IS MASS STORAGE, THE NESTED @FILE STATEMENT IS IGNORED. IF THE OUTPUT MEDIUM IS TAPE, THE CURRENT FILE IS CLOSED AS IF THE @ENDF STATEMENT WERE READ EXCEPT THAT THE TAPE IS NOT REWOUND AND ONLY ONE TAPE MARK IS WRITTEN FOLLOWED BY THE NEW LABEL BLOCK. FILE STATEMENTS CAN NOT BE NESTED.

5.5.10. THE @ENDF STATEMENT

THIS CONTROL STATEMENT IS ONLY USED TO TERMINATE FILE GENERATION INITIATED WITH

THE @FILE CONTROL STATEMENT. IF @ENDF APPEARS IN THE CONTROL STREAM WITHOUT A PREVIOUSLY ENCOUNTERED @FILE STATEMENT, IT IS IGNORED BY THE SYSTEM. THE FORMAT OF THE @ENDF STATEMENT IS:

```
    @ENDF                   COMMENT
```

THIS STATEMENT CANNOT BE CONTINUED ON A SECOND LINE.

5.5.11. THE @QUAL CONTROL STATEMENT

THE @QUAL STATEMENT ALLOWS THE USER TO SPECIFY A FILE NAME QUALIFICATION FOR IMPLIED USAGE ON SUCCEEDING CONTROL STATEMENTS INVOLVING FILE NAMES. THE FORMAT OF THIS STATEMENT IS:

```
    @QUAL   QUALIFIER
```

WHERE 'QUALIFIER' IS A SEQUENCE OF 12 OR FEWER CHARACTERS USED TO QUALIFY SUBSEQUENT FILE NAMES WHICH ARE HEADED BY AN ASTERISK (*). THE 'QUALIFIER' IS LIMITED TO THE CHARACTER SET A...Z, 0...9, -, AND $. AN EXAMPLE ON THE USE OF THE @QUAL STATEMENT FOLLOWS:

```
    @QUAL    JIM
        .
        .
        .
```

THE SUBSEQUENT STATEMENT:

```
    @FOR   *FILEA.JOE/ABC
```

WOULD BE INTERPRETED AS:

```
    @FOR   JIM*FILEA.JOE/ABC
```

WHERE 'FOR' IS A CALL ON THE FORTRAN COMPILER AND THE SPECIFICATIONS FIELD NAMES A PARTICULAR ELEMENT IN A PARTICULAR FILE (SEE SECTION ON PROCESSOR CALL STATEMENTS FOR METHOD OF REFERENCING AN ELEMENT).
FURTHER, THE SUBSEQUENT STATEMENT:

```
    @FREE *P
```

WOULD BE INTERPRETED AS

```
    @FREE   JIM*P
```

ANY NUMBER OF @QUAL STATEMENTS MAY APPEAR THROUGHOUT THE CONTROL STREAM. EACH WILL OVERRIDE THE EFFECT OF THE PREVIOUS ONE.

5.6. PROCESSOR CALL STATEMENTS

5.6.1. NOTATION FOR PROGRAM FILE ELEMENTS

A CONSISTENT NOTATION IS USED THROUGHOUT THE SYSTEM TO REFERENCE ELEMENTS OF A PROGRAM FILE. USING THE COBOL SYNTAX DESCRIPTION NOTATION, A REFERENCE TO AN ELEMENT HAS THE FORM:

```
    [[[QUALIFIER]*]FILE.]NAME[/VERSION][(CYCLE)]
```

AN EXTENSIVE SERIES OF DROPOUT RULES USUALLY ALLOW ABBREVIATION OF REFERENCES TO PROGRAM FILE ELEMENTS FROM THE FULL FORM SHOWN TO SOMETHING QUITE MANAGEABLE.

THE OMISSION OF 'QUALIFIER' WITH THE '*' PRESENT CAUSES THE @QUAL STATEMENT TO SUPPLY THE QUALIFIER USED. IF THE @QUAL STATEMENT HAS NOT OCCURRED, THE 'PROJECT' FIELD FROM THE @RUN STATEMENT IS USED AS THE QUALIFIER. THE OMISSION OF BOTH THE 'QUALIFIER, AND THE '*' CAUSES THE 'PROJECT' FIELD FROM THE @RUN STATEMENT TO BE USED AS THE QUALIFIER, PROVIDED THE 'FILE', IF SPECIFIED, IS NOT AN 'ATTACHED' NAME WHICH POINTS TO A PARTICULAR FILENAME. IF THE 'FILE.' SUB-FIELD IS ALSO OMITTED, THEN THE RUN TEMPORARY PROGRAM FILE IS INTENDED. THE SUB-FIELD 'NAME' MUST ALWAYS BE PRESENT WHEN REFERRING TO AN ELEMENT. THE 'VERSION' SUB-FIELD IS REQUIRED ONLY IN THE CASE THAT MORE THAN ONE VERSION OF A PARTICULAR ELEMENT EXISTS WITHIN THE PROGRAM FILE AS IS COMMON WHEN A PROGRAM IS IN CHECKOUT.

AN 'F-CYCLE' NUMBER MAY BE ATTACHED TO 'FILE' FIELD SHOWN ABOVE. ITS USE IS SIMILAR TO THAT OF THE 'CYCLE' FIELD DISCUSSED BELOW AND IS DESCRIBED IN THE SECTION ON '@ASG STATEMENTS'. LIKEWISE, THE TWO KEYS MAY BE ATTACHED TO THE 'FILE' FIELD. THEIR USE AND DESCRIPTION ARE DESCRIBED IN THE SECTION ON '@ASG STATEMENTS', (SECT. 5.5.1).

(ON THE VARIOUS CONTROL STATEMENTS, SUCH AS @ADD OR @START, WHICH CAN SPECIFY EITHER A 'FILE' OR AN 'ELEMENT' NAME, A METHOD IS ESTABLISHED WHICH DISTINGUISHES BETWEEN THEM. A PERIOD FOLLOWING THE 'NAME' WILL SPECIFY A 'FILE', AND NO PERIOD WILL SPECIFY AN 'ELEMENT').

THE CYCLE NUMBER SERVES TO DIFFERENTIATE SUCCESSIVE UPDATES OF A SYMBOLIC ELEMENT. OMISSION OF THE CYCLE NUMBER WHEN REFERRING TO A SYMBOLIC ELEMENT IMPLIES THAT THE MOST RECENTLY CONSTRUCTED COPY IS INTENDED. A COMPACTING METHOD, AS DESCRIBED LATER, IS EMPLOYED TO PREVENT THE RETENTION OF SEVERAL CYCLES OF A SYMBOLIC ELEMENT FROM APPROPRIATING AN EXCESSIVE AMOUNT OF SPACE ON WHATEVER STORAGE MEDIUM IS EMPLOYED. SOME EXAMPLES WILL HELP MAKE THIS A BIT CLEARER.

SORT	THE ELEMENT SORT IN THE RUN TEMPORARY FILE.
COST*PROG.EDIT	THE ELEMENT EDIT IN THE FILE COST*PROG.
*BACKUP.TLU/TWO	VERSION TWO OF ELEMENT TLU IN FILE BACKUP. THE QUALIFIER FOR BACKUP IS TAKEN FROM THE @QUAL CONTROL CARD.
PCF6.INTL(14)	THE 14TH GENERATION OF THE ELEMENT INTL IN THE FILE PCF6 BELONGING TO THE CURRENT PROJECT.

THE NOTATION GIVEN HERE FOR PROGRAM FILE ELEMENTS DOES NOT PROVIDE COMPLETE IDENTIFICATION OF THE PARTICULAR DATA DESIRED SINCE AN ELEMENT CAN EXIST IN MORE THAN ONE FORM; FOR EXAMPLE, SOURCE LANGUAGE AND RELOCATABLE. THIS IS ONLY AN APPARENT AMBIGUITY, HOWEVER, SINCE IN ALL INSTANCES THE SYSTEM IS AWARE OF THE TYPE OF ELEMENT DESIRED.

5.6.2. STATEMENT FORMAT

THERE ARE SEVERAL SYSTEM PROCESSORS WHICH PROCESS A SOURCELANGUAGE ELEMENT TO PRODUCE A RELOCATABLE BINARY ELEMENT. THE GENERAL FORMAT OF THE STATEMENT FOR CALLING THESE PROCESSORS IS AS FOLLOWS:

```
@PROCESSOR,OPTIONS  ELEMENT1,ELEMENT2,ELEMENT3,....
```

THE FIELD 'PROCESSOR' MAY CONTAIN ANY ONE OF THE ACRONYMS FOR, COB, ASM, ALG, PDP, OR WHATEVER IS AVAILABLE, IN WHICH CASE THE INDICATED PROCESSOR IS CALLED.

THE FIELD 'OPTIONS' MAY CONTAIN ANY ONE OR SEVERAL OF THE ALPHABETIC CHARACTERS 'A' THROUGH 'Z'. THE USE OF ANY OF THESE CHARACTERS BY A PROCESSOR IS DEFINED IN THE PERTINENT PROCESSOR USERS MANUAL. HOWEVER, THE FOLLOWING HAVE A COMMON DEFINITION FOR ALL PROCESSORS.

U - UPDATE. PRODUCE NEW CYCLE OF SOURCE LANGUAGE ELEMENT.

I - INSERT. INTRODUCE SOURCE LANGUAGE INTO PROGRAM FILE FROM CONTROL STREAM.

L - PRODUCE A COMPLETE PRINTED LISTING AS DEFINED FOR THE PERTINENT PROCESSOR.

N - (OR LACK OF ANY OTHER PRINTING OPTION) SUPPRESS ALL PRINTING BY THE PROCESSOR. IF 'N', DISREGARD PRESENCE OF ANY OTHER PRINT OPTION.

S - PRODUCE A PARTIAL LISTING AS DEFINED FOR THE PARTICULAR PROCESSOR.

W - LIST CORRECTION LINES BEING APPLIED.

THE FIELD 'ELEMENT 1' SPECIFIES THE PARTICULAR PROGRAM FILE ELEMENT TO BE USED FOR THE SOURCE LANGUAGE TO BE PROCESSED.
IF PRESENT, AND THERE IS NO I OPTION, THE LINES IMMEDIATELY FOLLOWING THE CONTROL STATEMENT ARE TAKEN TO BE CORRECTIONS TO THE SOURCE LANGUAGE ELEMENT. IF AN I OPTION IS PRESENT, THEN THE LINES FOLLOWING THE CONTROL STATEMENT ARE GIVEN TO THE PROCESSOR AND ARE INSERTED INTO THE PROGRAM FILE AS WELL.

THE FIELD 'ELEMENT2' IS THE NAME OF THE ELEMENT WHICH IS THE RELOCATABLE OR 'ABSOLUTE' CODE PRODUCED BY A PROCESSOR. THIS NAME (AND THE NAMES ASSOCIATED WITH 'ELEMENT1' AND ELEMENT3' FIELDS) MAY INCLUDE PROGRAM FILE NAMES, F-CYCLE, AND VERSION IF DESIRED. IF THE 'I' OPTION IS ON, THE NAME IS NOT REQUIRED BECAUSE THE NAME IN THE FIRST SPECIFICATION FIELD WILL BE USED IF THE FIELD IS BLANK . IF THE 'I' OPTION IS NOT ON A NAME MUST BE PRESENT.

THE FIELD 'ELEMENT 3' IS THE NAME OF THE SOURCE-LANGUAGE ELEMENT PRODUCED BY UPDATING THE INPUT SOURCE LANGUAGE ELEMENT. IF THIS FIELD IS VOID, NO UPDATED SOURCE LANGUAGE ELEMENT WILL BE PRODUCED UNLESS A 'U' OPTION IS SPECIFIED. IN THAT CASE, AN UPDATED ELEMENT IS PRODUCED, WITH THE SAME NAME AND VERSION AS THE INPUT ELEMENT, BUT WITH A CYCLE NUMBER ONE GREATER. NO 'ELEMENT 3' FIELD NEED EXIST WHEN THE I OPTION IS USED.

FURTHER SPECIFICATIONS CAN BE PRESENT IN A PROCESSOR CALL STATEMENT TO INDICATE A PARTICULAR PROGRAM FILE NAME TO BE USED FOR LIBRARY RETRIEVALS, SUCH AS SOURCE LANGUAGE SEGMENT COPIES, PROC SAMPLES AND SUCH. IF UNSTATED, THESE FILES ARE ASSUMED TO BE THE SAME AS THE FILE CONTAINING THE INPUT SOURCE-LANGUAGE ELEMENT, OR LACKING THAT, THE RUN-TEMPORARY PROGRAM FILE.

FOR THE THREE MOST COMMON CASES THE SPECIFICATIONS REDUCE TO TRIVIALITIES. IF SOURCE LANGUAGE IS COMING FROM THE CONTROL STREAM AND NO REFERENCE IS MADE TO PROGRAM FILES ON FASTRAND OR DRUM, THE PROCESSOR CALL STATEMENT (ASSUMING FORTRAN AS AN EXAMPLE) WILL REDUCE TO

```
@FOR
...............
...............
SOURCE LANGUAGE IMAGES
...............
...............
...............
```

IN THIS CASE, THE SOURCE-LANGUAGE PROGRAM IS COMPILED AND THE RESULTING RELOCATABLE ELEMENT PUT INTO THE RUN-TEMPORARY FILE, READY TO BE ACCESSED BY THE COLLECTOR AND LOADER.

A PROCESSOR MAY BE USED TO INTRODUCE A SOURCE-LANGUAGE ELEMENT INTO A PROGRAM FILE FOR THE FIRST TIME FROM THE CONTROL STREAM. IN THIS CASE THE I OPTION IS SPECIFIED AND THERE IS NO 'ELEMENT 3' FIELD. AS AN EXAMPLE, CONSIDER THE INITIAL PROCESSING OF THE ELEMENT WINDUP TO BE INSERTED INTO PROGRAM FILE PF3.

```
@FOR,I PF3.WINDUP
...............
...............
...............
SOURCE LANGUAGE IMAGES
...............
...............
...............
```

IN THIS CASE PROGRAM FILE PF3 WOULD BE LEFT WITH THE SOURCE LANGUAGE AND RELOCATABLE IMAGES OF ELEMENT WINDUP.

IF AN UPDATE IS BEING MADE TO SOME ELEMENT, SAY WINDUP, IN A PROGRAM FILE, (PF3) THEN THE PROCESSOR CALL STATEMENT WOULD READ:

```
@FOR,U PF3.WINDUP,.WINDUP
...............
...............
...............
SOURCE LANGUAGE CORRECTIONS
...............
...............
...............
```

IN THIS CASE, THE SOURCE-LANGUAGE ELEMENT SPECIFIED BY PF3 WINDUP IS UPDATED BY THE GIVEN CORRECTION LINES AND COMPILED.

THE RESULTING RELOCATABLE ELEMENT IS INSERTED BACK INTO PROGRAM FILE PF3, ALONG WITH THE NEXT CYCLE OF THE SOURCE-LANGUAGE ELEMENT WINDUP. IF THE INPUT SOURCE-LANGUAGE ELEMENT HAD A CYCLE NUMBER OF, SAY 72, THE NEW SOURCE-LANGUAGE ELEMENT HAS A CYCLE NUMBER OF 73. THE ENTIRE ELEMENT AND CORRECTION LINES ARE WRITTEN IN THE PROGRAM FILE, AND THE OLD ELEMENT IS MARKED DELETED. IF, FOR EXAMPLE, THREE CYCLES OF SOURCE-LANGUAGE ELEMENTS ARE BEING KEPT, THE PROGRAM FILE PF3 WILL CONTAIN BEFORE THE ABOVE STATEMENT IS EXECUTED, THE INFORMATION:

```
WINDUP (70)      COMPLETE ELEMENT
CORRECTION LINES CONVERTING WINDUP (70) TO WINDUP (71)
CORRECTION LINES CONVERTING WINDUP (71) TO WINDUP (72)
```

AFTER COMPILING, PF3 CONTAINS:

```
WINDUP (71) COMPLETE ELEMENT
CORRECTION LINES IN SDF FORMAT CONVERTING WINDUP (71) TO WINDUP (72)
CORRECTION LINES CONVERTING WINDUP (72) TO WINDUP (73)
```

THE NUMBER OF CYCLES RETAINED, SAY N, IS A SYSTEM STANDARD SET AT SYSTEM GENERATION TIME; THUS, A COMPLETE ELEMENT AND THE N-1 MOST RECENT SETS OF CORRECTIONS ARE KEPT. NORMALLY THIS WILL INVOLVE CONSIDERABLY LESS MASS STORAGE SPACE THAN EVEN TWO COMPLETE ELEMENTS AND PROVIDES CONSIDERABLY MORE FLEXIBILITY IN BACKING UP TO SOME PARTICULAR POINT IN THE HISTORY OF A PROGRAM.

THE NUMBER OF CYCLES KEPT IS SET AT THE SYSTEM STANDARD (WHICH CAN BE ALTERED AT SYSTEM GENERATION TIME, SEE SYSTEM GENERATION STATEMENTS - RETAIN CYCLES AND F-CYCLES, AND INITIAL SYSTEM), UNLESS SOME DIFFERENT NUMBER IS SPECIFIED BY THE PROGRAM FILE UTILITY ROUTINE FOR THE PARTICULAR ELEMENT OR PARTICULAR FILE. THE MAXIMUM NUMBER OF CYCLES THAT CAN BE RETAINED IS LIMITED ONLY BY THE STORAGE SPACE AVAILABLE, ALTHOUGH THE PROCESS BECOMES INEFFICIENT FOR AN EXCESSIVE NUMBER OF CYCLES. IT IS POSSIBLE TO REFERENCE ANY PARTICULAR AVAILABLE CYCLE OF A SOURCE-LANGUAGE ELEMENT. SUPPOSE THAT CYCLES 70 THROUGH 72 OF WINDUP ARE AVAILABLE. THE PROCESSOR CALL STATEMENT:

```
@FOR, U  PF3.WINDUP(70),.WINDUP
```

WOULD CREATE A NEW CYCLE 71 AND WOULD DELETE CYCLE 72. ON THE OTHERHAND, THE

PROCESSOR CALL STATEMENT:

```
@FOR  PF3.WINDUP(71),,WINDUP,PF3.WINDUP/NEW
```

WOULD LEAVE CYCLE 72 OF WINDUP INTACT BUT WOULD PRODUCE AN ENTIRELY NEW SOURCE-LANGUAGE ELEMENT WINDUP/NEW WHICH WOULD HAVE A CYCLE NUMBER ONE INTEGER LARGER THAN THE LARGEST IN FILE PF3 FOR WINDUP/NEW, IF ANY CYCLES OF THIS ELEMENT EXIST; IF NONE, A CYCLE NUMBER OF (1) IS ASSIGNED. IF THERE WERE ANY OTHER CYCLES OF WINDUP/NEW, THEY WOULD BE DELETED, REGARDLESS OF THEIR CYCLE NUMBERS.

5.6.3. FORMAT OF CORRECTION LINES

EACH PROCESSOR LISTS THE SOURCE LANGUAGE INPUT ON WHICH IT IS OPERATING. ON THIS LISTING, SUCCESSIVE LINES ARE LABELED BY SUCCESSIVE INTEGRAL NUMBERS. WHEN ALTERING A SOURCE-LANGUAGE ELEMENT IN A PROGRAM FILE, THESE NUMBERS ARE USED TO INDICATE WHERE CORRECTIONS ARE TO BE INSERTED. A LINE OF THE FORM:

```
-N,M
```

WITH THE '-' IN THE FIRST COLUMN INDICATES THAT SOURCE LINES 'N' THROUGH 'M' ARE TO BE REPLACED BY ALL SUCCEEDING LINES IN THE CONTROL STREAM UP TO THE NEXT LINE WITH A '-' IN COLUMN ONE, OR THE NEXT CONTROL STATEMENT.

A LINE OF THE FORM:

```
-K
```

INDICATES THAT SUCCEEDING CORRECTIONS ARE TO BE INSERTED INTO THE SOURCE LANGUAGE ELEMENT FOLLOWING LINE K.

FOR EXAMPLE, THE CONTROL STREAM:

```
@ASM,U WEEKLY.REPORT,.REPORT
-30,31
CORRECTION LINE A
-100,115
-120
CORRECTION LINE B
CORRECTION LINE C
CORRECTION LINE D
```

WILL REPLACE LINES 30 AND 31 BY THE CORRECTION LINE A, DELETE LINES 100 THROUGH 115, AND INSERT CORRECTION LINES B, C AND D FOLLOWING LINE 120.

IF THE USER WISHES TO INSERT CORRECTIONS BEFORE THE FIRST LINE ITEM OF HIS OLD SOURCE INPUT, HE MUST PLACE THEM IMMEDIATELY AFTER THE PROCESSOR CALL STATEMENT WITHOUT SPECIFYING A CORRECTION LINE.

WHEN CORRECTIONS FOLLOW A PROCESSOR CALL STATEMENT IN A CONTROL STREAM, THE SOURCE INPUT ROUTINE (SIR) INTERPRETS A MINUS SIGN '-' IN THE FIRST COLUMN OF A LINE AS A CORRECTION LINE. IN CERTAIN SITUATIONS WHERE THE USER MAY HAVE DATA WITH THE '-' IN COLUMN ONE, THIS IS NOT DESIRABLE. THIS MIGHT HAPPEN WHEN MAKING CORRECTIONS TO A @RUN OR @ADD STREAM WITH THE @DATA OR @ELT PROCESSORS. THE USER MAY WISH TO INSERT A SET OF CORRECTIONS THAT ARE ACTUALLY CORRECTIONS FOR A PROCESSOR CALL IN THE @RUN OR @ADD STREAM. THESE CORRECTIONS ARE NOT BE INTERPRETED UNTIL THE @RUN OR @ADD IS PROCESSED. TO GET AROUND THIS PROBLEM, SIR IS PREPARED. TO HANDLE THE FOLLOWING CORRECTION LINE:

```
-=X
```

WHICH SAYS FROM HERE ON, SIR IS TO USE 'X' TO IDENTIFY CORRECTION LINES. 'X' MAY BE 1,2, OR 3 CHARACTERS IN LENGTH BUT MUST NOT CONTAIN A SPACE OR NUMERIC CHARACTER. THE USER MAY CHANGE CORRECTION LINE IDENTIFIERS AS OFTEN AS HE WISHES BUT SIR WILL RECOGNIZE ONLY ONE IDENTIFIER AT A TIME. INITIALLY SIR IS SET TO RECOGNIZE '-' AS THE CORRECTION LINE IDENTIFIER.

THE FOLLOWING EXAMPLE ILLUSTRATES THE USE OF IDENTIFIER CHANGES.

```
@DATA     FILE1,FILE2
-2                  FOLLOW LINE 2
CORRECTIONS         WITH CORRECTIONS.
-=*                 CHANGE IDENTIFIER TO *.
*11,13              DELETE LINES 11,12,AND 13
CORRECTIONS         AND INSERT CORRECTIONS.
*=+++               CHANGE IDENTIFIER TO +++.
+++22               FOLLOW LINE 22
CORRECTIONS         WITH CORRECTIONS.
@END
```

5.6.4. SYSTEM LIBRARIES (PROGRAM FILES)

RELOCATABLE LIBRARY (SYS$*RLIB$). THIS FILE CONTAINS RELOCATABLE ELEMENTS AND PROCEDURE ELEMENTS AS NEEDED BY THE SYSTEM PROCESSORS (ASSEMBLER, COMPILERS, COLLECTOR, ETC.). THE INSTALLATION MAY INCLUDE ADDITIONAL ELEMENTS; HOWEVER, THE FILE EXISTS PRIMARILY AS A PLACE FOR STANDARD RELOCATABLE TO BE USED BY THE COLLECTOR IN PUTTING TOGETHER PROGRAMS AND AS A PLACE FOR STANDARD PROCEDURES TO BE PICKED UP BY THE ASSEMBLER AND COMPILERS.

SYSTEM LIBRARY (SYS$*LIB$). THIS FILE CONTAINS ABSOLUTE ELEMENTS ONLY. THIS INCLUDES SYSTEM PROCESSORS LIKE THE COLLECTOR OR ASSEMBLER AS WELL AS THOSE ELEMENTS THAT MAY BE ADDED BY THE USER. LIB$ IS VARIABLE IN LENGTH AT SYSTEM GENERATION TIME.

ADDITIONALLY, THERE IS A TABLE IN CORE CONTAINING THE NAME, RELATIVE ADDRESS IN LIB$, AND PROGRAM SIZE OF SELECTED PROCESSORS. THIS TABLE, LIBT, IS REFERENCED BEFORE LIB$ IS REFERENCED TO SAVE MASS STORAGE ACCESSES. LIBT IS VARIABLE IN LENGTH AND IS CREATED AT SYSTEM GENERATION TIME.

TEMPORARY PROGRAM FILE (PROJECT*TPF$). THIS FILE IS CREATED AUTOMATICALLY BY THE EXECUTIVE WHEN A RUN IS INITIATED. THE USER DOES NOT HAVE TO SPECIFY THE FILE NAME SINCE A VOID FILENAME CAN BE USED TO REFERENCE THE FILE. THE FILE QUALIFIER IS TAKEN FROM THE PROJECT FIELD OF THE RUN STATEMENT.

IF A FILENAME IS NOT GIVEN, THE ORDER IN WHICH THE EXECUTIVE SEARCHES PROGRAM FILES IS AS FOLLOWS:

ON PROCESSOR CALL STATEMENT	- LIB$, THEN TPF$
ON LINK$ REFERENCE	- LIB$, THEN TPF$
ON @XQT STATMENT	- TPF$

IF A FILE OTHER THAN TPF$ OR LIB$ IS INTENDED ON A PROCESSOR CALL STATEMENT OR TPF$ ON AN @XQT STATEMENT, THE FILENAME MUST BE GIVEN. AN EXAMPLE WOULD BE:

```
@XQT   FILEB.PROGA
```

WHERE THE PROGRAM 'PROGA' IS BEING EXECUTED FROM FILEB.

THE ORDER OF SEARCH USED BY THE COMPLILERS AND THE ASSEMBLER IN FINDING PROCEDURES IS AS FOLLOWS:

FIRST THE FILE FROM WHICH THE SYMBOLIC ELEMENT WAS TAKEN,
AND THEN RLIB$

THE SYSTEM GENERATION ROUTINE DISCUSSED LATER IN THIS MANUAL PROVIDES A MEANS OF AUGMENTING THE EXECUTIVE SYSTEM WITH NEW PROCESSORS. WITHIN THE FRAMEWORK PROVIDED BY THE SYSTEM, NEW PROCESSORS MAY BE APPENDED BY SPECIFYING THEIR NAMES AND CHARACTERISTICS WITH STATEMENTS TO THE SYSTEM GENERATION ROUTINE (SEE SYSTEM GENERATION STATEMENTS - ADD PROCESSOR).

5.6.5. A GENERALIZATION

THE USE OF A CONTROL STATEMENT OF THE FORM:

```
@PROGRAM,OPTIONS SPECIFICATIONS
```

TO INVOKE A PROCESSOR IS ONLY A PARTICULAR CASE OF A MORE GENERAL MECHANISM WITHIN THE EXECUTIVE SYSTEM. ANY SYMBOL IN THE 'PROGRAM' FIELD NOT OTHERWISE GIVEN SPECIFIC MEANING BY THE SYSTEM IS INTERPRETED AS A CALL ON AN ABSOLUTE PROGRAM BY THAT NAME TO BE TAKEN FROM THE SYSTEM LIBRARY; THAT IS, THE 'PROGRAM' FIELD WOULD BE INTERPRETED AS THE ABSOLUTE ELEMENT:

```
SYS$*LIB$.PROGRAM
```

WHERE 'SYS$*LIB$' IS THE IMPLIED FILE NAME AND 'PROGRAM' IS THE NAME OF THE ELEMENT TO BE EXECUTED. NO VERSION AMBIGUITY MAY EXIST. A FURTHER GENERALIZATION IS POSSIBLE BY CODING IN THE COMMAND FIELD THE ENTIRE SPECIFICATION OF AN ABSOLUTE ELEMENT DEFINING ITS RESIDENCE IN A FILE. AN EXAMPLE WOULD BE:

```
@USER*FILE.PROG,P   ABC
```

WHERE 'PROG' FROM THE FILE 'USER*FILE' IS EXECUTED AS A PROCESSOR. THE COMPLETE FORM OF ELEMENT NOTATION MAY BE USED.

THE 'OPTIONS' FIELD AND THE 'SPECIFICATION' FIELD OF THE CONTROL STATEMENT ARE MADE AVAILABLE TO THE PROGRAM REFERENCED BY THE PROGRAM'S FIRST REFERENCE TO READ$, BUT THE SERVICES OF THE PROCESSOR INTERFACE ROUTINES ARE NOT AVAILABLE UNLESS THE PROGRAM HAS BEEN SPECIFICALLY DESIGNATED AS A PROCESSOR AS DESCRIBED IN THE PRECEDING SECTION.

5.7. PROGRAM EXECUTION STATEMENTS

THE PROGRAM EXECUTION STATEMENTS ARE USED TO CONTROL THE CONSTRUCTION, RUNNING, AND DIAGNOSIS OF A PROGRAM CREATED BY A USER.

5.7.1. THE @MAP STATEMENT

THE MAP STATEMENT IS USED FOR CALLING THE COLLECTOR TO COLLECT A SERIES OF RELOCATABLE PROGRAMS FROM ONE OR MORE PROGRAM FILES AND TO COMBINE THEM INTO AN EXECUTABLE PROGRAM. THE FORMAT OF THIS STATEMENT IS:

```
@MAP,OPTIONS   FILE1. ELEMENT1, FILE2. ELEMENT2, FILE3. ELEMENT3
```

THE 'OPTIONS' FIELD IS ESSENTIALLY THE SAME AS FOR A PROCESSOR CALL STATEMENT. THE 'ELEMENT' FIELDS ARE USED TO SPECIFY THE PROGRAM FILE ELEMENTS TO BE USED IN THE ALLOCATION PROCESS. A DETAILED DESCRIPTION OF THE @MAP STATEMENT AND EXAMPLES OF ITS USE ARE INCLUDED IN THE SECTION ON THE COLLECTOR.

5.7.2. THE @XQT STATEMENT

THE @XQT STATEMENT IS USED TO INITIATE THE EXECUTION OF AN ABSOLUTE PROGRAM PREPARED BY THE COLLECTOR. IT HAS THE FORMAT:

```
@XQT,OPTIONS        ELEMENT
```

THE OPTIONS SUB-FIELD MAKES A 26-BIT MASK (EACH BIT THAT IS SET REPRESENTS AN ALPHABETIC CHARACTER THAT WAS SPECIFIED, A IS REPRESENTED BY THE LEFT-MOST BIT, AND Z BY THE RIGHT-MOST) AVAILABLE TO THE PROGRAM THROUGH THE EXECUTIVE REQUEST FUNCTIONOPT$. THE 'ELEMENT' FIELD OF THE STATEMENT NAMES THE PROGRAM FILE ELEMENT TO BE EXECUTED. VARIATIONS OF THE USE OF THE @XQT STATEMENT ARE GIVEN IN THE SECTION ON THE COLLECTOR (SECTION 15.1.2).

DATA CARDS TO BE INPUT BY THE PROGRAM MAY FOLLOW THE @XQT STATEMENT. THE PROGRAM USES THE SYSTEM REFERENCE 'READ$' IN GAINING ACCESS TO ALL IMAGES PRIOR TO THE NEXT EXECUTIVE CONTROL STATEMENT. WHEN AN EXECUTIVE CONTROL STATEMENT(OTHER THAN AN @EOF, SEE BELOW) IS DETECTED BY READ$, FURTHER READING BY THE USER (OR PROCESSOR) IS INHIBITED AND AN END-OF-DATA RETURN IS GIVEN. THOSE IMAGES NOT REQUESTED BY THE PROGRAM ARE BYPASSED WHEN THE PROGRAM IS FINISHED (A MESSAGE DENOTING THIS IS PLACED IN RUN PRINT FILE). AN EXAMPLE OF THE USE OF THE @XQT STATEMENT WOULD BE

```
@XQT,BA   FILE1.PAYDAY
••••••
USER DATA IMAGES
••••••
ENDED BY NEXT CONTROL STATEMENT
```

WHERE THE OPTIONS FOR CONTROLLING THE PROGRAM ARE 'B' AND 'A', THE FILE 'QUALIFIER' IS TAKEN AS THE 'PROJECT-ID', THE 'FILE' PORTION OF THE 'EXTERNAL' FILE NAME IS 'FILE1', AND THE ELEMENT TO BE EXECUTED IS 'PAYDAY', TAKEN FROM 'FILE1'. IF THE ELEMENT TO BE EXECUTED IS IN THE RUN-TEMPORARY FILE, THE FILE NAME IS NOT NEEDED. IF SUCH WERE THE CASE, AND OPTIONS WERE NOT REQUIRED, THEN THE ABOVE @XQT STATEMENT WOULD REDUCE TO:

```
@XQT PAYDAY
```

ADDITIONAL EXAMPLES OF THE @XQT STATEMENT ARE GIVEN IN THE SECTION ON THE COLLECTOR.

5.7.3. THE @EOF STATEMENT

THE @EOF STATEMENT IS USED AS A FILE DIVIDER (GENERAL SENTINEL) WITHIN THE DATA STREAM WHICH FOLLOWS THE @XQT STATEMENT(OR PROCESSOR CALL STATEMENT). IT IS THE ONLY CONTROL STATEMENT THAT CAN BE BYPASSED(READ) BY A USER PROGRAM. THE FORMAT OF THE @EOF STATEMENT IS

```
@EOF S
```

WHERE 'S' IS A ONE-CHARACTER SENTINEL TO BE PASSED TO THE REQUESTING PROGRAM AT THE TIME THE STATEMENT IS REQUESTED. WHEN THE @EOF IS DETECTED BY READ$, AN ABNORMAL RETURN IS MADE TO THE REQUESTOR WITH THE CHARACTER FOUND AT 'S' MADE AVAILABLE. A SUBSEQUENT REQUEST WILL CAUSE THE NEXT IMAGE TO BE TRANSMITTED. AN @EOF IS NEVER TRANSMITTED AS SUCH.

FOR THE EXACT MANNER OF USING @EOF STATEMENTS AND DETECTING END-OF-DATA, THE READER SHOULD SEE THE SECTION ON READ$.

AN EXAMPLE WHERE THE @EOF STATEMENT IS USED IS:

```
@XQT  PROGX
••••••
DATA OF PART 1
••••••
@EOF A
••••••
DATA OF PART 2
••••••
@XQT PROGY
```

ALL CARDS BETWEEN THE TWO @XQT STATEMENTS ARE TO BE INPUT BY PROGX. THE @EOF STATEMENT SERVES AS A MARKER BETWEEN THE TWO FILES.

5.7.4. THE @PMD STATEMENT

THE @PMD STATEMENT MAY BE USED TO OBTAIN A POST-MORTEM DUMP OF ALL OR PART OF THE CORE STORAGE USED BY AN EXECUTION TASK. THE FORMAT OF THIS STATEMENT MAY TAKE ONE OF TWO FORMS:

```
@PMD,OPTIONS  NAME1,NAME2,NAME3,...,NAMEN
OR:
@PMD,OPTIONS NAME,START,LENGTH,FORMAT
```

DETAILED DISCUSSIONS OF EACH OF THESE FORMS ARE GIVEN IN THE SECTION ON DIAGNOSTIC AIDS, HENCE ONLY A BRIEF SUMMARY IS GIVEN AT THIS POINT. ALL @PMD STATEMENTS FOLLOWING AN EXECUTION ARE HONORED UNTIL A CONTROL STATEMENT IS ENCOUNTERED WHICH IS NOT A CONDITIONAL CONTROL STATEMENT (I.E., @SETC, @TEST OR @JUMP) OR A @PMD STATEMENT (ANY OTHER STATEMENT WILL CAUSE THE TERMINATION OF THE PMD MODE). THE AVAILABLE 'OPTIONS' ARE DIVIDED INTO TWO CLASSES-SPECIAL AND STANDARD. IF A SPECIAL OPTION IS USED, THE FIRST FORM OF THE @PMD STATEMENT IS REQUIRED, AND 'NAME1', 'NAME2', ETC., ARE CONTROL, SEGMENT, OR ELEMENT DATA DESCRIPTION ELEMENTS WHICH ARE TO BE DUMPED ACCORDING TO THE 'OPTIONS' SPECIFIED. IF ONLY STANDARD OPTIONS ARE SPECIFIED, THE SECOND FORM IS USED. THE FIELD 'NAME' MAY SPECIFY AN ELEMENT OR SEGMENT TO BE DUMPED, OR IT MAY BE VOID, IN WHICH CASE ALL OF THE USER'S AREA OF CORE IS DUMPED. IF 'NAME' IS SPECIFIED, 'START' AND 'LENGTH' SPECIFY AN AREA OF THE ELEMENT OR SEGMENT TO BE DUMPED, AND 'FORMAT' SPECIFIES A FORMAT TO BE USED FOR THE DUMP LISTING.

STANDARD OPTIONS ALLOW FOR CONDITIONAL DUMPS, DEPENDING UPON THE TERMINATION OF THE RUN, FOR CHANGED-WORD DUMPING, AND FOR DUMPING ALL OF BLANK COMMON. SPECIAL OPTIONS ALLOW FOR DUMPING ALL OF AN ELEMENT OR SEGMENT; OR ONLY BANK 1 OR BANK 2 PORTIONS OF AN ELEMENT; AND FOR SPECIFYING ONLY ELEMENTS WHICH ARE NOT TO BE DUMPED.

5.8. CONDITIONAL STATEMENTS

5.8.1. PURPOSE OF CONDITIONAL STATEMENTS

THE CONDITIONAL STATEMENTS ARE SET APART FROM OTHER EXECUTIVE CONTROL STATEMENTS BECAUSE THEY ARE SPECIAL-USE FEATURES AND NEED NOT BE OF CONCERN IN MANY APPLICATIONS.

THE CONDITIONAL CONTROL STATEMENTS ARE USED TO ACCOMPLISH DYNAMIC ADJUSTMENT OF THE CONTROL STREAM AS IT IS BEING EXECUTED. A COMMON 'CONDITION' WORD IS MAINTAINED BY THE SYSTEM THROUGHOUT THE COURSE OF A RUN. THE VALUE IN THE 'CONDITION' WORD IS REFERENCED (TESTED OR SET) FROM WITHIN THE CONTROL STREAM VIA THE CONDITIONAL STATEMENTS, CAUSING PORTIONS OF THE STREAM TO BE BY-PASSED. IN ADDITION, ALL USER PROGRAMS WITHIN THE RUN AND THE EXECUTIVE HAVE THE ABILITY TO ACCESS THE WORD AND/OR RESET THEIR RESPECTIVE THIRDS (SEE THE SECTION ON THE 'CONDITION' WORD). THIS METHOD MAY CAUSE THE USER PROGRAM TO TAKE DIFFERENT PATHS AND/OR TO SET PARTS OF THE WORD SUCH THAT PORTIONS OF THE CONTROL STREAM ARE SKIPPED. THIS CONDITIONAL NETWORK ALLOWS A GIVEN CONTROL STREAM TO PRODUCE MUCH DIFFERENT RESULTS WITH ONLY A SLIGHT MODIFICATION TO THE STREAM OR WITH NO MODIFICATION IF THE EFFECTIVE STREAM IS DICTATED BY USER PROGRAMS REACTING TO STIMULI SUCH AS AMOUNT OF DATA, DAY OF MONTH, TIME OF DAY, ETC.

5.8.2. STATEMENT LABELS

THE EXECUTIVE LANGUAGE IS SUCH THAT CONTROL STATEMENTS MAY BE LABELED. THIS FEATURE IS PROVIDED IN ORDER TO ALLOW FUNCTIONS (STATEMENTS) TO BE SKIPPED WITH CONTROL BEING PASSED TO A STATEMENT WITH A PARTICULAR LABEL. THE @JUMP CONTROL STATEMENT (DESCRIBED LATER) IS USED TO MOVE CONTROL TO A STATEMENT WITH A PARTICULAR LABEL.

AS DESCRIBED IN THE FIRST PART OF THIS CHAPTER, THE LABELED EXECUTIVE CONTROL STATEMENT HAS THE FORMAT:

```
    @LABEL:COMMAND,OPTIONS  SPEC1,SPEC2,...,SPECN   COMMENT
```

WHERE THE LABEL IS LIMITED TO SIX CHARACTERS FROM THE ALPHANUMERIC SET(A...Z,0...9), BEGINS WITH AN ALPHABETIC, AND IS IMMEDIATELY FOLLOWED BY THE COLON(:).

AN EXAMPLE OF AN @XQT STATEMENT THAT IS LABELED IS:

```
    @A:XQT  PROGX
```

WHERE 'A' IS THE LABEL AND 'PROGX' IS THE ELEMENT TO BE EXECUTED.

A LABEL SPECIFICATION ON CERTAIN CONTROL STATEMENTS IS MEANINGLESS AND WILL BE IGNORED. THOSE STATEMENTS WHICH FALL INTO THIS CLASS ARE, @RUN,@EOF,@FILE,@ENDF, AND @COL.

A LABEL(OR LABELS) MAY BE ATTACHED TO AN EXISTING CONTROL STATEMENT WITHOUT PHYSICALLY CHANGING THE STATEMENT, BY USE OF THE @LABEL STATEMENT(SEE BELOW).

5.8.3. THE LABEL STATEMENT

A LABEL CAN BE PLACED ON AN EXISTING CONTROL STATEMENT BY PLACING A @LABEL STATEMENT IMMEDIATELY PRECEDING THE EXISTING STATEMENT. THE FORMAT OF THE STATEMENT IS:

```
@LABEL:
```

WHERE 'LABEL' IS THE TAG TO BE ATTACHED. IF A LABEL IS ALSO PRESENT ON THE EXISTING STATEMENT, THE STATEMENT IS RECOGNIZED BY BOTH LABELS. IF MORE THAN ONE @LABEL STATEMENT IS PRESENT, ALL ARE ATTACHED. AS AN EXAMPLE, THE @XQT STATEMENT BELOW CAN BE REFERENCED BY BOTH THE LABEL 'A', AND THE LABEL 'B'.

```
@A:
@B:XQT  PROGX
```

IF THE SAME LABEL APPEARS MORE THAN ONCE WITHIN A RUN, THE FIRST FORWARD OCCURRENCE IS TAKEN AS THE PROPER LABEL.

5.8.4. THE 'CONDITION' WORD

THE SYSTEM MAINTAINS A 'CONDITION' WORD (COMPUTER WORD OF 36 BITS) FOR EACH ACTIVE RUN. THE 'CONDITION' WORD IS SET TO ZERO AT THE BEGINNING OF A RUN (IN THE ABSENCE OF A 'SET' SPECIFICATION ON THE @START STATEMENT). THIS WORD IS DIVIDED INTO THREE PARTS FROM LEFT TO RIGHT. THE LEFT THIRD MAY BE SET BY THE EXECUTIVE ONLY (FOR ERROR CONDITIONS, ETC.), THE MIDDLE THIRD MAY BE SET EXTERNALLY IN THE CONTROL STREAM VIA

@SETC, AND THE RIGHT THIRD IS SET BY THE INTERNAL USER PROGRAM VIA AN INTERNAL REFERENCE TO SETC$. USER PROGRAMS CAN RETRIEVE THE ENTIRE WORD (VIA AN INTERNAL REFERENCE TO COND$) AND THE WORD CAN BE TESTED FROM WITHIN THE CONTROL STREAM, CAUSING BRANCHING TO A PARTICULAR STATEMENT, VIA THE @TEST CONTROL STATEMENT. A @JUMP STATEMENT IS PROVIDED FOR BRANCHING WHEN A PARTICULAR TEST IS MET.

THE STATE OF THE 'CONDITION' WORD, WHETHER SET FROM THE CONTROL STREAM OR BY USER PROGRAMS, CAN BE MONITORED AT ANY POINT WITHIN A RUN TO DECIDE HOW THE RUN SHOULD BEST PROCEED.

THE EXECUTIVE USES THE LEFT THIRD OF THE CONDITION WORD TO INDICATE THE TYPE OF PROGRAM TERMINATION. THE VALUES THAT MAY BE FOUND IN THIS PORTION OF THE CONDITION WORD, AND THEIR CORROSPONDING MEANINGS, ARE GIVEN BELOW:

0. EXIT$ TERMINATION OF ALL ACTIVITIES.

1. EXIT$ TERMINATION OF LAST ACTIVITY, ERR$ TERMINATION OF ONE OR MORE PREVIOUS ACTIVITIES.

2. ERR$ TERMINATION OF LAST ACTIVITY, EXIT$ TERMINATION OF ALL PREVIOUS ACTIVITIES, IF PRESENT.

3. ERR$ TERMINATION OF LAST ACTIVITY, ERR$ TERMINATION OF ONE OR MORE PREVIOUS ACTIVITIES.

4. ABORT$ TERMINATION OF LAST ACTIVITY, EXIT$ TERMINATION OF ALL PREVIOUS ACTIVITIES, IF PRESENT.

5. ABORT$ TERMINATION OF LAST ACTIVITY, ERR$ TERMINATION OF ONE OR MORE PREVIOUS ACTIVITIES.

A VALUE OF 2 OR 3 CAUSES A BATCH RUN TO BE TERMINATED AFTER PROCESSING PMD CONTROL STATEMENTS AND CONDITIONAL STATEMENTS. A VALUE OF 4 OR 5 WILL CAUSE IMMEDIATE TERMINATION OF THE RUN.

5.8.5. THE @SETC STATEMENT

THE @SETC CONTROL STATEMENT IS USED TO STORE (SET) A VALUE IN THE SECOND THIRD OF THE 'CONDITION' WORD. THE FORMAT OF THE STATEMENT IS:

```
@SETC  VALUE/J
```

WHERE 'VALUE' MUST BE SPECIFIED AND 'J' IS OPTIONAL BUT ASSUMED TO BE T2 IF ABSENT. THE 'VALUE' SUB-FIELD CONTAINS A POSITIVE, OCTAL NUMBER NOT TO EXCEED 4 DIGITS. IT IS TREATED AS 36 BITS(RIGHT JUSTIFIED, ZERO FILLED) PRIOR TO THE PARTIAL-WORD STORE IN THE 'CONDITION' WORD. IF THE MAGNITUDE OF THE NUMBER IS GREATER THAN CAN BE CONTAINED IN THE 'J' DESIGNATED PORTION OF THE 'CONDITION' WORD, TRUNCATION OCCURS.

ALLOWABLE 'J' DESIGNATORS ARE:

T2	MIDDLE THIRD
S3	THIRD SIXTH FROM LEFT
S4	FOURTH SIXTH FROM LEFT

EXAMPLES OF THE @SETC STATEMENT ARE:

```
@SETC  6
```

WHERE THE SECOND THIRD OF THE 'CONDITION' WORD IS SET TO 6 OR 0006, AND

```
@SETC  10/S3
```

WHERE THE THIRD SIXTH IS SET TO 10 OCTAL, WITH THE REST OF THE WORD LEFT UNDISTURBED.

5.8.6. THE @JUMP STATEMENT

THE @JUMP CONTROL STATEMENT IS USED WHEN STATEMENT EXECUTION IS TO BE BRANCHED TO A PARTICULAR LABELED STATEMENT. THE FORMAT OF THE STATEMENT IS:

```
@JUMP  LABEL
```

WHERE 'LABEL' APPEARS AS A LABEL ON A SUBSEQUENT CONTROL STATEMENT, OR IS A DECIMAL NUMERIC(N) SPECIFYING THAT CONTROL IS TO BE PASSED TO THE NTH CONTROL STATEMENT THAT FOLLOWS, EXCEPT THAT THOSE STATEMENTS WHICH CANNOT HAVE LABELS ARE NOT CONSIDERED IN THE COUNT. NOTE THAT THE @JUMP STATEMENT MUST REFERENCE IN THE FORWARD DIRECTION (TO A STATEMENT NOT YET PROCESSED). A NUMERIC OF ZERO(0) IS ILLEGAL.

5.8.7. THE @TEST STATEMENT

THE @TEST CONTROL STATEMENT IS USED TO TEST THE VALUE OF THE 'CONDITION' WORD FOR THE PURPOSE OF SELECTING PARTICULAR CONTROL STATEMENTS TO BE EXECUTED (OR SKIPPED). THE FORMAT OF THE @TEST STATEMENT IS:

```
@TEST  F/VALUE/J,F/VALUE/J,...
```

WHERE 'VALUE' CONTAINS A POSITIVE, OCTAL NUMBER NOT EXCEEDING 12 DIGITS. THE 'J' FIELD IS OPTIONAL AND WHEN IT IS NOT SPECIFIED THE MIDDLE THIRD IS ASSUMED. ALLOWABLE 'J' DESIGNATORS ARE:

W	WHOLE WORD
H1	LEFT HALF
H2	RIGHT HALF
T1 THRU T3	LEFT THIRD THRU RIGHT THIRD
S1 THRU S6	LEFT SIXTH THRU RIGHT SIXTH

THE 'F' FIELD(FUNCTION FIELD) SPECIFIES THE TEST TO BE MADE. IF MORE THAN ONE FUNCTION APPEARS ON THE STATEMENT, SCANNING CONTINUES UNTIL A TEST IS MET OR ALL FUNCTIONS ARE EXHAUSTED. THE CONTROL STATEMENT IMMEDIATELY FOLLOWING THE @TEST STATEMENT WILL BE SKIPPED IF A TEST IS MET; OTHERWISE, IT WILL BE EXECUTED. ALLOWABLE FUNCTIONS ARE:

```
TE        TEST EQUAL
                  (SKIP THE NEXT CONTROL STATEMENT IF THE
                  'J' DESIGNATED PORTION OF THE 'CONDITION'
                  WORD IS EQUAL TO 'VALUE' OR IN SIMPLER
                  TERMS, SKIP IF C EQUALS V)
TNE       TEST FOR NOT EQUAL
                  (SKIP IF C NOT EQUAL TO V)
TG        TEST FOR GREATER
                  (SKIP IF C GREATER THAN V)
TLE       TEST FOR LESS THAN OR EQUAL
                  (SKIP IF C LESS THAN OR EQUAL TO V)
```

THE SPECIFIED 'VALUE' IS INTERPRETED IN THE SAME MANNER AS FOR THE @SET STATEMENT(FULL 36 BITS), HOWEVER IT WILL APPEAR NEGATIVE IF THE UPPERMOST BIT IS SET. THIS IS ALSO TRUE FOR THE 'CONDITION' WORD WHEN THE ENTIRE WORD OR A THIRD IS BEING TESTED.

NOTE: THE @SETC STATEMENT IS EQUIVALENT TO THE MACHINE INSTRUCTION 'STORE A'(SA) WHERE THE 36 BIT 'VALUE' IS FOUND IN THE 'A' CONTROL REGISTER.

THE TEST FUNCTIONS ARE EQUIVALENT TO THE MACHINE INSTRUCTIONS 'TE,TNE,TG,AND TLE' WHERE THE 36 BIT 'VALUE' IS IN THE CONTROL REGISTER.

AN EXAMPLE OF THE USE OF THE @TEST CONTROL STATEMENT WOULD BE:

```
@TEST TE/6/T2,TG/12/H2
@XQT  PROGX
```

IF THE MIDDLE THIRD OF THE 'CONDITION' WORD IS EQUAL TO 6, OR IF THE RIGHT HALF IS GREATER THAN 12, THE @XQT STATEMENT WOULD BE SKIPPED, OTHERWISE IT WOULD BE EXECUTED.

CONSIDER THE FOLLOWING RUN WHICH UTILIZES ALL THREE OF THE CONDITIONAL CONTROL STATEMENTS(@SETC, @TEST AND @JUMP):

```
@RUN   ID6,PROJ1,888294,10
@SETC  6                    INITIAL SET OF 'CONDITION' WORD
......
......
@TEST  TE/6
@XQT   PROGX
......
......
@TEST  TE/6,TE/3
@JUMP 2
```

```
@JUMP A
@TEST TE/10,TE/4
@JUMP 3
@SETC  4
@JUMP B
@TEST TE/11
@JUMP  C
@XQT   PROGY
••••••
••••••
@A:XQT  PROGA
••••••
••••••
@B:XQT  PROGB
••••••
••••••
@C:XQT  PROGC
@FIN
```

BY CHANGING THE VALUE (NOW 6) ON THE INITIAL @SETC STATEMENT, THE RUN CAN BE MADE TO PRODUCE DIFFERENT RESULTS. AS THE RUN IS NOW 'SET', THE PROGRAMS A, B AND C WILL BE EXECUTED. IF THE INITIAL 'SET' VALUE WERE 3, THEN PROGRAM X WOULD ALSO BE EXECUTED. IF OCTAL 10 OR 4, PROGRAMS Y AND A WOULD BE SKIPPED. IF 11, ALL PROGRAMS ARE EXECUTED. IF SOME OTHER NUMBER, PROGRAMS Y, A AND B ARE SKIPPED.

ALTHOUGH NOT SHOWN IN THE EXAMPLE, IT IS IMPORTANT TO NOTE THAT PROGX, IF EXECUTED, COULD HAVE SET SOME PART OF THE RIGHT THIRD OF THE 'CONDITION' WORD. IN THE EXAMPLE ABOVE, THIS WOULD NOT HAVE AFFECTED THE PATHS TAKEN, BUT IF ANY PART OF THIS THIRD WERE TESTED VIA @TEST, IT WOULD HAVE HAD A PART IN DETERMINING WHETHER THE TESTS WERE MET OR NOT. THE SAME IS TRUE CONCERNING THE EXECUTIVE THIRD.

5.9. STATEMENT SYNTAX ERROR DIAGNOSTICS

WHILE THE CONTROL STATEMENT INTERPRETER IS CONVERTING THE CONTROL STATEMENTS FROM EXTERNAL TO INTERNAL FORMAT, IT PERFORMS A SYNTAX CHECK ON EACH STATEMENT. BELOW ARE THE ERROR MESSAGES THAT MAY OCCUR ON THE PRINTER, IMMEDIATELY FOLLOWING THE STATEMENT, WHEN A SYNTAX ERROR IS DETECTED.

```
1. XX   ILLEGAL OPTION  Z
2. XX   ILLEGAL CHARACTER  Z
3. XX   MAX NUMBER OF CHARACTERS EXCEEDED
4. XX   MAX NUMBER OF FIELDS OR SUB-FIELDS EXCEEDED
5. XX   REQUIRED FIELD OR SUB-FIELD MISSING
6.      @ IN COLUMN 1 OF CONTINUATION CARD
```

WHERE:
XX = THE CHARACTER POSITION AT WHICH THE ERROR WAS DETECTED.
Z = THE ILLEGAL CHARACTER OR OPTION.

5.10. STATEMENT FORMAT SUMMARY

THE EXECUTIVE CONTROL STATEMENTS ARE SUMMARIZED BELOW.

```
@RUN,PRIORITY/RUN-OPTIONS RUN-ID,ACCOUNTING,PROJECT,;
     RUNNING-TIME/DEADLINE,PAGES/CARDS,START-TIME
@FIN
@LOG  INFORMATION
@MSG,OPTIONS MESSAGE   . COMMENTS
@HDG,OPTIONS  HEADING TEXT
```

```
   @ADD  FILENAME   . COMMENTS
   @START  NAME,SET
OR @START,PRIORITY/RUN-OPTIONS  NAME,SET,RUN-ID,ACCOUNTING,;
     PROJECT,RUNNING-TIME/DEADLINE,PAGES/CARDS,START-TIME
   @SYM,OPTIONS  FILENAME/N,TYPE,SYMBIONT,FILE-LABELS
   @SYM,OPTIONS  SYMBIONT,NAME,FILE LABELS,REMOTE STATIONS
   @COL,OPTIONS  XX
   @CKPT,OPTIONS  FILENAME
   @RSTRT,PRIORITY/OPTION RUNID,ACCOUNTING,FILENAME,CKPT#,REEL#
   @ASG,OPTIONS  NAME/KEY1/KEY2,TYPE/RESERVE/GRANULE/MAXIMUM
   @ASG,OPTIONS  NAME/KEY1/KEY2,TYPE/UNITS/LOG/NOISE,;
    REEL1/REEL2/,.,/REELN
   @ASG  NAME,TYPE/LOCATIONS
   @ASG,OPTIONS  NAME,TYPE
   @MODE,OPTIONS  NAME/NOISE            (TAPE ONLY)
   @CAT,OPTIONS  NAME/KEY1/KEY2,TYPE/UNITS/LOG/NOISE,;
     REEL1/REEL2/.,./REELN
OR @CAT,OPTIONS  NAME/KEY1/KEY2,TYPE/RESERVE/GRANULE/MAXIMUM
   @FREE,OPTIONS  NAME1
   @USE   INTERNAL,EXTERNAL
OR @USE   INTERNAL,INTERNAL
   @ELT,OPTIONS    ELEMENT1,ELEMENT2,SENTINEL
   @DATA,OPTIONS  FILENAME1,FILENAME2,SENTINEL
   @END   SENTINEL
   @FILE,OPTIONS  FILENAME,DEVICE-TYPE
   @ENDF
   @QUAL  QUALIFIER
   @PROCESSOR,OPTIONS ELEMENT1,ELEMENT2,ELEMENT3,...
   @MAP,OPTIONS  ELEMENT1,ELEMENT2,ELEMENT3
   @XQT,OPTIONS  ELEMENT
   @EOF  S
   @PMD,OPTIONS  NAME,START,LENGTH,FORMAT
OR @PMD,OPTIONS  NAME1,NAME2,NAME3,...,NAMEN
   @LABEL:
   @SETC VALUE/J
   @JUMP LABEL
   @TEST  F/VALUE/J,F/VALUE/J,...
```

6. BASIC OPERATIONAL DESCRIPTIONS

6.1. BATCH PROCESSING

6.1.1. RUN PREPARATION

THE ACTIVITIES REQUIRED FOR SUBMITTING RUN INPUT TO THE 1108 EXECUTIVE SYSTEM FOR PROCESSING INCLUDE THE FOLLOWING:

1) PREPARATION OF THE NECESSARY PROGRAM DECKS AND DATA CARDS;

2) PREPARATION OF EXECUTIVE CONTROL STATEMENTS WHICH WILL DIRECT THE PROCESSING; AND

3) PREPARATION OF ANY ADDITIONAL EXECUTIVE CONTROL STATEMENTS NEEDED TO SPECIFY FACILITIES ASSIGNMENTS, POST MORTEM DUMPS DESIRED, ETC.

THIS INFORMATION IS THEN INPUT BY THE USER IN THE ORDER WHICH WILL INSURE PROPER PROCESSING OF THE RUN.

6.1.1.1. TYPICAL RUN INPUT ARRANGEMENT

TO ACQUAINT THE USER WITH THE OPERATION OF THE SYSTEM, A TYPICAL ARRANGEMENT OF RUN INPUT IS ILLUSTRATED BELOW; SECTIONS SUBSEQUENT TO THIS EXAMPLE REFER TO THIS RUN SETUP OCCASIONALLY, FOR PURPOSES OF ILLUSTRATION.

```
@RUN  AK4,888,OPTICS,5,75
@ASG,T  ATMOS,T,A341
@FOR
.....
.....
FORTRAN SOURCE
.....
@XQT
.....
.....
DATA
.....
.....
@PMD
@FIN
```

THIS PARTICULAR RUN WILL RESULT IN A FORTRAN COMPILATION, FOLLOWED BY THE EXECUTION OF THE PROGRAM CONSTRUCTED, AND THEN BY A POST-MORTEM DUMP. THE PROGRAM EXECUTED IS SUPPLIED DATA FROM THE CONTROL STREAM AND FROM AN INPUT TAPE OPTIC*ATMOS WHICH IS ON REEL A341. NOTE THE MANNER IN WHICH THE DROPOUT RULES REDUCE THE COMPLEXITY OF THE CONTROL STATEMENTS.

AS A SOMEWHAT MORE COMPLEX EXAMPLE, SUPPOSE THAT A PROGRAM FILE OPTICS*PROGS CONTAINS A COLLECTION OF SOURCE LANGUAGE AND THE CORRESPONDING RELOCATABLE ELEMENTS, WHICH COMBINE TO FORM A PROGRAM WHICH DETERMINES THE RESOLUTION OF AN OPTICAL SYSTEM. AN OLDER GENERATION, CYCLE 15, OF THE ELEMENT MURK IS TO BE ALTERED TO TAKE INTO ACCOUNT CERTAIN ADDITIONAL ABERRATIONS OF INTEREST. A SPECIAL FILE OF ANSWERS IS PRODUCED BY THE PROGRAM THAT IS TO BE EDITED SEPARATELY FROM THE RUN LISTING NORMALLY PRODUCED. THE SETUP MIGHT BE:

```
@RUN   AL5,888,OPTICS,10
@ASG,T  ATMOS,T,A341
@ASG    SPEC,F                           SPECIAL FILE
@FOR     PROGS.MURK(15), PROGS.MURK/ABER
..............
..............
CORRECTIONS TO CREATE MURK/ABER FROM MURK(15)
..............
..............
@MAP
IN     PROGS.MURK/ABER
@XQT
@SYM   PRNT,SPEC
@FIN
```

THE SPECIAL FILE, NAMED 'SPEC', WAS ASSIGNED(CREATED) ON FASTRAND AND USED BY THE PROGRAM 'MURK/ABER' FOR PRINT OUTPUT (BY MEANS OF PRINTA$). THE PRINTING OF THE FILE BY THE PRINT SYMBIONT 'PRNT' WAS THEN CALLED FOR BY USE OF THE @SYM CONTROL STATEMENT.

6.1.2. RUN SUBMISSION

SUBMISSION OF A RUN TO THE SYSTEM FOR PROCESSING MAY BE FROM EITHER A PRIMARY INPUT DEVICE LOCAL TO THE 1108, OR FROM A REMOTE TERMINAL. THE SYSTEM INPUT SYMBIONTS CONTROL ALL PRIMARY INPUT DEVICES FROM WHICH RUNS CAN BE SUBMITTED. THE CHAPTER ENTITLED 'SYMBIONTS' CAN BE REFERENCED FOR A COMPLETE DESCRIPTION OF THE OPERATION OF THE INPUT SYMBIONTS.

WHEN AN INPUT SYMBIONT IS ACTIVATED, THE CONTROL STREAM ACCEPTED MAY CONSIST OF ANY NUMBER OF 'STACKED' RUNS. ALL RUNS IN THE STREAM ARE HEADED BY A @RUN STATEMENT AND ENDED BY A @FIN STATEMENT. FOR EACH RUN SUBMITTED, THE PARTICULAR INPUT SYMBIONT WILL CONSTRUCT A FILE, PLACE IT ON MASS STORAGE IF NECESSARY, AND QUEUE IT FOR INSPECTION BY THE COARSE SCHEDULER. THE SYMBIONT EXTRACTS THE RUN-ID AND PRIORITY SPECIFICATIONS FROM THE @RUN STATEMENT FOR INCLUSION IN THE QUEUE ENTRY. FOR EACH @RUN STATEMENT PROCESSED BY THE SYMBIONTS, AN ENTRY IS MADE IN THE SYSTEM LOG AS TO THE TIME THE RUN ENTERED THE SYSTEM.

6.1.3. RUN SCHEDULING

THE COARSE SCHEDULER IS RESPONSIBLE FOR THE SELECTION OF RUNS TO BE INITIATED. SELECTION IS MADE FROM THOSE OF THE HIGHEST PRIORITY GROUP. WITHIN THIS GROUP THE RUNS ARE EXAMINED, IN THE ORDER OF RUN SUBMISSION, IN SEARCH OF A RUN WHOSE FACILITY REQUIREMENTS CAN BE MET. THE DEADLINE AND START-TIME ARE ALSO CONSIDERED IN THE SELECTION PROCESS.

WHEN A RUN IS SELECTED THE OPERATOR IS IMMEDIATELY NOTIFIED (BY MEANS OF CONSOLE MESSAGES) OF THE INITIAL, EXTERNAL FACILITY ASSIGNMENTS, IF ANY. THE OPERATOR CAN THEN PERFORM THE REQUIRED SETUP, SUCH AS REEL MOUNTING, ETC. ONCE A RUN HAS BEEN SELECTED (OPENED), ITS FIRST TASK IS CONSIDERED AVAILABLE FOR EXECUTION AS SOON AS SUFFICIENT CORE SPACE IS AVAILABLE. ACCOUNTING INFORMATION IS ALWAYS OUTPUT TO THE SYSTEM LOG FILE INDICATING THE TIME AT WHICH A RUN WAS OPENED. A MORE DETAILED EXPLANATION OF SCHEDULING PROCESS CAN BE FOUND IN THE 'SUPERVISOR' CHAPTER, SECTION 7.

6.1.4. RUN EXECUTION

ALL OF THE TASKS OF A RUN ARE PROCESSED IN THE ORDER IN WHICH THEY ENTER THE SYSTEM. HOWEVER, THE PROCESSING OF A RUN MAY BE INTERRUPTED BETWEEN TASKS TO ALLOW ANOTHER RUN OR A TASK FROM ANOTHER RUN, TO BE PROCESSED. DURING THE EXECUTION OF ANY TASK, THE TASK MAY BE INTERRUPTED MANY TIMES TO ALLOW TASKS FROM OTHER RUNS (AS WELL AS NECESSARY EXECUTIVE ACTIVITIES) TO GAIN TEMPORARY CONTROL OF THE PROCESSOR. SUCH ACTION USUALLY OCCURS WHENEVER THE TASK BECOMES INTERLOCKED AGAINST THE COMPLETION OF SOME ACTIVITY SUCH AS AN INPUT/OUTPUT OPERATION, OR A TASK OF HIGHER PRIORITY HAD A PREVIOUSLY IMPOSED INTERLOCK REMOVED, THUS MAKING IT READY FOR ADDITIONAL PROCESSING. ACCOUNTING INFORMATION IS COMPILED IN SUCH A MANNER THAT EACH RUN IS CHARGED FOR ONLY THE TIME IN WHICH IT ACTUALLY HAS CONTROL OF THE PROCESSOR.

6.1.5. POST-RUN PROCESSING

TERMINATION OF A RUN IS NORMALLY TRIGGERED BY THE COMPLETION OF THE LAST TASK WITHIN THE RUN. AT THIS POINT, SPACE RESERVED FOR ANY TEMPORARY FILES OR MASS STORAGE DEVICES IS RELEASED AND THE CORE STORAGE OCCUPIED BY THE FINAL TASK IS RELEASED. ALSO, ANY PERIPHERAL EQUIPMENT, SUCH AS TAPE UNITS OR PAPER TAPE EQUIPMENT ASSIGNED TO THE RUN, ARE RETURNED TO THE POOL OF AVAILABLE FACILITIES. AN ENTRY IS ALSO MADE IN THE SYSTEM LOG INDICATING THE COMPLETION TIME OF THE RUN.

THE PROCESSING OF THE PRINT FILES CREATED BY A RUN IS INITIATED AT THE TERMINATION OF THE RUN, IF FACILITIES ARE AVAILABLE. IF FACILITIES ARE NOT AVAILABLE, SUCH FILES ARE MAINTAINED ON MASS STORAGE DEVICES BY THE EXECUTIVE SYSTEM, AND THEIR PROCESSING IS INITIATED WHENEVER SUCH FACILITIES DO BECOME AVAILABLE.

6.2. DEMAND PROCESSING

THE EXECUTIVE SYSTEM DEFINES DEMAND PROCESSING AS A DEMAND AND RESPOND TYPE OF ACTIVITY, I.E., CONVERSATIONAL. CONVERSATION VIA A REMOTE CONSOLE CAUSES THE SYSTEM, A DEMAND PROCESSOR, OR AN ACTIVE PROGRAM TO IMMEDIATELY REACT AND RESPOND.

DEMAND PROCESSING TERMINALS CAPABLE OF SUBMITTING RUNS ARE TYPICALLY REMOTE FROM THE COMPUTER SITE. THESE TERMINALS CAN OPERATE IN EITHER THE BATCH OR DEMAND MODE AT THE OPTION OF THE USER; E.G., TELETYPE* KEYBOARD OR PAPER TAPE INPUT THROUGH THE TELETYPE. THE FOLLOWING SECTIONS ARE CONCERNED ONLY WITH THE DEMAND MODE. OTHER REMOTE TERMINALS USED FOR DATA TRANSMISSION ARE CONSIDERED LATER.

THE DISTINCTION IN THE SYSTEM BETWEEN BATCH-MODE PROCESSING AND DEMAND PROCESSING LIES IN THE FREQUENT INTERACTION WITH THE USER THAT OCCURS DURING DEMAND PROCESSING. THE TERMINAL USER CAN BE CONSIDERED TO BE IN CONVERSATION WITH THE EXEC, SPECIAL DEMAND PROCESSING FUNCTIONS, USER PROGRAMS, OR THE BATCH FUNCTIONS OF THE SYSTEM ON A UNIT BASIS.

TASKS EXECUTED BY THE DEMAND USER HAVE FREQUENT BUT SHORT BURSTS OF COMPUTATION. PROGRESS IS ALWAYS INSISTED UPON; HOWEVER, TO RECEIVE A SUBSTANTIAL AMOUNT OF COMPUTATION MAY REQUIRE A LONG PERIOD OF TIME. (ACCESS TO COMPUTATION IS A PERCENTAGE OF THE TOTAL COMPUTING FACILITY AND IS SCHEDULED IN SMALL INCREMENTS OF TIME AT FREQUENT INTERVALS TO PROVIDE IMMEDIATE RESPONSES, GIVING THE APPEARANCE OF TOTAL SYSTEM CONTROL TO THE USER AND THE IMPRESSION THAT HE IS THE ONLY USER CURRENTLY RUNNING). THE MORE A USER INTERACTS WITH A DEMAND PROGRAM, THE SHORTER THE BURSTS OF COMPUTATION, BUT OF COURSE, THE MORE HE IS GIVEN CONTROL. THE LESS INTERACTION, THE LONGER THE BURSTS BUT THE FEWER TIMES HE IS GIVEN CONTROL. THUS, IN REALITY, THE BURSTS OF COMPUTATION ARE OPTIMIZED TO PROVIDE AN APPARENT IMMEDIATE RESPONSE, WITH THE PROGRAM PLACED IN A DORMANT MODE DURING IDLE PERIODS AWAITING RESPONSE FROM THE USER.

WHILE A DEMAND PROGRAM IS IN A DORMANT MODE, DURING IDLE PERIODS AWAITING RESPONSE FROM THE USER THE EXECUTIVE MAY DEEM IT NECESSARY TO 'SWAP' CORE. NORMALLY THIS WILL HAPPEN ONLY WHEN CORE IS FULL AND ANOTHER DEMAND PROGRAM, WHICH IS CURRENTLY ON MASS STORAGE, HAS WORK TO DO.

*TRADEMARK OF TELETYPE CORPORATION

6.2.1. CONTROL STATEMENTS

CONVERSATIONAL PROCESSING IS ALWAYS CONTROLLED BY THE USER THROUGH THE USE OF THE EXECUTIVE CONTROL LANGUAGE WHICH WAS DESIGNED WITH THE PARTICULAR NEEDS OF THE DEMAND PROCESSING USER IN MIND. THESE STATEMENTS ALLOW HIM TO SET THE MODE OF OPERATION, USE LIBRARY FACILITIES, INTERRUPT, ALTER, STOP OR RESET THE OPERATION OF HIS RUN ETC. ALL OF THE CONTROL STATEMENTS ARE AVAILABLE TO THE DEMAND PROCESSING USER; HOWEVER, CERTAIN BATCH-PROCESSING FUNCTIONS WILL GIVE A UNIT RESPONSE TO THEIR INPUT RATHER THAN A LINE BY LINE CONVERSATION EXPECTED OF TRUE DEMAND FUNCTIONS.

A SIMPLE EXAMPLE FOLLOWS TO DEMONSTRATE THE USE OF THE SYSTEM IN A DEMAND PROCESSING MODE. THE RUN PRESENTED SHOWS THE COMMUNICATION THAT OCCURS AT A DEMAND TERMINAL AS A NEW FILE IS ESTABLISHED, A SIMPLE PROGRAM ASSEMBLED, THE PROGRAM PLACED IN THE FILE, AND THEN EXECUTED. THE COMMENTS ON THE RIGHT EXPLAIN THE ACTION TAKING PLACE AND ARE NOT PART OF THE ACTUAL COMMUNICATIONS. THE NON-PRINTING CHARACTERS DO NOT APPEAR (CARRIAGE RETURN AT THE END OF EACH LINE). THIS EXAMPLE BEGINS FOLLOWING THE INITIAL CONTACT WITH THE SYSTEM:

```
AFTER DIALING THE CENTRAL SITE           (THE SYSTEM IS READY FOR
AND RECEIVING THE READY TONE,            INPUT)
THE USER SHOULD TYPE IN A SITE
ID:

U1108A

UNIVAC 1108 TIME SHARING SYSTEM
#RUN XYZ,311202,DEMO                     (THE RUN BEGINS
                                         WITH RUNID, ACCOUNT,
                                         AND PROJECT NUMBER
                                         TO IDENTIFY THE USER).
#ASG,C PF,F/5                            (A 5 TRACK FILE
                                         'DEMO*PF' IS ASSIGNED,
                                         TO BE CATALOGUED AT
                                         THE END OF RUN).

#ASM,I   PF.ODDEVEN,ODDEVEN
                                         ELEMENT CALLED
                                         'ODDEVEN').
ASM 1/1/67                               (THE ASSEMBLER IS
                                         READY TO ACCEPT
                                         INPUT).
     AXR$                                (A PROC TO DEFINE
                                         REGISTER NAMES IS
                                         CALLED FROM THE
                                         SYSTEM LIBRARY).

P    FORM   12,6,18
ST*  P$RINT (P 5,4,STMSG)
     R$EAD  (QUIT, INPUT)
     L      A1,INPUT?                    (FORGOT ',S1';
                                         DELETE IMAGE AND
                                         TRY AGAIN).
     L,S1   A1,INPUT
     L      A0,(P 1,4,ODD)
     JB     A1,ST+1
     L      A0,(P 1,4E',EVEN)            (WENT BACK TO FIX
                                         A MISSING COMMA.
                                         (DOUBLE QUOTE-TTY).)
     J      ST+1
QUIT ER     EXIT$
                                         (REQUEST EXECUTION.)
INPUT RES   14
STMSG 'TYPE A SINGLE NUMBER.'
ODD   'IT'S ODD; TRY ANOTHER.'
EVEN  'IT'S EVEN; TRY ANOTHER.'
     END    ST
```

```
ASM COMPLETE                            (THE ASSEMBLY IS
                                        FINISHED).
#MAP,I    A,ODDEVEN
     IN ODDEVEN
#XQT  ODDEVEN



TYPE A SINGLE NUMBER.                   (NOW THE PROGRAM
1                                       AND THE USER
IT'S ODD; TRY ANOTHER.                  CONVERSE).
4
IT'S EVEN; TRY ANOTHER.
A
IT'S EVEN; TRY ANOTHER.                 (SMART PROGRAM--).
#FIN                                    (THAT'S ENOUGH).


27/ 3/67   0945

RUNID:    XYZ ACCOUNT:      311202 PROJECT:        DEMO
TIME: 0000.02 IN: 00023 OUT: 00000 PAGES: 0001


(EOT)                                   (END OF TRANSMISSION
                                        REQUEST TO QUIT THE
                                        LINE).
LINE RELEASED                           (LAST WORDS FROM
                                        SYSTEM).
```

6.2.2. CONVERSATIONAL COMPONENTS

THE FOLLOWING DESCRIBES THE COMPONENTS OF THE CONVERSATIONAL SYSTEM AND THEIR GENERAL FUNCTION.

6.2.2.1. SYMBIONT COMPLEX

ALL CONVERSATION BETWEEN A PROGRAM AND A REMOTE OR ON-SITE TERMINAL IS HANDLED BY THE EXECUTIVE'S SYMBIONT COMPLEX. IT CONSISTS OF A COMMON CONTROL, DEVICE HANDLERS FOR EACH DEVICE TYPE, A POOL FOR STORING IMAGES(HOWEVER, WHENEVER POSSIBLE OUTPUT WILL BE IMMEDIATELY TRANSFERRED TO THE TERMINAL WITHOUT GOING THROUGH A TEMPORARY STORE), AND INTERFACE TO PROGRAMS.
IT SHOULD BE NOTED THAT THE PROGRAM INTERFACE FOR THE SYMBIONTS IS CONSISTENT FOR ALL DEVICE TYPES.

6.2.2.2. RE-ENTRANT CONTROL

DURING SYSTEM GENERATION TIME A LIST IS MADE OF THE RE-ENTRANT PROCESSORS (SEE SYSTEM GENERATION STATEMENTS - RE-ENTRANT ROUTINES, AND INITIAL SYSTEM). AT EACH CALL, THE SUPERVISOR LOADS THE CORRECT RE-ENTRANT PROCESSOR, PROVIDES THE LINKAGE TO IT, AND SETS THE PROCESSOR'S ENTRY IN RE-ENTRANT CONTROL AS NON-AVAILABLE FOR SWAPPING. AFTER THE OPERATION IN THE PROCESSOR HAS BEEN COMPLETED, IT IS DETACHED FROM THE WORKER PROGRAM AND THE PROCESSOR'S ENTRY IN RE-ENTRANT CONTROL IS SET AS AVAILABLE FOR SWAPPING. IF CORE SPACE NEEDS REQUIRE THAT A RE-ENTRANT PROCESSOR BE REMOVED FROM CORE, THE SUPERVISOR WILL SELECT THE ONE TO BE SWAPPED FROM THOSE PROCESSORS WHICH ARE SET AS AVAILABLE FOR SWAPPING.

6.2.2.3. PROGRAM LIBRARY SERVICES

INCLUDED IN THE SYSTEM IS THE FACILITY TO FETCH AND FORMAT A SYMBOLIC PROGRAM FROM A FILE, COMPILE OR ASSEMBLE IT, AND PUT THE RESULTING SYMBOLIC, RELOCATABLE, AND/OR

ABSOLUTE ELEMENTS INTO A FILE. ALSO AVAILABLE ARE THE SERVICES OF FUR/PUR INCLUDING THE MANIPULATIONS OF TAPE FILES.

6.3. REAL TIME PROCESSING

THE PROCEDURES ON RUN SUBMISSION FOR REAL TIME PROCESSING ARE VERY SIMILAR TO BATCH PROCESSING. INITIALLY THE RUN PRIORITY CONTROLS EXECUTION, BUT AT ANY POINT THE REAL-TIME OR NON-REAL TIME STATE MAY BE ESTABLISHED OR MODIFIED BY MAKING A SPECIAL STATUS CHANGE EXECUTIVE REQUEST (RT$ OR NRT$).

6.3.1. PRIORITY

THE EXECUTIVE TASKS OF HIGHER PRIORITY THAN THE REAL TIME ESI COMPLETION ACTIVITIES ARE RESTRICTED TO INTERRUPT QUEUING ON ALL STANDARD CHANNELS, INTERRUPT PRE-PROCESSING ON ALL ESI CHANNELS, AND ACTIVATION OF ANY OTHER ESI COMPLETION ACTIVITY. AN ESI COMPLETION ACTIVITY WILL LOSE ITS PRIORITY STATUS AT THE MOMENT AN ER REQUEST IS SUBMITTED BY THE ACTIVITY OR AFTER A TIMED INTERVAL WHICHEVER OCCURS FIRST. WHEN THE COMPLETION ACTIVITY LOSES ITS HIGH PRIORITY STATUS, IT WILL DROP TO THE PRIORITY LEVEL OF THE PARENT ACTIVITY THAT INITIALLY REGISTERED THE ESI COMPLETION ACTIVITY. THEREFORE, PROGRAMS UTILIZING COMMUNICATION EQUIPMENT DIRECTLY WITH MULTIPLE BUFFERS MUST BE CLASSIFIED AS REAL TIME IN ORDER TO RECEIVE THE PROPER RESPONSE.

REAL TIME USER ACTIVITIES RECEIVE SWITCHING PRIORITY DIRECTLY BELOW INTERRUPT PROCESSING AND I/O FUNCTION INITIATION FOR ALL STANDARD CHANNELS AS WELL AS BELOW THE FUNCTIONS MENTIONED ABOVE(INTERRUPT QUEUING, ESI INTERRUPT PROCESSING, ACTIVATION OF ESI COMPLETION ACTIVITIES AND THE PROCESSING OF ESI COMPLETION ACTIVITIES). REAL TIME PROGRAMS RECEIVE TOP PRIORITY IN REGARD TO I/O REQUEST INITIATION AND ARE DECLARED INELIGIBLE FOR SWAPPING. THE PROCEDURES INVOLVED IN CHANGING THE SWITCHING LEVEL OF A REAL TIME WORKER PROGRAM IS FOUND IN SEC. 8(RT$)

A USER PROGRAM MAY GO INTO A SUSPENDED STATE AWAITING A TIMED INITIATION, BUT IF REAL TIME RESPONSE INITIATION IS REQUIRED, THE PROGRAM MUST HAVE A REAL-TIME CLASSIFICATION WHEN SUSPENDED.

BY EXECUTIVE REQUEST, THE REAL-TIME PROGRAM CAN EXPAND AND CONTRACT AT WILL, WITH THE DELAY FOR EXPANSION HELD BY THE EXECUTIVE SYSTEM TO THE TIME REQUIRED TO SUSPEND OTHER LOWER PRIORITY PROGRAMS AND SWAP THEM OUT TO MASS STORAGE.

6.3.2. TIMING

ESI INTERRUPT PROCESSING AND SWITCHING TIMES UNDER VARIOUS CONDITIONS HAVE BEEN CALCULATED FOR REAL TIME PROGRAMS BY COUNTING THE ACTUAL INSTRUCTIONS INVOLVED. THESE COUNTS MUST BE CONSIDERED AS ESTIMATES BECAUSE OF LATER CODING ADDITIONS, DELETIONS AND/OR ENHANCEMENTS.

6.3.2.1. ESI INTERRUPTS

WHEN AN ESI INTERRUPT OCCURS VARIOUS FUNCTIONS MUST BE ACCOMPLISHED BEFORE INTERRUPTS CAN BE ENABLED. THE AMOUNT OF TIME REQUIRED BEFORE INTERRUPTS ARE ALLOWED DEPENDS UPON THE TYPE OF INTERRUPT (INPUT MONITOR, OUTPUT MONITOR OR EXTERNAL) AS WELL AS THE TYPE OF OPERATION (SINGLE OR MULTIPLE BUFFER MODE). A MAXIMUM OF APPROXIMATELY 45 MICROSECONDS OF INSTRUCTION EXECUTION TIME WILL ELAPSE BEFORE INTERRUPTS ARE ENABLED.

1) IF AN ESI ACTIVITY HAS BEEN INTERRUPTED, CONTROL WILL BE RETURNED TO THE ACTIVITY WITHIN THE 45 MICROSECONDS MENTIONED ABOVE.

2) IF CONTROL IS TO BE GIVEN TO A NEW ESI ACTIVITY, APPROXIMATELY 120-180 MICROSECONDS WILL ELAPSE BETWEEN THE TIME THE ESI INTERRUPT OCCURRED AND THE NEW ACTIVITY RECEIVES CONTROL. THE TIME REQUIRED DEPENDS UPON THE NUMBER OF CONTROL REGISTERS TO BE SAVED IF A PROGRAM HAS BEEN INTERRUPTED AS WELL AS THE AMOUNT OF POST-PROCESSING REQUIRED FOR THE ACTIVITY IN THE COMMUNICATIONS HANDLER.

6.3.2.2. REAL-TIME WORKERS

IF A REAL-TIME WORKER PROGRAM HAS BEEN INTERRUPTED TO PROCESS AN INTERRUPT, APPROXIMATELY 230 MICROSECONDS ELAPSE BETWEEN THE TIME THE INTERRUPT OCCURRED AND THE REAL-TIME PROGRAM RECEIVES CONTROL (THE ACTUAL TIME DEPENDS UPON THE TYPE OF INTERRUPT BEING PROCESSED). IF A NEW I/O FUNCTION IS AVAILABLE FOR INITIATION BEFORE THE REAL-TIME PROGRAM RECEIVES CONTROL, APPROXIMATELY 480 MICROSECONDS OF INSTRUCTION EXECUTION ARE REQUIRED BETWEEN INTERRUPT OCCURRENCE AND PROGRAM CONTROL.

7. SUPERVISOR

7.1. GENERAL

THE SUPERVISOR IS THE 1108 EXECUTIVE SYSTEM COMPONENT THAT CONTROLS THE SCHEDULING AND EXECUTION OF ALL RUNS ENTERING THE 1108. IT IS DESIGNED TO CONTROL THE EXECUTION OF AN UNLIMITED NUMBER OF PROGRAMS IN A MULTI-PROGRAMMING ENVIRONMENT, WHILE ALLOWING EACH PROGRAM TO BE UNAFFECTED BY THE COEXISTENCE OF OTHER PROGRAMS.

RUNS ENTERING THE 1108 ARE SORTED INTO INFORMATION FILES BY THE INPUT SYMBIONTS, AND THESE FILES ARE SUBSEQUENTLY USED BY THE SUPERVISOR FOR RUN SCHEDULING AND PROCESSING. THUS, RUNS ARE SEGREGATED AT INPUT TIME, AND CONTROL STATEMENTS FOR EACH ARE SUBSEQUENTLY RETRIEVED AND SCANNED BY THE CONTROL STATEMENT INTERPRETER TO FACILITATE THE SELECTION OF RUNS BY THE COARSE SCHEDULER. THE COARSE SCHEDULING OF EACH RUN IS PRIMARILY DEPENDENT ON TWO FACTORS, THE PRIORITY OF THE RUN AND ITS FACILITY REQUIREMENTS. AT APPROPRIATE TIMES, THE SCHEDULER CONTROLS THE SETUP OF EACH RUN, BY INDIRECTLY FORCING TAPE MOUNTING MESSAGES, ETC.

THE DYNAMIC ALLOCATOR IS RESPONSIBLE FOR TAKING RUNS SET UP BY THE COARSE SCHEDULER AND ALLOTTING CORE SPACE ACCORDING TO THE NEEDS OF THE INDIVIDUAL TASKS OF A RUN. EACH RUN MAY BE THOUGHT OF AS BEING MADE UP OF TASKS, WHERE A TASK IS DEFINED TO BE A SINGLE OPERATION OF A SYSTEM PROCESSOR OR THE EXECUTION OF A USER PROGRAM. ALL TASKS FOR A GIVEN RUN WILL BE PROCESSED SERIALLY; HOWEVER, TASKS OF SEPARATE RUNS WILL BE INTERLEAVED.

THE PROCESSOR DISPATCHING ROUTINE IS RESPONSIBLE FOR THE ALLOCATION OF PROCESSOR TIME FOR ALL ACTIVE TASKS. THROUGH THE FUNCTIONS OF THIS ROUTINE, A TRUE MULTI-PROGRAMMING ENVIRONMENT IS ACHIEVED.

VARIOUS OTHER COMPONENTS OF THE SUPERVISOR PROVIDE THE USER WITH ACCESS TO THE CAPABILITIES OF THE EXECUTIVE SYSTEM ONCE A PROGRAM IS EXECUTING. AMONG THESE ARE THE EXECUTIVE REQUEST ROUTINE AND THE INTERRUPT PROCESSING ROUTINES. THE EXECUTIVE REQUEST ROUTINE IS THE INTERFACE BY WHICH THE PROGRAM COMMUNICATES WITH THE SUPERVISOR AND MAKES REQUESTS FOR EXECUTIVE SERVICES, WHILE THE INTERRUPT PROCESSING ROUTINE PROVIDES THE USER WITH THE CAPABILITY OF REACTING TO THE OCCURRENCE OF PROCESSOR INTERRUPTS.

BASICALLY, THE SUPERVISORY PART OF THE EXECUTIVE SYSTEM MAY BE DIVIDED INTO TWO CLASSES OF FUNCTION--THOSE WHICH ARE RESIDENT, AND THOSE WHICH ARE TRANSIENT. THE RESIDENT KERNEL COMPRISES THE FOLLOWING:

- INTERRUPT SUPERVISOR.
- CPU DISPATCHER.
- INPUT/OUTPUT CONTROL.
- DEVICE HANDLERS FOR TAPE, FASTRAND, COMMUNICATIONS SUB-SYSTEMS, ETC.(RECOVERY SEQUENCES ARE TRANSIENT).
- DRUM HANDLER, INCLUDING RECOVERY SEQUENCES.
- DYNAMIC ALLOCATOR.
- CORE CONTENTS CONTROL.
- EXECUTIVE REQUEST SUPERVISOR.
- REAL-TIME CLOCK AND DAY CLOCK ROUTINES.
- BLOCK BUFFERING PACKAGE.
- TASK AND SEGMENT LOADER.
- CONSOLE HANDLER.
- POOLING ROUTINE AND DATA POOL.
- READ$ AND PRINT$.
- LOGGING CONTROL.
- ERROR INTERRUPT SUPERVISOR.
- CORE PARITY RECOVERY ROUTINE.

- POWER-LOSS CONTROL ROUTINE.

THE TRANSIENT ROUTINES COMPRISE THE FOLLOWING:

- CONTROL STATEMENT INTERPRETER.
- COARSE SCHEDULER.
- FACILITIES INVENTORY.
- FASTRAND SPACE ASSIGNMENT.
- COMMUNICATIONS INTERFACE ROUTINES.
- CLT DIAL-UP AND AUTOMATIC-ANSWER.
- SYMBIONT PROBE ROUTINES.
- SYMBIONTS.
- LOGGING AND ACCOUNTING.
- I/O ERROR RECOVERY SEQUENCES FOR TAPE, FASTRAND, ETC.
- ABSOLUTE DUMP ROUTINE.

7.2. MULTI-PROCESSING

THE SCHEDULING AND PROCESSOR SWITCHING TECHNIQUES USED IN THE 1108 EXECUTIVE SYSTEM ARE DESIGNED TO PROVIDE MULTI-PROCESSING CAPABILITIES. THE EXTENSION TO MULTI-PROCESSING LEADS NATURALLY FROM THE MULTI-PROGRAMMING ASPECTS OF THE SYSTEM WHERE MANY INDEPENDENT TASKS IN WORKER PROGRAMS ARE AVAILABLE AT ANY INSTANT FOR PROCESSING. WITHIN THE EXEC ITSELF A SIMILAR SITUATION EXISTS WHERE AT ANY MOMENT MORE THAN ONE INDEPENDENT TASK IS WAITING TO BE DONE. PREVIOUS COMPUTING SYSTEMS USING INTERRUPT SCHEMES SIMILAR TO THE 1108 HAVE A LOGICAL STRUCTURE WHICH IS PREPARED FOR MULTI-PROCESSING AND IS PREVENTED ONLY BY THE LACK OF HARDWARE. CONSIDER AN OPERATING PROGRAM THAT HAS AN OUTSTANDING REQUEST FOR AN I-O OPERATION. WHEN THE I-O COMPLETION INTERRUPT OCCURS, THE EXEC INTERRUPTS THE PROGRAM AT A POINT UNKNOWN TO IT. THE PROGRAM ENVIRONMENT IS SAVED, THE INTERRUPT PROCESSED, AND CONTROL RETURNED TO THE PROGRAM. THE PROGRAM IS NEVER AWARE OF THE EVENT, AND WOULD NOT BE THE WISER HAD THE INTERRUPT TAKEN THE EXEC TO ANOTHER PROCESSOR WHERE THE TASK WAS PERFORMED TO COMPLETE THE I-O OPERATION.

IN THE MULTI-PROCESSING CONFIGURATION ONE EXEC IN THE SHARED CORE MEMORY OF THE 1108 CONTROLS ALL PROCESSING. THE MULTI-PROCESSING EXTENSIONS TO THE 1108 PROVIDE THE ABILITY TO ISOLATE EACH OF THE AVAILABLE PROCESSORS. EACH IN TURN ACTING AS THE EXEC PROCESSOR INSPECTS THE LIST OF CURRENT ACTIVITIES AND SELECTS A TASK TO BE DONE. ONE PROCESSOR MAY INTERLOCK THE OTHERS WHILE REFERENCING THE CRITICAL AREAS OF COMMON DATA.

THE MANY RUNS BEING INPUT TO THE 1108 EXECUTIVE SYSTEM PROVIDE A NUMBER OF TASKS TO BE MULTI-PROCESSED. WITHIN ANY RUN, THE INDIVIDUAL TASKS ARE EXECUTED IN A SERIAL MANNER AS DIRECTED BY THE USER. AMONG THE MANY RUNS, THE EXEC USES THE PROCESSORS OF THE SYSTEM TO WORK ON TASKS OF MORE THAN ONE RUN.

WITHIN ANY PROGRAM, THE EXEC PROVIDES THE ABILITY TO CODE PROCEDURES WHICH SPLIT A PROGRAM INTO AN ARBITRARY NUMBER OF INDEPENDENT PATHS, AVAILABLE FOR PROCESSING ON ANY OF THE PROCESSORS OF THE SYSTEM. SYSTEM DESIGN PROVIDES THE ABILITY TO USE ALL AVAILABLE PROCESSORS ON THE EXECUTION OF A PARTICULAR PROGRAM. THE EXECUTIVE IS OF COURSE FREE TO USE THE AVAILABLE PROCESSORS FOR PROGRAM-RELATED BUT INDEPENDENT TASKS NECESSARY WITHIN THE EXEC.

THE EXEC ROUTINES INTERFACING WITH THE RESIDENT WORKER PROGRAMS ARE RE-ENTRANT IN DESIGN. FOR MINOR TASKS REQUESTED OF THE EXEC MANY OF THE ROUTINES ARE TOTALLY RE-ENTRANT. OTHERS, WHEN IN THE MULTI-PROCESSING ENVIRONMENT, WILL QUEUE THE WORKER PROGRAM REQUEST WHERE SERIAL PROCESSING IN A PARTICULAR AREA OF THE EXEC IS REQUIRED. AT THE LOWEST LEVEL, THE EXEC MUST QUEUE INTERRUPTS USING THE DESIGNATED PROCESSOR. IT MUST ALSO ISSUE I-O REQUESTS ON PARTICULAR PROCESSORS ACCORDING TO THE AVAILABLE HARDWARE LINKAGES. ABOVE THIS BASIC LEVEL, ANY OF THE AVAILABLE PROCESSORS CAN PERFORM A GIVEN TASK, WITH SELECTION BASED ON THE PRIORITY OF THE TASKS CURRENTLY BEING EXECUTED BY THE PROCESSORS. THE AREAS OF EXEC CODING WHICH REFERENCE COMMON DATA AND OTHERS WITH SPECIALIZED CODING METHODS MUST BE PROTECTED FROM SIMULTANEOUS

EXECUTION, BUT MANY'AREAS WILL BE OPEN AND MULTI-PROCESSED AS NECESSARY.

7.3. SCHEDULING

7.3.1. GENERAL

THE GENERAL SCHEDULING TECHNIQUE USED BY THE EXECUTIVE REMOVES ANY DIFFICULTY IN THE ADVANCEMENT OF AN INSTALLATION INTO THE USE OF MULTI-PROGRAMMING. THE TECHNIQUE IS EASILY UNDERSTOOD, AND ANY INSTALLATION CAN READILY MODIFY OR EXTEND ITS CAPABILITIES IF NECESSARY.

THE SCHEDULING ROUTINES OF THE SUPERVISOR ARE RESPONSIBLE FOR THE CONTROL OF FACILITIES AS WELL AS THE ACTUAL SCHEDULING OF RUNS AND TASKS. THIS INCLUDES BOTH THE ASSIGNMENT AND RELEASE OF FACILITIES. THERE ARE FIVE ROUTINES WITHIN THE SYSTEM FOR HANDLING THE SCHEDULING OF RUNS AND THE TASKS WITHIN RUNS. THE IDENTITIES ARE:

FACILITIES INVENTORY
CONTROL STATEMENT INTERPRETER(CSI)
COARSE SCHEDULER
DYNAMIC ALLOCATOR
CORE CONTENTS CONTROL

EACH ROUTINE IS DISCUSSED INDIVIDUALLY IN THE SECTIONS WHICH FOLLOW.

7.3.2. FACILITIES INVENTORY

THE FACILITIES AT THE DISPOSAL OF THE EXECUTIVE SYSTEM INCLUDE THE INPUT/OUTPUT CHANNELS, ALL PERIPHERAL EQUIPMENT ATTACHED TO THESE CHANNELS, INCLUDING AVAILABLE COMMUNICATION LINE TERMINALS AND ALL CORE STORAGE SPACE. AVAILABLE FACILITIES AND THEIR DISPOSITION ARE INDICATED TO THE SYSTEM AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - CONNECT AND DISCONNECT CHANNELS - CORE SIZE, AND INITIAL SYSTEM); THEREAFTER, THE EXECUTIVE SYSTEM ASSIGNS THESE FACILITIES, AS NEEDED AND AS AVAILABLE, TO FULFILL THE FACILITY REQUIREMENTS OF ALL RUNS ENTERING THE 1108. THE EXECUTIVE SYSTEM MAINTAINS AND CONTINUALLY UPDATES INVENTORY TABLES THAT REFLECT WHAT FACILITIES ARE AVAILABLE FOR ASSIGNMENT, AND WHICH RUNS ARE USING THE CURRENTLY UNAVAILABLE FACILITIES.

AS DEMONSTRATED IN THE FOLLOWING TEXT, THE FACILITIES INVENTORY ROUTINE IS A BASIC CONSTITUENT OF THE SCHEDULING SECTION OF THE SUPERVISOR. ONLY BY ESTABLISHING AN EFFICIENT AND CONVENIENT (FOR THE USER) MEANS OF FACILITIES CONTROL IS THE MULTI-PROGRAMMING ENVIRONMENT PRACTICAL. THE ROUTINES CONTROLLING THE FACILITIES AVAILABLE TO THE 1108 ARE DESIGNED TO OPTIMIZE UTILIZATION OF THOSE FACILITIES WHILE REQUIRING A MINIMUM OF USER-GENERATED STATEMENTS CONCERNING OPERATIONAL REQUIREMENTS.

DEVICES SUCH AS MAGNETIC TAPES ARE NORMALLY ASSIGNED BEFORE RUN-EXECUTION TIME, SINCE THEY CANNOT BE SHARED BY TWO OR MORE RUNS, AND, SINCE THEY NORMALLY REQUIRE OPERATOR SET UPS. SUCH DEVICES ARE ALWAYS RELEASED AUTOMATICALLY BY THE TERMINATION OF THE RUN; HOWEVER THEY MAY ALSO BE RELEASED DURING THE COURSE OF THE RUN BY THE USER.

MAGNETIC TAPES ARE ALWAYS ASSIGNED WITH THE FOLLOWING RULES OF ALLOCATION.

TYPE DESIGNATION	ORDER OF SATISFYING REQUEST
T	8C,8CB,6C,6CB,8C9,6C9,4C,4CB,3A,2A.
C	8C,8CB,6C,6CB,4C,4CB
U	8C,8CB,6C,6CB
8C	8C,8CB
6C	6C,6CB
4C	4C,4CB
CB	8CB,6CB,4CB

U8	8CB,6CB
U9	8C9,6C9

OTHER 'TYPE' MAGNETIC TAPE ASSIGNMENTS DO NOT HAVE A SECOND CHOICE. (8CB,6CB,4CB,8C9,6C9,3A AND 2A)

BOTH CORE STORAGE SPACE AND MASS STORAGE SPACE (FASTRAND, FH-432, FH-880, ETC) ARE DYNAMICALLY ASSIGNABLE BY THE EXECUTIVE SYSTEM AND DYNAMICALLY RELEASABLE BY THE USER. THE ALLOCATION OF CORE SPACE IS DISCUSSED IN A SUBSEQUENT SECTION CALLED 'CORE CONTENTS CONTROL'. MASS STORAGE SPACE IS TREATED MUCH THE SAME AS CORE STORAGE SPACE BECAUSE SPACE CAN BE DYNAMICALLY REQUESTED AND RELEASED. THE USER IS ENCOURAGED TO DO SO WHENEVER POSSIBLE, SINCE NO RELOCATION OF INFORMATION IS NECESSARY IN ALLOCATING MASS STORAGE FACILITIES(AS MAY BE THE CASE FOR CORE STORAGE).

DRUM SPACE IS AVAILABLE IN CONTINUOUS BLOCKS IN ORDER TO ALLOW USERS TO TAKE ADVANTAGE OF THE HARDWARE CHARACTERISTICS OFFERED BY A WORD-ADDRESSABLE DRUM UNIT LIKE THE FH-432 AND FH-880. THE PORTION OF DRUM SPACE NOT USED IN CONTINUOUS BLOCKS AND NOT USED FOR SYSTEM RESIDENCE IS THE SIMULATED FASTRAND AREA USED FOR FILE STORAGE. THIS PERMITS FILES TO BE INDEPENDENT OF DEVICE, THUS RESULTING IN INCREASED OVERALL EFFICIENCY, BECAUSE FASTER DEVICES CAN BE USED WHENEVER SPACE IS AVAILABLE. A PORTION OF DRUM IS SET ASIDE AT SYSTEM GENERATION TIME FOR THE RESIDENCE OF THE SYSTEM AND THE PROCESSORS (SEE SYSTEM GENERATION STATEMENTS - SYSTEM RESIDENCE - MOVE PROCESSORS - DIVIDE DRUM). NORMALLY, ALL FILES CREATED BY A USER ON A DRUM DURING THE COURSE OF A RUN, ARE PURGED AT THE COMPLETION OF THE RUN, AND THE SPACE USED FOR SUCH FILES IS RETURNED TO THE POOL OF AVAILABLE FACILITIES. IF SO SPECIFIED, HOWEVER, THE FILE MAY BE RETAINED FOR FUTURE REFERENCE.

FASTRAND SPACE IS NORMALLY ALLOCATED IN GRANULES OF A SINGLE TRACK. THE SINGLE SECTOR ALLOCATION IS RESERVED FOR SYSTEM USAGE ONLY. BY USE OF THE @ASG EXECUTIVE CONTROL STATEMENT, A USER SPECIFIES THE NUMBER OF GRANULES TO BE INITIALLY RESERVED FOR THE FILE. EACH DYNAMIC REQUEST FOR ADDITIONAL SPACE WILL THEN RESULT IN THE ASSIGNMENT OF AN ADDITIONAL GRANULE. WHEN USING THE SYSTEM FILE CONTROL ROUTINES, THE USER WILL NOT HAVE TO REQUEST ADDITIONAL MASS STORAGE SPACE BECAUSE THIS PROCEDURE IS TAKEN CARE OF AUTOMATICALLY BY THE SYSTEM. FASTRAND SPACE IS TREATED DIFFERENTLY FROM DRUM SPACE, IN THAT IT MAY BE ASSIGNED FOR UNLIMITED DURATIONS. A USER'S PROGRAM FILE, FOR INSTANCE, MAY BE FORMED ON A FASTRAND UNIT AND LEFT THERE INDEFINITELY, WHEREAS USER FILES ON THE DRUM ARE NORMALLY EITHER PURGED AT THE COMPLETION OF A RUN OR ARE TRANSCRIBED TO FASTRAND.

IN ADDITION TO MAINTAINING COGNIZANCE OF SYSTEM INFORMATION CONCERNING DEVICE ERRORS, ETC., THE FACILITIES INVENTORY ROUTINE IS ABLE TO ACCEPT DIRECTION FROM AN OPERATOR CONCERNING DEVICE RELIABILITY, ETC. THE OPERATOR MAY INDICATE WEAK DEVICES, AND HE MAY REQUEST THAT DEVICES OR CHANNELS BE REMOVED FROM THE POOL OF AVAILABLE FACILITIES. THESE CAPABILITIES ARE FURTHER DISCUSSED IN A LATER SECTION.

7.3.3. CONTROL STATEMENT INTERPRETER(CSI)

THE CONTROL STATEMENT INTERPRETER IS USED BY THE SYMBIONT COMPLEX FOR THE PURPOSE OF INTERPRETING THE INPUT CONTROL STREAM. IT IS ALSO USED BY THE FUNCTION CSF TO INTERPRET CONTROL STATEMENT IMAGES PASSED TO THAT FUNCTION INTERNALLY VIA ER CSF$ REQUESTS. FOR EACH CONTROL STATEMENT IMAGE PASSED TO THE CONTROL STATEMENT INTERPRETER, CSI PERFORMS THE FOLLOWING MAIN TASKS:

1. FORMAT CHECKS THE STATEMENT
2. REQUESTS CONTINUATION STATEMENTS (CONTINUATION CHARACTER IS TREATED AS AN ILLEGAL CHARACTER IF REQUEST WAS TO CSF)
3. CHANGES THE EXTERNAL FORMAT OF THE CONTROL STATEMENT TO AN INTERNALLY FORMATTED TABLE ACCEPTABLE TO THE SYSTEM

THE CONTROL STATEMENT INTERPRETER IS THE SINGLE COMPONENT OF THE SUPERVISOR THAT DICTATES THE FORMAT OF THE EXECUTIVE CONTROL LANGUAGE. IF IT BECAME NECESSARY TO INPUT A CONTROL STREAM DIFFERENT FROM THAT SPECIFIED BY THE EXECUTIVE CONTROL LANGUAGE, THE CONTROL STATEMENT INTERPRETER COULD BE MODIFIED TO ACCEPT THIS INPUT AS

LONG AS THE INTERPRETATION PRESENTED FUNCTIONS KNOWN TO THE COARSE SCHEDULER AND IN THE PROPER ORDER AND GROUPING. THE COARSE SCHEDULER IS THE LEVEL AT WHICH THE CAPABILITIES AND FUNCTIONS OF THE SYSTEM ARE DEFINED. THE COARSE SCHEDULER SCANS THE INPUT STREAM VIA ER READ$ REQUESTS IN SEARCH OF THE NEXT LOGICAL TASK OR PARAMETER ON WHICH IT MUST ACT. IF THE NEXT STATEMENT IS A CONTROL STATEMENT, READ$ ACTIVATES CSI AND THE INTERNAL TABLE IT BUILDS IS PASSED BACK TO THE COARSE SCHEDULER. FOR DATA STATEMENTS, THE STATEMENT IMAGE IS PASSED BACK TO THE COARSE SCHEDULER. THE INTERFACE BETWEEN CSI AND THE SYSTEM IS FIXED, BUT INPUT TO CSI IS FIXED ONLY IN THE SENSE THAT THE EXECUTIVE CONTROL LANGUAGE IS DEFINED TO BE OF A PARTICULAR FORMAT.

7.3.4. COARSE SCHEDULER

COARSE SCHEDULING OCCURS AT THE CONTROL STATEMENT LEVEL AND CONCERNS THE INTRODUCTION OF NEW RUNS INTO THE OPERATING ENVIRONMENT, AS WELL AS CONTINUED PROCESSING OF PARTIALLY COMPLETED RUNS. THE RESULT OF SCHEDULING IS A QUEUE OF TASKS MADE AVAILABLE TO THE DYNAMIC ALLOCATOR AS THOSE READY TO BE INTRODUCED INTO THE OPERATING MIX. THE SYSTEM EXERCISES THE COARSE SCHEDULER WHEN A NEW RUN IS SUBMITTED OR WHEN A CHANGE IN THE PRIORITY, DEADLINE, OR START-TIME OF EXISTING RUNS IS GIVEN BY THE OPERATOR. MORE FREQUENT ARE THE INITIATIONS OF SCHEDULING BECAUSE OF PROGRAM OR TASK TERMINATIONS AND CHANGE IN PERIPHERAL FACILITY STATUS. ANY REDUCTION IN THE TOTAL STORAGE OR FACILITY REQUIREMENTS MIGHT ALLOW A WAITING TASK TO PROCEED.

THE FOLLOWING DISCUSSION CONCERNING COARSE SCHEDULING ASSUMES THAT THE READER IS FAMILIAR WITH THE EXECUTIVE CONTROL LANGUAGE AND WITH THE SCHEDULING PARAMETERS OF THE @RUN STATEMENT IN PARTICULAR.

7.3.4.1. BATCH SCHEDULING

THERE ARE THREE BASIC QUEUES USED BY THE COARSE SCHEDULER IN PROCESSING BATCH RUNS. THESE ARE:

RUN QUEUE - QUEUE OF RUNS NOT YET INITIATED.

STATEMENT QUEUE - CONTAINS AN ENTRY AT THE CONTROL STATEMENT LEVEL FOR EACH INITIATED RUN. EACH RUN IS NECESSARILY REPRESENTED IN THIS QUEUE AND THE PARTICULAR STATEMENT BEING PROCESSED MARKS THE DEPTH TO WHICH PROCESSING HAS PROCEEDED FOR THE RUN. THE TASKS WITHIN THIS QUEUE CAN BE IN VARIOUS STATES. THEY MAY BE IN A WAIT STATE BECAUSE OF FACILITY REQUIREMENTS, AS IN THE CASE OF THE @ASG STATEMENT, OR THEY MAY BE REPRESENTED IN THE CORE QUEUE WHERE THEY ARE CANDIDATES FOR CORE ALLOCATION AND EXECUTION. MANY TASKS CAN BE HANDLED ENTIRELY BY THE COARSE SCHEDULER AND NEVER REACH THE CORE QUEUE.

CORE QUEUE - QUEUE OF TASKS(WORKER PROGRAMS, ETC.) THAT ARE EITHER READY FOR CORE ALLOCATION AND EXECUTION OR ARE BEING EXECUTED. THE CONTROL STATEMENTS REPRESENTED HERE ARE A SUBSET OF THOSE IN THE STATEMENT QUEUE.

THE RUN QUEUE IS INITIALLY BUILT BY THE INPUT SYMBIONTS IN THE COURSE OF PROCESSING INCOMING RUNS. THE NUMBER OF RUNS CURRENTLY BEING PROCESSED BY THE SYSTEM IS LIMITED ONLY BY THE MASS STORAGE SPACE AVAILABLE TO HOLD THE INPUT STREAMS. FOR EACH RUN IN THE RUN QUEUE, THE COARSE SCHEDULER IS AWARE OF THE FOLLOWING SCHEDULING INFORMATION:

PRIORITY
DEADLINE

ESTIMATED RUNNING TIME
START-TIME
ORDER OF RUN SUBMISSION

WITHIN THE STATEMENT QUEUE, EACH ENTRY REPRESENTS THE TASK(ONE OR MORE FUNCTIONS) TO BE PERFORMED AS A RESULT OF ONE CONTROL STATEMENT INTERPRETATION. THE TASKS(STATEMENTS) IN THE STATEMENT QUEUE ARE IN ONE OF FOUR STATES:

1. WAITING FOR FACILITIES OTHER THAN CORE.
2. BEING PROCESSED BY THE COARSE SCHEDULER ITSELF--A CASE WHERE CORE STORAGE IS NOT REQUIRED.
3. REPRESENTED IN THE CORE QUEUE WHERE THEY ARE CANDIDATES FOR CORE ALLOCATION AND EXECUTION. ALL OTHER FACILITY REQUIREMENTS HAVE BEEN MET.
4. IN A WAIT STATE FOR SOME OTHER REASON, SUCH AS OPERATOR ACTION.

AS FACILITIES BECOME AVAILABLE, THE STATEMENT QUEUE IS SCANNED IN PRIORITY ORDER IN AN ATTEMPT TO SATISFY A STATEMENT THAT IS IN A FACILITY-WAIT STATE. THE FACILITIES INVENTORY ROUTINE IS RESPONSIBLE FOR MAINTAINING A LIST OF AVAILABLE FACILITIES, AND KEYS THE COARSE SCHEDULER WHEN ADDITIONAL FACILITIES BECOME AVAILABLE. IN SOME CASES,THE COARSE SCHEDULER WILL LOG A REQUEST WITH THE FACILITIES INVENTORY ROUTINE WHICH WILL AUTOMATICALLY BE HONORED WHEN THE PARTICULAR FACILITY BECOMES AVAILABLE. IF A DEADLINE RUN IS PRESENT, THE COARSE SCHEDULER WILL IMPOSE A HOLD ON FURTHER FACILITY ASSIGNMENTS UNTIL SUCH TIME AS THE REQUIREMENTS OF THE DEADLINE RUN HAVE BEEN MET. IF A PARTICULAR STATEMENT REQUIRES CORE AREA AND ALL OTHER FACILITY REQUIREMENTS HAVE BEEN MET, IT IS ALSO PLACED IN THE CORE QUEUE SO THAT IT MAY BE INTRODUCED INTO THE OPERATING MIX BY THE DYNAMIC ALLOCATOR. WHEN THE TASK HAS TERMINATED, IT IS TURNED BACK TO THE COARSE SCHEDULER FOR POST-PROCESSING.

WHEN ALL OF THE WORK FOR A TASK(STATEMENT) HAS BEEN COMPLETED, THE COARSE SCHEDULER CALLS ON THE CONTROL STATEMENT INTERPRETER FOR FURTHER INTERPRETATION OF THE CONTROL STREAM, WHICH CAUSES THE NEXT STATEMENT OF THAT RUN TO BE ADDED TO THE STATEMENT QUEUE OR THE END-RUN TO BE DETECTED. IN THE LATTER CASE, CLOSEOUT OCCURS. FURTHER CONTROL STATEMENT INTERPRETATION IS ASSUMED TO BE INSTANTANEOUS. SO, IN THAT SENSE, ONE STATEMENT FOR EACH RUN IS ALWAYS REPRESENTED IN THE STATEMENT QUEUE.

THE TASKS IN THE CORE QUEUE ARE IN ONE OF THREE STATES AS FAR AS THE COARSE SCHEDULER IS CONCERNED. THESE ARE:

1. ACTIVE - IN CORE AND BEING EXECUTED BY THE DYNAMIC ALLOCATOR.
2. SUSPENDED - PREVIOUSLY ACTIVATED BUT THEN SUSPENDED FROM CORE BY THE DYNAMIC ALLOCATOR TO CREATE CORE AREA FOR MORE IMPORTANT TASKS. CAN BE EXECUTED AGAIN AS SOON AS CORE IS AVAILABLE.
3. READY - NOT YET INITIATED BUT READY FOR CORE ALLOCATION AND EXECUTION.

EXCLUDING DEADLINE RUNS FOR A MOMENT, IN CONSIDERING THE RUN QUEUE THE COARSE SCHEDULER ATTEMPTS TO SELECT A RUN FOR INITIATION ONLY FROM THE GROUP WITH THE HIGHEST PRIORITY LETTER. WITHIN THIS GROUP, THE RUNS ARE CONSIDERED IN THE ORDER OF RUN SUBMISSION, WITH SELECTION TAKING PLACE ON THE FIRST RUN WHOSE INITIAL FACILITY REQUIREMENTS CAN BE MET. THE COARSE SCHEDULER WILL ATTEMPT TO OPEN A NEW RUN ONLY WHEN ALL STATEMENTS OF CURRENT RUNS ARE REPRESENTED IN THE ACTIVE PORTION OF THE CORE QUEUE. IN OTHER WORDS, A NEW RUN IS OPENED ONLY WHEN EXECUTION IS PROCEEDING FOR ALL CURRENTLY OPENED RUNS. THE MORE TASKS THAT CAN BE EXECUTED AT ONCE, THE MORE RUNS THAT WILL BE OPENED. SINCE THE CORE REQUIREMENTS FOR CONSECUTIVE TASKS WITHIN A RUN MAY VARY GREATLY, IT IS POSSIBLE THAT FOR A GIVEN SET OF RUNS, ALL CURRENT TASKS MIGHT BE ACTIVE AT ONE INSTANCE AND A SHORT TIME LATER MOST OF THE NOW CURRENT TASKS ARE INACTIVE BECAUSE OF LARGER CORE REQUIREMENTS. IT IS ALSO POSSIBLE THAT BECAUSE OF FACILITY REQUIREMENTS, ONLY A FEW OF THE OPENED RUNS CAN HAVE TASKS PLACED IN THE READY OR ACTIVE STATE. THE NUMBER OF OPENED RUNS THEN VARIES WITH THE PARTICULAR MIX OF TASKS AS WELL AS THE FACILITY MESH. WHEN ALL OPENED RUNS ARE PROGRESSING, AN

ADDITIONAL RUN IS INITIATED IF POSSIBLE. IF THE INITIAL FACILITY REQUIREMENTS OF THOSE RUNS OF THE HIGHEST PRIORITY CANNOT BE MET, RUN SELECTION IS HELD UNTIL A FACILITY CHANGE OCCURS.

IF IT BECOMES EVIDENT THAT A DEADLINE RUN WILL NOT BE COMPLETED ON TIME VIA THE NORMAL SCHEDULING ALGORITHM(VIA THE POWER OF ITS PRIORITY LETTER ONLY), THE RUN IS INITIATED IMMEDIATELY, REGARDLESS OF ITS PRIORITY, AND IF NECESSARY ITS TASKS ARE GIVEN HIGH PRIORITY IN THE STATEMENT AND CORE QUEUES. THE ESTIMATED RUNNING TIME OF THE ACTIVE RUNS IS USED AS A GAUGE AS TO WHEN A DEADLINE RUN SHOULD BE INITIATED. THE AMOUNT OF CPU TIME AVAILABLE TO BATCH RUNS, AS OPPOSED TO DEMAND AND REAL TIME, ALSO COMES INTO PLAY.

IN ORDER TO ACCOMMODATE DEADLINE RUNS, THE 'READY' TASKS OF THE CORE QUEUE(THOSE ENTERED BY THE COARSE SCHEDULER) FALL INTO ONE OF FOUR CLASSES AS SHOWN BELOW. THE DYNAMIC ALLOCATOR CONSIDERS THE CLASSES IN THE ORDER SHOWN AND IS RESTRICTED TO SELECTING FROM THE FIRST CLASS THAT IS NOT VOID.

1. CRITICAL DEADLINE - THOSE FOR WHICH CORE MUST BE MADE AVAILABLE IMMEDIATELY, AT THE EXPENSE OF TEMPORARILY SUSPENDING OTHER PROGRAMS FROM CORE IF NECESSARY(EXCLUDING REAL-TIME). SELECTION IS RESTRICTED TO A PARTICULAR TASK.

2. SEMI-CRITICAL DEADLINE - THOSE FOR WHICH CORE IS TO BE MADE AVAILABLE AT THE EXPENSE OF IMPOSING A WAIT ON FURTHER CORE ASSIGNMENT UNTIL SUCH TIME AS CORE IS RELEASED VIA TASK TERMINATIONS. SELECTION IS RESTRICTED TO A PARTICULAR TASK.

3. DEADLINE THOSE THAT HAVE BEEN DECLARED 'DEADLINE' BUT FROM WHICH THE DYNAMIC ALLOCATOR IS STILL FREE TO CHOOSE THE PARTICULAR TASK, FOR BEST 'CORE-FIT' AND MINIMUM OVERHEAD. SAME AS ABOVE CLASS EXCEPT SELECTION IS NOT RESTRICTED TO A PARTICULAR TASK.

4. NORMAL - THOSE FROM WHICH THE DYNAMIC ALLOCATOR IS FREE TO CHOOSE ON A 'BEST-FIT' BASIS, GIVING PREFERENCE TO THOSE WITH THE HIGHEST PRIORITY LETTER IF CORE REQUIREMENTS ARE THE SAME.

A DEADLINE TASK MAY BE FOUND IN ANY ONE OF THE FOUR CLASSES, DEPENDING ON HOW CRITICAL IT IS. CLASS 4 IS CONSIDERED THE NORMAL WAY IN WHICH TASKS ARE SELECTED FOR EXECUTION. TASKS MARKED AS REAL-TIME ARE AUTOMATICALLY PLACED IN CLASS 2 UNLESS A DEADLINE IS ALSO SPECIFIED IN WHICH CASE THEY MAY BE MOVED TO CLASS 1 WHERE CORE AREA IS IMMEDIATELY MADE AVAILABLE(BARRING THE PRESENCE OF OTHER CLASS 1 TASKS). AS AN EXAMPLE, REAL TIME RUNS MAY HAVE A HIGH PRIORITY, 'A' FOR INSTANCE, IN WHICH CASE THEY ARE IN A HIGH PRIORITY GROUP FOR RUN INITIATION AND CORE IS ACQUIRED AS SOON AS IT BECOMES AVAILABLE VIA OTHER TASK TERMINATIONS. A DEADLINE MAY ALSO BE SPECIFIED WHICH CAN CAUSE IMMEDIATE RUN INITIATION, ACQUISITION OF FACILITIES FOR THE HIGHEST PRIORITY IN THE STATEMENT QUEUE, AND THE SUSPENSION OF EXECUTING TASKS IN ORDER TO CREATE CORE SPACE. THE DEADLINE CAN OF COURSE BE APPLIED TO ANY TYPE OF BATCH RUN, HOWEVER IT SHOULD BE USED WITH SOME FORETHOUGHT SINCE IT CAUSES INTERRUPTION OF THE NORMAL SCHEDULING METHODS.

AS MENTIONED PREVIOUSLY, THE DYNAMIC ALLOCATOR RETURNS SUSPENDED TASKS TO THE CORE

QUEUE FOR LATER SELECTION. THE SUSPENDED TASKS FALL INTO TWO CLASSES:

1. SUSPENDED CRITICAL DEADLINE - SUSPENDED BECAUSE OF REAL-TIME OR DEADLINE CORE EXPANSIONS.
2. NORMAL SUSPENDED - SUSPENDED FOR ANY NUMBER OF REASONS(REAL-TIME, DEADLINE, DEMAND, AS WELL AS 'NORMAL' TASK EXPANSIONS).

THE CORE QUEUE THEN IS A SEVEN LEVEL QUEUE WITH ONE LEVEL TO INDICATE THE ACTIVE TASKS(NO SELECTION FROM THIS LEVEL) AND SIX LEVELS FOR THE SIX CLASSES OF TASKS THAT ARE WAITING FOR CORE ALLOCATION. WHEN THE DYNAMIC ALLOCATOR SELECTS A TASK, IT MUST BE TAKEN FROM THE LOWEST NUMBERED LEVEL THAT CONTAINS AN ENTRY.

LEVEL A - ACTIVE

LEVEL 1 - SUSPENDED CRITICAL DEADLINE
LEVEL 2 - CRITICAL DEADLINE
LEVEL 3 - NORMAL SUSPENDED (INCLUDES THOSE STILL CLASSED AS LEVEL 4 OR 5)
LEVEL 4 - SEMI-CRITICAL DEADLINE
LEVEL 5 - DEADLINE
LEVEL 6 - NORMAL

WITHIN LEVELS 1,2,4 AND 5 EACH TASK HAS A SELECTION 'WEIGHT' BASED ON HOW CRITICAL THE RUN'S DEADLINE TIME APPEARS TO THE SYSTEM. THE SELECTION WEIGHT IS SUBJECT TO CHANGE AS THE DYNAMIC ALLOCATOR (OR COARSE SCHEDULER) PERIODICALLY EXAMINES THE SYSTEM LOAD. AT LEVELS 1,2,AND 4, THE DYNAMIC ALLOCATOR IS RESTRICTED TO SELECTING THE TASK THAT HAS THE HIGHEST 'WEIGHT'. AT LEVEL 5 ANY TASK MAY BE SELECTED, WITH DEADLINE 'WEIGHT' GIVEN CONSIDERATION IF CORE REQUIREMENTS ARE THE SAME. A TASK AT LEVEL 3 OR 6 EITHER HAS NO DEADLINE SPECIFICATION ON ITS RUN STATEMENT, OR THE TIME ALLOWED TO COMPLETE THE RUN IS SUCH THAT THE DEADLINE SPECIFICATION IS AS YET BEING IGNORED IN FAVOR OF NORMAL SCHEDULING BASED ON THE PRIORITY LETTER. AS INDICATED EARLIER, AT THESE LEVELS THE DYNAMIC ALLOCATOR IS FREE TO CHOOSE, IN PRIORITY-LETTER ORDER, ANY TASK THAT WILL 'BEST-FIT' OR CAUSE THE LEAST TASK-SHIFT TO MAKE CORE AVAILABLE.

ONCE A TASK HAS BEEN ACTIVATED BY THE DYNAMIC ALLOCATOR AND SUBSEQUENTLY TERMINATES, THE TASK'S CORE AREA IS DUMPED TO MASS STORAGE FOR A POSSIBLE POST-MORTEM DUMP. IF SNAPSHOT DUMPS WERE TAKEN DURING THE EXECUTION, THE DIAGNOSTIC EDITOR IS CALLED TO PROCESS THEM. IF, DURING THE INSPECTION OF THE CONTROL STREAM, @PMD STATEMENTS ARE FOUND, THE PMD PROCESSOR IS CALLED TO PROVIDE POST-MORTEM DUMPS OF THE LAST PROGRAM OR PROCESSOR EXECUTED. SEVERAL @PMD STATEMENTS MAY APPLY TO A GIVEN PROGRAM; HOWEVER, THEY MUST BE GROUPED TOGETHER AS THE PMD PROCESSOR GIVES UP CONTROL WHEN IT READS A NON-PMD STATEMENT (CONDITIONAL STATEMENTS MAY BE INTERLEAVED WITH @PMD STATEMENTS AS THEY ARE PROCESSED BY CSI AND NEVER PASSED TO THE PMD PROCESSOR).

7.3.4.2. DEMAND SCHEDULING

THE COARSE SCHEDULING OF DEMAND RUNS IS SOMEWHAT DIFFERENT FROM THAT OF BATCH RUNS. IT IS ASSUMED THAT THE READER IS FAMILIAR WITH THE PRECEDING SECTION.

AS DEMAND RUNS APPEAR IN THE RUN QUEUE, THEY ARE INITIATED IMMEDIATELY. IN THE COARSE SCHEDULING OF DEMAND RUNS, THE DEADLINE, START-TIME, AND RUNNING TIME ARE OF NO SIGNIFICANCE AND NEED NOT BE SUBMITTED ON THE @RUN STATEMENT. ONLY THE PRIORITY LETTER, ONE OF THE LETTERS A THRU Z, IS USED. IT IS USED TO RESOLVE CONFLICTS ON GAINING EXTERNAL FACILITY ASSIGNMENTS AND TO GIVE CERTAIN PRIVILEGES IN CPU SWITCHING IF AN OVERLOAD SITUATION EXISTS SUCH THAT NOT ALL REMOTE TERMINALS CAN BE GIVEN THE DESIRED RESPONSE TIME (DISCUSSED IN SECTION ON DYNAMIC ALLOCATOR).

AT A GIVEN TIME, THE DEMAND USER IS IDLE, IN CONVERSATION WITH THE EXECUTIVE ON A

CONTROL STATEMENT LEVEL, OR IN CONVERSATION WITH A PARTICULAR CONVERSATIONAL PROCESSOR OR WORKER PROGRAM ON A SOURCE LINE LEVEL. THIS DISCUSSION IS CONCERNED MAINLY WITH THE CASE OF THE DEMAND USER BEING CONVERSATIONAL WITH THE SYSTEM ON A CONTROL STATEMENT LEVEL.

THE PROCEDURES FOLLOWED DURING THE INITIATION AND EXECUTION OF A DEMAND RUN ARE STRAIGHT FORWARD. THE USER MAKES CONTACT WITH THE INPUT SYMBIONT AND SUBMITS A @RUN STATEMENT. THE @RUN REQUEST IS PROCESSED BY THE COARSE SCHEDULER AND THE USER IS FREE TO SUBMIT ANOTHER CONTROL STATEMENT. THE COARSE SCHEDULER IS AGAIN KEYED WHEN THE NEXT CONTROL STATEMENT HAS BEEN RECEIVED BY THE INPUT SYMBIONT AND THE PREVIOUS TASK HAS BEEN COMPLETED. IN MOST CASES, AT LEAST FOR THE CONTROL STATEMENT ITSELF, THE INPUT WILL NOT GO TO MASS STORAGE BUT WILL BE LEFT IN CORE BY THE SYMBIONT FOR IMMEDIATE ACCESS BY THE COARSE SCHEDULER. SINCE MANY USERS MAY BE SUBMITTING STATEMENTS AT ONCE, A QUEUE OF CONTROL STATEMENTS(ONE PER DEMAND USER) MAY BUILD UP AT THE COARSE SCHEDULER LEVEL. THE COARSE SCHEDULER IS EITHER IN THE PROCESS OF ACTING ON A REQUEST OR STANDING BY FOR THE NEXT STATEMENT TO BE SUBMITTED.

EACH CONTROL STATEMENT SUBMITTED BY THE DEMAND USER IS IMMEDIATELY PLACED IN THE STATEMENT QUEUE (DESCRIBED EARLIER). (ONCE A RUN IS OPENED, DEMAND OR BATCH, IT HAS AN ACKNOWLEDGED ENTRY IN THE STATEMENT QUEUE, EVEN THOUGH THIS ENTRY SIMPLY MARKS THAT THE RUN IS IN A WAIT-STATE UNTIL SUCH TIME AS THE NEXT STATEMENT IS INTERPRETED.) NORMALLY A CONTROL STATEMENT CAN BE SATISFIED IMMEDIATELY (WITHIN A REASONABLE RESPONSE TIME). AN EXCEPTION TO THIS IS WHEN FACILITIES, SUCH AS MAGNETIC TAPES, ARE CALLED FOR BUT NOT AVAILABLE. IN THIS CASE THE USER IS NOTIFIED AND CAN ELECT TO TAKE AN ALTERNATE ACTION OR WAIT UNTIL THE ASSIGNMENT CAN BE MADE. IF MORE THAN ONE DEMAND RUN IS WAITING FOR THE SAME FACILITY AND THAT FACILITY BECOMES AVAILABLE, THE ASSIGNMENT IS MADE ACCORDING TO THE PRIORITY LETTERS OF THE DEMAND RUNS. THE MORE LIKELY CASE, FOR DEMAND MODE RUNS, IS THAT ONLY CORE AND MASS STORAGE ARE NEEDED IN ORDER TO SATISFY THE ACTION CALLED FOR ON THE USERS COMMAND. MASS STORAGE IS NORMALLY AVAILABLE, AND CORE IS ALWAYS AVAILABLE FOR FREQUENT BUT SHORT PERIODS OF TIME.

FOR BATCH RUNS, ONCE A TASK IS READY FOR EXECUTION IT IS PLACED IN THE CORE QUEUE (DESCRIBED EARLIER) WHERE IT IS EXECUTED IN THE GENERAL MIX OF BATCH TASKS BUT NOT NECESSARILY IMMEDIATELY OR WITHIN A CERTAIN RESPONSE TIME (EXCEPT FOR DEADLINE JOBS). FOR DEMAND MODE RUNS, WHEN A TASK IS READY FOR CORE SPACE, IT IS INTRODUCED INTO A SPECIAL QUEUE CALLED THE CORE-SWAP QUEUE, WHERE IT IS GIVEN CORE SPACE AND CPU TIME AS SOON AS ITS TURN COMES UP. THIS WILL BE ALMOST IMMEDIATE SINCE EACH TASK IN THE CORE-SWAP QUEUE IS GIVEN A CONTROLLED AMOUNT OF CPU TIME PRIOR TO BEING INTERRUPTED AND POSSIBLY SWAPPED OUT TO MASS STORAGE. THERE MAY BE CORE AVAILABLE, OF COURSE, WITHOUT HAVING TO SWAP OUT OTHER DEMAND TASKS, DEPENDING ON THE CURRENT SYSTEM LOAD. IT IS ALSO POSSIBLE THAT BATCH JOBS WILL GET SUSPENDED FROM CORE (SUSPENSION BEING FOR A LONGER PERIOD OF TIME THAN A SWAP) IN ORDER TO MEET THE RESPONSE TIME FOR DEMAND REQUESTS. NORMALLY HOWEVER, ASSUMING A HEAVY BATCH LOAD, THE TOTAL CORE SPACE BEING USED BY DEMAND TASKS IS SUPPRESSED AS MUCH AS POSSIBLE AS LONG AS REASONABLE RESPONSE TIMES ARE BEING MET. IF THE CORE-SWAP QUEUE WERE TO BE EXAMINED, AT ONE TIME IT MIGHT SHOW ALL TASKS RESIDING IN CORE, AND AT ANOTHER THAT ALL TASKS BUT ONE HAVE BEEN SWAPPED OUT TO MASS STORAGE, DEPENDING ON THE CURRENT SYSTEM LOAD. THE CORE-SWAP QUEUE THEN, SHOWS DEMAND TASKS TO BE IN ONE OF THE FOLLOWING STATES:

1. ACTIVE - IN CORE AND AVAILABLE FOR EXECUTION.
2. SWAPPED-OUT - PREVIOUSLY ACTIVATED BUT MOMENTARILY SWAPPED TO MASS STORAGE IN FAVOR OF ANOTHER DEMAND TASK - BUT CAN BE EXECUTED AGAIN AS SOON AS CORE IS MADE AVAILABLE.
3. READY - NOT YET INITIATED BUT READY FOR CORE ALLOCATION. THIS STATE IS THE SAME IN ALL RESPECTS TO THE 'SWAPPED-OUT' STATE, EXCEPT FOR AN INITIAL-LOAD RATHER THAN A RE-LOAD.

OF THE SWAPPED-OUT TASKS, ONLY THOSE THAT ARE STILL IN NEED OF EXECUTION ARE REPRESENTED IN THE CORE-SWAP QUEUE. AN ADDITIONAL STATE IS THE

INPUT-WAIT STATE

WHERE THE TASK MAY ALSO BE SWAPPED OUT, BUT HAVING REACTED TO THE LAST INPUT AND HAVING NOT YET RECEIVED THE NEXT STATEMENT, IT IS NOT IN NEED OF CPU TIME. WHEN THE INPUT IS RECEIVED, THE TASK IS MOVED TO THE 'SWAPPED-OUT' STATE IN THE CORE-SWAP QUEUE. THE DYNAMIC ALLOCATOR SELECTS TASKS FROM THE CORE-SWAP QUEUE IN A MANNER SUCH AS TO BEST MEET USER RESPONSE TIMES.

WHEN THE USER IS IN CONVERSATION WITH A PROCESSOR OR USER PROGRAM, ON A SOURCE LINE LEVEL, IT IS THE JOB OF THE DYNAMIC ALLOCATOR AND REMOTE SYMBIONTS TO HANDLE THESE COMMUNICATIONS (AS OPPOSED TO THE COARSE SCHEDULER WHEN THE CONVERSATION IS ON A CONTROL STATEMENT LEVEL). THE DYNAMIC ALLOCATOR CONTROLS THE CPU TIME GIVEN TO A DEMAND TASK WITH THE IDEA THAT THE TASK WILL REACT (OUTPUT) AND GO INTO A VOLUNTARY WAIT-STATE UNTIL SUCH TIME AS AN INPUT SOURCE LINE IS RECEIVED. WHEN THE TASK REACTS, THE DYNAMIC ALLOCATOR NOTIFIES THE OUTPUT SYMBIONT OF THE PRESENCE OF AN OUTPUT MESSAGE AND HOLDS THE TASK FOR FURTHER EXECUTION UNTIL SUCH TIME AS KEYED BY THE INPUT SYMBIONT THAT AN INPUT MESSAGE IS PRESENT. AT THIS TIME THE TASK IS PUT BACK INTO THE ACTIVE CYCLE AND GIVEN ITS SHARE OF CPU TIME. THE TIME-SHARED TASK MAY OR MAY NOT HAVE BEEN SWAPPED OUT PRIOR TO REACTING AND/OR BETWEEN INTERACTION WITH THE USER. THE POINT BEING THAT THE COARSE SCHEDULER IS BY-PASSED WHEN THE CONVERSATION IS NOT ON THE EXECUTIVE CONTROL STATEMENT LEVEL. WHEN THE TASK EVENTUALLY TERMINATES, THE COARSE SCHEDULER IS PUT BACK INTO PLAY FOR POST-PROCESSING AND IN ANTICIPATION THAT THE NEXT CONTROL STATEMENT WILL BE FORTHCOMING FROM THE REMOTE USER. THE NEXT CONTROL STATEMENT COULD CALL FOR THE EXECUTION OF ANOTHER TASK OR FOR RUN CLOSEOUT.

IT IS IMPORTANT TO NOTE THAT THE DEMAND USER HAS THE ENTIRE SYSTEM AT HIS DISPOSAL AND HAS THE SAME CAPABILITIES AS DOES THE BATCH USER; BUT MORE IMPORTANT, THE USE OF THE SYSTEM IS AVAILABLE 'ON DEMAND'.

7.3.4.3. SUMMARY

IN SUMMARY, THE TASKS (PROGRAMS) OF DEMAND RUNS GET ENTERED INTO THE CORE-SWAP QUEUE, AS OPPOSED TO THE CORE QUEUE FOR BATCH RUNS. BOTH QUEUES, THE

```
CORE QUEUE      - FOR BATCH TASKS, AND THE
CORE-SWAP QUEUE - FOR DEMAND TASKS,
```

ARE BASIC QUEUES USED BY THE DYNAMIC ALLOCATOR IN CONTROLLING CORE SPACE AND CPU TIME ALLOCATION.

FOR EACH RUN INITIATED (OPENED), THE COARSE SCHEDULER PREPARES A PROGRAM CONTROL TABLE (PCT) WHICH IS MAINTAINED DURING THE PROCESSING OF THE RUN. THIS TABLE CONTAINS CERTAIN FIXED INFORMATION, SUCH AS RUN-ID, ESTIMATED RUNNING TIME, ETC., AS WELL AS VARIABLE INFORMATION SUCH AS THE CURRENT FACILITIES ASSIGNED, THE CORE REQUIREMENTS OF THE PARTICULAR TASK BEING EXECUTED, ETC. THE CONTROL TABLE IS MAINTAINED BY THE DYNAMIC ALLOCATOR DURING THE EXECUTION OF A TASK, AND RETURNED TO THE COARSE SCHEDULER WHEN THE TASK TERMINATES.

7.3.5. THE DYNAMIC ALLOCATOR

7.3.5.1. GENERAL

THE FUNCTION OF THE DYNAMIC ALLOCATOR (DA) IS THE 'DYNAMIC' ALLOCATION OF CPU TIME AND CORE SPACE TO THE CURRENT MIX OF TASKS (PROGRAMS), WHERE 'THE CURRENT MIX' IS DETERMINED BY THE PARTICULAR GROUP OF RUNS PRESENTLY BEING PROCESSED. IN THE COURSE OF ALLOCATING CPU TIME FOR COMPUTATION, BASED ON PRIORITY LETTERS, DEADLINE TIMES, ETC., THE DYNAMIC ALLOCATOR NECESSARILY CONTROLS THE ALLOCATION OF CORE SPACE FOR EXECUTION, AS WELL AS THE MOVEMENT OF PROGRAMS TO AND FROM MASS STORAGE AS DEMAND PROGRAMS ARE SWAPPED OR BATCH PROGRAMS ARE SUSPENDED AND SUBSEQUENTLY REINITIATED. ASSUMING THAT MORE THAN ENOUGH REQUESTS EXIST FOR THE FACILITIES OF THE COMPUTER (FACILITIES BEING MAINLY SPACE AND TIME), IT IS THE JOB OF THE DYNAMIC ALLOCATOR TO MAKE AN EQUITABLE ALLOCATION OF THESE FACILITIES IN ORDER TO BEST SERVE THE VARIED INTERESTS OF ALL USERS. THE ALLOCATION IS BASED ON THE TYPE OF TASKS (REAL-TIME, DEMAND, AND BATCH), AS WELL AS ON THE PRIORITIES AND RESPONSE TIMES WITHIN A PARTICULAR TYPE. THE BASIC IDEA UNDER WHICH THE DYNAMIC ALLOCATOR OPERATES IS THAT THE PRIMARY CONCERN OF ANY COMPUTING INSTALLATION IS THE COMPLETION OF BATCH RUNS AT THE REQUIRED DEADLINE (WITHIN THE LIMITATIONS OF THE OPERATING ENVIRONMENT) WHILE AT THE SAME TIME ATTEMPTING TO MAINTAIN THE REQUIRED RESPONSE TIMES FOR DEMAND USERS. WITHIN THIS DYNAMIC OPERATING ENVIRONMENT, THE DIVIDING LINE BETWEEN DEMAND AND BATCH PROGRAMS IS SUBJECT TO CONSTANT CHANGE AS EMPHASIS IS PLACED UPON ALLOCATING TIME TO BATCH RUNS APPROACHING THE REQUIRED COMPLETION TIME.

THE DYNAMIC ALLOCATOR PREPARES THE SWITCH LIST USED BY THE DISPATCHER IN GIVING CONTROL TO PROGRAMS RESIDENT IN CORE AND SWITCHING AMONG THEM AS VARIOUS EVENTS AND CONTINGENCIES ARISE. THE DA (DYNAMIC ALLOCATOR) PERIODICALLY ADJUSTS THE SWITCHING LEVEL OF PROGRAMS OR CLASSES OF PROGRAMS SO AS TO FORCE THE CPU TIME TO BE USED IN A PARTICULAR MANNER BASED ON DEADLINES, PRIORITIES, AND INTERACTION RATES AS WELL AS CERTAIN OVERALL CONSTRAINTS AS TO HOW CPU TIME SHOULD BE SHARED AMONG THE DIFFERENT TYPES OF PROGRAMS (DEMAND VERSUS BATCH, FOR EXAMPLE).

THE DA IS STIMULATED BY CHANGES IN THE SYSTEM ENVIRONMENT, AND BY A TIMED INTERVAL THAT ALLOWS A PERIODIC APPRAISAL OF THE SYSTEM STATUS. EXAMPLES OF ENVIRONMENTAL CHANGES WHICH CAUSE THE DA TO BE KEYED ARE ACTIONS SUCH AS LISTED BELOW:

1. THE COARSE SCHEDULER MADE A NEW TASK AVAILABLE FOR EXECUTION.
2. AN EXECUTING PROGRAM IS REQUESTING ADDITIONAL CORE SPACE OR DESIRES TO RELEASE CORE SPACE.
3. THE SYSTEM HAS RECEIVED THE NEXT COMMAND FOR A DEMAND PROGRAM AND THE PROGRAM IS TO BE CONSIDERED FOR EXECUTION.

ON A TIMED INTERVAL (OR WHEN KEYED AS ABOVE), THE DA MAY DISCOVER THAT CERTAIN ADJUSTMENTS MUST BE MADE IN THE WAY CPU TIME (AND/OR CORE SPACE) IS BEING USED. SOME OF THE CONDITIONS WHICH THE DA DETECTS ARE AS FOLLOWS:

1. AN ADJUSTMENT MUST BE MADE SO THAT A PARTICULAR BATCH PROGRAM WILL BE COMPLETED AT THE REQUIRED DEADLINE.
2. THE TOTAL CPU TIME WILL NOT BE SHARED PROPERLY AMONG THE DIFFERENT TYPES OF PROGRAMS (DEADLINE BATCH VERSUS DEMAND, FOR EXAMPLE) UNLESS AN ADJUSTMENT IS MADE.
3. THE RESPONSE TIMES FOR DEMAND USERS WILL DEGENERATE UNLESS AN ADJUSTMENT IS MADE (BECAUSE OF A RECENT CHANGE IN THE REQUIREMENTS OF OTHER PROGRAMS).

THE DYNAMIC ALLOCATOR STIMULATES OTHER PORTIONS OF THE EXECUTIVE AS CHANGES IN SYSTEM STATUS OCCUR. FOR EXAMPLE, THE COARSE SCHEDULER IS KEYED WHEN ALL BATCH TASKS IN THE CORE QUEUE HAVE BEEN INITIATED. THIS ALLOWS THE COARSE SCHEDULER THE OPPORTUNITY TO OPEN A NEW RUN.

7.3.5.2. CORE ALLOCATION

WHEN THE TIME-SHARING OF CORE IS APPROPRIATE, THE DA IS RESPONSIBLE FOR INITIATING THE CORE SWAP; I.E., THE WRITING OF ONE PROGRAM TO MASS STORAGE AND REPLACING IT IN CORE WITH ANOTHER, SO THAT REASONABLE RESPONSE TIMES CAN BE ASSURED. THE DA CALLS ON THE CORE CONTENTS CONTROL ROUTINE FOR THE MOVEMENT OF PROGRAMS TO AND FROM MASS STORAGE AS WELL AS FOR THE MAINTENANCE OF THE TABLE REFLECTING THE CURRENT USE OF CORE (THE CORE MAP). IN BRIEF, CORE CONTENTS CONTROL (CCC) CONSISTS OF A SET OF SUBFUNCTIONS FOR CARRYING OUT SPECIFIC OPERATIONS, SUCH AS INITIAL LOADING OF A TASK OR THE MOVEMENT OF A PROGRAM FROM CORE TO MASS STORAGE. THE DA ASSESSES CORE STORAGE AND THE SWITCH LIST AND CONSTRUCTS A LIST OF OPERATIONS TO BE EXECUTED BY CCC. THE OPERATIONS PERFORMED BY CCC FOR THE DA ARE AS FOLLOWS.

1. INITIAL LOAD PROGRAM
2. RELOAD PROGRAM
3. SWAP-OUT PROGRAM
4. RELOCATE PROGRAM
5. EXPAND PROGRAM

CORE STORAGE IS ALLOCATED IN A MANNER SIMILAR TO THE ALLOCATION OF CPU TIME BY THE DISPATCHER. THIS METHOD IS USED TO HELP INSURE THE PROPER USE OF CPU TIME (SEE SHARING BETWEEN PROGRAM TYPES AND CPU ALLOCATION). EXECUTIVE, ESI, AND REAL-TIME REQUESTS ARE HONORED BEFORE ALL OTHERS; AND HONORED IN ORDER OF SWITCH LIST LEVELS. DEADLINE REQUESTS ARE THEN HONORED IN ORDER OF DEADLINE WEIGHTS. DEMAND AND BATCH REQUESTS ARE THEN ATTEMPTED IN ORDER OF SWITCH LIST LEVELS.

7.3.5.3. PROGRAM TYPES

AMONG THE FOUR TYPES OF PROGRAMS, THE PRIORITY ON GAINING PROCESSOR TIME IS AS SHOWN BELOW.

REAL-TIME TASKS
CRITICAL-DEADLINE TASKS
DEMAND TASKS
BATCH TASKS

THE ABOVE ORDER DOES NOT NECESSARILY REPRESENT THE ORDER IN WHICH THE TYPES OF PROGRAMS APPEAR ON THE SWITCH LIST. IN PRACTICE, THE TYPES OVERLAP EACH OTHER DEPENDING ON SUCH FACTORS AS:

1. AMOUNT OF CPU TIME TO BE GIVEN TO A PARTICULAR CLASS.
2. PRIORITY OF A PROGRAM (OR ACTIVITY) WITHIN A CLASS (TYPE).
3. THE CPU TIME NEEDED BY A PARTICULAR PROGRAM (OR ACTIVITY) PRIOR TO ITS RELEASING CONTROL FOR I/O ACTION.

THE SWITCH LIST IS ORDERED SUCH THAT THOSE PROGRAMS THAT ARE HIGHLY REACTIVE(NEED ONLY SHORT BURSTS BETWEEN I/O REFERENCES) TEND TO HAVE A HIGH PRIORITY REGARDLESS OF THEIR TYPE, HOWEVER THE DA IS FREE TO REORDER THE LIST SUCH THAT THE AMOUNT OF CPU TIME GIVEN TO EACH CLASS OF PROGRAMS IS AS SPECIFIED BY THE USER INSTALLATION(DISCUSSED BELOW). A SUCCEEDING SECTION CALLED 'CPU ALLOCATION' DESCRIBES THE PLACEMENT OF PROGRAMS AT SWITCHING LEVELS AND POSSIBLE MOVEMENT TO NEW LEVELS. THE FUNCTIONS OF THE EXECUTIVE ITSELF ALSO APPEAR ON THE SWITCH LIST AND THESE 'EXECUTIVE FUNCTIONS' ARE INTERSPERSED WITH THE FOUR TYPES OF PROGRAMS. THE SWITCHING PRIORITY OF EXECUTIVE FUNCTIONS RANGE FROM OPERATIONS WHICH ARE AT A HIGHER PRIORITY THAN REAL-TIME PROGRAMS(MONITORING I/O CHANNELS IN ESI MODE, I/O INTERRUPT QUEUING, ETC.), TO THOSE OPERATIONS WHICH HAVE A PRIORITY JUST ABOVE BATCH PROGRAMS(SCHEDULING OPERATIONS, SYMBIONT OPERATIONS, ETC.). CONTROL OVER THE SWITCH LIST BY THE EXECUTIVE(DISPATCHER) IS EXERTED AS A FUNCTION OF TIME(REAL-TIME CLOCK INTERRUPTS) AND AS A FUNCTION OF PERIPHERAL DEVICE ACTIVITY(I/O INTERRUPTS).

7.3.5.4. SHARING BETWEEN PROGRAM TYPES

AS MENTIONED EARLIER THE DYNAMIC ALLOCATOR INSURES THAT CERTAIN OVERALL RULES ARE ENFORCED AS TO HOW CPU TIME IS SHARED AMONG THE FOUR TYPES OF PROGRAMS. REAL-TIME PROGRAMS ARE GIVEN CPU TIME AS NEEDED WITH DEADLINE BATCH, DEMAND, AND BATCH SHARING THE REMAINDER. ONCE THE DEADLINE TIME OF A BATCH RUN BECOMES CRITICAL, THE TASKS OF THE RUN ARE ASSURED A PERCENTAGE OF THE REMAINING CPU TIME IN ORDER TO SATISFY THE COMPLETION TIME.

NORMALLY, DEMAND PROGRAMS ARE ALLOWED TO OPERATE IN THE TIME REMAINING AFTER SATISFYING THE REAL-TIME AND EXECUTIVE REQUIREMENTS AND THE TIME NECESSARY TO INSURE COMPLETION OF CRITICAL DEADLINE RUNS. THIS COULD MEAN THE COMPLETE SUPPRESSION OF DEMAND RUNS BY DEADLINE BATCH RUNS UNLESS SOME MINIMUM IS GUARANTEED. THE EXECUTIVE PROVIDES FOR THE 'SHARING PERCENTAGES' SHOWN BELOW TO BE SPECIFIED AT SYSTEM GENERATION TIME (SEE SYSTEM GENERATION STATEMENTS - SHARING PERCENTAGES, AND INITIAL SYSTEM) AND VIA OPERATOR DIRECTION.

DMIN - DEMAND MINIMUM PERCENTAGE
DMAX - DEMAND MAXIMUM PERCENTAGE

THESE VALUES ARE USED BY THE DYNAMIC ALLOCATOR ACCORDING TO THE FOLLOWING RULES:

1. REAL-TIME PROGRAMS - GIVEN CPU TIME AS NEEDED.

2. CRITICAL DEADLINE - GIVEN THAT PART OF THE REMAINING TIME NOT GUARANTEED TO DEMAND BY DMIN (SEE BELOW).
3. DEMAND(DMIN) - DEMAND PROGRAMS ARE GUARANTEED A MINIMUM PERCENTAGE(DMIN) OF THE TIME REMAINING AFTER REAL-TIME. THIS INSURES THAT DEADLINE OR NORMAL BATCH RUNS WILL NOT COMPLETELY SUPPRESS DEMAND RUNS.
4. DEMAND(DMAX) - DEMAND PROGRAMS ARE GIVEN A MAXIMUM PERCENTAGE(DMAX) OF THE TIME REMAINING AFTER REAL-TIME, WHICH WILL BE EXCEEDED ONLY IN THE EVENT OF OTHERWISE IDLE TIME. THIS IS ACTUALLY THE DIVIDING LINE BETWEEN BATCH AND DEMAND RUNS. IF SOME MINIMUM AMOUNT OF CPU TIME IS GIVEN TO EACH CLASS OF PROGRAMS IN THE SWITCH LIST, THIS GUARANTEES THAT AT LEAST THE HIGHLY REACTIVE PROGRAMS (ACTIVITIES) ARE GIVEN A CHANCE TO FUNCTION; KEEPING I/O GEAR IN USE, AND IN THE CASE OF DEMAND RUNS, KEEPING THE HIGHLY CONVERSATIONAL RUNS IN PROGRESS. THE SPECIFICATIONS DMIN AND DMAX SHOULD BE SET TO BEST MEET THE NEEDS OF THE PARTICULAR INSTALLATION. AN EXAMPLE FOR A PARTICULAR INSTALLATION MIGHT BE AS FOLLOWS:

REAL-TIME PROGRAMS WHEN PRESENT USE APPROXIMATELY 20 PERCENT OF THE TOTAL CPU TIME. DMIN IS SET TO 15 PERCENT OF THE TIME REMAINING, WITH DMAX SET TO 60 PERCENT. WITH THESE SPECIFICATIONS WE SEE THAT IF A CRITICAL-DEADLINE TASK THAT IS COMPUTE LIMITED WERE TO ENTER THE SYSTEM, IT WOULD GAIN A MAXIMUM OF 85 PERCENT OF THE CPU TIME LEFT AFTER REAL-TIME OPERATIONS - ASSUMING THAT THERE WERE ENOUGH DEMAND PROGRAMS TO USE 15 PERCENT. ON THE OTHER HAND, IN THE ABSENCE OF CRITICAL-DEADLINE RUNS, A HEAVY LOAD OF DEMAND RUNS CAN GAIN NO MORE THAN 60 PERCENT OF THE CPU TIME AFTER REAL-TIME, ASSUMING THERE ARE ENOUGH BATCH PROGRAMS TO USE 40 PERCENT.

7.3.5.5. PROGRAM STATES

THE VARIOUS PROGRAMS CURRENTLY BEING EXECUTED FALL INTO SEVERAL STATES DEPENDING ON THEIR ABILITY TO ACCEPT CONTROL, OR ON SOME ACTION TAKEN BY THE DYNAMIC ALLOCATOR IN SUPERVISING THE USE OF CPU TIME. THESE STATES ARE AS FOLLOWS:

1) TERMINATED - PROCESSING BY THE PROGRAM IS COMPLETED, THE ONLY FURTHER ACTION TO BE UNDERTAKEN IS HOUSEKEEPING BY THE EXECUTIVE AND RECORDING SPECIFIED POST MORTEM DUMPS FOR FUTURE EDITING AND LISTING.

2) SUSPENDED - PROCESSING BY THE PROGRAM IS CURRENTLY INTERRUPTED TO ACCOMMODATE A HIGHER PRIORITY JOB. THIS MAY RESULT FROM EXPANSION OF A REAL-TIME PROGRAM, EXCESSIVELY LONG WAIT USER SPECIFICATION, PRECEDENCE GIVEN TO A MORE URGENT CATEGORY, ETC. USER SPECIFIED SUSPENSION MUST BE RESUMED BY THE USER; OTHER SUSPENDED PROGRAMS ARE RESUMED AUTOMATICALLY BY THE EXECUTIVE. TASKS IN THIS STATE MAY STILL BE RESIDENT IN CORE, HOWEVER THEY ARE NORMALLY THOUGHT OF AS BATCH TASKS THAT HAVE BEEN SUSPENDED FROM CORE OR DEMAND TASKS THAT HAVE BEEN SWAPPED OUT INVOLUNTARILY.

3) WAITING - PROCESSING BY THE PROGRAM IS CURRENTLY INTERRUPTED PENDING COMPLETION OF AN EXTERNAL EVENT. THIS DIFFERS FROM THE SUSPENDED STATE IN THAT THE INTERRUPTION IS EXPECTED TO BE OF A RELATIVELY SHORT DURATION. THE WAIT STATE MAY BE IMPOSED BY UNSATISFIED I/O REQUESTS, UNSATISFIED INTERNAL REQUESTS FOR ACQUISITION OF FACILITIES OR OTHER REQUIREMENTS WHICH CANNOT BE IMMEDIATELY RESOLVED.

4) INPUT-WAIT - PROCESSING BY THE PROGRAM IS CURRENTLY INTERRUPTED PENDING ANALYSIS AND RESULTING ACTION TO BE TAKEN WHEN INPUT IS INTRODUCED ON THE DEMAND INPUT DEVICE. THE PROGRAM IS NOT A CANDIDATE FOR CPU TIME UNTIL THE INPUT COMMAND IS RECEIVED. TASKS IN THIS STATE MAY OCCUPY CORE WHILE WAITING FOR THE INPUT COMMAND, HOWEVER THEY ARE NORMALLY THOUGHT OF AS HAVING BEEN SWAPPED OUT IN ORDER TO TIME-SHARE CORE WITH OTHER USERS.

5) ACTIVE - PROCESSING BY THE PROGRAM IS CURRENTLY IN PROGRESS OR CAN BE PLACED IN PROGRESS UPON EXAMINATION OF THE QUEUE BY THE SWITCHING FUNCTION.

THE EXECUTIVE MAINTAINS A LIST OF OPERATIONAL PROGRAMS (THE FIVE STATES DESCRIBED

ABOVE) AND SWITCHES CONTROL OF THE COMPUTER BETWEEN THE CURRENTLY ACTIVE PROGRAMS TO ACHIEVE THE BEST FACILITY UTILIZATION WITHIN THE REALM OF NECESSARY REAL-TIME RESPONSE AND DESIRED INTERACTION RATES OF THE MIX OF PROGRAMS.

7.3.5.6. TABLES AND QUEUES

IN THE COURSE OF ALLOCATING TIME AND SPACE, THE DYNAMIC ALLOCATOR MAINTAINS NOT ONLY THE SWITCH LIST BUT ALSO A SET OF TABLES AND QUEUES REFLECTING THE STATE OF ALL AVAILABLE TASKS, INCLUDING THOSE NOT YET LOADED AND THOSE TEMPORARILY SUSPENDED ON MASS STORAGE. THE MAIN TABLES AND QUEUES, AND THEIR USES, ARE AS FOLLOWS:

1. SWITCH LIST - USED BY THE DISPATCHER TO SWITCH CONTROL AMONG THE VARIOUS TASKS. MAINTAINED BY BOTH THE DA AND THE DISPATCHER.

2. CORE QUEUE - USED BY THE DA IN THE COURSE OF SELECTING NEW OR SUSPENDED BATCH TASKS. MAINTAINED BY BOTH THE DA AND COARSE SCHEDULER. (DISCUSSED IN COARSE SCHEDULER SECTION).

3. CORE-SWAP QUEUE - USED BY THE DA IN THE COURSE OF SELECTING NEW OR SWAPPED-OUT DEMAND TASKS.

4. PROGRAM CONTROL TABLE(PCT) - SUPPLIED BY THE COARSE SCHEDULER AND USED BY THE DA TO MAINTAIN THE CURRENT STATUS OF A TASK AS IT PROCEEDS THROUGH EXECUTION.

5. CORE MAP - USED AND MAINTAINED BY THE DA(VIA THE CORE CONTENTS CONTROL ROUTINE) IN ALLOCATING CORE SPACE. IT REFLECTS THE CURRENT USE OF CORE.

6. TIME MAP - CONTAINS ALL HISTORICAL INFORMATION AS TO HOW THE CPU TIME IS BEING USED. VALUES ARE UPDATED BY BOTH THE DISPATCHER AND THE DYNAMIC ALLOCATOR. USED MAINLY BY THE DA IN DECIDING HOW CPU TIME SHOULD BE ALLOCATED IN THE FUTURE.

7. MASS STORAGE MAP - REFLECTS THE LOCATION OF ALL PROGRAMS CURRENTLY RESIDING ON MASS STORAGE AS WELL AS THE AREAS AVAILABLE FOR PLACING ADDITIONAL TASKS. USED BY THE COURSE SCHEDULER, DYNAMIC ALLOCATOR, AND THE SWAP ROUTINE.

THE DYNAMIC ALLOCATOR SUPERVISES THE USE OF CPU TIME BASED ON HISTORICAL INFORMATION FOUND IN THE TIME MAP. THE INFORMATION AVAILABLE IN THE TIME MAP IS AS FOLLOWS:

1. THE AMOUNT OF CENTRAL PROCESSOR TIME BEING USED BY EACH CLASS OF PROGRAMS(REAL-TIME, CRITICAL-DEADLINE, DEMAND, AND BATCH). AVAILABLE FOR THE LAST 6 SECOND PERIOD AS WELL AS AN AVERAGE OVER SEVERAL PERIODS.
2. THE CPU TIME USED FOR EXECUTIVE FUNCTIONS, OTHER THAN THE SWAPPING OF DEMAND TASKS.
3. IDLE TIME.
4. TOTAL CPU TIME USED THUS FAR BY EACH RUN.

IN ADDITION TO THE ABOVE INFORMATION, THE TIME MAP CONTAINS THE FOLLOWING FIXED INFORMATION:

1. ESTIMATED RUNNING TIME(CPU TIME) FOR EACH BATCH TYPE RUN.
2. DEADLINE TIME, IF SUPPLIED WITH BATCH RUN.

THE TIME MAP, ALONG WITH ALL OTHER INFORMATION MAINTAINED, ALLOWS THE DA TO MAKE JUDGEMENTS AS TO WHETHER ADJUSTMENTS SHOULD BE MADE IN THE SWITCHING PRIORITIES, IN THE BURSTS OF CPU TIME ALLOWED AT A PARTICULAR PRIORITY, AND IN THE USE OF CORE

STORAGE. FOR EXAMPLE, THE DA MONITORS THE PROGRESS OF A DEADLINE RUN AND, AFTER MAKING PROJECTIONS BASED ON CURRENT INFORMATION, DECIDES WHETHER THE RUN WILL BE COMPLETED ON TIME UNDER THE PRESENT CONDITIONS. IF NOT, THE DA MAKES WHAT APPEARS TO BE THE NECESSARY ADJUSTMENTS IN THE USE OF TIME AND SPACE, THE RESULTS OF WHICH WILL BE EVALUATED AT A LATER TIME.

OTHER FUNCTIONS PERFORMED BY THE DYNAMIC ALLOCATOR, BUT NOT SPECIFICALLY MENTIONED BEFORE ARE:

1. MAINTAINS COMPLETE SUPERVISION OVER THE DEMAND ENVIRONMENT. THIS INCLUDES COMMUNICATING WITH THE INPUT/OUTPUT SYMBIONTS, DRIVING THE SWAPPING MECHANISM, AND MAINTAINING THE STATUS OF RE-ENTRANT PROCESSORS.

2. ACCEPTS UNSOLICITED KEYINS AS TO HOW THE COMPUTER SHOULD BE SHARED AMONG THE DIFFERENT TYPES OF PROGRAMS AS WELL AS CHANGES IN THE PRIORITY AND/OR DEADLINE OF PARTICULAR RUNS.

3. RESPONDS TO EXTERNAL INQUIRIES CONCERNING THE PREDICTED OR CURRENT STATUS OF CURRENT RUNS.

7.3.5.7. CPU ALLOCATION

AS INDICATED EARLIER, THE DYNAMIC ALLOCATOR IS RESPONSIBLE FOR BUILDING AND MAINTAINING THE SWITCH LIST USED BY THE DISPATCHER IN GIVING CONTROL TO PROGRAMS(ACTIVITIES) THAT ARE RESIDENT IN CORE AND ABLE TO ACCEPT CONTROL. THE JOB OF THE DYNAMIC ALLOCATOR INCLUDES THE PERIODIC RESTRUCTURING OF THE LIST SUCH THAT DURING THE NEXT PERIOD, THE DISPATCHER WILL DISTRIBUTE CPU TIME IN THE MOST SUITABLE MANNER. DURING A GIVEN PERIOD, THE DISPATCHER SWITCHES CONTROL ACCORDING TO THE 'CURRENT' STRUCTURE OF THE LIST, ALWAYS GIVING CONTROL TO THE HIGHEST PRIORITY PROGRAM READY TO RUN. THE SWITCH LIST IS CONSTRUCTED SUCH THAT IN THE COURSE OF MAKING ADJUSTMENTS, VERY LITTLE WORK MUST BE DONE BY THE DYNAMIC ALLOCATOR. SWITCH LIST ENTRIES THEMSELVES ARE NEVER MOVED, BUT RATHER THE 'POINTERS' TO THE ENTRIES ARE SIMPLY REORDERED. IN THE NORMAL CASE, ONLY MINOR ADJUSTMENTS ARE MADE TO THE SWITCH LIST AT ANY ONE TIME.

IN DISCUSSING THE ALGORITHMS OF THE DYNAMIC ALLOCATOR, IT IS ESSENTIAL THAT THE NATURE OF THE SWITCH LIST AND THE MECHANICS OF SWITCHING BE UNDERSTOOD. THE READER IS REFERRED TO SECTION 7.4, PRIOR TO CONTINUING WITH THIS DISCUSSION. IN BRIEF, THE SWITCH LIST IS A N LEVEL LIST WHICH ALLOWS ANY NUMBER OF ENTRIES AT ANY LEVEL. LEVEL 1 IS THE HIGHEST PRIORITY LEVEL WITH LEVEL N THE LOWEST, WHERE N IS A REASONABLE VALUE. THE DISPATCHER SWITCHES ACCORDING TO THE FOLLOWING RULES:

1. PROGRAMS(ACTIVITIES) AT LEVEL L HAVE PRIORITY OVER TASKS AT LEVEL L+1.

2. PROGRAMS WITHIN A LEVEL ARE TREATED EQUALLY AS FAR AS THE PRIORITY ON GAINING CPU TIME IS CONCERNED. THEY MAY HOWEVER BE GRANTED DIFFERENT AMOUNTS OF CPU TIME ONCE THEY GAIN CONTROL(SEE BELOW).

3. A PROGRAM LOSES CONTROL EITHER BY VOLUNTARY RELEASE OR BY HAVING CONTROL TAKEN BECAUSE A TIME LIMIT WAS EXCEEDED.

THE TIME LIMIT(QUANTUM) FOR A PARTICULAR TASK IS DETERMINED AS FOLLOWS:

1. EACH LEVEL L OF THE SWITCH LIST HAS ASSIGNED A UNIQUE TIME-FACTOR CALLED T(MORE PROPERLY T SUB L). T IS LARGER AS THE LEVEL NUMBER INCREASES. CURRENTLY, THE VALUE OF T FOR LEVEL L IS 2 TO THE POWER L. LEVEL 1 THEN HAS A T VALUE OF 2, LEVEL 2 A VALUE OF 4, ETC.

2. EACH PROGRAM(ACTIVITY) ON THE SWITCH LIST HAS AN ASSOCIATED ALLOCATION-FACTOR(A) AS ASSIGNED BY THE DYNAMIC ALLOCATOR. THROUGH SYSTEM GENERATION PARAMETERS THIS VALUE MAY BE

ALTERED. A PRIORITY FACTOR (F) MAY ALSO BE SUPPLIED WHOSE EFFECT IS TO ALLOCATE VARYING AMOUNTS OF TIME BASED ON PRIORITY.

3. THE TIME LIMIT , OR QUANTUM(Q), FOR A TASK IS THE PRODUCT OF THE LEVEL TIME-FACTOR(T), ADJUSTED PRIORITY, AND THE TASK ALLOCATION FACTOR(A), OR Q=A*(1+P/F)*T.

IT IS IMPORTANT TO NOTE THAT AS A PROGRAM IS MOVED TO A DIFFERENT LEVEL, ITS QUANTUM IS AUTOMATICALLY CHANGED, ASSUMING 'A' IS NOT ADJUSTED.

IF A TASK DOES NOT VOLUNTARILY RELEASE CONTROL PRIOR TO EXCEEDING THE QUANTUM ALLOWED AT THE CURRENT LEVEL, IT IS MOVED TO THE NEXT HIGHER NUMBERED LEVEL(PRIORITY DECREASED BUT QUANTUM INCREASED). (A 'VOLUNTARY RELEASE OF CONTROL' IS DEFINED AS FOLLOWS: FOR BATCH PROGRAMS, THE PROGRAM HAS RELEASED CONTROL WHILE WAITING FOR COMPLETION OF AN I/O REQUEST - OF ANY TYPE. FOR DEMAND PROGRAMS, THE PROGRAM HAS RELEASED CONTROL WHILE WAITING FOR INPUT FROM THE REMOTE CONSOLE ONLY.) ALL TASKS HAVE AN 'INITIAL' LEVEL AT WHICH THEY ARE REGISTERED PRIOR TO ANY ACTION, AND THIS IS THE LEVEL TO WHICH THEY ARE RETURNED WHEN REACTIVATED ON SATISFACTION OF THE WAIT CONDITION.

THE MOVE FUNCTION IS DONE AUTOMATICALLY BY THE DISPATCHER, IN THAT THE DISPATCHER TAKES CONTROL FROM A TASK WHEN IT EXCEEDS THE QUANTUM FOR THE CURRENT LEVEL. A DEMAND PROGRAM MAY UNDERGO SWAPPING PRIOR TO BEING GIVEN CONTROL AT THE NEW LEVEL.

THE INITIAL LEVEL FOR A BATCH OR DEMAND PROGRAM IS AT LEVEL TWO.

THE SWITCH LIST IS CONSTRUCTED SUCH THAT THE FOLLOWING FUNCTIONS ARE AVAILABLE TO THE DA.

ENTER - ENTER A TASK AT A PARTICULAR INITIAL LEVEL.
SET - SET OR RESET THE ALLOCATION-FACTOR(A) FOR A PARTICULAR PROGRAM.
MOVE - MOVE A TASK TO A DIFFERENT LEVEL.
MOVE1 - MOVE, FROM A GIVEN LEVEL, ALL TASKS OF A PARTICULAR TYPE(SUCH AS DEMAND) TO A DIFFERENT INITIAL LEVEL.
MOVE2 - MOVE ALL TASKS OF A PARTICULAR TYPE DOWN OR UP A GIVEN NUMBER OF LEVELS(TO NEW INITIAL LEVELS). THIS SERVES TO CHANGE THE RELATIONSHIP BETWEEN OTHER TYPES AND THE TYPE BEING MOVED.

IN STUDYING THE CPU ALLOCATION ALGORITHM, IT IS IMPORTANT TO NOTE THAT THE QUANTUM ALLOWED A PROGRAM AT A PARTICULAR LEVEL IS NOT NECESSARILY THE SAME AS IT IS FOR OTHER PROGRAMS AT THAT LEVEL.

7.3.6. CORE CONTENTS CONTROL(CCC)

THE NATURE OF THE 1108 HARDWARE MAKES FEASIBLE THE DYNAMIC RELOCATION OF PROGRAMS RESIDING IN CORE STORAGE IN ORDER TO PROVIDE A MORE EFFECTIVE MULTI-PROGRAMMING ENVIRONMENT. THIS MAKE POSSIBLE THE RE-ARRANGEMENT OF THE CONTENTS OF CORE STORAGE WHENEVER NECESSARY TO PROVIDE CONTIGUOUS BLOCKS OF UNUSED CORE. WITHIN THE 1108 EXECUTIVE SYSTEM, THE CORE CONTENTS CONTROL ROUTINE IS RESPONSIBLE FOR THE NECESSARY MOVEMENT OF PROGRAMS WITHIN CORE STORAGE, AS WELL AS FOR MAINTAINING INFORMATION DESCRIBING THE MAKE-UP OF CORE STORAGE-I.E., WHAT PROGRAMS ARE IN CORE, WHERE EACH OF THESE IS LOCATED, AND WHAT AREAS OF CORE STORAGE ARE AVAILABLE FOR USE. THIS ROUTINE IS ALSO RESPONSIBLE FOR INITIATING THE LOADING OF ANY REQUIRED PROGRAMS, AND FOR CARRYING OUT ANY 'CORE-SWAPS' CALLED FOR BY THE DYNAMIC ALLOCATOR.

CORE STORAGE IS INITIALLY ASSIGNED TO EACH TASK ACCORDING TO ITS IMMEDIATE REQUIREMENTS. AS THESE REQUIREMENTS CHANGE, THE USER MAY REQUEST ADDITIONAL CORE STORAGE SPACE, OR HE MAY RELEASE SPACE TO AID IN OPTIMIZING SYSTEM PERFORMANCE.

THE ASSIGNMENT OF CORE STORAGE SPACE FOR THE TRANSIENT PORTIONS OF THE RESIDENT IS HANDLED DIFFERENTLY WHENEVER SUFFICIENT CORE STORAGE FOR LOADING SUCH A ROUTINE IS NOT AVAILABLE. IT OCCASIONALLY BECOMES NECESSARY TO SWAP SOME USER PROGRAM (OR A LOWER-PRIORITY TRANSIENT) OUT TO TEMPORARY DRUM STORAGE, IN ORDER TO LOAD A TRANSIENT ROUTINE; SUCH ACTION IS CARRIED OUT BY THE CORE CONTENTS ROUTINES WHENEVER NECESSARY. ANY TIME A SYSTEM PROCESSOR OR A USER PROGRAM COMPLETES ITS CURRENT OPERATION, THE CORE STORAGE USED BY THE PROCESSOR OR PROGRAM IS RETURNED TO THE POOL OF AVAILABLE CORE STORAGE. THE COMPACTING OF CORE STORAGE IS NOT CARRIED OUT AT THIS TIME; INSTEAD, IT IS PERFORMED LATER, WHENEVER ANOTHER PROCESSOR, USER PROGRAM, OR SYSTEM TRANSIENT IS TO BE INITIATED, OR WHEN A PROGRAM REQUESTS MORE CORE STORAGE. HOWEVER, COMPACTING IS NEVER PERFORMED UNNECESSARILY, AS THE DA ALWAYS ATTEMPTS TO FIT PROGRAMS INTO GAPS IN THE IN-USE CORE STORAGE.

WHENEVER A SYSTEM TRANSIENT ROUTINE COMPLETES ITS CURRENT OPERATION, THE CORE STORAGE AREA IT OCCUPIES IS NOT ACTUALLY RELEASED, BUT IS PLACED IN A 'RELEASE IF NECESSARY' CONDITION. SUCH A ROUTINE IS THEREFORE STILL AVAILABLE FOR USE, IF NECESSARY, UNTIL THE CORE STORAGE SPACE IT OCCUPIES IS REQUIRED FOR SOME OTHER OPERATION. IF THE TRANSIENT ROUTINE IS REQUESTED AGAIN BEFORE SUCH AN EVENT, ITS CORE STORAGE IS RETURNED TO THE IN-USE CONDITION.. EACH SUCH TRANSIENT ROUTINE HAS ASSOCIATED WITH IT A 'STICKING PRIORITY', SO THAT THE MORE FREQUENTLY A TRANSIENT IS USED, THE LONGER IT TENDS TO RETAIN ITS CORE STORAGE SPACE AFTER EACH PERIOD OF OPERATION. THIS PROCEDURE PREVENTS UNNECESSARY LOADING OF TRANSIENT ROUTINES, SINCE THEY REMAIN IN CORE STORAGE AS LONG AS IT IS POSSIBLE TO DO SO WITHOUT INTERFERING WITH THE OVERALL OPERATION OF THE SYSTEM.

REAL TIME PROGRAMS ALSO RECEIVE SPECIAL CONSIDERATIONS, INSOFAR AS THE CORE CONTENTS CONTROL ROUTINE IS CONCERNED. A PROGRAM WITH REAL-TIME REQUIREMENTS IS NEVER MOVED ABOUT IN CORE. AN ATTEMPT IS MADE TO POSITION SUCH PROGRAMS OPTIMALLY WITHIN CORE STORAGE. WHEN THEY REQUEST REAL-TIME STATUS THROUGH THE APPROPRIATE ER FUNCTION, THEY REMAIN STATIONARY THROUGHOUT THEIR OPERATION UNTIL REAL-TIME STATUS IS RELEASED. A REAL-TIME PROGRAM NEED NOT BE CONCERNED WITH THE TIME TAKEN BY THE CORE CONTENTS CONTROL ROUTINE IN MOVING OTHER PROGRAMS, SINCE THIS PROCEDURE IS INTERRUPTABLE, AND CONTROL CAN BE RETURNED TO THE CURRENTLY-OPERATING REAL-TIME PROGRAM WHENEVER NECESSARY.

CORE SPACE ASSIGNED IS NECESSARILY SUBJECT TO THE RESTRICTION THAT ITS D-BANK AREA CAN NEVER BE ASSIGNED NEARER THE BEGINNING OF PHYSICAL CORE THAN THE LENGTH OF ITS I-BANK AREA, THUS AVOIDING NEGATIVE RELOCATION ASSIGNMENTS OF STORAGE.

7.4. DISPATCHER

THE DISPATCHER SWITCHES WITHIN THE PROGRAMS RESIDENT IN CORE ACCORDING TO THE STRUCTURE OF THE SWITCH LIST PROVIDED BY THE DYNAMIC ALLOCATOR. SWITCHING IS ALWAYS ACCOMPLISHED BY EXECUTING THE HIGHEST PRIORITY PROGRAM CURRENTLY ABLE TO RUN. THE SWITCHING LEVEL OF EACH PROGRAM IS DETERMINED BY DECISIONS MADE BY THE DYNAMIC ALLOCATOR BASED ON PRIORITY, COMPUTER USAGE RATIO AMONG THE DIFFERENT TYPES OF PROGRAMS (DEMAND, BATCH, ETC.) AND INTERACTION RATE. SINCE THE FACTORS WHICH DETERMINE THE APPROPRIATE SWITCHING LEVEL MAY CHANGE, THE SWITCHING LEVEL OF A PARTICULAR PROGRAM MAY BE CHANGED BY THE DYNAMIC ALLOCATOR THROUGHOUT THE EXECUTION. ACTIVITIES WITH REAL-TIME REQUIREMENTS, OF COURSE, WILL TEND TO OCCUPY THE HIGHEST LEVELS. ALL OTHER PROGRAMS(DEMAND AND BATCH) ARE NORMALLY INTERMIXED THROUGHOUT THE REMAINING LEVELS ALTHOUGH A LOW PRIORITY BATCH PROGRAM MAY REQUIRE A HIGH RESPONSE FOR A SPECIFIC ACTIVITY. THE ONLY RESTRICTION ON THE PRIORITY AT WHICH AN ACTIVITY MAY BE REGISTERED IS THE PRIORITY LIMIT ASSIGNED TO THE ACCOUNT NUMBER OF THE MAIN PROGRAM. EXEC ACTIVITIES ARE INTERSPERSED THROUGHOUT ALL SWITCHING LEVELS.

PROGRAMS ON ALL LEVELS WILL LOSE CONTROL IN FAVOR OF ANOTHER PROGRAM WITHIN THE SAME LEVEL UPON THE EXPIRATION OF A TIME LIMIT SET BY THE DYNAMIC ALLOCATOR OR BY VOLUNTARILY RELEASING CONTROL AWAITING COMPLETION OF I/O OPERATIONS. ONCE A PROGRAM ON A GIVEN SWITCHING LEVEL HAS LOST CONTROL, IT WILL NOT REGAIN CONTROL UNTIL AN ENTIRE CYCLE HAS BEEN ACCOMPLISHED ON THIS LEVEL. IN THE MEANTIME ACTIVITIES ON A

HIGHER PRIORITY LEVEL MAY REQUIRE CONTROL AS THE RESULT OF I/O COMPLETIONS.

7.4.1. SWITCHING

7.4.1.1. REAL TIME SWITCHING

ESI ACTIVITIES WHICH ARE ACTIVATED AS THE RESULT OF ESI INTERRUPTS RECEIVE THE HIGHEST SWITCHING PRIORITY. INTERRUPT QUEUING IS THE ONLY EXECUTIVE FUNCTION THAT TAKES PRECEDENCE. REAL-TIME WORKER PROGRAMS RECEIVE SWITCHING PRIORITIES DIRECTLY BELOW THE EXEC FUNCTIONS FOR INITIATING I/O REQUESTS AND PROCESSING INTERRUPTS. SINCE MORE THAN ONE PROGRAM WITH REAL-TIME REQUIREMENTS OF IDENTICAL PRIORITIES MAY BE PRESENT SIMULTANEOUSLY ON THE SWITCHING CYCLE, IT BECOMES THE DUTY OF THESE PROGRAMS TO SHARE CONTROL AS REQUIRED.

ONLY WHEN ALL ACTIVITIES OF THE HIGHER PRIORITY LEVELS HAVE RELINQUISHED CONTROL DOES THE DISPATCHER SWITCH TO THE NEXT LEVEL. IF A PROGRAM OF A LOWER PRIORITY LEVEL IS INTERRUPTED DUE TO AN I/O COMPLETION WHICH CAUSES THE REACTIVATION OF A HIGHER PRIORITY PROGRAM, CONTROL WILL BE SWITCHED TO THE HIGHER PRIORITY PROGRAM.

FOR THE BENEFIT OF REAL/TIME PROGRAMS THE EXEC IS DESIGNED TO QUEUE ALL INTERRUPTS WHICH IT MUST HANDLE. A SIMPLE PRIORITY SCHEME IS THEN AVAILABLE TO EXTRACT FIRST FOR PROCESSING ANY WHICH ARE RELATED TO THE REAL TIME TASK.

7.4.2. SWITCH LISTS

INFORMATION WHICH MUST BE MAINTAINED BY THE DISPATCHER FOR EACH PROGRAM ENTERED ON THE SWITCHING CYCLE INCLUDES THE FOLLOWING:

1) ENTRY POINT - THE ADDRESS AT WHICH THE PROGRAM WILL NEXT RECEIVE CONTROL.
2) RUN IDENTITY - THE FIELDDATA IDENTIFICATION OF THE RUN ASSOCIATED WITH THIS PROGRAM.
3) PROCESSOR STATE REGISTER IMAGE - SPECIFIES THE WORD WHICH MUST BE LOADED INTO THE PROCESSOR STATE REGISTER BEFORE CONTROL IS GIVEN TO THE PROGRAM. INFORMATION CONTAINED IN THIS WORD INCLUDES SPECIAL DESIGNATORS FOR: CARRY AND OVERFLOW, GUARD MODE, MEMORY PROTECTION, EXEC MODE AND VALUES USED IN BASE INDEXING.
4) ACTIVITY MASK - SPECIFIES WAIT CONDITIONS FOR THIS ACTIVITY. ENFORCES A HOLD ON THIS ACTIVITY UNTIL ANY NUMBER OF ACTIVITIES OF THIS SET HAVE BEEN DEACTIVATED.
5) MEMORY LOCKOUT - SPECIFIES THE QUANTITY WHICH MUST BE LOADED INTO THE STORAGE LIMIT REGISTER BEFORE THE PROGRAM REGAINS CONTROL TO INSURE THAT ALL AREAS, EXCEPT THE PROGRAM CURRENTLY OPERATING, ARE LOCKED OUT.
6) RUNNING TIME - AN ACCUMULATION OF THE PROGRAM'S TOTAL COMPUTE TIME IN MILLISECONDS. (INCLUDES INTERRUPT PROCESSING).
7) ESTIMATED RUN TIME - THE ESTIMATED PROGRAM RUN TIME IN MINUTES.
8) CR INDICATOR - INDICATES TO THE DISPATCHER THE AMOUNT OF CONTROL REGISTERS WHICH MUST BE RESTORED BEFORE THE PROGRAM REGAINS CONTROL.
9) WAIT INDICATORS - THESE INDICATORS ARE SET BY VARIOUS SECTIONS OF THE EXEC TO TEMPORARILY RETIRE A PROGRAM FROM THE SWITCHING CYCLE.
10) PROGRAM CONTROL TABLE ADDRESS - ADDRESS OF THE TABLE USED BY THE EXEC FOR MAINTAINING SWITCH LIST INFORMATION AS WELL AS THE SAVE AREA FOR CONTROL REGISTERS.

7.4.3. COMMON DATA PROTECTION

UNDER CONTROL OF THE 1108 EXECUTIVE SYSTEM INDEPENDENT ACTIVITIES CAN BE ESTABLISHED (USING THE FORK$ FUNCTION) WITHIN A PROGRAM AND THEREFORE EACH ACTIVITY

HAS ACCESS TO THE SAME DATA. ON EARLIER COMPUTERS, UNLESS INTERRUPTS WERE DISABLED, CONFLICTS COULD OCCUR WHEN REFERENCING COMMON LOCATIONS. ON THE UNIVAC 1108 A SPECIAL HARDWARE INSTRUCTION, TEST AND SET (73 - 17), HAS BEEN PROVIDED TO PROTECT AGAINST SUCH CONFLICTS. THIS INSTRUCTION(TS) SHOULD BE USED IN PLACE OF THE 'PREVENT ALL INTERRUPTS AND JUMP' INSTRUCTION. IT IS ILLEGAL TO 'EXECUTE REMOTE' THIS INSTRUCTION. THE INSTRUCTION FUNCTIONS AS FOLLOWS:

1) IF BIT 30 OF THE OPERAND IS EQUAL TO ONE, THE PROGRAM IS INTERRUPTED, CONTROL IS GIVEN TO ANOTHER ACTIVITY AND THE A LOWER LEVEL.
2) IF BIT 30 OF THE OPERAND IS EQUAL TO ZERO, THE NEXT INSTRUCTION WILL BE EXECUTED.
3) WHEN EXECUTING THE TS INSTRUCTION, REGARDLESS OF THE STATUS OF BIT 30, BIT 30 WILL BE SET TO ONE, BITS 31 THRU 35 ARE CLEARED AND BITS 0 THRU 29 REMAIN UNDISTURBED.

WHEN THE PROTECTED SEQUENCE HAS BEEN COMPLETED A SIMPLE 'STORE ZERO' INSTRUCTION WILL CLEAR THE 'TEST AND SET' CONDITION. THE INSTRUCTION SEQUENCE FOR PROTECTING COMMON DATA IS AS FOLLOWS:

```
TS      IND
- - - - - - - - -
PROTECTED AREA
- - - - - - - - -
SZ      IND,0,S1
```

WHERE,

IND IS ANY CORE MEMORY LOCATION.

7.5. INTERRUPT PROCESSING

THE INTERRUPT HANDLING ROUTINES OF THE 1108 EXEC CONTROL ALL INTERRUPTS. THESE INTERRUPTS ARE RECEIVED FROM A PERIPHERAL SUBSYSTEM OR FROM THE CONTROL SECTION OF THE CENTRAL PROCESSOR.

7.5.1. INPUT/OUTPUT INTERRUPTS

THE EXEC ENCODES THE TYPE AND CHANNEL NUMBER OF THE INTERRUPT AND ROUTES IT TO THE APPROPRIATE PROCESSING ROUTINE. THIS CATEGORY OF INTERRUPTS INCLUDES EXTERNAL REQUESTS, INPUT DATA TERMINATION AND OUTPUT DATA TERMINATION FOR BOTH ESI AND NON-ESI CHANNELS, FUNCTION TERMINATION FOR NON-ESI CHANNELS, CLOCK AND EXTERNAL SYNCHRONIZATION. INTERRUPTS ARE QUEUED IF AN INTERRUPT IS BEING PROCESSED OR IF HIGHER PRIORITY INTERRUPTS PRESENTLY EXIST IN THE QUEUE. WITH THE EXCEPTION OF REAL TIME REQUIREMENTS, WHEN IT IS NECESSARY TO QUEUE AN INTERRUPT, IT IS REMOVED FROM THE QUEUE BY CHANNEL PRIORITY (CHANNEL 0 BEING TOP PRIORITY).

7.5.2. COMPUTER/CORE MALFUNCTIONS

7.5.2.1. CONTROL MEMORY PARITY ERROR

IF AN I/O CONTROL WORD FAILURE OCCURS, THE CURRENT STATUS OF I/O ACTION ON ALL PERIPHERALS ON THE CHANNEL IS LOGGED. IF IT IS DETERMINED THAT THE ERROR IS TRANSIENT (FAILURE DOES NOT REOCCUR AFTER THE EXECUTIVE SYSTEM EXERCISES THE LOCATION), THE FUNCTION IS REINITIATED AND CONTROL IS RETURNED TO THE INTERRUPTED ADDRESS. IF THE ERROR IS NOT TRANSIENT AND EQUIPMENT ON ANOTHER CHANNEL CANNOT ASSUME THE RESPONSIBILITY OF THIS CHANNEL, THE PROGRAMS ASSIGNED TO THIS CHANNEL WILL BE TERMINATED AND THE CHANNEL WILL BE DECLARED DOWN.

2) IF A TRANSIENT ERROR EXISTS IN AN A, B OR R REGISTER AND THE RECOVERY ROUTINE DOES NOT USE THE FAULTY REGISTER, AUTOMATIC RECOVERY IS ATTEMPTED IF A RESTART POINT IS

PROVIDED. IF THE ERROR IS NOT TRANSIENT INVOLVING AN A, B OR R REGISTER, THE SYSTEM IS TERMINATED.

7.5.2.2. INPUT/OUTPUT DATA PARITY

IF A TRANSIENT ERROR OCCURS, THE PROGRAM WITH THE DATA ERROR IS GIVEN CONTROL AT A RESTART POINT IF ONE IS PROVIDED. IF A RESTART POINT HAS NOT BEEN PROVIDED, THE PROGRAM IS TERMINATED. THE EXECUTIVE SYSTEM RESUMES NORMALLY. IF THE ERROR OCCURRED WITHIN EXEC AND RESTART IS IMPOSSIBLE THE SYSTEM WILL STOP.

2) IF THE ERROR IS NOT TRANSIENT, THE PROGRAM INTERRUPTED IS TERMINATED AND THE BLOCK OF MEMORY INVOLVED IS DECLARED DOWN. IF THE DAMAGED CODING WAS CRITICAL (AN UNRECOVERABLE PART OF THE EXEC), THE SYSTEM WILL STOP.

7.5.2.3. CORE STORAGE PARITY ERROR

SAME ACTION AS INPUT/OUTPUT DATA PARITY.

7.5.2.4. POWER FAILURE

WHEN A POWER FAILURE OCCURS, THE EXEC INITIATES THE FOLLOWING SHUT-DOWN SEQUENCE:

1) UPON OCCURENCE OF THE INTERRUPT, THE INTERRUPT ADDRESS AND CONTROL MEMORY ARE SAVED. CHANNELS CONTAINING 'I/O ACTION' ARE FLAGGED.
2) IF THE COMPUTER IS RESTARTED WITHOUT CLEARING, THE I/O ACTION RESTART FLAGS ARE CLEARED, CONTROL MEMORY IS RESTORED AND RETURN IS MADE TO THE INTERRUPTED ADDRESS.
3) IF THE COMPUTER IS RESTARTED AFTER CLEARING, THE NECESSARY FLAGGED I/O ACTIONS ARE REQUEUED ON THE I/O REQUEST LIST. CONTROL MEMORY IS RESTORED AND CONTROL IS GIVEN TO THE PROGRAM INTERRUPTED.

7.5.2.5. MESSAGE FORMAT

IF, BECAUSE OF A COMPUTER/CORE MALFUNCTION, THE EXECUTIVE SYSTEM MUST TERMINATE A WORKER PROGRAM OR THE EXECUTIVE SYSTEM MUST BE REINITIALIZED, AN APPROPRIATE MESSAGE WILL BE DISPLAYED ON THE MONITOR PRINTER. IN THE REMOTE CASE, A MESSAGE MAY NOT APPEAR IF CONSOLE CHANNEL PROBLEMS, FAILURE OF EXEC A, B, OR R REGISTERS USED IN THE INTERRUPT ANALYSIS ROUTINE OR CORE PARITY ERRORS IN THE INTERRUPT ANALYSIS ROUTINE OCCUR. A RECORD OF THE TYPE AND NUMBER OF CORE AND DATA PARITY ERRORS IS MAINTAINED IN THE I/O ERROR LOG.

THE GENERAL MESSAGE FORMAT IS :

```
   (MALFUNCTION) - (ERROR ADDRESS) - (STATUS) - (ACTION)
 WHERE,

   MALFUNCTION IS : CONTROL MEMORY FAILURE, I/O DATA PARITY
                    ERROR, CORE STORAGE PARITY ERROR OR POWER
                    FAILURE.

   STATUS IS :      RUN IDENTITY TERMINATED(ITEM MAY BE ABSENT)
                    EXEC DOWN              (ITEM MAY BE ABSENT)
                    CHANNELS DOWN          (ITEM MAY BE ABSENT)
 AND

   ACTION IS :      INITIALIZE SYSTEM OR MAINTENANCE REQUIRED.
```

THIS MESSAGE IS DISPLAYED ONLY IF A WORKER PROGRAM HAS BEEN TERMINATED BECAUSE OF

HARDWARE PROBLEMS AND/OR THE EXECUTIVE SYSTEM REQUIRES OPERATOR ACTION.

7.5.3. PROGRAM CONTINGENCY

A PROGRAM CONTINGENCY IS A CONDITION WITHIN A RUNNING PROGRAM WHICH CAUSES A COMPUTER INTERRUPT OR PSEUDO-INTERRUPT. THE USER PROGRAM IS ABLE TO SPECIFY THAT IT WISHES TO PROCESS THE INTERRUPT RATHER THAN ACCEPT THE STANDARD ACTION PROVIDED BY THE SUPERVISOR, BY INITIATING A CORRESPONDING EXECUTIVE REQUEST FUNCTION BEFORE THE CONDITION OCCURS. THE 1108 EXECUTIVE SYSTEM RECOGNIZES THE FOLLOWING PROGRAM CONTINGENCIES:

CONTINGENCY	CONTINGENCY TYPE	MNEMONIC
ILLEGAL OPERATION	1	IOPR
GUARD MODE	2	IGDM
FLOATING POINT OVERFLOW	3	IFOF
FLOATING POINT UNDERFLOW	4	IFUF
DIVIDE OVERFLOW	5	IDOF
RESTART	6	IRST
ABORT	7	IABT
CONSOLE INTERRUPT	10	IINT
TEST AND SET INTERRUPT (R/T ONLY)	11	ITS
'ERR MODE' ENTRY	12	IERR$

ERROR NAME	ERROR TYPE	MNEMONIC
I/O CALL ERROR	1	I/O
SYMBIONT CALL ERROR	2	SYMB
ERR$ CALL	3	ERR$
ILLEGAL OR BAD ER	4	ER
CONSOLE CALL ERROR	5	CONS
COMMUNICATIONS ERRORS	6	COM2
COMMUNICATIONS ERRORS	7	COMM

UNDER THE 'CONSOLE INTERRUPT' ENTRY (CONTINGENCY TYPE 010) FOLLOWING CONDITIONS CAN OCCUR

ERROR (CONDITION) NAME	ERROR TYPE	MNEMONIC
II ON-SITE KEYIN	1	II
REMOTE BREAK KEY	2	RBK

EXCEPT FOR THE CASE WHERE THE USER MAKES A REFERENCE TO ERR$,THE EXEC AUTOMATICALLY CHANNELS CONTROL TO THE 'ERR MODE' CONTINGENCY ON DETECTING A CONDITION UNDER WHICH THE EXECUTIVE REQUEST CANNOT BE SATISFIED. FOR EACH OF THESE CASES (CONSOLE, I/O, SYMB, ETC.) AN 'ERROR CODE' MAY ALSO BE GENERATED. THESE ERROR CODES ARE DESCRIBED IN THE SECTION APPROPRIATE TO THE ERROR TYPE (E.G., I/O ERROR CODES IN THE I/O CHAPTER, ETC.) IN ORDER TO SPECIFY THAT THE USER DESIRES TO RECEIVE CONTROL ON THE OCCURENCE OF CONTINGENCIES, THE FOLLOWING ER IS USED:

```
L    A0,ADDR             T1         S3          H2
                     ------------------------------
ER   IALL$           5  MASK   :  APPL.   :CONT.ADR.:
                     ------------------------------
```

WHERE THE WORD AT ADDR HAS THE FOLLOWING FORMAT AND MEANING:

T1 SELECTION MASK

THIS MASK INDICATES WHICH OF THE 10 CONTINGENCY TYPES ARE TO BE PASSED TO THE PROGRAM. CONTINGENCY TYPE 1 (IOPR) IS BIT 24, TYPE 2 (IGDM) IS BIT 25, ETC. WITH TYPE 012 (IERR$) BEING BIT 33. IF T1 IS ZERO, NO CONTINGENCIES ARE PASSED TO THE PROGRAM. TEST AND SET CONTINGENCY SELECTION IS ALLOWED ONLY FOR REAL-TIME ACTIVITIES, NEVER FOR THE ENTIRE PROGRAM OR BATCH ACTIVITIES.

S3 CONTINGENCY APPLICATION (APPL.)

THIS FIELD SPECIFIES WHETHER THE CONTINGENCY SETTING IS TO APPLY TO THE ENTIRE PROGRAM, TO JUST THE REFERENCING ACTIVITY, OR TO ESI ACTIVITIES (SEE SUBSEQUENT

DISCUSSION ON ESI CONTINGENCIES). THE INDICATORS ARE :

0	ENTIRE PROGRAM
1	THIS ACTIVITY
2	ESI ACTIVITIES

IF A CONTINGENCY OCCURS AND NO ACTIVITY CONTINGENCY ROUTINE IS SPECIFIED (OR SELECTED FOR THIS CONTINGENCY), THE PROGRAM CONTINGENCY ROUTINE WILL BE REFERENCED BY DEFAULT. IF NO PROGRAM CONTINGENCY ROUTINE EXISTS (OR SELECTED FOR THIS CONTINGENCY), STANDARD ACTION WILL RESULT (STANDARD ACTION IS DISCUSSED LATER).

H2 CONTINGENCY ROUTINE ADDRESS

THIS FIELD CONTAINS THE ADDRESS OF THE FIRST WORD OF THE CONTINGENCY ROUTINE. IF THIS FIELD IS ZERO, THE CONTINGENCY ROUTINE SETTING FOR THE CASE SPECIFIED IN S3 (PROGRAM,ACTIVITY OR ESI) IS CANCELLED. A SECOND REQUEST WITH AN ADDRESS OTHER THAN ZERO WILL CAUSE A RESETTING OF THE LOCATION AND CONDITION OF CONTINGENCY PROCESSING. A PROGRAM CONTINGENCY ROUTINE IS MERELY A DIVERSION FROM THE ACTIVITY WHICH CAUSES IT TO OCCUR.

USING THE EXEC REFERENCE DESCRIBED ABOVE, THE USER MAY HAVE AS MANY CONTINGENCY POINTS (ROUTINES) AS THERE ARE ACTIVITIES, PLUS ONE FOR THE PROGRAM, PLUS ONE FOR ESI ACTIVITIES. THE CONTINGENCY ROUTINE HAS THE FOLLOWING CHARACTERISTICS:

1. THE FIRST TWO WORDS ARE RESERVED FOR THE EXEC TO INSERT PARAMETERS ON THE OCCURRENCE OF AN INTERRUPT.

2. CONTROL IS ALWAYS PASSED TO THE THIRD WORD OF THE ROUTINE.

3. SINCE A CONTINGENCY ROUTINE EXECUTION IS A DIVERSION FROM AN ACTIVITY, IT IS THE RESPONSIBILITY OF THE CONTINGENCY ROUTINE TO SAVE REGISTERS BEFORE USING THEM.

4. THE CONTINGENCY ROUTINE IS OBLIGATED TO NOTIFY THE EXEC WHEN THE CONTINGENCY STATE EXPIRES. THIS NORMALLY OCCURS AT THE TIME THE PARAMETER AREA (FIRST TWO WORDS) CAN BE DESTROYED BY THE EXEC, BUT DEPENDS ON THE LOGIC OF THE PROGRAM. THE REFERENCES TO NOTIFY THE EXEC THAT THE CONTINGENCY STATE HAS EXPIRED ARE:

   ```
   1)  ER     CEND$    OR    2)  ANY OTHER ER INSTRUCTION
   ```

   CONTROL IS RETURNED FOLLOWING THE REFERENCE TO CEND$.

5. IN MAKING THE CEND$ OR 'OTHER' EXEC REFERENCE, THE CONTINGENCY ROUTINE (ACTIVITY) IS RETURNED TO NORMAL STATUS. IT IS ASSUMED THAT THE REFERENCE IS MADE PRIOR TO SIGNIFICANT PROCESSING.

THE EXECUTIVE HANDLES CONTINGENCY SITUATIONS IN SERIAL, REGARDLESS OF THE CONTINGENCY POINT OF THE ACTIVITY. IT IS THEREFORE MANDITORY THAT THE 'RELEASE' REFERENCE OCCUR, AND PREFERABLY IN A TIMELY FASHION. IT IS POSSIBLE FOR A CONTINGENCY ROUTINE TO GENERATE CONTINGENCY CONDITIONS EITHER BY DESIGN OR BY AN ERROR IN CODING. WHENEVER AN ACTIVITY (CONTINGENCY ROUTINE) IS PROCESSING AN INTERRUPT OF ANY SORT, ALL CONTINGENCY SETTINGS FOR THAT ACTIVITY ARE RETURNED TO THE STANDARD ERR$ TERMINATION FOR THE DURATION OF THE CONTINGENCY STATE. IF SUCH SHOULD OCCUR, THE CEND$ REFERENCE IS AUTOMATICALLY ACCOMPLISHED BY THE EXEC 8 IN CASES WHERE THE ACTIVITY TERMINATES.

IN THE CASE OF CONSOLE INTERRUPTS AND THE RESTART INTERRUPT, THE ENTIRE PROGRAM CONTINGENCY MUST BE SET BECAUSE THE EXEC IS FREE TO DIVERT ANY ACTIVITY TO THE CONTINGENCY POINT. IF NO CONTINGENCY POINT IS DEFINED THEN CONTROL RETURNS TO THE LATEST RE-ENTRY POINT FOR EACH ACTIVITY.

THE EXEC STORES THE FOLLOWING INFORMATION IN THE FIRST TWO WORDS (PARAMETER AREA) OF THE CONTINGENCY ROUTINE UPON EXECUTION OF THE ROUTINE.

	S1	S2	S3	H2
WORD 1	ERROR TYPE	ERROR CODE	CONT. TYPE	ERROR (REENTRY) ADDRESS
WORD 2	TS FOR ESI	NOT USED	NOT USED	PACKET ADDRESS

THE CONTINGENCY TYPES, ERROR TYPES AND ERROR CODES WERE DESCRIBED EARLIER. THE CONTINGENCY TYPE CAN BE USED BY THE CONTINGENCY ROUTINE AS AN INDEX TO DETERMINE THE TYPE OF INTERRUPT, IF NECESSARY. THE SAME HOLDS TRUE FOR THE ERROR TYPE IN THE CASE OF AN ERR MODE CONTINGENCY.

IN ALL CASES EXCEPT GUARD MODE, THE ERROR ADDRESS PLUS ONE IS THE ADDRESS THAT WOULD HAVE RECEIVED CONTROL IF THE ERROR WERE IGNORED. THUS, IF AN ERROR IS TO BE BYPASSED, THE CONTINGENCY ROUTINE SHOULD DO A JUMP TO ADDRESS-PLUS-ONE, AFTER THE REFERENCE TO CEND$.

FOR GUARD MODE ERRORS, THE HARDWARE DOES NOT GUARANTEE THE ADDRESS PROVIDED ON AN INTERRUPT; THUS THE ERROR ADDRESS MAY NOT BE MEANINGFUL FOR GUARD MODE ERRORS. THE PROGRAM MUST TAKE PRECAUTIONS TO AVOID AN INFINITE LOOP. THE PACKET ADDRESS IS MEANINGFUL ONLY FOR I/O AND CONSOLE ERRORS.

STANDARD CONTINGENCY ACTION

FLOATING POINT OVERFLOW AND UNDERFLOW AND DIVIDE OVERFLOW RESULT IN CLEARING OF THE TWO REGISTERS WHICH WERE USED IN THE COMMAND WHICH GENERATED THE INTERRUPT. THIS DOES NOT APPLY IF AN EXECUTE REMOTE WAS USED TO REFERENCE THE INSTRUCTION GENERATING THE INTERRUPT. THE USE OF THE EXECUTE REMOTE WILL RESULT IN TWO ARBITRARY REGISTERS BEING CLEARED (USUALLY A0 AND A1).

ILLEGAL OPERATION,GUARD MODE,AND THE 'ERROR MODE' CONDITIONS RESULT IN AN ERROR MESSAGE AND REGISTER DUMP IN THE PRINT FILE AND TERMINATION OF THE ACTIVITY WHICH CAUSED THE ERROR. THE ERROR MESSAGE HAS THE FORMAT:

```
(CONT OR ERROR MNEMONIC)-(ERROR ADDRESS)-(ERROR CODE)-(PACKET ADDRESS, IF
APPLICABLE)
```

THE ABORT CONDITION RESULTS IN AN ABORT MESSAGE IN THE PRINT FILE AND TERMINATION OF ALL ACTIVITIES IN THE PROGRAM, AND WITH THE EXCEPTION OF DEMAND RUNS, THE RUN IS TERMINATED. THE FORMAT OF THE ABORT MESSAGE IS:

```
ABORT-(ABORT ADDRESS)
```

THIS INTERRUPT MAY COME FROM THE PROGRAM OR THE OPERATOR (X KEYIN). IF A PROGRAM IS NOT IN EXECUTION, THE RUN IS SIMPLY TERMINATED.

THE CONSOLE INTERRUPT IS IGNORED IF NO CONTINGENCY ROUTINE IS PRESENT FOR THE PROGRAM OR A PROGRAM IS NOT CURRENTLY BEING EXECUTED. (THE COARSE SCHEDULER IS IN CONTROL).

THE RESTART CONDITION RESULTS IN A RESTART MESSAGE IN THE PRINT FILE. CONTROL IS THEN GIVEN TO THE REINITIATION ADDRESS. THE FORMAT OF THE MESSAGE IS:

```
RESTART-(REINITIATION ADDRESS)
```

STANDARD ACTION FOR A TEST & SET INTERRUPT IS TO SIMPLY DROP THE PROGRAM ONE LEVEL ON THE SWITCH LIST AND RETURN CONTROL TO THE TS LOCATION WHEN THE ACTIVIY NEXT RECEIVES CONTROL. FOR REAL-TIME ACTIVITIES, LEVEL DEGRADATION DOES NOT OCCUR. THE TS INSTRUCTION SHOULD NOT BE USED IN CONTINGENCY CODING UNLESS THE USER CAN PREDICT THAT ON FAILURE, THE ACTIVITY NOW POSSESSING THE COMMON DATA WILL NOT ALSO CAUSE A

CONTINGENCY, THEREBY CAUSING IT TO BE QUEUED WAITING FOR CONTINGENCY ENTRANCE, IN WHICH CASE A LOCK UP SITUATION OCCURS.

A PROGRAM IS TERMINATED 'IN ERROR' ONLY IF THE LAST EXISTING ACTIVITY TERMINATES VIA STANDARD CONTINGENCY ACTION.

7.5.4. ESI CONTINGENCY

THE 1108 COMMUNICATION HANDLER PROVIDES THE USER THE CAPABILITY OF PROCESSING VARIOUS CONTINGENCY CONDITIONS THAT MIGHT OCCUR WHILE EXECUTING UNDER AN ESI ACTIVITY. TO ESTABLISH AN ESI CONTINGENCY, THE USER REAL-TIME PROGRAM WILL REGISTER THE CONTINGENCY VIA THE EXECUTIVE REQUEST IALL$. THE CALLING SEQUENCE IS AS FOLLOWS:

```
L          A0,ADDR
ER         IALL$
```

WHERE ADDR IS DEFINED AS FOLLOWS:

```
----------------------------------------
:                  :APPL : CONTINGENCY ADDR :
----------------------------------------
```

H1-APPLICATION SET TO 2 FOR ESI CONTINGENCY.
H2-CONTINGENCY ROUTINE BEGINNING ADDRESS.

IT IS NOT NECESSARY FOR THE REAL-TIME PROGRAM TO PROVIDE A SELECTIVE MASK IN THE ABOVE REFERENCE, AS REQUIRED FOR NON-ESI CONTINGENCY, AS ALL CONTINGENCY TYPES WITHIN AN ESI ACTIVITY WILL BE PROCESSED BY THE SPECIFIED ESI CONTINGENCY.

THE ESI CONTINGENCY ROUTINE ADDRESS POINTS TO THE BEGINNING OF A TWO WORD PACKET WHICH WILL CONTAIN PERTINENT INFORMATION CONCERNING THE ESI CONTINGENCY. SAME AS FOR BATCH AND REAL TIME.

ON THE OCCURRENCE OF AN ESI CONTINGENCY, CONTROL WILL BE GIVEN TO THE LOCATION PLUS 2. IF THE REAL-TIME PROGRAM DESIRES AN ESI CONTINGENCY, THE ABOVE EXECUTIVE REQUEST MUST BE EXECUTED PRIOR TO ANY REFERENCE TO THE COMMUNICATION HANDLER.

THERE MAY BE AS MANY ESI ACTIVITIES AS THERE ARE CPU'S IN A SYSTEM. ESI CONTINGENCY ACTIVITIES FOR A PROGRAM EXECUTE SERIALLY. ONCE AN ESI CONTINGENCY ACTIVITY IS GIVEN CONTROL, ANY OTHER CONTINGENCY THAT MAY OCCUR ON ANOTHER CPU WILL BE INHIBITED VIA A LOGICAL TEST AND SET ON S1 OF THE SECOND WORD OF THE CONTINGENCY PACKET. THESE CONDITIONS EXIST UNTIL THE ESI (CURRENT) CONTINGENCY IS TERMINATED BY EXECUTING AN EXECUTIVE REQUEST, BY CLEARING THE TEST AND SET INDICATOR, OR BY ALLOWING A TIME OUT. SHOULD A CONTINGENCY OR TIME-OUT OCCUR WHILE EXECUTING IN THE ESI CONTINGENCY, THE ESI CONTINGENCY ACTIVITY WILL BE TERMINATED AND THE COMMUNICATION LINE ASSOCIATED WITH THE CONTINGENCY WILL BE DEACTIVATED FOR INPUT AND/OR OUTPUT OPERATIONS. TO ADVISE THE REAL-TIME ACTIVITY OF TERMINAL DEACTIVATION, THE FIRST REFERENCE TO THE DEACTIVATE TERMINAL BY THE REAL-TIME PROGRAM WILL CAUSE THE ACTIVITY TO ENTER NON-ESI CONTINGENCY MODE WITH AN ERROR CODE OF 0710.

TO EXIT FROM AN ESI CONTINGENCY ACTIVITY, ONLY THOSE EXECUTIVE REQUEST SPECIFIED FOR NORMAL ESI ACTIVITIES MAY BE USED: EXIT$, ACT$, CADD$, AND ADACT$. ANY REFERENCE OTHER THAN THOSE INDICATED ABOVE WILL CAUSE A CONTINGENCY WITHIN A CONTINGENCY, CAUSING TERMINAL DEACTIVATION

THE FORMAT OF THE ESI CONTINGENCY PACKET AND FIELD DEFINITIONS ARE AS FOLLOWS:

```
       35        29        23        17                    0
       ----------------------------------------------------
  0    :  07     : CODE   :CTG TYPE:     ERROR LOCATION   :
       ----------------------------------------------------
  1    :  TS     :                  : PACKET ADDRESS      :
       ----------------------------------------------------
  2    CONTROL POINT
       WORD-0
```

```
                S1-ERROR TYPE EQUAL 07.

                S2-ERROR CODE
                   060-INDICATES CONTINGENCY TYPES 1-5.
                   061-ESI ACT$ OR ADACT$ REQUEST ERROR.
                   062-ESI CADD$ OR ADACT$ REQUEST ERROR.
                   063-ILLEGAL ER REQUEST.
                   064-ESI TIME-OUT.

                S3-CONTINGENCY TYPE
                     1-ILLEGAL OPERATION
                     2-GUARD MODE
                     3-FLOATING POINT OVERFLOW
                     4-FLOATING POINT UNDERFLOW
                     5-DIVIDE OVERFLOW
                   012-'ERR MODE', CODES 061-063.
                H2-LOCATION OF ERROR OR RE-ENTRY POINT FOR
                   CONTINGENCY TYPE 012

        WORD-1
                S1-TEST AND SET (TS) INDICATOR.
                H2-PACKET ADDRESS FOR A CONTINGENCY TYPE 012
```

7.6. TIMING ROUTINES

7.6.1. GENERAL

WITHIN THE SUPERVISOR, THE TIMING ROUTINES MAKE PROVISIONS FOR THEIR USE BY AN OBJECT PROGRAM AS WELL AS BY THE SYSTEM. THESE ROUTINES ARE AVAILABLE TO THE USER BY MEANS OF EXECUTIVE REQUESTS. THE TIMING ROUTINES SERVE AS THE BASIS FOR ALL ACCOUNTING AND LOGGING FUNCTIONS, AS WELL AS A SOURCE OF CONTROL FOR MANY REAL-TIME APPLICATIONS.

7.6.2. REAL-TIME CLOCK

THE REAL-TIME CLOCK ROUTINE IS USED BY THE SYSTEM FOR TIMING VARIOUS ACTIVITIES SUCH AS INPUT/OUTPUT FUNCTIONS, OPERATOR RESPONSES, CPU USAGE TIME FOR EACH RUN. THIS ROUTINE IS ALSO USED BY THE SYSTEM TO FORCE INTERRUPTS AFTER VARIABLE AMOUNTS OF TIME SO THAT SUCH EVENTS AS NON-RESPONSIVE I/O DEVICES, UNBALANCED USAGE OF CPU TIME, ETC. ARE DETECTED. THE FREQUENCY OF INTERRUPT DEPENDS ON THE NEEDS OF THE SYSTEM. THE ROUTINE IS SO DESIGNED THAT MANY EVENTS MAY BE SIMULTANEOUSLY TIMED, AND MANY INTERRUPTS MAY BE SIMULTANEOUSLY REQUESTED. SINCE ONLY ONE REAL-TIME CLOCK AND ONE CLOCK INTERRUPT ARE AVAILABLE ON THE 1108, THE REAL TIME CLOCK ROUTINE ESSENTIALLY ACTS AS A MULTIPLEXOR AND CREATES AN ENVIRONMENT SUCH THAT ANY ROUTINE MAY OPERATE AS THOUGH IT HAD EXCLUSIVE ACCESS TO THE REAL-TIME CLOCK AND TO THE ASSOCIATED INTERRUPT.

7.6.3. DAY CLOCK

THE DAY CLOCK ROUTINE IS USED BY THE SYSTEM TO MAINTAIN AN ACCURATE, STANDARD TIME. THIS TIME IS USED BY ALL PROCESSORS FOR ANNOTATING LISTINGS, BY THE FILE CONTROL SUPERVISOR FOR MAINTAINING HISTORICAL INFORMATION ABOUT ALL FILES, BY THE ACCOUNTING AND LOGGING ROUTINES FOR TIME-TAGGING EVENTS, AS WELL AS, BY OTHER ROUTINES FOR OTHER FUNCTIONS.

7.7. LOGGING AND ACCOUNTING

AN EXTENSIVE LOGGING SYSTEM IS MAINTAINED FOR COLLECTING INFORMATION PERTAINING TO EACH RUN AND PROGRAM. THE INFORMATION LOGGED IS LATER USED FOR ACCOUNTING AND POST-EXECUTION PURPOSES. A SUMMARY ACCOUNTING LOG IS UPDATED THROUGHOUT ALL RUNS.

```
.................
CONSOLE MESSAGES:
................:
SYMBIONTS       :
................:
I/O ERROR LOG   :
................:              .............
COARSE SCHEDULER:    .........  :           :
................:    :       :  :TEMPORARY  :
ER FUNCTIONS    :..>:LOG     :.>:MASTER     :(FASTRAND)
................:    :CONTROL:  :RUN LOG    :
USER(BY ER CSF$):    :.......:  :           :
................:               :...........:
TERMINATION     :                     .
................:                     .
CONTINGENCY     :               .............
................:               :ACCOUNTING :
FACILITY USAGE  :               :ROUTINE    :
................:               :...........:
                               .      .      .
                              .       .       .
                    ............  .........  .............
                    :RUN       :  :MASTER :  :SUMMARY    :
                    :PRINT     :  :RUN LOG:  :ACCOUNTING:
                    :FILE      :  :.......:  :FILE       :
                    :..........:  (FASTRAND  :...........:
                                   OR TAPE)   (FASTRAND)
```

7.7.1. LOG CONTROL

THIS FUNCTION OF THE 1108 EXEC CONTROLS THE FLOW OF ALL LOGGING INFORMATION GENERATED BY THE VARIOUS EXEC FUNCTIONS AND BY THE USER PROGRAM. USER REQUESTS ARE GRANTED THROUGH THE CONTROL STREAM @LOG STATEMENT AND THE EXECUTIVE ENTRANCE CSF$. EACH EXECUTING RUN IS ASSIGNED A CORE LOG BUFFER. AS EACH BUFFER IS FILLED, THE LOG CONTROL ROUTINE CHAINS THE NEW BLOCK OF INFORMATION BY RUN IN THE TEMPORARY MASTER RUN LOG.

7.7.2. RUN TERMINATION ACCOUNTING

AT EACH RUN TERMINATION, THE SUMMARY ACCOUNTING ROUTINE IS ACTIVATED. APPLICABLE INFORMATION IN THE PROGRAM CONTROL TABLE AND THIS RUN'S ENTRIES IN THE TEMPORARY RUN LOG ARE COLLECTED.

INFORMATION WHICH PERTAINS TO A RUN IS PLACED AT THE END OF THE PRINT FILE. THE FOLLOWING INFORMATION IS AVAILABLE FOR THE REQUESTOR OF THE RUN:

RUN IDENTITY
CONTROL LANGUAGE LOG STATEMENTS
CONSOLE MESSAGES PERTAINING TO THE RUN
EXECUTIVE REQUEST LOG MESSAGES
PROJECT IDENTITY
ACCOUNT NUMBER
TOTAL RUN TIME
PAGES OF PRINTING APPLICABLE TO THE RUN
NUMBER OF CARDS READ IN AND PUNCHED OUT
TIME AND DATE OF RUN INITIATION

A CONSECUTIVE GROUP OF BLOCKS ARE WRITTEN TO THE MASTER LOG. ALL INFORMATION COLLECTED IS SAVED IN SEQUENCE WITH THE ORDER OF EVENTS DURING THE RUN.

THE ACCOUNTING ROUTINE THEN READS AND UPDATES ALL TOTALS MAINTAINED. THE NUMBER OF RUNS IS INCREMENTED. CURRENT TIME AND DATE ARE RECORDED. THIS SUMMARY GIVES FOR EACH ACCOUNT A TOTAL FACILITY USAGE UPON WHICH A BILLING MAY BE MADE.

7.7.3. BILLING

THE ACCOUNTING FILE IS PERMANENTLY ASSIGNED AS A FASTRAND FILE. CONTINUING TOTALS ARE KEPT UNTIL CLEARED BY THE BILLING ROUTINE OR THE FILE IS REPLACED DURING SYSTEM LOADING. A PROCEDURE FOR EXECUTING THE BILLING ROUTINE MUST BE ESTABLISHED BY THE USER.

THE BILLING ROUTINE SUPPLIED IS INTENDED FOR USE AS A BASE FOR AN INSTALLATION'S OWN UNIQUE BILLING PROGRAM. THE FORMAT FOR OUTPUT IS VERY SIMPLE IN THE ROUTINE SUPPLIED. THE PRIMARY PURPOSE IS TO PERFORM THE EXECUTIVE RELATED FUNCTIONS NECESSARY FOR PROPER ACCOUNTING FILE MAINTENANCE. THE ROUTINE PERFORMS THE FOLLOWING FUNCTIONS:

READ EACH ENTRY IN THE ACCOUNT FILE.

WHILE THE FILE IS LOCKED OUT FROM SUMMARY ACCOUNTING, RESET EACH ITEM TO THE CLEARED STATE.

PRINT EACH TOTAL FOR THE VARIOUS FACILITIES USED BY THE ACCOUNT.

FLAG EACH ENTRY WHICH HAD BEEN ADDED THROUGH OPERATOR ACTION. GIVE THE DATE ON WHICH THE ITEM WAS ADDED.

TOTAL EACH ENTRY TYPE FOR ALL ACCOUNTS.

PRINT A SUMMARY FOR THE ENTIRE SYSTEM USAGE SINCE THE PREVIOUS BILLING RUN. THIS INCLUDES TOTAL RUNS PROCESSED, TIME CHARGED, AND PERIPHERAL USAGE.

EXECUTIVE TIME AND FACILITY USAGE IS LOGGED UNDER THE ACCOUNT MAINTAINED FOR EXEC. THE TOTALS APPEAR JUST AS IF THE EXEC WAS A USER PROGRAM.

7.7.4. MASTER RUN LOG

THE MASTER RUN LOG MAINTAINS ALL INFORMATION IN 224 WORD BLOCKS. EACH BLOCK CONTAINS EIGHT 28-WORD LOG ENTRIES. THE STANDARD LOG MEDIUM IS FASTRAND;, HOWEVER, THE USER MAY REQUEST TAPE AS THE SPECIFIC LOG MEDIUM AT SYSTEMS GENERATION TIME. THE BASIC TYPES OF INFORMATION ENTERED IN THE RUN LOG ARE FACILITY USAGE, RUN TERMINATION DATA AND LOGGING ENTRIES MADE BY LOG CONTROL CARDS OR THE EXECUTIVE REQUEST CSF$.

THE RUN IDENTITY IS SPECIFIED IN EACH BLOCK. THE FIRST WORD OF EACH NEW ITEM IN THE BLOCK DEFINES THE TYPE OF DATA IN THE ITEM AND THE NUMBER OF WORDS PERTAINING TO THIS ITEM.

7.7.4.1. LOG CONTROL CARD OR EXECUTIVE REQUEST LOG ENTRY

LOGGING ENTRIES SPECIFIED BY THE LOG CONTROL CARD STATEMENT (@LOG) AND THE EXECUTIVE REQUEST(CSF$) WILL BE PLACED IN THE RUN LOG IN THE ORDER IN WHICH THEY OCCUR. THE FORMAT OF THESE ENTRIES IS:

```
      35      30 29      24 23                                  0
     -------------------------------------------------------------
   1 :    A    :    B    :                                       :
     -------------------------------------------------------------
   2 :                            C                              :
     -------------------------------------------------------------
  28 :                                                           :
     :                                                           :
     /                                                           /
     /                                                           /
     /                                                           /
  31 :                                                           :
     -------------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (LOGGING ENTRIES =01)
B = NUMBER OF WORDS IN ENTRY ITEM BEGINNING WITH NEXT WORD.
C = MESSAGE TEXT

7.7.4.2. FACILITY USAGE LOG ENTRY

WHENEVER THE CONFIGURATION OF A RUN IS CHANGED, EXCEPT FOR MASS STORAGE, AN ENTRY IS MADE INTO THE ACCOUNTING LOG. THE FORMAT OF THE USAGE LOG IS :

```
     35        29        23        17                          0
    -------------------------------------------------------------
  1 :    A    :    B    :                                      :
    -------------------------------------------------------------
  2 :       C           :    D    :    E    :                  :
    -------------------------------------------------------------
  3 :          TIME     OF   DAY                               :
    -------------------------------------------------------------
```

WHERE,

A= ENTRY TYPE (FACILITY USAGE = 02)
B= NUMBER OF WORDS FOR THIS ENTRY (IF 1 FACILITY, B=2; IF TWO
   FACILITIES, B=4; ETC.)
C= CHANNEL NUMBER OF FACILITY
D= UNIT NUMBER
E= EQUIPMENT TYPE (01=8C, 02=6C, 05=8C 9 TRACK, 06=6C 9 TRACK
   07=4C, 16=3A, 17=2A,   40=CARD SUBSYSTEM,
   42=PAPER TAPE SUBSYSTEM, 44=PRINTER,
   50=1004, 70=CTS; 71=WTS, 72=CTMC).

THE FIRST ENTRY FOR A PARTICULAR CHANNEL AND UNIT INDICATES AN ASSIGNMENT AND THE SECOND ENTRY A RELEASE, IF NO SECOND ENTRY EXISTS, RELEASE IS AT RUN TERMINATION. THIRD, FIFTH, ETC ENTRIES ARE SUBSEQUENT ASSIGNMENTS OF THE SAME DEVICE.

7.7.4.3. CATALOGUED MASS STORAGE FILE USAGE ENTRY

WHENEVER A CATALOGUED MASS STORAGE FILE IS DROPPED FROM THE CATALOGUING ENVIRONMENT (FREE STATEMENT PROCESSED AFTER ASG WITH AN R OR K OPTION), OR CHANGE IN SPACE ASSIGNED IS MADE, AN ENTRY IS MADE IN THE MASTER RUN LOG IN THE FOLLOWING FORMAT:

```
      35        29        23                                    0
     -------------------------------------------------------------
   1 :    A    :    B    :                                       :
     -------------------------------------------------------------
   2 :                   FILE NAME                               :
     -------------------------------------------------------------
   3 :                   FILE NAME   (CONT.)                     :
     -------------------------------------------------------------
   4 :                  QUALIFIER                                :
     -------------------------------------------------------------
   5 :                  QUALIFIER (CONT.)                        :
     -------------------------------------------------------------
   6 :                  ACCOUNT NUMBER                           :
     -------------------------------------------------------------
   7 :                  ACCOUNT NUMBER (CONT.)                   :
     -------------------------------------------------------------
   8 :                 TIME OF CATALOGUING                       :
     -------------------------------------------------------------
   9 :           TRACK-MINUTES OF FH432 USAGE                    :
     -------------------------------------------------------------
  10 :           TRACK-MINUTES OF FH880 USAGE                    :
     -------------------------------------------------------------
  11 :           TRACK-MINUTES OF F-II USAGE                     :
     -------------------------------------------------------------
  12 :           TRACK-MINUTES OF FASTBAND USAGE                 :
     -------------------------------------------------------------
```

WHERE,

A=ENTRY TYPE (MASS STORAGE DROP=03)
B=NUMBER OF WORDS FOR THIS ENTRY

7.7.4.4. PROGRAM TERMINATION LOG ENTRY

FOR EACH PROGRAM IN THE RUN, TERMINATION INFORMATION IS ENTERED IN THE RUN LOG. THE FORMAT OF THIS ENTRY IS:

```
      35     30 29      24 23     18 17                        0
    -----------------------------------------------------------
  1 :   A    :    B    :                                       :
    -----------------------------------------------------------
  2 :        PROGRAM  IDENTITY                                 :
    -----------------------------------------------------------
  3 :        PROGRAM  IDENTITY  (CONT.)                        :
    -----------------------------------------------------------
  4 :        PROGRAM  IDENTITY  (CONT.)                        :
    -----------------------------------------------------------
  5 :        PROGRAM  IDENTITY  (CONT.)                        :
    -----------------------------------------------------------
  6 :        PROGRAM  INITIATION  TIME (TDATE$ FORMAT)         :
    -----------------------------------------------------------
  7 :        PROGRAM TERMINATION  TIME  (TDATE$ FORMAT)        :
    -----------------------------------------------------------
  8 :        ACTUAL  PROGRAM  RUN  TIME  (MILLISECONDS)        :
    -----------------------------------------------------------
  9 :    FINAL   IBANK  LENGTH    :    FINAL  DBANK  LENGTH    :
    -----------------------------------------------------------
 10 :   MAIN PROGRAM TYPE (R,D,B) :                            :
    -----------------------------------------------------------
 11 :   TERMINATION  TYPE         :  LAST REENTRY ADDRESS      :
    -----------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (PROGRAM TERMINATION ENTRY = 04)
B = NUMBER OF WORDS FOR THIS ENTRY (PROGRAM TERMINATION = 13 OCTAL)

7.7.4.5. RUN TERMINATION LOG ENTRY

AT THE COMPLETION OF EACH RUN, TERMINATION INFORMATION IS ENTERED IN THE RUN LOG. THE FORMAT OF THIS ENTRY IS:

```
     35     30 29     24 23     18 17                             0
    -------------------------------------------------------------
 1 :   A    :    B    :                                         :
    -------------------------------------------------------------
 2 :         PROJECT  IDENTITY                                  :
    -------------------------------------------------------------
 3 :         PROJECT  IDENTITY   (CONTINUED)                    :
    -------------------------------------------------------------
 4 :         ACCOUNT  NUMBER                                    :
    -------------------------------------------------------------
 5 :         ACCOUNT  NUMBER     (CONT.)                        :
    -------------------------------------------------------------
 6 :         INITIATION TIME-DATE                               :
    -------------------------------------------------------------
 7 :         TERMINATION TIME-DATE                              :
    -------------------------------------------------------------
 8 :   CARDS  IN                  :   CARDS  OUT                :
    -------------------------------------------------------------
 9 :        C     :            LINE COUNT                       :
    -------------------------------------------------------------
10 :         TOTAL  RUN  TIME  (MILLISECONDS)                   :
    -------------------------------------------------------------
11 :       TRACK-SECONDS OF FH432 RUN TEMPORARY AREA USED       :
    -------------------------------------------------------------
12 :       TRACK-SECONDS OF FH880 RUN TEMPORARY AREA USED       :
    -------------------------------------------------------------
13 :       TRACK-SECONDS OF F-II FASTRAND RUN TEMPORARY AREA    :
    -------------------------------------------------------------
14 :       TRACK-SECONDS OF FASTBAND RUN TEMPORARY AREA USED    :
    -------------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (RUN TERMINATION ENTRY = 05)
B = NUMBER OF WORDS FOR THIS ENTRY (RUN TERMINATION = 14 OCTAL)
C = PRIORITY OF RUN

THE MASS STORAGE USAGE LOGGED FOR THE RUN (WORDS 10-13) IS THE AREA UNDER THE CURVE OF MASS STORAGE ASSIGNMENT PLOTTED AS A FUNCTION OF TIME. THIS VALUE INCLUDES THE AREA ASSIGNED AS SCRATCH FILES (BOTH DRUM FORMAT AND FASTRAND FORMAT FOR FH432 AND FH880) AND THE VARIATIONS OF FILES TO BE CATALOGUED WHEN RELEASED.

7.7.4.6. I/O ERROR LOG ENTRY

WITHIN THE MASTER RUN LOG A RECORD IS MAINTAINED OF THE OCCURRENCE OF I/O ERRORS. THE COUNTS ARE KEPT IN CORE UNTIL SUCH TIME AS A 28 WORD BLOCK CAN BE DUMPED INTO THE LOGGING MEDIUM. ERRORS ARE COUNTED FOR MASS STORAGE AND MAGNETIC TAPE DEVICES. THE PARTICULAR ERRORS LOGGED ARE SPECIFIED IN CHAPTER 10. THE FORMAT OF THE I/O ERROR LOG IS:

```
      35         29         23         17                           0
    --------------------------------------------------------------
  1 :   A     :    B    :                                        :
    --------------------------------------------------------------
  2 :       C           :    D     :     E    :                  :
    --------------------------------------------------------------
  3 :            F                 :               G             :
    --------------------------------------------------------------
  4 :                             H                              :
    --------------------------------------------------------------
  5 :    I    :    J    :                   K                    :
    --------------------------------------------------------------
  6 :    A    :    B    :                                        :
    --------------------------------------------------------------
```

WHERE,

A= ENTRY TYPE (I/O ERROR = 06)
B= NUMBER OF WORDS FOR THIS ENTRY (N+3 WHERE N IS THE NUMBER OF DIFFERENT ERRORS WHICH HAVE OCCURRED PLUS THE NUMBER FOR WHICH THE COUNT HAS EXCEEDED 64)
C= CHANNEL NUMBER
D= UNIT NUMBER
E= EQUIPMENT TYPE:
   01= VIIIC
   02= VIC
   03= IVC
   20= FH432
   21= FH880
   30= FASTRAND
F= NUMBER OF INPUT FUNCTIONS
G= NUMBER OF OUTPUT FUNCTIONS
H= NUMBER OF INPUT AND OUTPUT WORD TRANSFERS DIVIDED BY 1000
I= THE EXTERNAL INTERRUPT CODE FROM THE STATUS WORD FOR DRUM OR FASTRAND OR A PSEUDO CODE FOR MAGNETIC TAPES (SEE CHAPTER 10 FOR PARTICULAR CODES)
J= COUNT OF OCCURRENCES OF THIS ERROR.
K= ADDRESS OF ERROR FOR MASS STORAGE ERRORS IF APPLICABLE OTHERWISE ZERO.

7.7.4.7. CONSOLE LOG ENTRIES

EACH CONSOLE MESSAGE IS PLACED IN THE MASTER LOG. AT RUN TERMINATION, EVERY MESSAGE PERTAINING TO THE RUN IS PRINTED AT THE END OF THE PROGRAM LISTING. THE FORMAT IF THE LOG ENTRY IS:

```
  35       29       23       17                            0
  ---------------------------------------------------------
1 :   A    :   B    :          :                          :
  ---------------------------------------------------------
2 :      :          :    C   :                            :
  ---------------------------------------------------------
3 :   D                                                   :
  ---------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (CONSOLE ENTRIES = 07)
B = NUMBER OF WORDS IN TEST
C = MESSAGE NUMBER
D = TEXT

7.7.4.8. RUN INITIATION

WHEN A RUN IN OPENED, AN ENTRY IS MADE IN THE LOG WITH PERTINENT INFORMATION CONCERNING THE RUN.

```
   35       29       23       17        11        5        0
  ---------------------------------------------------------
1 :  A      :   B    :                                    :
  ---------------------------------------------------------
2 :  C               : START TIME        : DEADLINE TIME  :
  ---------------------------------------------------------
3 : EST RUN TIME     :  EST P.O.         :  EST C.O.      :
  ---------------------------------------------------------
4 :      RUNID (GENERATED)                                :
  ---------------------------------------------------------
5 :      RUNID (SUBMITTED)                                :
  ---------------------------------------------------------
6 :      PROJECT                                          :
  ---------------------------------------------------------
7 :                PROJECT                                :
  ---------------------------------------------------------
8 :      ACCOUNT                                          :
  ---------------------------------------------------------
9 :                NUMBER                                 :
  ---------------------------------------------------------
10:      SEQUENCE RUNID                                   :
  ---------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (RUN INITIATION = 9)
B = NUMBER OF WORD IN LOG ENTRY
C = RUN LIMITING OPTIONS (S,C,P, OR T)

7.7.4.9. CONSOLE REPLIES

REPLIES TO CONSOLE TYPE AND READ MESSAGES ARE PLACED IN THE MASTER LOG. THE REPLIES AS WELL AS THE TYPE AND READ MESSAGES ARE PRINTED AT THE END OF THE PROGRAM LISTING. THE FORMAT OF THE CONSOLE REPLY IS:

```
   35       29       23       17        11        5       0
   -----------------------------------------------------------
1 :   A    :    B    :                                       :
   -----------------------------------------------------------
2 :  REPLY                                                   :
   -----------------------------------------------------------

                   .
                        .
                             .

   -----------------------------------------------------------
27:                                              : MSG ID :
   -----------------------------------------------------------
```

WHERE,

A = ENTRY TYPE (CONSOLE REPLIES = 012)
B = NUMBER OF WORDS IN TEXT

7.7.5. SUMMARY ACCOUNTING FILE

A SPECIAL ACCOUNTING FILE IS MAINTAINED BY THE EXECUTIVE SYSTEM FOR THE PURPOSE OF PROVIDING LIMITED SUMMARY ACCOUNTING INFORMATION. THE INFORMATION IS ACCUMULATED BY ACCOUNT NUMBER AT THE TIME OF RUN COMPLETION. THE TYPE OF TERMINATIONS WITHIN THE RUN ARE DISREGARDED.

THE SUMMARY WILL ACCUMULATE INFORMATION ON THE FOLLOWING ITEMS:

- RUN TIME APPLICABLE TO THE ACCOUNT NUMBER
- TIME AND DATE OF THE FIRST ENTRY IN THE ACCOUNT NUMBER
- TIME AND DATE OF THE LAST ENTRY IN THE ACCOUNT NUMBER
- NUMBER OF PAGES OF PRINTING
- NUMBER OF CARDS READ IN AND NUMBER PUNCHED OUT
- ELAPSED TIME AN I/O FACILITY WAS ASSIGNED TO THE ACCOUNT. THE I/O FACILITIES APPLY TO THE DRUM SCRATCH AREA AND TO THE EQUIPMENT DESCRIBED IN THE FACILITY USAGE LOG ENTRY.

THE SUMMARY ACCOUNTING FILE IS FIRST CONSTRUCTED DURING SYSTEM GENERATION. AN ENTRY FOR EACH ACCOUNT NUMBER IS CONSTRUCTED WHICH INCLUDES THE SCHEDULING LIMITATIONS FOR THIS ACCOUNT. THE ENTRY IS 54 WORDS LONG AND CURRENTLY HAS MANY UNUSED WORDS ALLOWING EXPANSION FOR NEW PERIPHERAL SUMMARY ITEMS, ETC.

ADDITIONAL ENTRIES ARE MADE IF OPERATOR APPROVAL IS GIVEN TO ACCEPT THE UNKNOWN ACCOUNT NUMBER OF A RUN. THE PRIORITY SUPPLIED IS PLACED IN THE 'P' AND 'Q' ENTRIES. DEADLINE AND REAL TIME ARE MADE ILLEGAL.

```
     35        29        23        17        11        05    00
    ------------------------------------------------------------
 00 :                ACCOUNT                                   :
    ------------------------------------------------------------
 01 :                NUMBER                                    :
    ------------------------------------------------------------
 02 :    P     :    Q    :   DL    :   RT    :   NEW   :       :
    ------------------------------------------------------------
 03 :                                                          :
    ------------------------------------------------------------
 04 :                                                          :
    ------------------------------------------------------------
 05 :                                                          :
    ------------------------------------------------------------
 06 :                                                          :
    ------------------------------------------------------------
 07 :                                                          :
    ------------------------------------------------------------
 08 :                                                          :
    ------------------------------------------------------------
 09 :                                                          :
    ------------------------------------------------------------
 10 :                DATE OF FIRST ENTRY                       :
    ------------------------------------------------------------
 11 :                TIME OF FIRST ENTRY                       :
    ------------------------------------------------------------
 12 :                DATE OF LAST ENTRY                        :
    ------------------------------------------------------------
 13 :                TIME OF LAST ENTRY                        :
    ------------------------------------------------------------
 14 :                DATE ENTRY LAST CLEARED                   :
    ------------------------------------------------------------
 15 :                TIME ENTRY LAST CLEARED                   :
    ------------------------------------------------------------
 16 :                TOTAL NUMBER OF RUNS                      :
    ------------------------------------------------------------
 17 :                TOTAL ELAPSED TIME OF RUNS                :
    ------------------------------------------------------------
 18 :                TOTAL COMPUTE TIME USED                   :
    ------------------------------------------------------------
```

```
19 :                   TOTAL CARDS/IMAGES IN                     :
   -----------------------------------------------------------
20 :                   TOTAL CARDS OUT                           :
   -----------------------------------------------------------
21 :                   TOTAL LINES OUT                           :
   -----------------------------------------------------------
22 :                                                             :
   -----------------------------------------------------------
23 :                                                             :
   -----------------------------------------------------------
24 :                                                             :
   -----------------------------------------------------------
25 :                   TRACK-MINUTES OF FH432 USAGE              :
   -----------------------------------------------------------
26 :                   TRACK-MINUTES OF FH880 USAGE              :
   -----------------------------------------------------------
27 :                   TRACK-MINUTES OF FASTRAND USAGE           :
   -----------------------------------------------------------
28 :                   TRACK-MINUTES OF FASTBAND USAGE           :
   -----------------------------------------------------------
29 :                                                             :
   -----------------------------------------------------------
30 :                                                             :
   -----------------------------------------------------------
31 :                   TOTAL TIME 8C ASSIGNED                    :
   -----------------------------------------------------------
32 :                   TOTAL TIME 6C ASSIGNED                    :
   -----------------------------------------------------------
33 :                   TOTAL TIME 4C ASSIGNED                    :
   -----------------------------------------------------------
34 :                   TOTAL TIME IIIA ASSIGNED                  :
   -----------------------------------------------------------
35 :                   TOTAL TIME IIA ASSIGNED                   :
   -----------------------------------------------------------
36 :                   TOTAL TIME FH432 ASSIGNED                 :
   -----------------------------------------------------------
37 :                   TOTAL TIME FH880 ASSIGNED                 :
   -----------------------------------------------------------
38 :                                                             :
   -----------------------------------------------------------
39 :                                                             :
   -----------------------------------------------------------
40 :                   TOTAL TIME CARD SUBSYSTEM ASSIGNED        :
   -----------------------------------------------------------
41 :                   TOTAL TIME PAPER TAPE SUBSYSTEM ASSIGNED  :
   -----------------------------------------------------------
42 :                   TOTAL TIME PRINTER ASSIGNED               :
   -----------------------------------------------------------
43 :                   TOTAL TIME 1004 ASSIGNED                  :
   -----------------------------------------------------------
44 :                   TOTAL TIME CTS ASSIGNED                   :
   -----------------------------------------------------------
45 :                   TOTAL TIME WTS ASSIGNED                   :
   -----------------------------------------------------------
46 :                   TOTAL TIME CTMC TERMINAL ASSIGNED         :
   -----------------------------------------------------------
47 :                                                             :
   -----------------------------------------------------------
48 :                                                             :
   -----------------------------------------------------------
49 :                                                             :
   -----------------------------------------------------------
50 :                                                             :
   -----------------------------------------------------------
```

```
51 :                                                          :
    ----------------------------------------------------------
52 :                                                          :
    ----------------------------------------------------------
53 :                                                          :
    ----------------------------------------------------------
54 :                                                          :
    ----------------------------------------------------------
```

WHERE:
  P = HIGEST ALLOWABLE PRIORITY.
  Q = PRIORITY USED WHEN NONE SPECIFIED.
  DL= NON-ZERO IF DEADLINE SPECIFICATION ALLOWED.
  RT= NON-ZERO IF REAL TIME SPECIFICATION ALLOWED.
 NEW= NON-ZERO IF ENTRY ADDED BY OPERATOR ACTION.

7.8. CHECKPOINT/RESTART

7.8.1. GENERAL

THE 1108 EXECUTIVE SYSTEM INCORPORATES A CHECKPOINT/RESTART FACILITY THAT WILL FUNCTION WITH ANY FILE FORMAT AT ANY LEVEL OF ACCESS. IT MAY BE INITIATED BY A USER PROGRAM, A CONTROL STATEMENT, OR AN UNSOLICITED KEYIN. ALL RUNS EXCEPT REAL-TIME MAY BE CHECKPOINTED.

THE CHECKPOINT ROUTINE WILL AUTOMATICALLY INCLUDE ALL PERTINENT INFORMATION NECESSARY FOR RESTART EXCEPT CATALOGUED OR 'TO BE CATALOGUED' FASTRAND FILES THAT ARE BEING UPDATED. THE USER MUST SPECIFY, BY AN OPTION ON THE ASG CONTROL CARD FOR EACH OF THESE CATALOGUED FILES, WHETHER THE CHECKPOINT ROUTINE SHOULD OR SHOULD NOT DUMP SUCH FILES. IF THEY ARE DUMPED, THE STATUS OF THE ORIGINAL FILE REMAINS UNCHANGED. (SEE CHAPTER ON THE ASG CONTROL CARD). ALL TEMPORARY FILES WILL BE AUTOMATICALLY DUMPED AND RELOADED ON RSTRT. IF THE TEMPORARY FILE DOES NOT NEED TO BE SAVED, IT SHOULD BE RELEASED BEFORE THE CHECKPOINT WITH A CALL TO CSF$.

CHECKPOINT INFORMATION MAY BE RECORDED ON TAPE OR FASTRAND. IF TAPE IS CHOSEN, THE INFORMATION CAN BE INTERSPERSED WITH DATA (CHECKPOINT ON OUTPUT DATA TAPE ONLY) OR RECORDED ON A NON-DATA TAPE (FILE INDEPENDENT CHECKPOINT). EACH CHECKPOINT TAKEN IS CAPABLE OF BEING USED AS A RESTART POINT AT SOME SUBSEQUENT TIME. IF CHECKPOINT INFORMATION IS RECORDED ON FASTRAND, ONLY THE LAST CHECKPOINT IS AVAILABLE FOR RESTARTING. ALSO IT IS NOT POSSIBLE TO INCLUDE DUMPS OF FILES WITHIN THIS CHECKPOINT.

THE RESTART ROUTINE RESTORES A PROGRAM TO THE OPERATIONAL STATUS IT ATTAINED WHEN THIS PARTICULAR CHECKPOINT WAS TAKEN. THE RESTORING PROCEDURE IS AUTOMATIC EXCEPT FOR CATALOGUED FASTRAND FILES THAT WERE DUMPED. AGAIN THE USER MUST SPECIFY, BY OPTION ON THE ASG CONTROL CARD FOR EACH CATALOGUED FILE THAT WAS DUMPED, WHETHER THESE FILES CAN BE RELOADED. WHEN RELOADING IS COMPLETE AND BEFORE REINITIATION, THE ROUTINE EXITS TO THE RESTART INTERRUPT LOCATION WHICH IS PROVIDED FOR THE USER WHO WISHES TO BE COGNIZANT OF ALL RESTARTS. IF THE INTERRUPT LOCATION IS NOT CODED THE PROGRAM IS AUTOMATICALLY REINITIATED.

7.8.2. USER-CHECKPOINT INTERFACE

THE USER INTERFACES WITH THE CHECKPOINT ROUTINE INTERNALLY BY A PROGRAM OR ACTIVITY, EXTERNALLY BY A CONTROL STATEMENT OR UNSOLICITED REQUEST.

7.8.2.1. PROGRAM REQUEST

THE PROGRAM REQUEST IS A PROCEDURE CODED IN THE USER SOURCE PROGRAM IN ANTICIPATION OF SOME CONTINGENCY. THE FREQUENCY OF THIS REQUEST AND THE RECORDING DEVICE FOR THE CHECKPOINT INFORMATION IS FIXED EACH TIME THIS RUN IS MADE. THE REQUEST FORM IS:

```
        C$KPT     'FILENAME',,'OPTIONS'
```

WHERE:

C$KPT IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FILENAME IS THE NAME OF AN ASSIGNED TAPE OR FASTRAND FILE INTO WHICH THE CHECKPOINT INFORMATION WILL BE RECORDED. IT MAY BE AN ACTUAL DATA FILE ALLOWING DIRECT ASSOCIATION OF CHECKPOINT AND FILE, OR IT MAY BE AN INDEPENDENT FILE CONTAINING CHECKPOINTS. IF THE CHECKPOINT INFORMATION IS TO BE RECORDED ON A FASTRAND FILE, IT MUST BE A SEQUENTIAL FILE. THE CHECKPOINT CAN NOT BE TAKEN ON A RANDOM FILE.

THE OPTIONS FIELD MAY INCLUDE P, T, OR PT. THESE SIGNIFY:

P THIS SPECIFIES THAT THE DUMP COMPLETION MESSAGE AND ALL ERROR MESSAGES ARE TO BE WRITTEN ON THE CONSOLE. IF NOT PRESENT NONE OF THESE MESSAGES WILL BE WRITTEN.

T THIS SPECIFIES THAT THE RUN IS TO BE TERMINATED AFTER THE CHECKPOINT IS TAKEN.

THE PROCEDURE WILL LOAD A0 WITH AN ENTRY CODE.

7.8.2.2. CONTROL STATEMENT REQUEST

THE CONTROL STATEMENT REQUEST APPEARS IN THE CONTROL STREAM OF A RUN.

7.8.2.3. UNSOLICITED REQUEST

THE UNSOLICITED REQUEST IS USED PRIMARILY BY THE OPERATOR TO CHECKPOINT A RUN. ALL RUNS IN THE OPERATIONAL ENVIRONMENT ARE SUSCEPTIBLE TO THIS REQUEST. THE REQUEST FORM IS:

```
        CK OPTIONS,FILENAME,RUNID
```

WHERE:

CK IS THE MESSAGE IDENTITY. FILENAME IS THE NAME OF THE TAPE FILE ASSIGNED TO THE RUNID INTO WHICH THE CHECKPOINT INFORMATION WILL BE RECORDED. FILENAME MAY ALSO BE THE NAME OF A FILE ASSIGNED BY THE OPERATOR FOR SYSTEM USE. IN SUCH A CASE THE DISPOSITION OF THE CHECKPOINT INFORMATION IS THE OPERATOR'S OPTION. THE OPERATOR CAN NOT TAKE A CHECKPOINT ON FASTRAND.

THE OPTIONS FIELD CAN INCLUDE TWO CHARACTERS. THESE ARE:

P THIS SPECIFIES THAT THE DUMP COMPLETION MESSAGE AND ALL ERROR MESSAGES ARE TO BE WRITTEN ON THE CONSOLE. IF NOT PRESENT NONE OF THESE MESSAGES WILL BE WRITTEN.

T THIS SPECIFIES THAT THE RUN IS TO BE TERMINATED AFTER THE CHECKPOINT IS TAKEN.

7.8.3. DUMPING TECHNIQUES

THE DUMPING TECHNIQUE VARIES DEPENDING ON WHETHER THE CHECKPOINT IS TO BE RECORDED ON TAPE OR FASTRAND. WHILE FASTRAND MAY CONTAIN ONLY ONE VALID DUMP PER RUN, TAPE MAY CONTAIN MANY. FILES ARE DUMPED WHEN CHECKPOINT IS ON TAPE AND NEVER WHEN ON FASTRAND.

THE SAVING OF PERTINENT INFORMATION NECESSARY FOR RESTARTING IS ACCOMPLISHED AUTOMATICALLY REGARDLESS OF THE DEVICE CHOSEN. HOWEVER, THE DISPOSITION OF THE FILES DEPENDS DIRECTLY ON BOTH THE DEVICE AND THE USER'S OPTION. IF A CARD READER OR PAPER TAPE IS ASSIGNED DIRECTLY TO THE PROGRAM, THE CHECKPOINT CAN NOT BE TAKEN.

THE FOLLOWING ARE SOME OF THE STEPS INVOLVED WHEN PROCESSING A CHECKPOINT REQUEST.

THEIR ORDER IN THE LIST IS NOT NECESSARILY THEIR ORDER OF OCCURRENCE.

(1) ALL REGISTERS ARE SAVED
(2) ALL I/O ACTIVITY FOR THE RUN IS COMPLETED
(3) THE REMAINING PORTION OF THE CONTROL STREAM IS SAVED
(4) EXEC CONTROL TABLES, SWITCH LIST POSITIONS, ETC., NECESSARY TO REESTABLISH THE RUN ARE SAVED
(5) A CHECKPOINT ID RECORD IS WRITTEN
(6) USER'S CORE AREA IS WRITTEN
(7) THE POSITIONS OF ALL TAPE FILES ARE NOTED
(8) TEMPORARY DRUM AND FASTRAND FILES ARE DUMPED
(9) CATALOGUED UPDATE FILES ARE DISPOSED OF ACCORDING TO ASG CARD OPTIONS
(10) AN END OF CHECKPOINT RECORD IS WRITTEN

IT SHOULD BE NOTED THAT WHEN CHECKPOINT IS ON FASTRAND, OPERATIONS (8) AND (9) ARE NOT PERFORMED.

7.8.3.1. TAPE

IF THE CHECKPOINT UNIT IS TAPE AND AN END OF REEL CONDITION IS ENCOUNTERED WHILE A DUMP IS BEING RECORDED, THE ROUTINE WILL AUTOMATICALLY SWAP REELS AND CONTINUE DUMPING ON THE ALTERNATE REEL. CONTROL WILL NOT BE RETURNED TO THE USER UNTIL ALL DUMPING IS COMPLETE. BECAUSE OF THIS, A DATA TAPE FILE SHOULD NOT BE DESIGNATED AS THE CHECKPOINT UNIT IF THE USER HAS PROVIDED FOR PHYSICAL END OF REEL PROCESSING. REEL SWAPPING OF INDEPENDENT TAPE FILES ASSIGNED TO THE PROGRAM OR TAPE FILES ASSIGNED TO THE SYSTEM (BY THE OPERATOR) DOES NOT AFFECT THE USER PROGRAM.

A REEL CONTAINING A PORTION OF A CHECKPOINT AND ITS ALTERNATE THAT CONTAINS THE BALANCE OF THE DUMP, ARE TREATED AS ONE LOGICAL REEL WHEN READ BY THE DATA HANDLING ROUTINE.

A CHECKPOINT CAN NOT BE TAKEN ON A TAPE IN WHICH THE MODE HAS BEEN CHANGED IN THE MIDDLE OF THE FILE. ALSO, CHECKPOINTS SHOULD NOT BE TAKEN ON A TAPE WRITTEN IN 200 PPI, BECAUSE IF THE END OF REEL IS REACHED, THE CHECKPOINT ROUTINE MIGHT WRITE OFF THE TAPE.

CHECKPOINT DUMPS ON A DATA TAPE OR AN INDEPENDENT TAPE MAY BE CATALOGUED BY THE USER IN THE NORMAL MANNER (ASG CONTROL CARD OPTION). HOWEVER, THE DISPOSITION OF THE SYSTEM ASSIGNED TAPE IS THE OPERATOR'S OPTION.

7.8.3.2. FASTRAND

CHECKPOINT ON FASTRAND IS LIMITED TO ONE ACTIVE DUMP PER RUN TO MINIMIZE ABUSE OF STORAGE CAPACITY. THIS TYPE OF CHECKPOINT WILL NOT CONTAIN DUMPS OF FILES. THIS IMPLIES THAT CHECKPOINT ON FASTRAND SHOULD BE CALLED FOR ONLY BY THOSE RUNS WHERE HARDWARE MALFUNCTIONS ARE THE EXPECTED CONTINGENCIES AND IMMEDIATE RESTART IS DESIRABLE. SUCH RUNS COULD BE CLASSIFIED AS PRODUCTION TYPES, ETC.

7.8.3.3. CONTROL MESSAGES

THE CHECKPOINT ROUTINE PRODUCES A CONTROL MESSAGE TO INDICATE THAT A DUMP WAS COMPLETED SUCCESSFULLY. NORMALLY THIS MESSAGE IS RECORDED IN THE RUN LOG; HOWEVER, IF THE PRINT OPTION IN THE CHECKPOINT REQUEST IS USED IT WILL ALSO APPEAR ON THE OPERATOR'S CONSOLE. THE MESSAGE FORM IS:

```
RUNID CKPT# FILENAME, REEL NBR
```

WHERE RUNID IS THE IDENTITY OF THE RUN. CKPT# IS THE NUMBER OF THE CHECKPOINT (TAKEN IN NUMERICAL ORDER). FILENAME IS THE NAME OF THE FILE TO CONTAIN THE CHECKPOINT DUMP, AND REEL NBR IS THE NUMBER OF THE REEL(USUALLY GIVEN BY THE OPERATOR IN RESPONSE TO A MOUNT MESSAGE) OF THE FILE THAT ACTUALLY CONTAINS THE DUMP, IF THE DUMP IS ON TAPE.

A SUBSEQUENT MESSAGE IS PRODUCED IN CONJUNCTION WITH THE ONE ABOVE BUT IT IS RECORDED ONLY IN THE PROGRAM LOG. THIS MESSAGE HAS THE FORMS:

```
      FILENAME(1) - REEL NBR(1), *REEL NBR(2),...,REEL NBR(N)
      FILENAME(2) - REEL NBR(1), REEL NBR(2),...*REEL NBR(N)
                  -      -          -       -        -
                  -      -          -       -        -
      FILENAME(N) - *REEL NBR(1), REEL NBR(2),... REEL NBR(N)
OR,
      FILENAME(1) - FS
          -
      FILENAME(N) - FS
OR,
      FILENAME(1) - DM
          -
      FILENAME(N) - DM
```

WHERE THE FILENAMES ARE THE NAMES OF THE FILES PRESENTLY ASSIGNED TO THE RUN. THE REEL NBRS ARE THE NUMBERS OF THE REELS OF THE FILES. AN ASTERISK BEFORE A REEL NBR INDICATES THE REEL THAT WAS ACTUALLY IN USE AT THE TIME OF THE DUMP. FILES RESIDING ON FASTRAND AND DRUM ARE DENOTED BY THE CHARACTERS FS AND DM.

WHEN HARDWARE AND PARAMETER FORMAT ERRORS OCCUR, MESSAGES WILL BE WRITTEN ON THE CONSOLE, BUT ONLY WHEN THE P OPTION IS SPECIFIED. FOR FORMAT AND LEGAL RESPONSES TO THESE MESSAGES, SEE OPERATOR COMMUNICATIONS, SECTION 11.

7.8.3.4. STATUS CODES

WHEN A PROGRAM REQUESTED CHECKPOINT HAS BEEN COMPLETED, A STATUS CODE IS RETURNED IN A0. FOR A RESTART THE CODE IS RETURNED IN BITS 35-30 OF THE FIRST WORD OF THE CONTINGENCY INTERRUPT ROUTINE. AFTER A RESTART THE CODES ARE:

```
00 - NORMAL COMPLETION
01 - UNRECOVERABLE TAPE ERROR
02 - UNRECOVERABLE DRUM ERROR
03 - CKPT# NOT ON FILE SPECIFIED
04 - CKPT# NOT ON REEL SPECIFIED
05 - DUMP FILE NOT CATALOGUED
06 - ERROR OPTION FOR NO RELOAD WAS SPECIFIED.  THE FILE
     IS NO LONGER CATALOGUED
```

AFTER A CHECKPOINT THE STATUS CODES ARE:

```
00 - NORMAL COMPLETION
01 - UNRECOVERABLE TAPE ERROR
02 - UNRECOVERABLE DRUM ERROR
03 - DUMP FILE IS NOT ASSIGNED TO THE PROGRAM AND IS
     NOT A SYSTEM FILE
```

7.8.4. USER-RESTART INTERFACE

THE RESTART PROCEDURE IS THE REVERSE OF THE CHECKPOINT PROCEDURE. THAT IS, IT RESTORES A RUN TO THE OPERATIONAL STATUS IT HAD ATTAINED WHEN THE DUMP WAS TAKEN. THIS DUMP FILE MUST BE CATALOGUED PRIOR TO RESTART, EITHER ON AN ASG CARD IN A PREVIOUS RUN OR BY A CAT CARD. THE REQUESTS ARE SIMILAR TO THE CHECKPOINT REQUESTS IN THAT THEY ALSO CAN BE INITIATED INTERNALLY BY A PROGRAM OR ACTIVITY, AND EXTERNALLY BY A CONTROL STATEMENT OR UNSOLICITED REQUEST. ANY CHECKPOINT DUMP IS CAPABLE OF BEING REESTABLISHED BY ANY RESTART REQUEST REGARDLESS OF THE TYPE OF CHECKPOINT REQUEST USED TO PRODUCE THE DUMP. FOR EXAMPLE, AN INTERNAL CHECKPOINT REQUEST MAY BE RESTARTED BY EITHER AN INTERNAL, CONTROL STATEMENT, OR UNSOLICITED RESTART REQUEST.

THE RESTART INTERRUPT LOCATION WILL BE CHECKED UPON COMPLETION OF THE LOAD TO ASCERTAIN IF THE USER WANTS CONTROL AT THIS INTERRUPT LOCATION BEFORE THE RUN IS

INITIATED. IF THE USER DOES ACCEPT CONTROL, IT IS HIS RESPONSIBILITY TO TRANSFER CONTROL TO REINITIATE HIS RUN AT THE COMPLETION OF THE INTERRUPT SEQUENCE.

7.8.4.1. PROGRAM REQUEST

THE PROGRAM REQUEST IS A PROCEDURE CODED IN THE USER'S SOURCE PROGRAM TO AFFORD AN IMMEDIATE RESTART CAPABILITY WITHOUT LEAVING THE OPERATIONAL ENVIRONMENT, AND/OR WITHOUT OPERATOR INTERVENTION. THE REQUEST FORM IS:

```
      R$STRT   'FILENAME',CKPT#,'P'
```

WHERE:

R$STRT IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. P SPECIFIES THAT ERROR MESSAGES ARE TO BE TYPED ON THE CONSOLE. FILENAME IS THE NAME OF A TAPE OR FASTRAND FILE ASSIGNED TO THE SYSTEM OR RUN THAT CONTAINS THE CHECKPOINT DUMP TO BE RESTARTED. CKPT# IS THE NUMBER OF THE PARTICULAR CHECKPOINT TO USE. WITH FILENAME AND CKPT# THE SYSTEM AUTOMATICALLY SELECTS THE REEL THAT ACTUALLY CONTAINS THE DUMP.

7.8.4.2. CONTROL STATEMENT REQUEST

THE CONTROL REQUEST APPEARS IN THE CONTROL STREAM OF A RUN OR, AS A SEPARATE RUN. (SEE THE CHAPTER ON CONTROL STATEMENTS FOR A DETAILED DESCRIPTION)

7.8.4.3. UNSOLICITED REQUEST

THE UNSOLICITED REQUEST IS USED PRIMARILY BY THE OPERATOR TO RESTART A PARTICULAR RUN OR GROUP OF RUNS. ONLY CHECKPOINT DUMPS ON TAPE ARE SUSCEPTIBLE TO THIS REQUEST. THE REQUEST FORM IS:

```
      RS PRIORITY/P,RUNID,ACCOUNTING,FILENAME,CKPT#
```

WHERE:

RS IS THE REQUEST IDENTITY AND MUST BE USED AS SHOWN. P, IF GIVEN, SPECIFIES THAT ERROR MESSAGES ARE TO BE TYPED ON THE CONSOLE. THE PRIORITY FIELD SPECIFIES THE NEW PRIORITY UNDER WHICH TO REESTABLISH AND RERUN THE RUN. IF NOT SPECIFIED THE STANDARD SYSTEM PRIORITY IS USED. RUNID IS THE IDENTITY OF THE CHECKPOINTED RUN. ACCOUNTING IS THE ACCOUNT TO WHICH THE CHARGE FOR RELOADING THE RUN WILL BE ATTRIBUTED. CKPT# IS THE NUMBER OF THE SPECIFIC DUMP TO RESTART. FILENAME IS THE NAME OF THE TAPE FILE THAT CONTAINS THE CHECKPOINT DUMP. THIS FILE MUST BE EITHER A CATALOGUED FILE OR A SYSTEM ASSIGNED FILE. FOR CATALOGUED FILES, ALL THE INFORMATION NECESSARY TO CORRECTLY READ THE DUMP TAPE WILL BE OBTAINED FROM THE DIRECTORY. SYSTEM ASSIGNED FILES ARE ALWAYS READ IN THE STANDARD MODE.

8. EXECUTIVE REQUEST FUNCTIONS

8.1. GENERAL

A SET OF FUNCTIONS ARE PROVIDED BY THE EXECUTIVE FOR USE BY USER PROGRAMS. THIS SET OF FUNCTIONS, GENERALLY SPEAKING, ENCOMPASSES THOSE OPERATIONS WHICH ARE EITHER ILLEGAL FOR THE USER PROGRAM TO INITIATE DIRECTLY OR REQUIRE THE COGNIZANCE AND DIRECTION OF THE EXECUTIVE SYSTEM. THE FUNCTIONS ARE CATEGORIZED AS FOLLOWS:

ACTIVITY REGISTRATION
TERMINATION
FILE SUPERVISION
REAL-TIME
RE-ENTRANT ROUTINES
DYNAMIC FACILITIES
SYSTEM LOG
ERROR INTERRUPTS
INPUT/OUTPUT REQUESTS
CONSOLE REQUESTS
SYMBIONT CONTROL
CHECKPOINT/RESTART
MISCELLEANOUS

THE EXECUTIVE SYSTEM MAY BE ENTERED TO INITIATE A FUNCTION THROUGH UTILIZATION OF THE EXECUTIVE REQUEST INSTRUCTION (ER) OF THE GENERAL FORM:

```
ER  FUNCTION-ID
```

THE FUNCTION -ID TO BE AN ABSOLUTE NAME AS GIVEN IN ERU$. ONLY THE U FIELD IS CODED. NO INDIRECT ADDRESSING, USE OF INDEX REGISTERS,OR REQUEST FOR INDEX MODIFICATIONS ARE INTERPRETED BY THE SOFTWARE PROCESSING OF AN ER FUNCTION. IN SOME CASES, THE PRESENCE OF PARAMETERS ARE REQUIRED IN CONTROL REGISTERS WHEN THE EXECUTIVE SYSTEM IS ENTERED. THE DISCUSSIONS IN THIS CHAPTER ON EACH OF THE INDIVIDUAL FUNCTIONS WILL SPECIFY THE REQUIRED PARAMETERS.

EXECUTIVE REQUEST FUNCTIONS MAY BE FURTHER CATEGORIZED AS ASYNCHRONOUS, SYNCHRONOUS, OR IMMEDIATE. AN ASYNCHRONOUS REQUEST ALLOWS THE RETURN OF CPU CONTROL TO THE REQUESTING PROGRAM AS SOON AS ITS PRIORITY PERMITS AND REQUIRES INTERROGATIVE ACTION BY THE USER PROGRAM TO DETERMINE WHETHER THE REQUEST HAS BEEN COMPLETED. SYNCHRONOUS REQUESTS SUSPEND OPERATION OF THE REQUESTING PROGRAM UNTIL THE REQUEST HAS BEEN COMPLETED. AT THAT TIME, SYSTEM CONTROL RETURNS TO THE PROGRAM. WHILE THE PROGRAM IS SUSPENDED, OTHER PROGRAMS ARE EXECUTING. AN IMMEDIATE REQUEST IS ONE WHICH IS PERFORMED ON THE SPOT BY THE SYSTEM. IMMEDIATE EXECUTIVE REQUESTS ARE CONFINED TO THOSE WHICH CAN BE FULFILLED IN A SUFFICIENTLY SMALL NUMBER OF INSTRUCTIONS THAT CPU DISPATCHING AT THIS TIME WOULD NOT SERIOUSLY DEGRADE PERFORMANCE OF THE SYSTEM.

FOR ASYNCHRONOUS AND SYNCHRONOUS REQUESTS, SUFFICIENT INFORMATION IS RECORDED IN A QUEUE TO ALLOW THE REQUEST TO BE PERFORMED. THIS QUEUE ENTRY IS MAINTAINED UNTIL THE REQUEST IS SATISFIED. IF THE SERVICE ROUTINE IS RE-ENTERABLE, IT ALLOWS SERVICE REQUESTS FROM SEVERAL USER OR SYSTEM PROGRAMS TO BE PERFORMED SIMULTANEOUSLY, AS INPUT/OUTPUT DEVICE AVAILABILITY PERMITS.

INSOFAR AS IS FEASIBLE, CPU DISPATCHING FOR SERVICE ROUTINES IS AT A PRIORITY LEVEL RELATED TO THAT OF THE REQUESTING ROUTINE.

8.2. ACTIVITY REGISTRATION

PROVISIONS ARE AVAILABLE TO ALLOW INDEPENDENT ACTIVITIES OF A PROGRAM TO BE REGISTERED AND EXECUTED CONCURRENTLY. THE SUBPROGRAM MAY BE REGISTERED AS EITHER A TIMED (TFORK$) OR AN UN-TIMED (FORK$) ACTIVITY.

THE USER PROGRAM HAS A MAIN ACTIVITY IN EVERY CASE. ITS CONTROL MEMORY ALWAYS CONSISTS OF ALL A, B, AND R REGISTERS WITH THE EXCEPTION OF R0. THIS DOES NOT IMPLY THAT MAJOR CONTROL OF THE TASK MUST BE WITHIN THIS PATH, BUT THAT EXEC FOR CONTROL PURPOSES DOES RECOGNIZE A MAIN ACTIVITY.

8.2.1. FORK$

THE FORK$ FUNCTION REGISTERS AN UN-TIMED ACTIVITY.

A UNIQUE SUBPROGRAM CONTROL WORD IS MAINTAINED FOR EACH MAIN PROGRAM BY THE DISPATCHER. THIS CONTROL WORD IS USED FOR CONTROL OVER THOSE ACTIVITIES THAT HAVE BEEN REGISTERED WITH AN IDENTITY ATTACHED. A LIMIT OF 35 ACTIVITIES WITH AN IDENTITY MAY BE REGISTERED AT ANY ONE TIME. AS EACH SUBPROGRAM IS REGISTERED, THE CORRESPONDING BIT IS SET ON THE CONTROL WORD. IF A SUBPROGRAM REGISTRATION IS ATTEMPTED WITH AN IDENTITY EQUAL TO A SUBPROGRAM PRESENTLY REGISTERED, AN 'ERROR MODE' CONDITION RESULTS.

IF IT IS DESIRED TO REGISTER A REAL-TIME ACTIVITY, A PRIORITY NUMBER MUST BE INCLUDED AS A PARAMETER. THIS PRIORITY NUMBER IS THE ACTUAL SWITCHING LEVEL ON THE REAL-TIME SWITCH LIST. THE PRIORITY VALUE IS CHECKED AGAINST THE PRIORITY LIMIT ASSIGNED TO THE ACCOUNT NUMBER. IF AN ESI COMPLETION ACTIVITY IS REGISTERED, THE ACTIVITY WILL REMAIN IN A WAIT STATE ON THE SWITCHING CYCLE UNTIL ACTIVATED AS THE RESULT OF AN ESI INTERRUPT.

INFORMATION WHICH MUST BE SUPPLIED IN CONTROL REGISTER, A0, UPON REGISTERING A NEW ACTIVITY INCLUDES: THE INITIAL ADDRESS AT WHICH THE NEW ACTIVITY IS TO BEGIN EXECUTION, AN INDICATION OF THE AMOUNT OF CONTROL REGISTERS WHICH MUST BE SAVED, IF DESIRED AN ACTIVITY IDENTITY AND FOR REAL-TIME REGISTRATIONS A SWITCHING LEVEL PRIORITY NUMBER. UPON RETURNING CONTROL TO THE REFERENCING PROGRAM THE EXEC WILL LOAD AN EXEC ACTIVITY IDENTITY IN THE LOWER HALF OF A0.

```
 35      30 29     24 23     18 17                            0
--------------------------------------------------------------
:    A    :    B    :    C    :              D               :
--------------------------------------------------------------
```

WHERE,

A - REAL-TIME SWITCHING LEVEL PRIORITY NUMBER. THIS FIELD IS USED ONLY WHEN REGISTERING A REAL-TIME WORKER OR REAL-TIME INTERRUPT ACTIVITY. THE NUMBER ONE (1) IS RESERVED FOR REAL-TIME INTERRUPT ACTIVITIES. SWITCHING LEVELS, (2-35), MAY BE REQUESTED FOR REAL-TIME WORKERS. THE PRIORITY REQUESTED IS CHECKED AGAINST THE PRIORITY LIMIT OF THE ACCOUNT NUMBER. ALL NON REAL-TIME ACTIVITIES REGISTERED INITIALLY ASSUME THE SWITCHING PRIORITY LEVEL OF THE MAIN PROGRAM. INTERRUPT ACTIVITIES WILL AUTOMATICALLY BE ASSIGNED A HIGHER SWITCHING PRIORITY LEVEL THAN ALL NORMAL ACTIVITIES FOR THIS SPECIFIC CLASS OF PROGRAM (BATCH, DEMAND, ETC.).

B - ACTIVITY IDENTIFICATION (1-35; A LIMIT OF 35 ACTIVITIES MAY BE REGISTERED WITH AN IDENTITY AT ANY ONE TIME). IF THIS FIELD IS BLANK THE ACTIVITY IS REGISTERED WITHOUT AN IDENTITY.

C - AMOUNT OF CR WHICH MUST BE SAVED. (0 = B11, A0 THRU A5, AND R1 THRU R3; 1 = ALL A, B, AND R REGISTERS WITH THE EXCEPTION OF R0). THE SET OF CONTROL REGISTERS CHOSEN WILL REMAIN WITH THE ACTIVITY THROUGHOUT THE LIFE OF THE ACTIVITY. UPON INITIALLY RECEIVING CONTROL, THE SET OF

CONTROL REGISTERS ASSIGNED TO THE PARENT PROGRAM WILL BE PASSED ON.

D - ENTRY ADDRESS

THE EXEC REFERENCE IS:

```
        L    A0,PARAMETER-WORD
        ER   FORK$
```

THE FOLLOWING ERROR CODES APPEAR UNDER THE CIRCUMSTANCES AND THE ACTIVITY IS TERMINATED.

0405 DUPLICATE OR INVALID ACTIVITY IDENTITY REQUESTED WHEN REFERENCING FORKING

0406 REAL-TIME PRIORITY IS NOT PERMITTED UNDER THE ACCOUNT NUMBER USED.

8.2.2. TIMED ACTIVITY REGISTRATION

THE TFORK$ FUNCTION REGISTERS A TIMED ACTIVITY. THE TIMED REGISTRATION IS IDENTICAL TO THE UN-TIMED FUNCTION WITH THE EXCEPTION THAT AN ADDITIONAL PARAMETER MUST BE SUPPLIED IN CONTROL REGISTER, A1, GIVING THE INCREMENT OF TIME IN MILLISECONDS TO DELAY INITIATION OF THE ACTIVITY.

THE EXEC REFERENCE IS:

```
        L    A0,PARAMETER-WORD
        LA   A1,WAIT TIME IN MILLISECONDS
        ER   TFORK$
```

8.2.3. TIMED WAIT

THE TWAIT$ FUNCTION PLACES A PROGRAM IN WAIT STATE FOR THE AMOUNT OF TIME SPECIFIED IN THE REGISTER,A1. THE EXEC REFERENCE IS:

```
        L'   A1,WAIT TIME IN MILLISECONDS
        ER   TWAIT$
```

8.2.4. WAIT CONDITIONS

USING THIS FUNCTION, THE SUBPROGRAM MAY WAIT UNTIL ANY NUMBER OF OTHER ACTIVITIES OF THIS SET ARE DEACTIVATED BEFORE IT IS GIVEN CONTROL. A 'CONDITION' MASK MUST BE SUPPLIED IN CONTROL REGISTER, A0. THIS MASK IS APPLIED AGAINST THE SUBPROGRAM CONTROL WORD BY THE DISPATCHER. FOR EXAMPLE IF ACTIVITIES 1 AND 4 MUST BE COMPLETED BEFORE AN ACTIVITY MAY GAIN CONTROL, A0 WOULD BE LOADED WITH A 22 (OCTAL), AND BITS 1 AND 4 WOULD BE MASKED ON THE SUBPROGRAM CONTROL WORD.

THE EXEC REFERENCE IS:

```
        L    A0,CONDITION-MASK
        ER   AWAIT$
```

NOTE: IOXI$ WILL NOT GIVE A NEW ACTIVITY ID.

THE FOLLOWING ERROR CODE APPLIES AND THE ACTIVITY IS TERMINATED.

0404 NO IDENTITY UPON AWAIT$ REQUEST

8.3. ACTIVITY NAMING AND INTERACTION.

8.3.1. NAMING TECHNIQUE.

A NAME MAY BE ATTACHED TO EACH NON-ESI ACTIVITY, IF DESIRED, FOR THE PURPOSE OF IDENTIFICATION. THIS NAME IS USED WHEN ONE ACTIVITY OF A PROGRAM DESIRES TO ACTIVATE OR DELETE ANOTHER ACTIVITY OF THE SAME PROGRAM. THIS NAME IS NOT THE SAME AS THE IDENTITY USED IN CONJUNCTION WITH AWAIT$.

THE NAME IS 36 BITS IN LENGTH. EIGHTEEN BITS ARE DEFINED BY THE USER AND THE REMAINING 18 BITS BY THE EXEC. THE USER PORTION MAY BE OF ANY VALUE DESIRED. IT IS NOT GUARANTEED UNIQUE BY EXEC8.

AN ACTIVITY NAME TABLE EXISTS IN THE PCT WHEN A NAME EXISTS.

THE CALLING SEQUENCE FOR NAMING AN ACTIVITY IS AS FOLLOWS:

```
ER   NAME$
```

WITH THE LOWER A0 (H2 PORTION) CONTAINING THE 18 BIT USER SUPPLIED SYMBOLIC NAME. THE FULL 36 BIT NAME IS RETURNED IN A0 AND MUST BE USED FOR ACT$.

A NAME IS NOT AUTOMATICALLY ATTACHED TO AN INTERRUPT ACTIVITY THROUGH AN I/O REFERENCE EXCEPT WHERE AN ACTIVITY IS CONVERTED TO AN INTERRUPT ACTIVITY. THE ONLY TWO REQUESTS WHICH CONVERT AN ACTIVITY ARE IOXI$ AND IOAXI$. THESE REQUESTS DO NOT GENERATE NEW ACTIVITIES BUT TERMINATE THE REQUESTORS PRIORITY LEVEL, REDUCE TO THE MINOR REGISTER SET IF NECESSARY, AND RAISE IT TO THE INTERRUPT ACTIVITY LEVEL. ANY ACTIVITY GENERATED THROUGH AN I/O REFERENCE MAY MAKE A REFERENCE TO NAME$ TO ATTACH A NAME.

8.3.2. ACTIVITY ACTIVATION (ACT$)

A NAMED ACTIVITY MAY BE ACTIVATED BY ANOTHER ACTIVITY VIA THE REFERENCE:

```
ER ACT$
```

WITH A0 CONTAINING THE 36 BIT NAME RETURNED IN A0 ON A NAME$ REFERENCE.

8.3.3. ACTIVITY DEACTIVATION (DACT$

A NAMED ACTIVITY MAY DEACTIVATE ITSELF VIA THE REFERENCE:

```
ER DACT$
```

THE ONLY WAY THAT THE ACTIVITY CAN BE REACTIVATED IS VIA THE ACT$ REFERENCE. CONTROL IS RETURNED FOLLOWING THE ER ON ACTIVATION. THE PROGRAM MUST INSURE ALL ACTIVITIES ARE ACTIVATED BEFORE TERMINATION OCCURS.

8.3.4. ACTIVITY DELETION (ADLT$)

A NAMED ACTIVITY MAY BE DELETED VIA THE REFERENCE

```
ER ADLT$
```

WITH A0 CONTAINING THE 36 BIT NAME RRTURNED IN A0 ON A NAME$ REFERENCE. CONTROL IS RETURNED FOLLOWING THE ER UNLESS THE NAME CANNOT BE FOUND, IN WHICH CASE THE CONTINGENCY PATH IS TAKEN. AN ACTIVITY SUPPLYING ITS OWN NAME WILL BE TREATED AS THOUGH IT HAD MADE AN EXIT$ REFERENCE.

8.4. TERMINATION

8.4.1. THE EXIT FUNCTION

THE FINAL EXIT TO EXEC OR THE TERMINATION OF ANY ACTIVITY MAY BE THE EXIT FUNCTION. THE EXEC DELETES THE ACTIVITY AND CHECKS FOR THIS BEING THE LAST. IF ALL ACTIVITIES HAVE BEEN DELETED, THE PROGRAM IS TERMINATED.

THE EXEC REFERENCE IS:

```
        ER  EXIT$
```

IF ANY ASYNCHRONOUS I/O HAS BEEN ISSUED BY THE EXITING ACTIVITY, THE I/O REQUESTS ARE DELETED FROM THE CHANNEL QUEUE OR, IF BEING CURRENTLY SERVICED, ARE STOPPED AT THE FIRST POSSIBLE BREAK POINT. NO FURTHER UPDATING IS DONE FOR THE I/O REQUESTS AND NO I/O INTERRUPT ACTIVITIES ARE REGISTERED FOR THE BYPASSED REQUESTS.

ALL ESI COMPLETION ACTIVITIES USE THE EXIT$ REQUEST HOWEVER THESE ACTIVITIES ARE RETURNED TO THE SYSTEM AND ARE EXECUTED AGAIN AS ESI ACTIVITIES OR ARE TERMINATED VIA THE COMMUNICATIONS HANDLER TERMINATE.

8.4.2. THE ABORT FUNCTION

THE ABORT FUNCTION IS AN EXIT INDICATING THAT THE ACTIVITY IS ENDING IN AN ABORT CONDITION. ALL CURRENT ACTIVITIES ARE TERMINATED, AND THEN THE RUN IS TERMINATED (IF IT IS NOT DEMAND). THE EXEC REFERENCE IS:

```
        ER    ABORT$
```

NOTE: A PMD IS NOT HONORED AT THIS POINT.

8.4.3. THE ERR EXIT

THE ERR FUNCTION SETS THE 'ERR MODE' CONDITION. IF THE WORKER HAS ESTABLISHED AN 'ERR MODE' ROUTINE, CONTROL WILL BE RETURNED TO IT. OTHERWISE, STANDARD 'ERR MODE' ACTION WILL OCCUR. FOR INFORMATION ON THE STANDARD 'ERR MODE' ACTION REFER TO THE PROGRAM CONTINGENCY PARAGRAPH IN THE SUPERVISOR CHAPTER.

THE EXEC REFERENCE IS:

```
        ER   ERR$
```

NOTE: A PMD IS HONORED AT THIS POINT.

8.5. FILE SUPERVISION

SEVERAL ENTRANCES INTO THE EXEC ARE PROVIDED FOR FILE SUPERVISION:

BJOIN$	-	BUFFER POOL EXPANSION
BPOOL$	-	BUFFER POOL SETUP
CADD$	-	COMMUNICATIONS BUFFER POOL ADDITION
CGET$	-	COMMUNICATIONS BUFFER POOL REMOVAL
CPOOL$	-	COMMUNICATIONS BUFFER POOL SETUP
DITEM$	-	DIRECTORY ITEM RETRIEVAL
FACIL$	-	FACILITY DETERMINATION
TINTL$	-	TAPE REEL NUMBER SET INITIALIZATION
TSWAP$	-	TAPE SWAPPING

A SUMMARY DESCRIPTION OF EACH FOLLOWS. FOR ADDITIONAL INFORMATION, REFER TO THE CHAPTERS ON FILE CONTROL SYSTEM AND DATA HANDLING.

8.5.1. BUFFER POOL EXPANSION

THE BJOIN$ FUNCTION PROVIDES THE MEANS FOR ADDING A CORE AREA TO A PREVIOUSLY ESTABLISHED BUFFER POOL. A TWO WORD PACKET MUST BE PROVIDED TO IDENTIFY THE AREA AS FOLLOWS:

```
 35                            17                           00
-------------------------------------------------------------
:  NOT USED                    :ADDR OF INIT CONTROL PACKET  :
-------------------------------------------------------------
:LENGTH OF ADDITIONAL AREA     :ADDRESS OF ADDITIONAL AREA   :
-------------------------------------------------------------
```

THE ADDRESS OF THIS PACKET MUST BE IN REGISTER A0 WHEN THE EXECUTIVE IS ENTERED.

THE EXEC REFERENCE IS:

```
L     A0,PACKET ADDRESS
ER    BJOIN$
```

8.5.2. BUFFER POOL SETUP

THE BPOOL$ FUNCTION PROVIDES THE MEANS FOR THE INITIAL SETUP OF A BUFFER POOL AT PROGRAM EXECUTION TIME. IT SHOULD BE NOTED THAT A BUFFER POOL CAN ALSO BE CREATED BY THE SYSTEM PROCEDURE B$GPUL IF THE MAIN STORAGE AREA FOR THE BUFFER POOL CAN BE ALLOTTED BY ASSEMBLY TIME.

THE BPOOL$ FUNCTION REQUIRES A TWO WORD PACKET AS FOLLOWS TO DEFINE THE AREA.

```
 35                            17                           00
-------------------------------------------------------------
:  BUFFER SIZE                 :  ADDRESS OF FIRST BUFFER    :
-------------------------------------------------------------
:  NOT USED                    :  LENGTH OF CORE AREA        :
-------------------------------------------------------------
```

THE ADDRESS OF THE PACKET MUST BE IN REGISTER A0 WHEN THE EXECUTIVE IS ENTERED.

THE EXEC REFERENCE IS:

```
L     A0,PACKET ADDRESS
ER    BPOOL$
```

8.5.3. COMMUNICATIONS BUFFER POOL ADDITION

THE CADD$ FUNCTION PROVIDES THE MEANS FOR RETURNING BUFFERS TO THE PREVIOUSLY ESTABLISHED CHARACTER ORIENTED BUFFER POOL FROM WHICH THE BUFFERS HAD BEEN REMOVED. A TWO WORD PACKET MUST BE PROVIDED TO IDENTIFY THE BUFFERS AS FOLLOWS:

```
    35                            17                           00
   -------------------------------------------------------------
01 : NUMBER TO BE ADDED           : EXEC LINKING VALUE FOR POOL  :
   -------------------------------------------------------------
02 : NOT USED                     : ADDRESS OF FIRST BUFFER      :
   -------------------------------------------------------------
```

THE ADDRESS OF THE PACKET MUST BE IN REGISTER A0 WHEN THE EXECUTIVE IS ENTERED.

THE EXEC REFERENCE IS:

```
L     A0,PACKET ADDRESS
ER    CADD$
```

8.5.4. COMMUNICATIONS BUFFER POOL REMOVAL

THE CGET$ FUNCTION PROVIDES THE MEANS FOR REMOVING BUFFERS FROM A PREVIOUSLY ESTABLISHED CHARACTER ORIENTED BUFFER POOL FOR SPECIFIC USE OF THOSE BUFFER AREAS BY THE REQUESTOR WITH THE ASSURANCE THAT ALL OF THE BUFFERS REMOVED WILL NOT BE USED BY ANY OTHER PORTION OF THE OPERATING SYSTEM. REGISTER A0 IS TO CONTAIN THE FOLLOWING INFORMATION WHEN THE EXECUTIVE IS ENTERED.

```
 35                          17                            00
--------------------------------------------------------------
: NUMBER TO BE REMOVED       : EXEC LINKING VALUE FOR POOL  :
--------------------------------------------------------------
```

THE EXEC REFERENCE IS:

```
        L     A0,PARAMETER WORD
        ER    CGET$
```

8.5.5. COMMUNICATIONS BUFFER POOL SETUP

THE CPOOL$ FUNCTION PROVIDES THE MEANS FOR THE INITIAL SETUP OF A CHARACTER ORIENTED BUFFER POOL WITH THE APPROPRIATE LINKAGE FOR USE BY THE REAL-TIME INTERRUPT PROCESSING ROUTINES. THE CPOOL$ FUNCTION REQUIRES A TWO WORD PACKET AS FOLLOWS TO DEFINE THE BUFFER AREA.

```
    35                  23       17                            00
   --------------------------------------------------------------
01 : BUFFER SIZE IN CHARACTERS   : ADDRESS OF FIRST BUFFER      :
   --------------------------------------------------------------
02 : NOT USED           : METHOD : LENGTH OF AREA TO BE USED    :
   --------------------------------------------------------------
```

METHOD IS THE FLAG TO INDICATE THE DESIRED BUFFER POOLING TECHNIQUE. THE ADDRESS OF THE PACKET MUST BE IN REGISTER A0 WHEN THE EXECUTIVE IS ENTERED.

THE EXEC REFERENCE IS:

```
        L     A0,PACKET ADDRESS
        ER    CPOOL$
```

8.5.6. DIRECTORY ITEM RETRIEVAL

THE MASTER DIRECTORY ITEMS FOR CATALOGUED FILES CURRENTLY ASSIGNED TO THE REFERENCING RUN CAN BE RETRIEVED VIA THE EXECUTIVE RETURN DITEM$. A COMPLETE DESCRIPTION OF THIS LINKAGE IS GIVEN IN THE FILE REFERENCING SECTION OF THE FILE CONTROL SYSTEM CHAPTER.

8.5.7. FACILITY DETERMINATION

THE FACIL$ FUNCTION PROVIDES THE MEANS FOR DETERMINING THE HARDWARE DESCRIPTION OF A PARTICULAR NAMED FILE. THE A0 REGISTER IS LOADED WITH THE ADDRESS OF A PACKET WITH THE FIRST TWO WORDS CONTAINING THE INTERNAL NAME OF THE FILE OF CONCERN. THE REMAINDER OF THE PACKET IS SUPPLIED BY THE REQUEST. COMPLETE DETAILS ARE GIVEN IN THE FILE CONTROL SYSTEM CHAPTER.

THE EXEC REFERENCE IS:

```
        L     A0,PACKET ADDRESS
        ER    FACIL$
```

8.5.8. TAPE REEL NUMBER SET INITIALIZATION

THE EXECUTIVE RETURN TINTL$ CAN BE USED TO RESET THE REEL NUMBER INDEX FOR A SPECIFIED FILE TO THE FIRST REEL AND CAUSE LOAD MESSAGES TO BE ISSUED ACCORDING TO

THE TYPE OF ASSIGNMENT (ONE OR TWO UNITS) AND CURRENT STATUS OF THE ASSIGNMENT. COMPLETE DETAIL FOR THIS FUNCTION IS GIVEN IN THE FILE CONTROL SYSTEM-FILE REFERENCING SECTION.

8.5.9. TAPE SWAPPING

DURING THE COURSE OF A RUN THE USER CAN REQUEST RELOADING OF A UNIT BY THE CALLING SEQUENCE:

```
LA A0,PKT ADDR
ER TSWAP$
```

WHERE THE PACKET ADDRESSED BY A0 IS A TWO-WORD PACKET CONTAINING THE INTERNAL FILE NAME OF THE ASSIGNMENT TO BE SWAPPED. THE EXECUTIVE WILL REWIND THE CLOSED-OUT REEL WITH INTERLOCK IF NOT ALREADY DONE, SWITCH UNITS IF TWO DRIVES ARE INVOLVED IN THE ASSIGNMENT AND REQUEST MOUNTING OF THE NEXT REEL OF THE FILE. IF THE ASSIGNMENT HAS TWO PHYSICAL UNITS INVOLVED, THE TSWAP$ ROUTINE WILL REQUEST MOUNTING AHEAD OF TIME FOR THE ALTERNATE UNIT. THE REEL NUMBER RESPONSE TO THE MOUNT MESSAGE IS NOT CONSIDERED FOR CATALOGUING UNTIL A SWAP IS MADE ONTO A UNIT; I.E., THE OPERATOR CAN RESPOND TO THE MOUNT MESSAGE FOR UNIT 2 ANY TIME WHILE UNIT 1 IS BEING REFERENCED, BUT IF A SWAP TO UNIT 2 DOES NOT OCCUR THE RESPONSE IS DISREGARDED. COMPLETE MOUNTING INSTRUCTIONS ARE GIVEN IN THE FILE CONTROL SYSTEM CHAPTER.

RETURN TO THE USER AFTER REFERENCE TO TSWAP$ IS IMMEDIATE AND IN LINE. THE A0 REGISTER IS SET NEGATIVE UPON RETURN IF THE FILE NAME IS NOT CURRENTLY ASSIGNED AS MAGNETIC TAPE. THIS IS THE ONLY ERROR CONDITION.

IT IS ESSENTIAL THAT THE TSWAP$ ROUTINE BE REFERENCED IF THE FILE IS TO BE CATALOGUED AND THE REEL NUMBERS ARE DESIRED IN THE DIRECTORY. OTHERWISE ONLY THE FIRST REEL NUMBER IS ENTERED IN THE MASTER DIRECTORY ITEM AND, EXCEPT WHEN NUMBERS APPEAR ON THE ASG STATEMENT, NO KNOWLEDGE IS AVAILABLE IN THE SYSTEM OF THE EXISTENCE OF MORE THAN ONE REEL (TWO IF TWO UNITS ARE ASSIGNED).

8.6. REAL-TIME

THE RT$ FUNCTION UPGRADES THE PROGRAM STATUS FROM NORMAL TO REAL-TIME IF THE ACCOUNT NUMBER WILL PERMIT SUCH ACTION. THIS FUNCTION ALSO PROVIDES A REAL-TIME ACTIVITY A MEANS OF CHANGING ITS SWITCHING PRIORITY LEVEL WITHIN THE REAL-TIME CLASS. THE USER MUST SUBMIT A SWITCHING LEVEL PRIORITY NUMBER (1-35) IN CR, A0. ACTIVITIES WHICH INITIATE FUNCTIONS ON COMMUNICATION (ESI) EQUIPMENT, MUST BE REGISTERED AS REAL-TIME IN ORDER TO RECEIVE THE PROPER RESPONSE. REAL-TIME PRIVILEGES INCLUDE TOP PRIORITY ON THE SWITCHING CYCLE, I/O INITIATION, AND INTERRUPT HANDLING. A PROGRAM IN THE REAL-TIME MODE WILL BE PREVENTED FROM ENTERING THE SWAP CYCLE.

THE NRT$ FUNCTION PROVIDES THE REAL-TIME ACTIVITY A MEANS OF ENTERING THE BATCH STATUS. THE REAL-TIME ACTIVITY COUNT IS DECREMENTED WHEN ALL ACTIVITIES FOR A PROGRAM HAVE ENTERED THE NORMAL BATCH MODE.

THE EXECUTIVE REFERENCES ARE:

```
ER   RT$
```

AND,

ER NT$ THE FOLLOWING ERROR CODES APPLY AND THE ACTIVITY IS TERMINATED.

0406 (RT) THE REAL-TIME PRIORITY IS NOT PERMITTED UNDER THE ACCOUNT NUMBER USED.

0412 (NRT) REQUESTED NRT$ FROM AN ACTIVITY WHICH IS NON REAL-TIME.

8.7. REENTRANT ROUTINES

A REENTRANT ROUTINE IS A ROUTINE CAPABLE OF PROCESSING REQUESTS FROM SEVERAL WORKER PROGRAMS CONCURRENTLY WITHOUT ANY INTERACTION BETWEEN THE REQUESTORS. THAT IS, THE REENTRANT ROUTINES PRESENT AN INTERFACE TO EACH REQUESTING PROGRAM SO AS TO APPEAR TO BE SERVICING ONLY THE REQUESTOR IN CONTROL AT ANY GIVEN POINT IN TIME. TO ACCOMPLISH THIS, A REENTRANT ROUTINE MAY CONTAIN ONLY AN IBANK AREA AND MAY STORE ONLY WITHIN THE DBANK OF THE WORKER PROGRAM.

TWO ENTRANCES ARE PROVIDED INTO THE EXECUTIVE FOR REENTRANT ROUTINE CONTROL.

8.7.1. REENTRANT ENTRY

THE EXECUTIVE REQUEST 'LINK$' PROVIDES THE WORKER PROGRAM WITH THE MEANS OF LINKING TO A REENTRANT ROUTINE. WHEN THE EXECUTIVE IS ENTERED, REGISTER A0 MUST CONTAIN THE NAME OF THE REENTRANT ROUTINE. IF THE ROUTINE IS NOT IN CORE, IT IS LOADED, AND THE PROPER LIMITS ARE CONSTRUCTED FOR THE ROUTINE WHEN USED FOR THIS PARTICULAR WORKER PROGRAM. THE IBANK OF THE REENTRANT ROUTINE AND THE DBANK OF THE WORKER PROGRAM ARE USED IN SETTING THE BASE REGISTER FOR THE REENTRANT ROUTINE CONTROL.

ALL REENTRANT ROUTINES WHICH ARE TO BE AVAILABLE TO THE WORKER PROGRAM BY THE EXECUTIVE REQUEST 'LINK$', MUST BE IDENTIFIED AT SYSTEM GENERATION TIME. THE EXECUTIVE MAINTAINS A RECORD OF ALL REENTRANT ROUTINES IN USE AND ONCE A REENTRANT ROUTINE IS LOADED, IT WILL REMAIN IN CORE EXCEPT FOR HIGHER PRIORITY SPACE REQUIREMENTS.

THE EXEC REFERENCE IS:

```
L     A0,REENTRANT ROUTINE NAME
ER    LINK$
```

8.7.2. REENTRANT RETURN

THE EXECUTIVE REQUEST, ULINK$, PROVIDES THE REENTRANT ROUTINE THE MEANS OF RETURNING CONTROL TO THE USER PROGRAM. THE BASE REGISTERS WILL BE ADJUSTED TO SATISFY THE WORKER PROGRAMS I-BANK AREA IN PLACE OF THE REENTRANT ROUTINES I-BANK. THE LOCKIN AREA WILL BE ALTERED TO INCLUDE THE WORKER PROGRAMS I-BANK AREA.

THE EXEC REFERENCE IS:

```
ER    ULINK$
```

8.8. DYNAMIC FACILITIES

CERTAIN FACILITIES MAY BE REQUESTED OR RELEASED BY AN EXECUTING PROGRAM THROUGH EXECUTIVE REQUEST FUNCTIONS. PRIMARILY, THESE FUNCTIONS ARE:

```
MCORE$ - REQUEST ADDITIONAL CORE
LCORE$ - RELEASE UNNEEDED CORE
CSF$   - GENERALIZED CONTROL STATEMENT ENTRANCE
```

THE CSF$ FUNCTION MAY BE UTILIZED TO LINK TO THE CONTROL STATEMENT INTERPRETER FOR EXECUTION TIME REQUESTS. THE CSF$ FUNCTION DESCRIPTION SHOULD BE REFERENCED FOR A COMPLETE DEFINITION OF THE LEGAL OPERATIONS THROUGH THE LINKAGE.

8.8.1. REQUEST ADDITIONAL CORE

A WORKER PROGRAM MAY REQUEST ADDITIONAL CORE BY LOADING THE HIGHEST RELATIVE ADDRESS REQUIRED (EITHER IBANK OR DBANK) IN REGISTER A0 AND ENTERING THE EXECUTIVE VIA 'MCORE$'. ADDITIONAL IBANK REQUESTS ARE RESTRICTED TO PREVIOUSLY RELEASED IBANK

AREAS. CONTROL IS RETURNED TO THE WORKER PROGRAM WHEN THE REQUEST HAS BEEN SATISFIED. IF THE REQUEST CANNOT BE SATISFIED, AN 'ERROR MODE' CONDITION WILL RESULT. POSSIBLE ERRORS ARE A REQUEST MAKING THE IBANK GREATER THAN 65536, OR AN EXTENSION WHICH REQUIRES MORE SPACE THAN PHYSICALLY AVAILABLE.

IF THE ADDITIONAL CORE IS NOT IMMEDIATELY AVAILABLE, THE REQUESTING PROGRAM OR SUBPROGRAM IS SUSPENDED UNTIL THE AREA IS AVAILABLE. THIS COULD REQUIRE WAITING FOR THE COMPLETION OF SEVERAL PROGRAMS. IF THE REQUESTOR IS A REAL-TIME PROGRAM, OTHER PROGRAMS WILL BE SUSPENDED IN ORDER TO SATISFY THE REQUEST. REAL-TIME PROGRAMS SHOULD NORMALLY REGISTER UNIQUE ACTIVITIES FOR MAKING CORE REQUESTS.

THE EXEC REFERENCE IS:

```
L     A0,HIGHEST ADDRESS NEEDED
ER    MCORE$
```

8.8.2. RELEASE OF UNNEEDED CORE

CORE AREAS NO LONGER NEEDED BY THE WORKER PROGRAM MAY BE RETURNED TO THE AVAILABLE POOL BY LOADING REGISTER A0 WITH THE MINIMUM ADDRESS REQUIRED IN EITHER THE IBANK OR DBANK AND ENTERING THE EXECUTIVE VIA 'LCORE$'. CONTROL WILL BE RETURNED TO THE WORKER PROGRAM WHEN THE REQUEST HAS BEEN SATISFIED.

THE EXEC REFERENCE IS:

```
L     A0,MINIMUM ADDRESS REQUIRED
ER    LCORE$
```

8.8.3. CONTROL STATEMENT FORMAT REQUESTS-CSF$

THE EXEC FUNCTION, CSF$, PROVIDES THE USER WITH A MEANS OF SUBMITTING AN EXECUTIVE CONTROL STATEMENT IMAGE FOR INTERPRETATION AND PROCESSING. THE IMAGE SUBMITTED MUST CONTAIN A CHARACTER STRING THAT IS IDENTICAL TO WHAT IT WOULD BE IF THE FUNCTION WERE SUBMITTED AS AN EXECUTIVE CONTROL STATEMENT IN THE INPUT STREAM. THE IMAGE IS ASSUMED TO BE -ONE- LINE OF VARIABLE LENGTH, AND THE CONTINUATION CHARACTER IS NOT USED, AS IT WAS IN THE CASE OF STREAM INPUT WHERE MULTIPLE LINES WERE ALLOWED. THE COMMAND PORTION OF THE STRING DICTATES THE FUNCTION TO BE CARRIED OUT BY CSF (PRIOR TO RETURN OF CONTROL), THE SAME AS IF THE CARD WERE IN THE INPUT STREAM. THE READER IS REFERRED TO THE SECTION ENTITLED 'EXECUTIVE CONTROL LANGUAGE' FOR THE FORMAT OF INPUT CONTROL STATEMENTS.

THE CONTROL REGISTER A0 MUST BE LOADED SUCH THAT THE RIGHT HALF CONTAINS THE IMAGE ADDRESS AND THE LEFT HALF CONTAINS THE CARD IMAGE LENGTH(NO. OF COMPUTER WORDS). IF THE LEFT HALF IS ZERO, THE WORD LENGTH OF THE IMAGE IS ASSUMED TO BE 14. THE IMAGE MUST BE CONTAINED WITHIN 40 WORDS OR LESS. PRESENTING CSF$ WITH AN IMAGE LENGTH GREATER THAN 40 WORDS WILL RESULT IN THE TERMINATION OF THE ACTIVITY. IN ALL CASES, SCANNING OF THE IMAGE WILL BE TERMINATED UPON ENCOUNTERING THE LAST ALLOWABLE SPECIFICATION FIELD OR BY THE TERMINATOR (SPACE-PERIOD-SPACE).

CONTROL REGISTER A0 WILL CONTAIN THE STATUS CODE WHEN CONTROL IS RETURNED. IF THE STATUS CODE IS 0440, 0441, 0442, OR 0443, THE ACTIVITY IS TERMINATED AND THE STATUS CODE IS NOT OBSERVED BY THE EXECUTING PROGRAM UNLESS AN ERROR TERMINATION INTERRUPT ACTIVITY IS CALLED FOR. THE STATUS CODES ARE AS FOLLOWS:

OCTAL ERROR CODE	DEFINITIONS
0440	A SYNTAX ERROR WAS DETECTED WHILE INTERPRETING THE IMAGE RESULT - ERROR TERMINATION
0441	THE IMAGE LENGTH WAS FOUND TO BE GREATER THAN 40 WORDS RESULT - ERROR TERMINATION
0442	THE IMAGE CONTAINED AN ILLEGAL COMMAND RESULT - ERROR TERMINATION

0443 THE ADDRESS OF THE IMAGE IS OUTSIDE OF THE ACTIVITIES CORE LIMITS
RESULT - ERROR TERMINATION

STATUS CODES PRESENT IN A0 UPON RETURN TO USER

BIT SETTING .	MEANING
35	REQUEST NOT ACCEPTED-EXAMINE REST OF BIT SETTINGS AS TO WHY.
34	FIELD ERROR IN STATEMENT OTHER THAN SYNTAX
33	FILE NAME HAS ALREADY BEEN ASSIGNED TO THIS RUN
32	FILE HAS PREVIOUSLY BEEN CATALOGUED
31	EQUIPMENT TYPE ON @ASG STATEMENT IS NOT COMPATIBLE WITH CATALOGUED EQUIPMENT TYPE
30	NAME ON @ASG STATEMENT FOUND IN ATTACHED NAME LIST
29	12 CHARACTER NAME IS NOT UNIQUE (THAT PORTION USED FOR INTERNAL I/O)
28	X OPTION SPECIFIED ON FILE ALREADY ASSIGNED TO THIS RUN
27	READ KEY INCORRECT
26	WRITE KEY INCORRECT
25	WRITE KEY EXISTS IN DIRECTORY, NONE SPECIFIED ON @ASG STATEMENT (READ MODE ONLY - WARNING)
24	READ KEY EXISTS IN DIRECTORY, NONE SPECIFIED ON @ASG STATEMENT (WRITE MODE ONLY - WARNING)
23	READ KEY EXISTS ON @ASG STATEMENT, NONE EXISTS IN DIRECTORY (INFORMATION - WARNING)
22	WRITE KEY EXISTS ON @ASG STATEMENT, NONE EXISTS IN DIRECTORY (INFORMATION - WARNING)
21	'A' OPTION SPECIFIED ON @ASG STATEMENT AND FILE NAME NOT FOUND IN DIRECTORY
20	INVALID REEL NUMBER ON @ASG STATEMENT FOR CATALOGUED FILE
19	MASS STORAGE FILE HAS BEEN ROLLED OUT
18	REQUEST ON WAIT STATUS FOR FACILITIES
17	UNCATALOGUED FILE OPTIONS 'CURPW' WERE SPECIFIED FOR CATALOGUED FILE
16	THIS FILE HAS BEEN ASSIGNED EXCLUSIVELY TO SOME OTHER RUN
15	FIND MADE IN DIRECTORY AND FILE ALREADY ASSIGNED TO SOME OTHER RUN

14	FIND MADE IN DIRECTORY AND NOT ASSIGNED
13	PROJECT I.D. INCORRECT FOR PRIVATE FILE
12	EQUIPMENT TYPE IS TAPE
11	READ ONLY FILE, CATALOGUED WITH 'R' OPTION
10	WRITE ONLY FILE, CATALOGUED WITH 'W' OPTION
9	EQUIPMENT IS DOWN

STATUS CODES PRESENT IN A0 UPON RETURN TO USER FOR @START COMMAND:

0	REQUEST PROCESSED NORMALLY
1	IMPROPER RUN STREAM IN FILE, REQUEST REJECTED
2	FILE UNOBTAINABLE, REQUEST REJECTED
3	ELEMENT UNOBTAINABLE, REQUEST REJECTED
4	FILE NAME NOT SPECIFIED, REQUEST REJECTED

THE EXEC REFERENCE IS CSF$, WITH THE IMAGE ADDRESS AND LENGTH(OPTIONAL) IN CONTROL REGISTER, A0.

ONLY A RESTRICTED SET OF COMMANDS(FUNCTIONS) ARE ALLOWED VIA THE CSF$ REFERENCE. THESE ARE LISTED BELOW.

@START	SCHEDULE AN INDEPENDENT RUN.
@ASG	ASSIGN A FILE AND ITS I/O DEVICE
@USE	ASSOCIATED INTERNAL FILE NAME WITH EXTERNAL FILE NAME.
@CAT	CATALOGUE A FILE
@FREE	DEASSIGN A FILE AND ITS I/O DEVICE
@LOG	MESSAGE TO LOG FILE
@QUAL	FILE QUALIFICATION
@SYM	QUEUE FILES
@ADD	INCLUDE FILE OR ELEMENT IN CONTROL STREAM
@BRKPT	BREAKPOINT SYMBIONT OUTPUT FILES

AN EXAMPLE OF THE USE OF THE @ASG COMMAND VIA THE CSF$ REFERENCE IS:

A) ASSUME THE USER WISHES TO ASSIGN A TEMPORARY FASTRAND SCRATCH FILE, NAME IT 'FILE', AND RESERVE 2 GRANULES.

B) THE IMAGE TO SUBMIT COULD BE

```
     --------
IADD :@ASG,T:
     : FILE,:
     :F,2 . :
     --------
```

WHICH IS A THREE WORD IMAGE AT ADDRESS 'IADD'. IN THIS CASE, THE PERIOD STOPS THE SCANNING. IF FOUND IN THE CONTROL STREAM, THIS STATEMENT WOULD APPEAR AS

```
@ASG,T FILE,F/2.
```

C) TO CAUSE THE ASSIGNMENT, CONTROL REGISTER A0 WOULD BE LOADED WITH 'IADD' PRIOR TO THE ER INSTRUCTION

REFERENCING CSF$.

8.9. INPUT/OUTPUT

BECAUSE OF THE COMPLEXITIES AND THE MULTITUDE OF OPTIONS IN REQUESTING INPUT /OUTPUT FUNCTIONS, NO ATTEMPT TO DEFINE THEM WILL BE MADE IN THIS CHAPTER. DETAILED INFORMATION ON THE REQUIRED PROCEDURES FOR REQUESTING I/O CAN BE FOUND IN THE CHAPTER ON INPUT/OUTPUT DEVICE HANDLERS.

8.10. CONSOLE COMMUNICATIONS

A USER PROGRAM MAY INITIATE CONSOLE COMMUNICATIONS BY ENTERING THE EXECUTIVE THROUGH THE 'COM$' EXECUTIVE REQUEST FUNCTION. TWO TYPES OF CONSOLE COMMUNICATIONS MAY BE INITIATED - TYPE OR TYPE AND READ. THE TYPE FUNCTION IS USED WHEN NO OPERATOR REPLY IS REQUIRED. THE TYPE AND READ FUNCTION REQUIRES AN OPERATOR RESPONSE.

TO ACTIVATE THE CONSOLE, THE WORKER PROGRAM MUST SUPPLY A CONTROL PACKET OF THE FORM:

```
35      30        24        18                       00
----------------------------------------------------------
:    S   :    C    :         :          NCI            :
----------------------------------------------------------
:       MCC                  :      TYPE ADDRESS       :
----------------------------------------------------------
:       ICC                  :      READ ADDRESS       :
----------------------------------------------------------
```

WHERE:

S	STATUS CODE
C	CONSOLE CLASSIFICATION NUMBER
NCI	THE NUMBER OF CHARACTERS RECEIVED AS INPUT ON A TYPE AND READ FUNCTION
MCC	THE NUMBER OF CHARACTERS IN THE OUTPUT MESSAGE TEXT
TYPE ADDRESS	THE ADDRESS OF THE FIRST WORD OF THE OUTPUT MESSAGE TEXT
ICC	THE NUMBER OF CHARACTERS EXPECTED AS INPUT ON A TYPE AND READ FUNCTION
READ ADDRESS	THE ADDRESS OF THE FIRST WORD OF THE BUFFER TO RECEIVE THE OPERATOR RESPONSE ON A TYPE AND READ FUNCTION

WHEN THE LAST WORD IN THE PACKET IS EQUAL TO ZERO, A TYPE FUNCTION IS INITIATED. A TYPE AND READ IS SPECIFIED BY ENTERING THE REQUIRED FIELDS IN THE LAST WORD. THE ADDRESS OF THE PACKET MUST BE IN REGISTER A0 WHEN THE EXECUTIVE IS ENTERED.

THE EXEC REFERENCE IS:

```
L    A0,PACKET ADDRESS
ER   COM$
```

ADDITIONAL INFORMATION ON THE USE OF THE CONSOLE CAN BE FOUND IN THE CHAPTER ENTITLED 'OPERATOR COMMUNICATIONS'.

8.11. SYMBIONT CONTROL

SIX FUNCTIONS ARE AVAILABLE FOR CONTROLLING THE SYMBIONT INTERFACE ROUTINES:

```
PRINT$ - SYSTEM INITIATED PRINT FILE
PRNTA$ - ALTERNATE PRINT FILE
PUNCH$ - SYSTEM INITIATED PUNCH FILE
PNCHA$ - ALTERNATE PUNCH FILE
READ$  - ENTRY FOR READING CONTROL STREAM
READA$ - ALTERNATE READ ENTRY FOR SYMBIONT CONTROLLED
         DEVICES
```

DETAILED INFORMATION ON THE CONSTRUCTION OF THE CONTROL PACKETS AND THE CODING CONVENTIONS REQUIRED FOR CALLING THESE FUNCTIONS CAN BE FOUND IN THE CHAPTER ON SYMBIONTS.

8.12. CHECKPOINT/RESTART

THE FOLLOWING IS A SUMMARY DESCRIPTION OF THE CHECKPOINT/RESTART FUNCTIONS. A DETAILED DESCRIPTION CAN BE FOUND IN THE CHECKPOINT/RESTART PARAGRAPH IN THE SUPERVISOR CHAPTER.

8.12.1. CHECKPOINT

WHEN A WORKER CALLS FOR A CHECKPOINT, ALL INFORMATION REGARDING THE STATUS OF HIS RUN IS SAVED ON A SPECIFIED TAPE FILE OR ON FASTRAND (CALLED BY FILENAME 'FS'). THIS RUN CAN BE REINITIATED FROM THIS POINT AT A LATER TIME. IF A 'P' OPTION IS SPECIFIED, INFORMATION REGARDING THIS CHECKPOINT WILL BE TYPED ON THE CONSOLE. IF A 'T' OPTION IS SPECIFIED, THE RUN WILL BE TERMINATED; IF NOT SPECIFIED, CONTROL WILL BE RETURNED TO THE WORKER AFTER CHECKPOINT.

EXEC REFERENCE IS MADE BY THE PROCEDURE CALL:

```
    C$KPT     'FILENAME','OPTIONS'
```

WHICH WILL GENERATE THE ER REQUEST. THE PROCEDURE WILL LOAD A0 WITH AN ENTRY CODE.

```
     L,U A0,PKT
     ER  CKPT$
```

WHERE PKT IS A THREE WORD PACKET GENERATED UNDER LOCATION COUNTER $(30).

```
          -------------------------------------
WORD 1    :  0  :  0  :  0  :  0  : 'P' : 'T' :  *
          -------------------------------------
WORD 2    :                                   :
          :              FILE NAME            :
          -------------------------------------
```

```
     UPON RETURN FROM CKPT, WORD 1 WILL BE SET AS FOLLOWS:
   S1 - =0 IF CHECKPOINT WAS TAKEN.
        = ERROR CODE IF CHECKPOINT ENDED IN ERROR AND COULD NOT BE TAKEN.
```

8.12.2. RESTART

A WORKER MAY CALL FOR A RESTART OF A RUN IF HE HAS TAKEN A RESTART PREVIOUSLY. IF A RUN RESTARTS ITSELF, IT WILL BE TERMINATED AND REINITIATED AT THE CHECKPOINT SPECIFIED. IF IT RESTARTS ANOTHER RUN, A START SITUATION WILL OCCUR.

EXEC REFERENCE IS MADE BY THE PROCEDURE CALL:

```
     R$STRT    'FILENAME',CKPT#,'P'
```

WHICH WILL GENERATE THE ER REQUEST.

```
     L,U    A0,PKT
     ER     RSTRT$
```

WHERE PKT IS A FOUR WORD PACKET GENERATED UNDER LOCATION COUNTER $(30).

```
         ------------------------------------
WORD 1  :                         :    P     : *
         ------------------------------------
WORD 2  :              FILE NAME             :
WORD 3  :                                    :
         ------------------------------------
WORD 4  :                          CKPT #    :
         ------------------------------------
```

* WHERE P IS ZERO FILLED IF NOT SPECIFIED.
RETURN FROM A RSTRT CALL WILL BE MADE TO THE LOCATION FOLLOWING THE CHECKPOINT CALL.

8.13. MISCELLANEOUS

8.13.1. UNLOCK

IN ORDER TO ALLOW AN INTERRUPT ACTIVITY TO REDUCE ITS PRIORITY IN THE ABSENCE OF ANY OTHER ER TO BE DONE THE ER UNLCK$ CAN BE USED. THE PRIMARY INTENT OF THE INTERRUPT ACTIVITY IS TO PLACE THIS ACTIVITY AT ITS PROPER LEVEL. PROLONGED EXECUTION AT THIS HIGH LEVEL NEGATES ANY BENEFIT HENCE THE USER SHOULD USE THE UNLCK$ FUNCTION BEFORE GOING INTO A LENGTHY COMPUTE LOOP.

8.13.2. OPTION LETTER RECOVERY

THE OPTION LETTERS ON THE @XQT CONTROL STATEMENT INITIATING A PROGRAM ARE MADE AVAILABLE BY CALLING EXECUTIVE REQUEST FUNCTION 'OPT$'. WHEN CONTROL IS RETURNED, THE OPTION LETTERS WILL BE ENCODED INTO REGISTER A0; I.E., IF THE LETTER Z IS PRESENT, BIT 0 WOULD BE SET; IF THE LETTERS A AND B ARE PRESENT, BITS 24 AND 25 WOULD BE SET.

THE EXEC REFERENCE IS:

```
     ER    OPT$
```

8.13.3. TIME AND DATE

THE EXECUTIVE FUNCTION 'DATE$' SUPPLIES THE USER WITH THE CURRENT DATE IN FIELDATA DECIMAL CODE IN REGISTER A0. THE CURRENT TIME IN FIELDATA DECIMAL CODE IS SUPPLIED IN REGISTER A1. THE FORMAT OF REGISTER A0 IS:

```
35               24 23                  12 11                  00
-----------------------------------------------------------------
:         A        :          B          :          C           :
-----------------------------------------------------------------
```

WHERE,

A - MONTH (1=JAN, 2=FEB, ETC.)
B - DAY OF THE MONTH
C - YEAR (LAST TWO DIGITS OF THE YEAR)

THE FORMAT OF REGISTER A1 IS:

```
35                  24 23                  12 11                  00
----------------------------------------------------------------------
:         D          :         E          :         F            :
----------------------------------------------------------------------
```

WHERE,

D - HOURS (00 - 24)
E - MINUTES (00 - 60)
F - SECONDS (00 - 60)

THE EXECUTIVE FUNCTION 'TDATE$' SUPPLIES THE USER WITH THE DATE AND TIME IN A0 IN THE FOLLOWING FORMAT:

```
35         29        23        17                          0
-------------------------------------------------------------
:   A     :    B    :   C    :             D               :
-------------------------------------------------------------
```

WHERE,

A = MONTH
B = DAY
C = YEAR (MODULO 1964)
D = TIME IN SECONDS FROM MIDNIGHT

THE EXECUTIVE FUNCTION 'TIME$' SUPPLIES THE USER WITH THE TIME OF DAY IN MILLISECONDS IN REGISTER A0.

8.13.4. READ PROGRAM CONTROL TABLE

INFORMATION STORED IN THE PROGRAM CONTROL TABLE COLLECTED BY THE EXECUTIVE FOR RUN CONTROL IS AVAILBLE WITHIN A PROGRAM USING THE REQUEST, PCT$. THE FORM IS:

```
L     A0,(N,A)
ER    PCT$
```

WHERE 'N' WORDS ARE MOVED BY THE EXECUTIVE TO THE LOCATION 'A' IN THE PROGRAM. THE FIRST WORD CONTAINS THE RUNID IN FIELDATA CODE. CONSULT SYSTEMS MANUAL FOR OTHER AVAILABLE DATA.

NOTE: THE MAXIMUM 'N' PERMITTED IS LIMITED TO ALL DATA BEFORE
THE FILE NAME LIST. FILE NAMES,USE NAMES AND OTHER INFORMATION
WILL NOT BE AVAILABLE TO THE PROGRAMMER.

8.13.5. SNAPSHOT DUMPS

A SNAPSHOT DUMP OF THE CONTENTS OF SELECTED CONTROL REGISTERS AND MAIN STORAGE CAN BE OBTAINED USING THE REQUEST 'SNAP$'. THE EXECUTIVE REQUEST IS CODED:

```
L,U    A0,PKT ADDR
ER     SNAP$
```

WHERE THE PACKET HAS THE FORM:

```
 35    32                      17                         00

-----------------------------------------------------------
:                      SNAPSHOT IDENTIFIER                :
-----------------------------------------------------------
:X:A:R:         LENGTH          :           ADDRESS       :
-----------------------------------------------------------
:                    FORMER CONTENTS OF A0                :
-----------------------------------------------------------
```

THE SNAPSHOT IDENTIFIER IS A SIX CHARACTER FIELDATA WORD USED TO IDENTIFY THE SNAPSHOT. THE TIME AND DATE WILL ALSO BE PRINTED ON THE IDENTIFICATION LINE.

'X','A', AND 'R' ARE ONE BIT FIELDS WHICH WHEN SET TO ONE CAUSE THE DESIGNATED CONTROL REGISTERS TO BE PRINTED. 'LENGTH' AND 'ADDRESS' DEFINE A MAIN STORAGE AREA TO BE DUMPED.

UPON RETURN FROM THE REQUEST, THE CONTENTS OF A0 WILL BE EQUAL

TO PKT ADDR+2. THE CONTENTS OF THIS CELL WILL ALSO BE USED FOR A0 IF THE SNAP REQUEST SPECIFIES AN A-REGISTER DUMP OR AN X-REGISTER DUMP.

8.13.6. THE CONDITION WORD

8.13.6.1. SET

THE EXEC FUNCTION, SETC$, PLACES(SETS) THE CONTENTS OF THE LOWER THIRD(BITS 11-00) OF CONTROL REGISTER A0 IN THE CORRESPONDING THIRD OF THE RUN 'CONDITION' WORD. THIS FUNCTION IS SIMILAR TO THAT OF THE @SETC CONTROL STATEMENT, EXCEPT THAT THERE IS NO CHOICE OF J-DESIGNATORS.

8.13.6.2. RETRIEVE

THE EXECUTIVE FUNCTION 'COND$' RETRIEVES THE CONDITION WORD AND MAKES IT AVAILABLE TO THE USER IN REGISTER A0. THIS FUNCTION IS SOMEWHAT SIMILAR TO THE @TEST CONTROL STATEMENT IN THAT A DECISION CAN BE MADE ON THE BASIS OF WHAT IS FOUND IN THE CONDITION WORD. THIS DOES NOT IMPLY, HOWEVER, THAT THE 'COND$' FUNCTION ITSELF PROVIDES ANY TESTING CAPABILITIES.

8.13.7. SEGMENT LOADING

THE LOAD$ FUNCTION PROVIDES THE WORKER PROGRAM A MEANS OF LOADING SEGMENTS OF ITSELF. THE SEGMENT NAME IS LOADED IN CR A0 AND THE RETURN ADDRESS IS LOADED IN CR A1 BEFORE THE REQUEST IS MODE. IN ADDITION, IF A0 IS POSITIVE, THE CORE WITH WHICH THE SEGMENT WILL BE LOADED IS ZERO FILLED PRIOR TO LOADING. IF A1 IS ZERO, CONTROL IS RETURNED TO THE NEXT INSTRUCTION OF THE PROGRAM. IF A RELOCATABLE SEGMENT IS TO BE LOADED, CR A2 MUST BE LOADED WITH THE ADDRESS WHERE THE SEGMENT IS TO BE LOADED

THE EXECUTIVE REQUEST IS CODED:

```
L,U    A0,SEGMENT NAME
L,U    A1,RETURN ADDRESS
ER     LOAD$
```

8.14. SUMMARY

TABLE 8.1 SUMMARIZES THE EXECUTIVE REQUESTS RECOGNIZED BY THE EXECUTIVE SYSTEM. THE TABLE CLASSIFIES ER FUNCTIONS AS TO WHETHER THEY ARE IMMEDIATE, ASYNCHRONOUS, OR SYNCHRONOUS.

FUNCTION	TYPE	DESCRIPTION	CROSS REFERENCE
ABORT$	I	ABORT EXIT	EXECUTIVE REQUEST FUNCTIONS

ACT$		ACTIVITY ACTIVATION	EXECUTIVE REQUEST FUNCTION.
ALTCL$	S	CLOSE ALTERNATE FILES	SYMBIONTS
ADLT$	I	ACTIVITY DELETION	EXECUTIVE REQUEST FUNCTIONS
ADACT$		COMPLETION ACTIVITIES	I/O DEVICE HANDLERS
AWAIT$	I	ACTIVITY WAIT	EXECUTIVE REQUEST FUNCTIONS
BJOIN$	I	BUFFER POOL EXPANSION	FILE CONTROL SYSTEM
BPOOL$	I	BUFFER POOL SETUP	FILE CONTROL SYSTEM
CADD$	I	COMMUNICATIONS BUFFER POOL ADDITION	FILE CONTROL SYSTEM
CEND$		PROGRAM CONTINGENCY	SUPERVISOR
BREAD$		BLOCK BUFFERING	DATA HANDLING
BRRED$		BLOCK BUFFERING	DATA HANDLING
BWRITE$		BLOCK BUFFERING	DATA HANDLING
BRWRT$		BLOCK BUFFERING	DATA HANDLING
BOPEN$		BLOCK BUFFERING	DATA HANDLING
BCLOR$		BLOCK BUFFERING	DATA HANDLING
BCLOF$		BLOCK BUFFERING	DATA HANDLING
BMARK$		BLOCK BUFFERING	DATA HANDLING
CGET$	I	COMMUNICATIONS BUFFER POOL REMOVAL	FILE CONTROL SYSTEM
CKPT$	S	CALL FOR CHECKPOINT	SUPERVISOR
COND$	I	RETRIEVE THE 'CONDITION WORD	EXECUTIVE REQUEST FUNCTIONS
CMD$	A	COMMUNICATION HANDLER, DIAL	INPUT/OUTPUT DEVICE HANDLERS
CMH$	A	COMMUNICATION HANDLER, HANGUP	INPUT/OUTPUT DEVICE HANDLERS
CMI$	A	COMMUNICATION HANDLER, INPUT	INPUT/OUTPUT DEVICE HANDLERS
CMO$	A	COMMUNICATION HANDLER, OUTPUT	INPUT/OUTPUT DEVICE HANDLERS
CMS$	I	COMMUNICATION HANDLER, INITIALIZE	INPUT/OUTPUT DEVICE HANDLERS
CMSA$	A	COMMUNICATION HANDLER, SEND AND ACKNOWLEDGE	INPUT/OUTPUT DEVICE HANDLERS
CMT$	I	COMMUNICATION HANDLER, TERMINATE	INPUT/OUTPUT DEVICE HANDLERS
CPOOL$	I	COMMUNICATIONS BUFFER POOL SETUP	FILE CONTROL SYSTEM
CSF$	S	CONTROL STATEMENT FORMAT ENTRY	EXECUTIVE REQUEST FUNCTIONS
COM$	S	CONSOLE COMMUNICATION ENTRANCE	OPERATOR COMMUNICATIONS
DACT$		ACTIVITY DEACTIVATION	EXECUTIVE REQUEST FUNCTION.
DATE$	I	REQUEST DATE	EXECUTIVE REQUEST FUNCTIONS
DITEM$	S	DIRECTORY ITEM RETRIEVED	FILE CONTROL SYSTEM
DREL$	I	DYNAMIC SEGMENT RELEASE	AUXILARY PROCESSORS
ERR$	I	ERROR EXIT	EXECUTIVE REQUEST FUNCTIONS
EXIT$	I	NORMAL EXIT	EXECUTIVE REQUEST FUNCTIONS
FACIL$	S	FACILITY DETERMINATION	FILE CONTROL SYSTEM
FORK$	I	REGISTER ACTIVITY	EXECUTIVE REQUEST FUNCTIONS
IALL$	I	INTERRUPT ADDRESS CHANGE	SUPERVISOR
IO$	A	INPUT/OUTPUT FUNCTION	INPUT/OUTPUT DEVICE HANDLERS
IOARB$	A	INPUT/OUTPUT FUNCTION	INPUT/OUTPUT DEVICE

		ON ARBITARY DEVICE	HANDLERS
IOAXI$	A	INPUT/OUTPUT FUNCTION ON ARBITARY DEVICE EXIT AND INTERRUPT ACTIVITY	INPUT/OUTPUT DEVICE HANDLERS
IOI$	A	INPUT/OUTPUT FUNCTION,	INPUT/OUTPUT DEVICE
NAME$		ACTIVITY NAMING	EXECUTIVE REQUEST FUNCTION.
		INTERRUPT ACTIVITY	HANDLERS
IOWI$	S	INPUT/OUTPUT FUNCTION, INTERRUPT ACTIVITY, WAIT	INPUT/OUTPUT DEVICE HANDLERS
IOW$	S	INPUT/OUTPUT FUNCTION AND WAIT FOR COMPLETION	INPUT/OUTPUT DEVICE HANDLERS
IOXI$	A	INPUT/OUTPUT FUNCTION, INTERRUPT ACTIVITY, EXIT	INPUT/OUTPUT DEVICE HANDLERS
IRST$	I	RESTART	SUPERVISOR
LCORE$	S	RELEASE CORE	EXECUTIVE REQUEST FUNCTIONS
LINK$	S	REQUESTS TO LINK TO REENTRANT ROUTINE	EXECUTIVE REQUEST FUNCTIONS
LOAD$	S	REQUEST FOR SEGMENT TO BE LOADED	AUXILIARY PROCESSORS
MCORE$	S	REQUEST ADDITIONAL CORE	EXECUTIVE REQUEST FUNCTIONS
NRT$	I	RELEASE REAL-TIME STATUS	EXECUTIVE REQUEST FUNCTIONS
OPT$	I	REQUEST OPTION LETTERS FROM XQT CARD	EXECUTIVE REQUEST FUNCTIONS
PCHCN$	S	PUNCH SYMBIONT CONTROL	SYMBIONTS
PCT$	I	PROGRAM CONTROL TABLE REFERENCE	EXECUTIVE REQUEST FUNCTIONS
PRINT$	S	SYSTEM INITIATED PRINT FILE ENTRY	SYMBIONTS
PRTCN$	S	PRINT SYMBIONT CONTROL	SYMBIONTS
PRNTA$	S	ALTERNATE PRINT FILE ENTRY	SYMBIONTS
PUNCH$	S	SYSTEM INITIATED PUNCH FILE ENTRY	SYMBIONTS
PNCHA$	S	ALTERNATE PUNCH FILE ENTRY	SYMBIONTS
READ$	S	ENTRY FOR READING CONTROL STREAM	SYMBIONTS
READA$	S	ALTERNATE READ ENTRY FOR SYMBIONT CONTROLLED DEVICES	SYMBIONTS
RSTRT$	S	CALL FOR RESTART DUMP	SUPERVISOR
RT$	I	REQUEST REAL-TIME STATUS	EXECUTIVE REQUEST FUNCTIONS
SETC$	I	SET A VALUE IN THE 'CONDITION' WORD	EXECUTIVE REQUEST FUNCTIONS
SNAP$	S	SNAPSHOT DUMP	EXECUTIVE REQUEST FUNCTIONS
TDATE$	I	REQUEST DATE/TIME	EXECUTIVE REQUEST FUNCTIONS
TFORK$	I	TIMED ACTIVITY REGISTRATION	EXECUTIVE REQUEST FUNCTIONS
TIME$	I	REQUEST TIME OF DAY IN MILLISECONDS	EXECUTIVE REQUEST FUNCTIONS
TINTL$	S	TAPE REEL NUMBER SET INITIALIZATION	EXECUTIVE REQUESTS
TSWAP$	S	TAPE SWAPPING	EXECUTIVE REQUEST FUNCTIONS
ULINK$	S	RETURN TO WORKER PROGRAM FROM REENTRANT ROUTINE	EXECUTIVE REQUEST FUNCTIONS
UNLCK$	I	REVERT TO NORMAL MODE	SUPERVISOR

			HANDLERS
WAIT$	S	WAIT FOR COMPLETION OF SPECIFIC I/O	INPUT/OUTPUT DEVICE HANDLERS
WANY$	S	WAIT FOR COMPLETION OF ANY I/O	INPUT/OUTPUT DEVICE HANDLERS

TABLE 8.2

9. SYMBIONTS

9.1. GENERAL DESCRIPTION

THE EXECUTIVE SYSTEM PROVIDES A COMPLEX OF ROUTINES CALLED SYMBIONTS WHICH INTERFACE THE PRIMARY UNIT RECORD EQUIPMENT OF THE 1108 WITH THE USER. THESE DEVICE ROUTINES ARE AVAILABLE FOR ALL STANDARD TERMINALS. ON-SITE DEVICES INCLUDE CARD READERS/PUNCHES, PRINTERS, PAPER TAPE READER/PUNCH AND 1004'S. REMOTE SITE TERMINALS ARE ACCOMODATED USING NORMAL COMMUNICATIONS SYSTEMS. DATA, TO AND FROM THESE DEVICES, IS BUFFERED USING MASS STORAGE TO PROVIDE AN EFFECTIVE LINKAGE TO THE ASYNCHRONOUS AND RELATIVELY SLOW DEVICES. THE ROUTINES WHICH INTERFACE DIRECTLY WITH THE USER ARE AVAILABLE THROUGH THE EXEC REQUESTS READ$, PRINT$ AND PUNCH$. FOR EACH THE USER SPECIFIES THE CORE AREA WHERE THE IMAGE IS, OR IS TO BE PLACED. A FILE IS AVAILABLE FOR ALL THREE PATHS, AND THE PROVISION IS MADE THROUGH DEFINITION BY THE USER OF OTHER FILES FOR THE CONSTRUCTION OF OTHER OUTPUTS AND INPUTS AS REQUIRED. AN OUTPUT CONTROL REQUEST IS ALSO AVAILABLE WHICH CAN BE USED TO PROVIDE CONTROL INFORMATION TO THE ROUTINES RELATED TO SPECIFIC OUTPUT DEVICES. THE OUTPUT CONTROL REQUEST ALSO PROVIDES THE ADDITIONAL FACILITY OF ADVISING THE OPERATOR OF THE OUTPUT DEVICE SYNCHRONIZED WITH THE ACTIVITY OF THE DEVICE. PROVISION ALSO EXISTS TO SKIP OR BACKSPACE OVER PORTIONS OF THE OUTPUT FILES.

THE READ$, PRINT$ AND PUNCH$ INTERFACE ROUTINES ARE CAPABLE OF MULTIPLE FILE OPERATIONS WITHIN ANY RUN. THE SYSTEM IS RESPONSIBLE FOR INITIATING ONE FILE FOR EACH INTERFACE ROUTINE, THE RUN INITIATING FILE(READ$), THE SYSTEM PRINT(PRINT$) AND PUNCH(PUNCH$) FILES. THESE ARE NECESSARILY DEFINED TO ALLOW THE VARIOUS EXEC 8 COMPONENTS TO RECEIVE INPUT AND TO DISPLAY OUTPUT. THESE INTERFACE ROUTINES ARE SUPPLEMENTED WITH ADDITIONAL ENTRANCES TO FACILITATE MULTIPLE FILE SYMBIONT INPUT/OUTPUT OPERATIONS.

DURING SYSTEM GENERATION, AN OUTPUT DEVICE OR CLASS OF DEVICES IS ASSOCIATED WITH EACH OF THE INPUT DEVICES (SEE SYSTEM GENERATION STATEMENTS - ASSOCIATED SYMBIONTS). THE LOCATION OF THE CONTROL STREAM INPUT FOR EACH RUN THEN DICTATES THE LOCATION OF THE PRIMARY PRINT OUTPUT DEVICE. THE @SYM STATEMENT PROVIDES A METHOD OF SELECTING AN ALTERNATE OUTPUT DEVICE FOR THE PRINT OR PUNCH FILES. IN GENERAL, ALL OF THE REMOTE TERMINALS WILL RECEIVE THEIR OUTPUT DIRECTLY, BUT WHERE VOLUME DICTATES OR A SUITABLE PRINTER IS NOT AVAILABLE, SOME OTHER REMOTE OUTPUT OR AN ON-SITE PRINTER COULD BE SELECTED.

INPUT TO THE SYSTEM IS SEPARATED ACCORDING TO @RUN AND @FILE STATEMENTS. EACH @RUN ENCOUNTERED AND ITS SUCCEEDING STATEMENTS ARE USED TO GENERATE A 'RUN FILE' TO BE SUBMITTED TO THE SCHEDULING ROUTINE. INFORMATION OF CONCERN TO THE COARSE SCHEDULER IS EXTRACTED FOR PRELIMINARY RUN SCHEDULING. THE INPUT SYMBIONTS MUST ALSO INTERPRET THE @ELT,D, @DATA, @END, @FILE AND @ENDF STATEMENTS TO PROVIDE UNRESTRICTED INPUT OF DATA FILES BETWEEN THESE TWO STATEMENTS.

ALL FILES CREATED OR PROCESSED BY THE SYMBIONT COMPLEX ARE IN THE SYSTEM DATA FILE (SDF) FORMAT. ANY FILE EXISTING IN THIS FORMAT CAN BE DIRECTLY PROCESSED WITH THE INPUT INTERFACE ROUTINES AND/OR WITH THE OUTPUT SYMBIONTS.

FILES OR ELEMENTS ARE INSERTED INTO THE RUN FILE WHEN AN @ADD CONTROL STATEMENT IS ENCOUNTERED BY THE READ$ ROUTINE. WHEN READ$ IS ENTERED AND AN @ADD CONTROL STATEMENT IS FOUND, THE FIRST IMAGE OF THE FILE NAMED WILL BE TRANSFERRED TO THE REQUESTOR IN LIEU OF THE @ADD. AN INTERNAL EXEC CALLING SEQUENCE TO PERFORM THE @ADD FUNCTION IS PROVIDED FOR DYNAMIC FILE INSERTION.

SYMBIONT OUTPUT FILES ARE SUBJECT TO VARIOUS CONTROL FUNCTIONS SUBMITTED BY AN INTERNAL CALLING SEQUENCE. THESE FUNCTIONS VARY ACCORDING TO THE OUTPUT DEVICE TYPE TO WHICH THE FILE IS BEING DIRECTED. AS THE CONTROL PARAMETERS ARE SUBMITTED THEY ARE PLACED INTO THE APPROPRIATE OUTPUT FILE, AND INTERPRETED WHEN THE FILE IS BEING

PROCESSED.

THE SYMBIONTS DESIGNED FOR USE WITH REMOTE DEVICES IN PARTICULAR ARE SELF-ADJUSTING. WHEN A NUMBER OF IMAGES ARE STACKED AWAITING TRANSMISSION, THE SYMBIONT AUTOMATICALLY MAKES USE OF A MASS STORAGE FILE FOR TEMPORARY STORAGE. ON THE OTHER HAND, WHEN ONLY A FEW ARE CURRENTLY IN TRANSIT, THE SYMBIONT SAVES THESE WITHIN THE CORE AREA AVAILABLE, DISPENSING THEM AS REQUIRED. OUTPUT FOR DEMAND RUNS USE CORE BUFFERING ONLY. WHEN A LIMITED NUMBER OF IMAGES ARE STACKED IN THE OUTPUT QUEUE, THE SUBMITTING RUN IS SWAPPED OUT.

COMMUNICATION BETWEEN THE 1108 SYMBIONTS AND THE 1004 OPERATES IN TWO MODES OF IMAGE HANDLING. THE ONE MODE TRANSFERS IMAGES IN THE FORMAT IN WHICH THEY ARE PRESENTED FROM THE USER. THIS IS DONE FOR ON-SITE 1004 OPERATIONS AND FOR THE REMOTE 1004 OPERATING AT HIGH SPEED TRANSMISSION RATES FASTER THAN 2400 BITS PER SECOND. THE OTHER MODE OF OPERATION COMPRESSES THE IMAGES BY ELIMINATING REDUNDANT SPACE CHARACTERS PRIOR TO TRANSMISSION ACROSS SLOWER SPEED REMOTE LINES. THE FOLLOWING TABLE IS A LIST OF THE FUNCTIONS ACCOMPANYING EACH TYPE OF 1108-1004 COMMUNICATION:

	ON-SITE	REMOTE NON-COMP	REMOTE COMP
CARD READ (80)	X	X	X
CARD PUNCH (80)	X	X	X
PRINT	X	X	X
CODE IMAGE READ (80)	X	X	
CODE IMAGE PUNCH (80)	X	X	
PAPER TAPE READ	X	X	
PAPER TAPE PUNCH	X	X	

THE SYMBIONT COMPLEX CONSISTS OF THE FOLLOWING ITEMS:

1. A MASTER CONTROL FOR ALL SYMBIONT OPERATIONS. INCLUDED ARE COMMON FUNCTIONS REQUIRED BY ALL VARIATIONS OF THE COMPLEX.
2. CONTROL TABLES ARE CONSTRUCTED DYNAMICALLY WITH ITEMS FOR ALL ACTIVE DEVICES AND FILES.
3. DEVICE ROUTINES, ONE FOR EACH OF THE HARDWARE DEVICES ATTACHED TO THE 1108. A SINGLE COPY IS BROUGHT IN WHICH HANDLES ALL DEVICES OF THE SAME CLASS.
4. IMAGE/BLOCK BUFFER CONTROL WHICH HANDLES ALL THE BUFFERED FILES ON MASS STORAGE.
5. INTERFACE ROUTINES WRITTEN IN A RE-ENTRANT MANNER TO DISPENSE INPUT IMAGES AND ACCEPT OUTPUT IMAGES FROM THE VARIOUS PROGRAMS AND THE EXEC.

MANY OF THE CONTROLLED DEVICES ARE INITIATED AUTOMATICALLY BY THE EXEC AS REQUIRED. OTHERS REQUIRE OPERATOR INITIATION SINCE INFORMATION IS NOT AVAILABLE TO EXEC FOR EXAMINATION. ACTIVITY MAY ALSO BE INITIATED WITH THE @SYM CONTROL STATEMENT.

9.2. READ$

RUN FILES GENERATED BY THE INPUT SYMBIONTS ARE AVAILABLE TO THE USER VIA THE READ$ ROUTINE. EACH REFERENCE TO READ$ RESULTS IN AN IMAGE TRANSMISSION TO THE REQUESTOR. THE IMAGES ARE EITHER TAKEN FROM THE SYMBIONT STORED RUN FILE OR ARE OBTAINED DIRECTLY FROM THE SYMBIONT. THE REQUIREMENT FOR ACCESSING DATA IN THIS MANNER IS THAT IT MUST FOLLOW THE EXECUTION CONTROL STATEMENT OF THE PROGRAM WHICH IS TO PROCESS THE DATA. IMAGE TRANSFERS ARE RESTRICTED TO NON-CONTROL CARDS ONLY, WITH THE EXCEPTION OF CLIST STATEMENTS AND THE PROCESSOR CALL CARD FOR PROCESSORS.

THE FOLLOWING LINKAGE IS USED TO OBTAIN AN IMAGE:

```
    LA  A0,PACKET
 ER   READ$
 NORMAL RETURN
```

REGISTER A0 CONTAINS:

```
 +  EOF,ADDRESS
```

THE FIELD 'ADDRESS' IS THE LOCATION TO WHICH THE INPUT IMAGE IS TRANSFERRED. THE IMAGE IS TAKEN FROM THE IMAGE POOL, AND TRANSFERRED TO 'ADDRESS'. CONTROL IS RETURNED TO THE ADDRESS FOLLOWING THE READ$ ENTRY WITH THE IMAGE WORD COUNT IN THE LOWER HALF OF A0. THE UPPER HALF OF A0 DEFINES THE TYPE AND SOURCE OF THE DATA IMAGE.

THE CONDITIONS POSSIBLE IN A0 ON RETURN FROM A READ$ OR TREAD$ REFERENCE ARE:

BIT 35 SET - ABNORMAL RETURN TAKEN BECAUSE A CONTROL STATEMENT CANNOT BE PASSED TO THE REQUESTOR. ANY FURTHER ATTEMPT TO DO A READ$ OR TREAD$ REQUEST WITHIN THIS PROGRAM WILL CAUSE AN ERROR TERMINATION. (WITHIN THE CLIST$ MODE THIS APPLIES ONLY TO THE @FIN

STATEMENT.)

BIT 34 SET - CURRENTLY READING AN ADD FILE. THIS IS NOT INFLUENCED BY OTHER BITS OR RETURNS.

BITS 33-32 - NOT USED, ALWAYS SET TO ZERO.

BIT 31 SET - IMAGE IN INFOR FORMAT. (SEE TECHNICAL DOCUMENTATION FOR DESCRIPTION OF INFOR.)

BIT 30 SET - PART OF AN INFOR TABLE, THE NEXT REFERENCE WILL RETRIEVE NEXT PART. IN OTHER WORDS, BIT 30 SET TO ZERO INDICATES THAT THIS IS THE LAST PART. BIT 31 IS SET REGARDLESS OF THE SETTING OF BIT 30.

BITS 29-18 - SET TO ZERO EXCEPT IN CLIST$ MODE WHEN IT IS SET TO ZERO FOR DATA STATEMENTS OR TO THE INDEX OF A COMMAND IN THE RELEVANT CLIST$.

BITS 17-0 - WORD COUNT IF IMAGE PASSED.
SENTINEL IF AN @EOF STATEMENT IS PASSED.
ZERO IF NO IMAGE PASSED.

@EOF STATEMENT RETURN
A0 IS SET TO THE SENTINEL CHARACTER (RIGHT-JUSTIFIED, ZEROFILLED) FROM COLUMN SIX OF THE STATEMENT. CONTROL IS RETURNED TO THE USER AT HIS ABNORMAL RETURN POINT. BIT 34 IS THE ONLY BIT THAT MAY BE SET ON THIS RETURN.

9.2.1. CLIST$

CLIST$ IS A MECHANISM TO ALLOW A PROGRAM TO READ A GIVEN SET OF CONTROL STATEMENTS. IF A USER CHOOSES TO USE CLIST$, THOSE CONTROL STATEMENTS IN HIS LIST WILL BE PASSED TO HIM, TRANSPARENT STATEMENTS WILL BE HANDLED BY THE EXECUTIVE, AND THE @EOF AND @FIN STATEMENTS WILL CAUSE AN END-OF-FILE RETURN; ALL OTHER CONTROL STATEMENTS WILL BE REJECTED BY THE EXECUTIVE UNTIL AN ACCEPTABLE STATEMENT IS ENCOUNTERED.

CLIST$ CAN BE USED IN ONE OF TWO WAYS; IT CAN CALL FOR A LIST OF COMMANDS THAT IS STORED IN THE EXEC, OR IT CAN REQUEST A LIST IN THE USER PROGRAM WHICH IS MOVED INTO THE EXEC FOR THE DURATION OF THE PROGRAM. THE CALLING SEQUENCE FOR CLIST$ IS

```
          L          A0,PKT
          ER         CLIST$
```

WHERE PKT CONTAINS THE ALPHANUMERIC NAME OF A CANNED LIST, LEFT-JUSTIFIED, SPACE-FILLED, OR THE ADDRESS OF A LIST WITHIN THE PROGRAM. IN EITHER CASE, THE LIST MAY CONTAIN UP TO 62 ALPHANUMERIC NAMES, LEFT-JUSTIFIED, SPACE-FILLED. THE LIST IS TERMINATED BY A MINUS ZERO. THE INDEX OF A NAME CORRESPONDS TO ITS POSITION IN THE LIST; THE FIRST NAME HAS INDEX 1; THE SECOND 2; AND SO ON. IF A LIST NAME IS GIVEN THAT IS NOT IN THE SYSTEM, IT IS NOT CONSIDERED TO BE IN ERROR, BUT THE LIST IS REGARDED AS AN EMPTY LIST. IF A MINUS ZERO IS NOT SUPPLIED AS A TERMINAL ITEM IN THE LIST, IT IS AUTOMATICALLY SUPPLIED IN PLACE OF THE SIXTY-THIRD NAME.

WHENEVER A PROGRAM HAS ENTERED THE CLIST$ MODE, THE CONTROL STATEMENT @ENDX IS AUTOMATICALLY ASSUMED TO BE PART OF THE LIST; ITS INDEX IS 77 OCTAL. THE PURPOSE OF THIS STATEMENT IS TO HAVE A COMMON TERMINATION COMMAND FOR PROGRAMS USING CLIST$.

THE INDEX SPECIFYING A GIVEN COMMAND FROM THE LIST IS RETURNED IN BITS 23 TO 18 OF A0 ON RETURN FROM THE READ$ REQUEST. THE IMAGE GIVEN FOR A STATEMENT CONTAINING A COMMAND IN THE LIST WILL BE THE COMPLETE STATEMENT

IMAGE.

IF A CONTROL STATEMENT IS REJECTED AND THE PROGRAM IS ATTEMPTING TO READ THE IMAGE VIA TREAD$, THE OUTPUT IMAGE ASSOCIATED WITH THE TREAD$ CALL WILL BE REPRINTED.

THE PROCEDURE

```
    R$EAD  PKT
```

GENERATES A TWO WORD CALLING SEQUENCE. THE FIRST WORD LOADS REGISTER A0 FROM THE LOCATION 'PKT' AND THE SECOND WORD IS AN 'ER READ$'.

9.2.2. TREAD$

THE LINKAGE TREAD$ CAN BE USED AS AN ALTERNATE TO THE READ$ REQUEST. TREAD$ IS DESIGNED FOR CONVERSATIONAL REMOTE TERMINAL INPUT AND PROVIDES THE ABILITY TO PRINT AT THE TERMINAL A REQUEST FOR AN INPUT IMAGE. THE REQUEST FORM IS:

```
         LA   A0,PACKET ADDRESS
        ER    TREAD$
```

WHERE PACKET ADDRESS REFERS TO A TWO WORD PACKET CONTAINING THE PRINT$ AND READ$ PARAMETERS:

```
        P       L,S,N,OUT
        +       EOF,ADDRESS
```

THE ACTION THAT TAKES PLACE IS:

1 'S' SPACING IS PERFORMED.
2 THE IMAGE AT 'OUT' OF LENGTH 'N' IS PRINTED.
3 AN IMAGE IS ACCEPTED FROM THE INPUT DEVICE.
4 IF THE IMAGE IS DIRECTED TO THE EXECUTIVE, IT IS ACTED UPON WHILE THE PROGRAM WAITS. SPACING AND THE PRINTOUT IS THEN RE-ISSUED AS A REQUEST FOR ANOTHER IMAGE FOR THE PROGRAM.
5 AN IMAGE FOR THE PROGRAM IS THEN TRANSMITTED WITH THE SAME INFORMATION IN A0 AS A READY CALL.

THE PRINTING IS SUSPENDED WHEN IMAGES ARE BEING TAKEN FROM AN @ADD FILE.

9.2.3. ALTERNATE INPUT FILES (READA$)

ASSIGNMENT OF SYMBIONT CONTROLLED INPUT DEVICES IS MADE AVAILABLE TO A RUN WITH THE ARBITRARY DEVICE FORMAT @ASG STATEMENT. THE OPERATING PROGRAM MAY EITHER USE THE DEFINED DEVICE SYMBIONT OR PROVIDE ITS OWN DEVICE HANDLER TO PERFORM THE INPUT OPERATION. WHEN USING THE SYMBIONT COMPLEX, INPUT IMAGES ARE OBTAINED FROM THE DEVICE THROUGH THE LINKAGE:

```
        L    A0,  PACKET ADDRESS
        ER   READA$
```

REGISTER A0 CONTAINS THE ADDRESS OF THE PACKET:

```
        +    EOF,ADDRESS
             'FILE NAME'
```

THE INTERNAL 'FILE NAME' MUST HAVE BEEN PREVIOUSLY ASSOCIATED WITH AN INPUT DEVICE PRIOR TO THE FIRST READA$ REFERENCE. IF THE INPUT DEVICE IS MASS STORAGE OR TAPE, IT IS ASSUMED

TO BE IN SDF FORMAT AND WILL SIMPLY BE READ. IF THE FILENAME HAS NOT BEEN DEFINED, THE PROGRAM IS PLACED IN THE 'ERR MODE'. THE FIRST REFERENCE TO READA$ INITIATES THE INPUT OPERATION AND CONTROL IS RETURNED UPON RECEIPT OF THE IMAGE. THE INPUT OPERATION IS CONTINUOUS WITH INTERMEDIATE MASS STORAGE BUFFERING IN SDF FORMAT.

EACH DEVICE TYPE HAS AN END OF INPUT SENTINEL WHICH IS DEFINED AS THE @FIN STATEMENT FOR UNIT RECORD EQUIPMENT, OR THE SDF END-OF-FILE SENTINEL FOR TAPE OR MASS-STORAGE. WHEN READA$ DETECTS THE END-OF-FILE SENTINEL, THE FILE IS CLOSED (BUT NOT FREED) AND CONTROL IS RETURNED TO THE USER AT HIS EOF RETURN.

IN ADDITION TO THE END OF INPUT SENTINEL, THE @COL CONTROL STATEMENT AND ITS CORRESPONDING END SENTINEL ARE INTERPRETED IN THE ALTERNATE MODE. ALL OTHER CONTROL STATEMENTS ARE NOT INTERPRETED.

READA$ FILE WILL BE CLOSED ON AN END-OF-FILE, A @FREE OF THE FILE, OR ON A @BRKPT FILENAME (SEE @BRKPT IN THE CHAPTER ON EXECUTIVE CONTROL LANGUAGE.
THE PROCEDURE:

```
R$EADA PKT ADDRESS
```

GENERATES A TWO-WORD CALLING SEQUENCE. THE FIRST WORD LOADS A0 WITH THE LOCATION 'PKT ADDRESS' AND THE SECOND WORD IS AN 'ER READA$'.

9.3. PRINT$

ASSOCIATED WITH EACH RUN IS A PRINT FILE WHICH IS ESTABLISHED BY THE COARSE SCHEDULER. THIS FILE CONTAINS THE PRINT OUTPUT GENERATED BY THE VARIOUS SYSTEM COMPONENTS AND THE USER PROGRAMS OF THE RUN. PRINT$ TRANSFERS EACH IMAGE FROM THE USER'S AREA TO THE SYSTEM CONTROLLED IMAGE POOL. FROM THERE THE IMAGES ARE BUFFERED TO AN INTERMEDIARY STORAGE DEVICE, OR ARE PROCESSED DIRECTLY BY THE PRINT SYMBIONT. THE BATCH PROCEDURE IS FOR PRINT$ TO BUFFER THE FILE TO MASS STORAGE, AND FOR THE SYMBIONT TO PROCESS THE FILE WHEN IT REACHES THE TOP OF THE QUEUE. WHEN MAGNETIC TAPE IS DEFINED AS THE REPOSITORY MEDIUM, THE FILE IS BUFFERED DIRECTLY TO THE TAPE UNIT. IN THE DEMAND MODE, THE FILE IS PROCESSED DIRECTLY BY THE SYMBIONT. THE FOLLOWING LINKAGE IS USED TO TRANSMIT A PRINT IMAGE:

```
ER    PRINT$
NORMAL RETURN
```

PRIOR TO ENTERING PRINT$, REGISTER A0 MUST BE LOADED WITH THE FOLLOWING WORD:

```
P  L,S,N,A
```

WITH P DEFINED AS

```
P    FORM  6,6,6,18
```

WHERE:
A- IS THE ADDRESS OF THE FIRST WORD OF THE PRINT IMAGE.
N- IS THE NUMBER OF WORDS TO BE TRANSFERRED TO THE PRINT FILE BEGINNING AT LOCATION 'A'.
S- IS THE NUMBER OF LINES TO BE SPACED BEFORE PRINTING THE IMAGE. FOR S=0, NO SPACING WILL OCCUR. WHEN S IS GREATER THAN THE NUMBER OF LINES REMAINING ON THE PRINTABLE PAGE, THE IMAGE WILL BE PRINTED ON LOGICAL LINE 1 OF THE FOLLOWING PAGE. WHEN THE COUNT OF AN IMAGE EXCEEDS THE NUMBER OF LINES REMAINING ON THE CURRENT PAGE, THE IMAGE SUBMITTED IS PRINTED ON LOGICAL LINE 1 OF THE SUCCEEDING PAGE.
L- IS THE CARRIAGE CONTROL LOOP VALUE.

THE @BRKPT STATEMENT PROVIDES THE MEANS TO ESTABLISH MASS-STORED PRINT FILE BREAKPOINTS OR TO RE-ASSIGN THE FILE'S OUTPUT MEDIUM. WHEN RE-ASSIGNMENT OF THE PRINT$ FILE IS SPECIFIED, ONLY THOSE PRINT IMAGES GENERATED AFTER THE @BRKPT STATEMENT WILL RESIDE ON THE LATEST DEFINED DEVICE. THE @SYM CONTROL STATEMENT, PREVIOUSLY DISCUSSED IN THE EXECUTIVE LANGUAGE CHAPTER OF THIS DOCUMENT , IS REQUIRED TO QUEUE ALL PRINT FILES WHICH HAVE BEEN GENERATED IN ANY FILE OTHER THAN THE SYSTEM DEFINED PRINT$ FILE (THIS INCLUDES BREAKPOINT TO MAGNETIC TAPE). IF THE PRINT$ IS INTENDED FOR OUTPUT OTHER THAN THE DEVICE ARRIVED AT BY DEVICE ASSOCIATION, IT MUST BE QUEUED FOR THE INTENDED SYMBIONT WITH THE @SYM CONTROL STATEMENT PRIOR TO THE END OF THE RUN.

THE PROCEDURE:

```
        P$RINT  PKT
```

WILL GENERATE A TWO WORD CALLING SEQUENCE REFERENCING PRINT$. THE FIRST WORD WILL LOAD REGISTER A0 FROM THE ADDRESS 'PKT', FOLLOWED BY AN 'ER PRINT$'.

9.3.1. ALTERNATE PRINT FILES (PRNTA$)

AN ADDITIONAL ENTRY INTO THE PRINT$ INTERFACE ROUTINE IS PROVIDED TO FACILITATE THE HANDLING OF PRINT FILES OTHER THAN THE SYSTEM INITIATED PRINT FILE. THE FOLLOWING LINKAGE IS USED TO TRANSMIT A PRINT IMAGE INTO AN ALTERNATE PRINT FILE

```
      ER  PRNTA$
      NORMAL RETURN
```

PRIOR TO ENTERING PRNTA$, REGISTER A0 MUST BE SET TO THE ADDRESS OF THE FOLLOWING PACKET:

```
           P  L,S,N,A
              'FILE-NAME' (12 CHARACTERS SPACE FILLED)
```

THE FIELDS 'A', 'N', 'S' AND 'L' AND THE FORM 'P' HAVE THE SAME DEFINITION AS PRINT$. THE ADDITIONAL FIELD 'FILE-NAME' IS THE INTERNAL REFERENCE TO A PREVIOUSLY ASSIGNED DEVICE. IF 'FILENAME' IS UNDEFINED, AN @ASG, CP FILE NAME, F2 IS PERFORMED AND THE FILE IS OPENED. THE FIRST REFERENCE TO PRNTA$ INITIALIZES THE PRINT OPERATION.

THE ALTERNATE FILES PRODUCED ARE IN SDF FORMAT. . IF THE ASSIGNED DEVICE IS A PRINTER SUBSYSTEM, PRINTING IS INITIATED WITH THE FIRST REQUEST. INTERMEDIATE MASS STORAGE BUFFERING IS UTILIZED WITH DIRECT PRINTING. WHEN THE SDF END-OF-FILE IS DETECTED, THE PRINTER IS RELEASED.

FILE LABELS FOR ALTERNATE PRINT FILES ARE GENERATED WITH THE 'FILENAME' FIELD OF THE PRNTA$ CALLING SEQUENCE. A @BRKPT CALL CAUSES THE FIRST PRINT FILE TO BE CLOSED AND THE TAPE POSITIONED FOR THE NEXT FILE. THE FIRST REFENCE TO PRNTA$ FOLLOWING THE ER @BRKPT INITIALIZES THE SUCCEEDING FILE.

PRINT FILES PROCESSED VIA THE PRNTA$ LINKAGE ARE NOT AUTOMATICALLY ENTERED INTO A SYMBIONT PRINT QUEUE, UNLESS THE EXEC PERFORMED THE ASSIGNMENT OF THE FILE IN WHICH CASE WHEN THE FILEIS CLOSED, IT IS QUEUED IN THE SAME MANNER AS THE PRINT$ FILE AND THE FILE IS DECATALOGUED AFTER IT IS PRINTED. IN ALL OTHER CASES THE @SYM CONTROL STATEMENT IS REQUIRED TO QUEUE THE GENERATED PRINT FILE.

THE PROCEDURE:

```
      P$RNTA  PKT
```

GENERATES A TWO-WORDCALLING SEQUENCE REFERENCING PRNTA$. THE FIRST WORD LOADS REGISTER A0 WITH THE ADDRESS 'PKT', FOLLOWED BY AN 'ER PRNTA$'.

9.4. PUNCH$

EACH RUN ENTERED INTO THE SYSTEM WILL HAVE A PUNCH FILE WHICH IS ESTABLISHED WITH THE FIRST REFERENCE TO PUNCH$. THE DEFINITION OF THIS PUNCH FILE IS SPECIFIED AS EITHER 80 COLUMN CARD IMAGES, OR AS 5,6,7 OR 8 LEVEL PAPER TAPE. THE DETERMINATION OF THE PUNCH TYPE IS BY INPUT DEVICE ASSOCIATION. IF A DIFFERENT DEFINITION IS REQUIRED, IT CAN BE ESTABLISHED BY A REFERENCE TO PCHCN$. REFERENCE TO THIS PUNCH FILE IS VIA THE LINKAGE:

```
    ER   PUNCH$
    NORMAL RETURN
```

PRIOR TO ENTERING PUNCH$, REGISTER A0 MUST CONTAIN THE FOLLOWING WORD:

```
    +   N,A
```

A- IS THE ADDRESS OF THE FIRST WORD OF THE PUNCH IMAGE
N- IS THE NUMBER OF WORDS TO BE TRANSFERRED TO THE PUNCH
   FILE BEGINNING AT LOCATION 'A'.

THE @BRKPT STATEMENT PROVIDES THE MEANS TO ESTABLISH MASS STORED PUNCH FILE BREAKPOINTS OR TO RE-ASSIGN THE FILE'S OUTPUT MEDIUM. THE @SYM CONTROL STATEMENT IS REQUIRED TO QUEUE ALL PUNCH FILES WHICH HAVE BEEN GENERATED IN ANY FILE OTHER THAN THE SYSTEM DEFINED PUNCH$ FILE.
IF THE PUNCH$ FILE IS INTENDED FOR OUTPUT OTHER THAN THAT DETERMINED BY DEVICE ASSOCIATION, IT MUST BE QUEUED FOR THE PROPER SYMBIONT BY THE @SYM CONTROL STATEMENT BEFORE THE END OF THE RUN.

THE PROCEDURE

```
        P$UNCH PKT
```

WILL GENERATE A TWO-WORD CALLING SEQUENCE REFERENCE TO PUNCH$. THE FIRST WORD WILL LOAD REGISTER A0 FROM THE ADDRESS 'PKT', FOLLOWED BY AN 'ER PUNCH$'.

9.4.1. ALTERNATE PUNCH FILES (PNCHA$)

AN ADDITIONAL ENTRY INTO THE PUNCH$ INTERFACE ROUTINE IS PROVIDED TO FACILITATE THE PROCESSING OF PUNCH FILES OTHER THAN SYSTEM INITIATED PUNCH FILE. THE FOLLOWING LINKAGE IS USED TO TRANSMIT A PUNCH IMAGE

```
    ER PNCHA$
    NORMAL RETURN
```

PRIOR TO ENTERING PNCHA$, REGISTER A0 MUST BE SET TO THE ADDRESS OF THE FOLLOWING PACKET

```
    + N,A
      'FILE NAME' (12 CHARACTERS SPACE FILLED)
```

'N' AND 'A' HAVE THE SAME DEFINITION AS PUNCH$.
THE RULES FOR 'FILENAME' AND FILE QUEUEING ARE THE SAME AS FOR PRNTA$ FILES.

THE PROCEDURE

```
    P$NCHA  PKT
```

GENERATES A TWO-WORD CALLING SEQUENCE REFERENCING PNCHA$. THE FIRST WORD LOADS REGISTER A0 WITH THE ADDRESS 'PKT', FOLLOWED BY AN 'ER PNCHA$'.

9.5. SYMBIONT CONTROL

THERE ARE SEVERAL CONTROL FUNCTIONS TO ALLOW THE USER A MEANS OF DIRECTING THE SYMBIONTS TO PERFORM SPECIAL FUNCTIONS ON A FILE.

9.5.1. OUTPUT CONTROL

THE OUTPUT CONTROL FUNCTIONS PROVIDE A GENERAL METHOD OF CONTROLLING SYMBIONT OUTPUTS. EACH REQUEST INSERTS IN THE OUTPUT FILE AN IMAGE OF THE SAME FORMAT AS STANDARD DATA, BUT WITH A FLAG INDICATING THIS IS CONTROL INFORMATION. THE SYMBIONT OUTPUT DEVICE ROUTINES INTERPRET THIS CONTROL IMAGE AND TAKE THE APPROPRIATE ACTION. NOTE THAT FOR MASS STORAGE OR TAPE BUFFERED FILES THIS ACTION OCCURS AT THE TIME OF THE PHYSICAL OUTPUTTING. FOUR EXECUTIVE REQUEST FUNCTIONS CORRESPONDING TO THE PRINT$, PUNCH$, PRNTA$, AND PNCHA$ REQUESTS ARE PROVIDED:

```
LA      A0,(N,A)
ER      PRTCN$          FOR PRINT$ FILES

LA      A0,(N,A)
ER      PCHCN$          FOR PUNCH$ FILES

LA      A0,(N,A)
ER      PRTCA$          FOR PRNTA$ FILES

LA      A0,(N,A)
ER      PCHCA$          FOR PNCHA$ FILES
```

FOR ALL CALLING SEQUENCES, 'N' IS THE LENGTH OF THE PACKET IN WORDS WHICH BEGINS AT ADDRESS 'A'. FOR PRINT$ AND PUNCH$ N IS OF THE CHARACTER STRING OR IMAGE WHICH DEFINES THE CONTROL FUNCTION. IT MAY BE CODED:

```
'IMAGE'
```

FOR ALTERNATE FILES THE PACKET CONTAINS TWO ADDITIONAL WORDS AT THE BEGINNING NAMING THE FILE:

```
'FILENAME'          (12 CHARACTERS SPACE FILLED)
'IMAGE'
```

EACH IMAGE CONSISTS OF ONE OR MORE CONTROL FUNCTIONS. SPACES ARE IGNORED PRIOR TO THE FIRST, OR BETWEEN FUNCTIONS. EACH FUNCTION BEGINS WITH A SINGLE LETTER, FOLLOWED BY A COMMA FOLLOWED BY ANY SPECIAL INFORMATION REQUIRED, AND ENDED BY A PERIOD THE FORMAT OF THE INFORMATION CHARACTER STRING VARIES ACCORDING TO THE FUNCTION BUT MUST NOT CONTAIN A PERIOD.

9.5.1.1. PRTCN$ CONTROL FUNCTIONS

THE FOLLOWING CONTROL FUNCTIONS ARE DEFINED FOR PRINT FILES:

L- SPACE PRINTER TO LOGICAL LINE 'NN', WHERE LOGICAL LINE IS DEFINED AS THE LINE NUMBER RELATIVE TO THE TOP MARGIN SETTING. (SEE M BELOW). ALL LINE POSITIONING AND PRINTING IS PERFORMED WITHIN THE DEFINED MARGIN SETTINGS. (THE BOTTOM LOGICAL LINE OF A PAGE IS IDENTICAL TO THE TOP LOGICAL LINE -1 OF THE SUCCEEDING PAGE.) POSITIONING TO A LOGICAL LINE ON PRINTERS WITH SPACE-PRINT OPERATION IS TO LOGICAL LINE N-1, THEREFORE WHEN N=1, THE LOGICAL LINE SETTING IS THE LAST LINE OF THE CURRENT PAGE. THIS IS ALSO TRUE WHEN N=0, OR WHEN N IS GREATER THAN THE LENGTH OF THE LOGICAL PAGE. WHEN N IS LESS THAN OR EQUAL TO THE CURRENT LINE OF THE CURRENT PAGE, THE SUCCEEDING PAGE IS POSITIONED TO THE LOGICAL LINE N-1.

THE FORMAT OF THIS FUNCTION AS IT APPEARS IN THE CONTROL PACKET IS:

'L,NN,'

H- INITIATE HEADING PRINTING. THIS FUNCTION PROVIDES THE USER WITH AN AUTOMATIC MEANS OF PRINTING A HEADING ON EACH SUCCEEDING PAGE OF HIS PRINT FILE. THE FORMAT OF THIS FUNCTION AS IT APPEARS IN THE CONTROL PACKET IS

'H,OPTION,PAGE#,TEXT OF HEADING,'

IF 'OPTION' CONTAINS THE LETTER 'X', A PAGE AND DATE WILL NOT BE PRINTED AS A PART OF THE HEADING. 'N' TURNS THE HEADING OFF.

A PAGE COUNT IS MAINTAINED BY THE PROCESSING SYMBIONT. WHEN THE 'PAGE#' FIELD IS BLANK, THE PAGE COUNT CURRENT TO THE FILE IS USED TO BEGIN PAGE NUMBERING. WHEN CODED, 'PAGE#' IS MADE THE PAGE NUMBER. IN ADDITION TO THE PAGE NUMBER, THE CURRENT DATE IS INCLUDED IN THE HEADING, AND BOTH WILL APPEAR IN THE UPPER RIGHT CORNER OF EACH PAGE. THE POSITION OF THE HEADING IS THE SECOND LINE ABOVE LOGICAL LINE 1. IF THE UPPER MARGIN IS ONE LINE OR NON-EXISTENT, NO HEADING WILL BE PRINTED. AS MANY AS 16 WORDS OF HEADING TEXT MAYBE SUPPLIED.

M- SET MARGINS. THIS FUNCTION SUPPLIES THE INFORMATION FOR RE-ADJUSTING PAGE LENGTH, AND TOP AND BOTTOM MARGINS. THE STANDARD PRINT PAGE DEFINITION IS 66 LINES PER PAGE WITH A TOP MARGIN SETTING OF SIX LINES, AND A BOTTOM MARGIN SETTING OF THREE LINES. NOTE THAT TOP AND BOTTOM MARGINS REFER TO THE NUMBER OF BLANK LINES AT THE TOP AND BOTTOM OF THE PAGE RESPECTIVELY. THUS THE STANDARD MARGIN SETTING IS 66,6,3 GIVING 57 PRINTABLE LINES. THIS PAGE DEFINITION IS ASSUMED AT THE BEGINNING OF EACH PRINT FILE. WHEN THE 'M' FUNCTION IS USED, A PAGE ALIGNMENT PROCEDURE IS INITIATED WITH THE PAGE LENGTH PARAMETER. THIS FUNCTION IS ALSO USED TO RETURN TO THE STANDARD PAGE LENGTH. THE FORMAT OF THIS FUNCTION AS IT APPEARS IN THE CONTROL PACKET IS:

'M,LENGTH,TOP,BOTTOM,'

S- SPECIAL FORMS REQUEST. THIS FUNCTION IS PROVIDED TO SPECIFY THE LOADING OF A SPECIAL FORM REQUEST TO PROCESS THE PRINT, OR PUNCH FILE. THE FORMAT OF THIS FUNCTION AS IT APPEARS IN THE CONTROL PACKET IS:

'S,MESSAGE TEXT,'

WHERE THE MESSAGE TEXT IS A MAXIMUM OF TEN WORDS. WHEN THIS FUNCTION IS ENCOUNTERED BY THE PROCESSING SYMBIONT, IT WILL BE DISPLAYED ON THE OPERATOR'S CONSOLE IN THE FORM:

SNAME/SITE I.D. MESSAGE DISPLAYED HERE AQ

THE OPTIONS AVAILABLE TO THE OPERATOR FOR ANSWERING THE MESSAGE ARE SYMBIONT DEPENDENT. THE FOLLOWING OPTIONS ARE INCLUDED IN THE 0755 HSP, CARD PUNCH AND THE 1004 PRINTER , CARD PUNCH, AND REMOTE SYMBIONTS.

A-BEGIN PROCESSING THE OUTPUT FILE.
Q-RETURN FILE TO SYMBIONT QUEUE. THE PRINT OR PUNCH FILE WILL BE PASSED TEMPORARILY AND PLACED AT THE END OF THE OF THE SYMBIONT .

9.5.1.2. PCHCN$ CONTROL FUNCTIONS

THE FOLLOWING CONTROL FUNCTIONS ARE DEFINED FOR PUNCH FILES:

C- THE OUTPUT SYMBIONTS REACT TO THIS FUNCTION IN THE SAME MANNER AS THE INPUT SYMBIONTS REACT TO A @COL CARD. THE FORMAT OF THIS FUNCTION AS IT APPEARS IN THE CONTROL PACKET IS:

'C,OPTIONS.'

WHERE THE OPTIONS ARE:

B- SWITCH TO COLUMN BINARY.
E- SWITCH TO EIGHTY COLUMN CARD CODE.
5- SWITCH TO 5 LEVEL PAPER TAPE.
6- SWITCH TO 6 LEVEL PAPER TAPE.
7- SWITCH TO 7 LEVEL PAPER TAPE.
8- SWITCH TO 8 LEVEL PAPER TAPE.
T- THE SYMBIONT WILL TRANSLATE FROM FIELDDATA TO THE STANDARD CODE DEFINED FOR THAT LEVEL OF PAPER TAPE. THIS OPTION IS VALID ONLY WHEN THE SYMBIONT IS IN ONE OF THE PAPER TAPE MODES OR AS A SECOND OPTION WITH ONE OF THE PAPER TAPE DESIGNATORS.

S- SPECIAL FORM REQUEST. THIS IS DEFINED AND TREATED EXACTLY THE SAME AS IT IS FOR PRNCN$.

9.5.1.3. INTERNAL BREAKPOINT FUNCTION

THE INTERNAL BREAKPOINT ALLOWS DYNAMIC BREAKPOINTING OF PRINT AND PUNCH FILES AS DESCRIBED IN THE SECTION ON EXECUTIVE CONTROL LANGUAGE. THE CALL IS TO CSF$ WITH AN IMAGE OF A @BRKPT CARD. THE CALLING SEQUENCE IS DEFINED IN THE SECTION ON EXECUTIVE REQUEST FUNCTIONS.

9.5.1.4. INTERNAL FILE QUEUEING

FILES MAY BE QUEUED INTERNALLY BY A REQUEST TO CSF$ WITH REGISTER A0 SET TO THE ADDRESS OF AN IMAGE OF A @SYM CONTROL STATEMENT. THE HANDLING OF THIS REQUEST IS DESCRIBED IN THE SECTION ON EXECUTIVE CONTROL LANGUAGE AND THE CALL TO CSF$ IS DESCRIBED IN THE SECTION ON EXECUTIVE REQUEST FUNCTIONS.

9.5.2. INPUT CONTROL

THE ONLY MEANINGFUL CONTROL ON INPUT FILES IS THE @ADD FUNCTION. THIS IS INVOKED INTERNALLY BY AN:

```
ER   CSF$
```

WITH REGISTER A0 SET TO:

```
+    N,ADDRESS
```

WHERE N IS THE NUMBER OF WORDS IN THE PACKET AT 'ADDRESS'. THE FORMAT IS IDENTICAL TO THE @ADD CARD. THE EFFECT OF THE DYNAMIC ADD IS DESCRIBED IN THE SECTION ON EXECUTIVE CONTROL LANGUAGE.

9.5.3. SYMBIONT ERROR CODE DEFINITIONS (EMODE TYPE 02)

-----	----------
01	READ WAS ATTEMPTED BEYOND FILE LIMITS
02	SECOND ABNORMAL RETURN FROM READ$
03	I/O ERROR ENCOUNTERED BY READ$
04	IMAGE LENGTH IN SDF FILE IS GREATER THAN 15 WORDS
05	ATTEMPT TO REFERENCE AN ADD FILE WITHOUT BEING ASSIGNED
06	ADDRESS IN A0 ON READ$ CALL IS NOT IN REQUESTORS CORE AREA
07	IMPROPER FILE CONTROL IMAGE (@COL ETC.)
020	ALTERNATE FILE BEING REFERENCED WAS NOT ASSIGNED FOR

	DEMAND
021	ALTERNATE FILE COULD NOT BE ASSIGNED FOR PRINT OR PUNCH
022	ALTERNATE ER REQUEST IS IMPROPER FOR TYPE OF FILE
023	ALTERNATE FILE PACKET IS NOT WITHIN REQUESTORS CORE AREA
024	ALTERNATE READ FILE BEING REFERENCED HAS NOT BEEN ASSIGNED
025	ERROR ON FIRST WORD IN READA$ FILE
026	ALTERNATE FILE NOT FAST OR TAPE
027	MAX. NO. OF ALT FILE EXCEEDED
030	MAX. NO. OF BREAKPOINTS FOR PR$ PU$ EXCEEDED
034	LENGTH OF ALTERNATE PRINT FILE HAS BEEN EXCEEDED
035	LENGTH OF ALTERNATE PUNCH FILE HAS BEEN EXCEEDED
036	PUNCH$ FILE HAS BEEN EXCEEDED
037	PRINT$ FILE HAS BEEN EXCEEDED
040	PRINT/PUNCH BUFFER IS NOT WITHIN REQUESTOR CORE AREA
041	MAX PAGES SPECIFIED ON RUN CARD HAS BEEN REACHED
042	MAX CARDS SPECIFIED ON RUN CARD HAS BEEN REACHED
077	ADD FILE NESTING GREATER THAN LIMITS.

9.6. ON-SITE DEVICE HANDLERS

9.6.1. INPUT OPERATION

USE OF THE CARD, PRINTER AND PAPER TAPE DEVICES ARE HANDLED BY THE SYSTEM UNDER SYMBIONT CONTROL. MANY OF THE OPERATIONS PERFORMED ON THESE DEVICES ARE INITIATED AUTOMATICALLY BY THE EXEC. THE INPUT DEVICES, CARD AND PAPER TAPE, ARE EXAMINED PERIODICALLY BY THE SYSTEM FOR A NON-INTERLOCK CONDITION ON A PARTICULAR UNIT. INPUT FUNCTIONS ARE SENT TO THE UNITS BY THE SYMBIONT CONTROL ROUTINE AND THE RESULTANT INTERRUPT IS INTERROGATED TO DETERMINE THE STATE OF THE DEVICE. (THE 1004 IS PROBED IN A SIMILAR MANNER BUT CONTROLLED BY AN ALTERATION SWITCH.) WHEN THE UNIT BEING PROBED RETURNS AN INTERRUPT, OTHER THAN INTERLOCK, THE APPROPRIATE SYMBIONT IS THEN LOADED AND GIVEN CONTROL. A @RUN CONTROL STATEMENT MUST THEN BE ENCOUNTERED. A RUN FILE OF THE INPUT STREAM IS GENERATED ON MASS STORAGE AND IS TERMINATED BY THE PRESENCE OF A SUCCEEDING @RUN OR @FIN STATEMENT. EACH @RUN STATEMENT ENCOUNTERED IN THE INPUT STREAM WILL GENERATE A UNIQUE RUN FILE. HOWEVER, WHEN A @FILE STATEMENT IS ENCOUNTERED AN ADDITIONAL FILE IS GENERATED, WHICH INCLUDES ALL FOLLOWING INPUT TO THE @ENDF STATEMENT. IF THE @FILE STATEMENT REQUIRES A DEVICE OTHER THAN IMMEDIATE ACCESS STORAGE, AN INPUT MODE IMPASSE IS MAINTAINED UNTIL THE PROPER DEVICE IS READIED FOR RECEIPT OF THE DATA FILE. A MOUNT MESSAGE DIRECTING THE LOADING OF A TAPE ACCORDING TO THE 'NAME' FIELD OF THE @FILE STATEMENT IS DISPLAYED ON THE OPERATOR'S CONSOLE.

IF @RUN IS NOT ENCOUNTERED AS THE FIRST STATEMENT OF THE INPUT STREAM, OR AS THE FIRST STATEMENT FOLLOWING @FIN, THE UNIT WILL BE LOGICALLY INTERLOCKED AND THE FOLLOWING MESSAGE WILL BE DISPLAYED ON THE CONSOLE

```
SNAME RUN SEARCH A
```

WHERE THE RESPONSE INTERPRETATION FOR

A- IS READ AND SEARCH THE INPUT FOR A @RUN STATEMENT. EACH STATEMENT READ IN THIS MODE WILL BE DISREGARDED UNTIL A @RUN IS ENCOUNTERED, AT WHICH POINT THE MESSAGE

```
SNAME END SEARCH 'RUNID'
```

IS DISPLAYED ON THE CONSOLE AND THE NORMAL MODE IS ASSUMED. THE 'RUNID' IS THE ID FIELD OF THE FIRST @RUN CARD WHICH IS FOUND. IF ALL AVAILABLE CARDS ARE READ AND NO @RUN CARD IS FOUND, THE

READER WILL BE RETURNED TO THE PROBE CYCLE.

INDIVIDUAL PERIPHERAL UNITS CAN BE ELIMINATED FROM THE INTERLOCK/PROBE CYCLE WITH THE FOLLOWING UNSOLICITED KEY-IN:

```
SM CRX L
```

INDIVIDUAL UNITS CAN BE RE-INSTATED TO THE INTERLOCK/PROBE CYCLE THE UNSOLICITED KEY-IN.

```
SM CRX I
```

9.6.2. OUTPUT QUEUEING.

ALL PRINT AND PUNCH FILES HANDLED BY SYMBIONTS, EXCEPT FOR ALTERNATE FILES ASSIGNED DIRECTLY TO AN ARBITRARY DEVICE, MUST GO THROUGH AN OUTPUT QUEUE. FILES ARE NEVER PLACED IN THE QUEUE UNTIL THEY ARE CLOSED. FILES MAY BE CLOSED BY A @BRKPT CARD, RUN TERMINATION, OR A @FREE CARD. SYSTEM DEFINED PRINT$ AND PUNCH$ FILES AND THOSE ALTERNATE FILES ASSIGNED BY THE EXEC ARE AUTOMATICALLY QUEUED FOR OUTPUT WHEN THEY ARE CLOSED. IF IT IS DESIRED TO DIRECT THE OUTPUT OF PRINT$ AND PUNCH$ FILES TO A DEVICE OTHER THAN THE ONE ARRIVED AT BY INPUT DEVICE ASSOCIATION, IT IS NECESSARY TO PERFORM A @SYM BEFORE CLOSING THE FILE. IF INPUT DEVICE ASSOCIATION IS NOT DESIRED FOR ALTERNATE FILES, IT IS NECESSARY FOR THE USER TO ASSIGN THE MASS STORAGE FILE. IF THE FILE IS NOT QUEUED AUTOMATICALLY, IT IS NECESSARY TO DO A @SYM ON THE FILE AFTER IT IS CLOSED IN ORDER TO PUT THE FILE IN THE INPUT QUEUE.

9.6.3. DEVICE TYPES AND RECOVERY PROCEDURES

THE DEVICES HANDLED BY SYMBIONT OPERATION AND THE RECOVERY PROCEDURES ARE INDIVIDUALLY DISCUSSED. THE GENERAL ERROR MESSAGE FORMAT DISPLAYED ON THE OPERATOR'S CONSOLE IS

```
SNAME  ERROR
```

WHERE SNAME IS THE SYMBIONT NAME AND ERROR IS THE SYMBOLIC IDENTIFICATION OF THE INTERRUPT. THE ANSWER IS IN THE FORM

```
SM  SNAME  F
```

SM IS TO IDENTIFY AND ROUTE ALL ON-SITE SYMBIONT MESSAGES TO THE CONTROL ROUTINE.

SNAME IS THE DEVICE TO WHICH THE MESSAGE APPLIES. IF SNAME IS ILLEGAL, THE MESSAGE 'KEYERR' IS DISPLAYED ON THE OPERATOR'S CONSOLE.

F IS THE ACTION TO BE TAKEN BY THE SYMBIONT. THE 'F' DEFINITIONS ARE

E- END OF INPUT, RUN FILE COMPLETE

I- INITIATE SYMBIONT IF PREVIOUSLY LOCKED (L KEYIN) SUSPENDED (S KEYIN)OR DOWNED. THE MESSAGE 'KEYERR' IS DISPLAYED AT THE CONSOLE IF THE SYMBIONT WAS NOT LOCKED, SUSPENDED OR DOWNED.

L- LOCK SYMBIONT. INHIBITS FURTHER PROCESSING ON THE ASSOCIATED DEVICE UNTIL THE SYMBIONT IS RE-INITIATED WITH A 'I' KEYIN. WHEN LOCKING AN ACTIVE PRINT OR PUNCH SYMBIONT, PROCESSING OF THE CURRENT FILE IS COMPLETED PRIOR TO EFFECTING THE 'LOCK' ON THE DEVICE. TO LOCK AN INPUT SYMBIONT, IT MUST BE IN AN INACTIVE

STATE. IF THE INPUT SYMBIONT IS ACTIVE THE OPERATOR IS NOTIFIED WITH THE MESSAGE 'SNAME ACTIVE'.

R- REPRINT OR REPUNCH (IN FORM RXX) WHERE XX IS THE NUMBER OF: 1) PAGES TO REPRINT OR, 2) CARDS TO BE REPUNCHED.

```
SM SNAME R3
```

REPRINT OR REPUNCH THE PREVIOUS THREE PAGES OR CARDS. THE MAXIMUM FOR REPRINTING OR REPUNCHING IS 63(DEC.)

SKIP (IN FORM R+XX) WHERE XX IS THE NUMBER OF: 1) PAGES TO SKIP OR, 2) CARDS TO SKIP.

```
SM  SNAME R+3
```

SKIP THE NEXT THREE PAGES OR CARDS. THE MAXIMUM FOR SKIPPING IS 63(DEC.)

T- TERMINATE CURRENT FILE. FOR INPUT THE RUN FILE IS DISCARDED. FOR OUTPUT THE CURRENT FILE IS TERMINATED AND PROCESSING WILL CONTINUE WITH THE NEXT FILE OF THE QUEUE.

S- SUSPEND SYMBIONT. THE SPECIFIED OUTPUT SYMBIONT IS IMMEDIATELY SUSPENDED FROM FURTHER PROCESSING. PROCESSING OF THE FILE IS CONTINUED WITH THE 'I' KEYIN. INPUT SYMBIONTS CANNOT BE SUSPENDED.

X- DELETE FILE FROM OUTPUT QUEUE IN THE FORM 'X F', WHERE 'F' IS THE FILE NAME TO WHICH THE MESSAGE IS BEING DIRECTED. THE 'SNAME' FIELD IS DROPPED FOR THIS MESSAGE. THIS MESSAGE HAS THE FORM

```
SM X RUNID/FILENAME
```

THE 'RUNID' IS REQUIRED TO SPECIFY THE PARTICULAR RUN TO WHICH THE MESSAGE APPLIES. FILENAME IS USED TO DENOTE THE SPECIFIC OUTPUT FILE OF A RUN THAT IS TO BE DELETED, AND IS EITHER THE NAME OF A FILE ENTERED INTO THE OUTPUT QUEUE WITH A @SYM CONTROL STATEMENT, OR IT IS DEFINED AS PRINT$ OR PUNCH$ IF THE FILE WAS QUEUED BY PRINT$ OR PUNCH$ RESPECTIVELY. THERE ARE VARIOUS WAYS OF SPECIFYING FILE DELETIONS, WHICH MAY BE BEST ILLUSTRATED WITH THE FOLLOWING EXAMPLES. FOR BREVITY IN THESE EXAMPLES, ONLY THE 'RUNID/FILENAME' PORTION OF THE MESSAGE IS USED.

```
RUNID
RUNID/FILENAME
RUNID/FILENAME-N
RUNID/FILENAME-NX
```

WHEN ONLY THE RUNID IS SPECIFIED, ALL FILES OR PORTIONS OF FILES RELATED TO THIS RUN AND STILL REMAINING IN THE OUTPUT QUEUE ARE DELETED. 'N' REFERS TO A BREAKPOINT NUMBER AND SPECIFIES THAT ONLY THE NTH SEGMENT OF THE OUTPUT FILE IS TO BE DELETED. 'NX' SPECIFIES THE DELETION OF ALL THE REMAINING SEGMENTS, INCLUDING N, FROM THE OUTPUT QUEUE. BREAKPOINT NUMBERS MUST BE PRECEDED WITH A MINUS (-)

SIGN.

9.6.3.1. 900/300 CARD SUBSYSTEM

CARD READER

SYMBIONT NAME 'CRX'

READER INITIATION

THE OPERATOR LOADS THE RUN CARDS INTO THE INPUT HOPPER THEN DEPRESSES THE READY (CLEAR) SWITCH AND THE SYSTEM START SWITCH. THE 'CR' SYMBIONT IS ACTIVATED VIA THE INTERLOCK PROBE CYCLE WHEN A CARD IMAGE IS TRANSFERRED FROM THE DEVICE.

CONSOLE MESSAGES

1) INTERLOCK (OCTAL 74)

```
SNAME  INTLK  ET
```

THIS TYPEOUT IS THE RESULT OF INTERLOCK DETECTION AT ANY POSITION WITHIN THE INPUT STREAM. THE CR SYMBIONT WILL PROBE THE READER FOR REMOVAL OF THE INTERLOCK CONDITION. WHEN FOUND TO BE INTERLOCK FREE, THE READ OPERATION WILL CONTINUE. IF ANY ACTION OTHER THAN CONTINUATION IS DESIRED, THE 'SM' MESSAGE MUST BE USED PRIOR TO CLEARING THE READER WITH ONE OF THE FOLLOWING OPTIONS:

T- DISCARD THE CURRENT FILE.
E- END OF INPUT, RUN FILE COMPLETE. (THIS IS THE SAME ACTION AS PLACING A @FIN CARD IN THE READER AND REMOVING INTERLOCK.)

2) ILLEGAL CHARACTER (OCTAL 70)

THE MESSAGE DISPLAYED BY THE SYMBIONT INCLUDES THE OPTIONS FOR RECOVERY. THE OPERATOR MUST RESPOND WITH

```
SNAME ILCHAR  AES
```

THE CARD CONTAINING THE ILLEGAL CHARACTER IS CHANNELED TO THE NORMAL STACKER. DEPENDING ON THE POSITION OF THE CARD IN ERROR IN THE RUN STREAM, THERE COULD BE 0, 1 OR 2 CARDS FOLLOWING. IF THE CARD IN ERROR IS AT LEAST 3 CARDS FROM THE END OF THE DECK, THERE WILL BE TWO TRAILING CARDS. THE TRAILING CARDS ARE FED THROUGH THE READER TO MAINTAIN THE PROPER CARD SEQUENCE. TO RE-READ THE ERROR AND ANY TRAILING CARDS, THEY SHOULD BE REPLACED IN THE INPUT HOPPER, DEPRESS THE 'READY' (CLEAR) SWITCH AND ANSWER THE MESSAGE WITH THE 'A' OPTION. THE AVAILABLE OPTIONS ARE:

A- RE-INITIATE READ CYCLE. CONTINUE PROCESSING CURRENT CARD DECK.
E- END OF INPUT. THIS RESPONSE SHOULD BE THE RESULT OF THE HOPPER BEING EMPTY AND THE ILLEGAL CHARACTER IN THE @FIN CARD. RUN FILE INVOLVED IS ENTERED INTO THE SCHEDULE QUEUE.
T- TERMINATES THE RUN CURRENTLY BEING READ.

3) LIGHT/DARK ERROR (OCTAL 54)

THE MESSAGE DISPLAYED BY THE SYMBIONT INCLUDING THE OPTIONS FOR RECOVERY IS:

```
SNAME  LTDRK AES
```

A- RE-INITIATES READ CYCLE. CONTINUE PROCESSING CURRENT RUN STREAM.
E- END OF INPUT. ALL PREVIOUSLY READ CARDS OF THIS

RUN FILE ARE ENTERED INTO THE SYSTEM. SAME ACTION AS @FIN CARD.
T- TERMINATES THE RUN CURRENTLY BEING READ.

TO RECOVER FROM THIS ERROR:

1- REMOVE CARDS FROM INPUT HOPPER,
2- DEPRESS OFF-LINE SWITCH (LIGHT ON),
3- DEPRESS THE MANUAL FEED SWITCH TO REMOVE CARD FROM READY STATION,
4- REMOVE CARDS FROM ALTERNATE STACKER AND PLACE AT BOTTOM OF INPUT HOPPER,
5- REPLACE REMAINDER OF DECK IN THE HOPPER,
6- DEPRESS OFF-LINE (LIGHT OFF), READY AND START SWITCHES, AND
7- ANSWER MESSAGE WITH 'A' OPTION.

4) SYNCRONIZER COUNTER ERROR (OCTAL 20)

SNAME CTRERR AET

5) SYNCRONIZER SEQUENCE ERROR (OCTAL 30)

SNAME SEQERR AET

6) ILLEGAL FUNCTION (OCTAL 50)

SNAME UNDFNC XX AET

THE LAST FUNCTION SENT TO THE SUBSYSTEM WAS AN ILLEGAL CODE AND IS DISPLAYED IN THE MESSAGE 'XX' FIELD OF THE IF 'XX' IS A LEGAL FUNCTION CODE, THIS IS PROBABLY AN INDICATION OF HARDWARE FAILURE. IF 'XX' IS NOT A LEGAL FUNCTION CODE, THIS PROBABLY INDICATES SOFTWARE FAILURE.

7) UNDEFINED INTERRUPT

SNAME UNEXP-FF EI AET

THIS MESSAGE IS THE RESULT OF AN UNDEFINED INTERRUPT RECEIVED FROM THE SUBSYSTEM. THE 'FF' FIELD IS THE LAST FUNCTION CODE USED WITH RESULTANT INTERRUPT REPRESENTED IN THE 'EI' FIELD

CARD PUNCH

SYMBIONT NAME 'CPX'

PUNCH INITIATION
THIS SYMBIONT IS ACTIVATED WHEN A CARD PUNCH FILE IS ENTERED INTO ITS QUEUE VIA THE PUNCH$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PUNCH TERMINATION
AT THE CONCLUSION OF PROCESSING EACH PUNCH FILE, THREE(3) CARDS ARE FED THROUGH THE PUNCH UNIT AS FILE SEPARATORS.

PCHCN$ CONTROL FUNCTIONS
THE CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'C' AND 'S'.

CONSOLE MESSAGES
1)INTERLOCK (OCTAL 74)

SNAME INTLK ART

THIS MESSAGE IS THE RESULT OF AN INTERLOCK CONDITION ON THE PUNCH UNIT AND REQUIRES A RESPONSE. THE PUNCH UNIT MUST BE CLEARED PRIOR TO ANSWERING THE MESSAGE.

A- CONTINUE.
R- REPUNCH XX CARDS
T- TERMINATE CURRENT PUNCH FILE

2) PUNCH CHECK ERROR (OCTAL 54)
THE MESSAGE DISPLAYED BY THE SYMBIONT INCLUDES THE OPTIONS FOR RECOVERY. THE OPERATOR MUST RESPOND WITH ONE OF TWO OPTIONS TO THE FOLLOWING MESSAGE

SNAME PNCHCK ART

THE CARD CONTAINING THE PUNCH CHECK ERROR AND THE TWO FOLLOWING PUNCHED CARDS WILL BE CHANNELLED INTO THE ERROR STACKER. THE 'PUNCH CHECK' SWITCH ON THE OPERATOR'S CONTROL MUST BE DEPRESSED PRIOR TO THE OPERATOR RESPONDING TO THE MESSAGE. THE OPTIONS ARE:

A- ATTEMPT RE-PUNCHING OF THE ERROR AND TWO FOLLOWING CARDS.
R- REPUNCH XX CARDS.
T- TERMINATES CURRENT PUNCH FILE.

DOWNING THE SUBSYSTEM

WHEN THE CARD READER IS DECLARED DOWN DURING THE PROCESS OF READING A RUN FILE, THE RUN FILE INVOLVED IS DISREGARDED. WHEN THE PUNCH UNIT IS DECLARED DOWN DURING THE PROCESSING OF A PUNCH FILE, THE SYSTEM WILL SAVE THE PREVIOUS PUNCHED 16 CARDS AND THE REMAINDER OF THE FILE. AN ATTEMPT WILL BE MADE TO REDIRECT THIS FILE TO OTHER PUNCH EQUIPMENT IF AVAILABLE.

9.6.3.2. 0755 HIGH SPEED PRINTER SUBSYSTEM

SYMBIONT NAME 'PRX'

PRINTER INITIATION
THIS SYMBIONT IS ACTIVATED WHEN A PRINT FILE IS ENTERED INTO ITS QUEUE VIA THE PRINT$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PRNCN$ CONTROL FUNCTIONS
THE CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'H','L','M' AND 'S'.

CONSOLE MESSAGES
1) INTERLOCK (OCTAL 74)

SNAME INTLK ART

THIS MESSAGE IS THE RESULT OF AN INTERLOCK CONDITION ON THE PRINTER AND REQUIRES ONE OF THE ABOVE RESPONSES. THE UNIT MUST BE CLEARED PRIOR TO ANSWERING THE MESSAGE.

A- CONTINUE PRINTING.
R- REPRINT XX PAGES
T- TERMINATE THE CURRENT PRINT FILE.

2) OUT OF PAPER WARNING (OCTAL 44)

SNAME OUTPAP ART

THIS MESSAGE IS DISPLAYED WHEN THE PRINTER HAS RUN OUT OF PAPER AND WHEN THE LAST LINE OF THE PAGE HAS BEEN PRINTED. POSITION THE PAPER IN THE PRINTER TO THE TOP LINE OF THE PAGE. PRINTING IS RESUMED WITH A NEW PAGE.

9.6.3.3. 1004 SUBSYSTEM

CARD READER

SYMBIONT NAME 'CRX'

READER INITIATION

TO PREPARE THE 1004 FOR CARD READING:

1. LOAD INPUT HOPPER WITH CARDS
2. ADD TERMINATION CONTROL CARDS
3. IF THE READ WAIT STATION IS UNOCCUPIED, DEPRESS THE STOP, FEED AND RUN SWITCHES.
4. DEPRESS ALTERATION SWITCH 1.

SENSING OF ALTERATION SWITCH 1 FOR READING INITIATION IS PASSED ON TO THE 1108 SYMBIONT CONTROL ROUTINE WHICH INITIATES THE LOADING OF THE SYMBIONT. THE ALTERATION SWITCH IS INTERNALLY CLEARED BY THE 1004 PRIOR TO READING THE FIRST CARD. (THE SWITCH LIGHT REMAINS LIT UNTIL CLEARED BY THE OPERATOR. I.E. PRIOR TO READING ANOTHER CARD DECK ALTERATION SWITCH 1 MUST BE DEPRESSED TWICE.)

THE FIRST CARD READ WILL ALWAYS BE IN THE MODE SPECIFIED AS STANDARD AT SYSTEMS GENERATION. TO READ SOME OTHER MODE THE FIRST CARD WOULD BE A @COL CONTROL CARD AND WOULD INDICATE THE MODE TO READ NEXT. WITH THE ADDITION OF THE EXTERNAL INTERRUPT FEATURE TO THE 1004, THE EMPTY HOPPER CONDITION CAN BE RECOGNIZED. AT THE END OF INPUT DECK FOR CARDS IS A @FIN CARD FOLLOWED BY AN EMPTY HOPPER INTERRUPT. FOR CODE IMAGE CARDS, THE END OF INPUT DECK IS SIGNALED BY A CARD CONTAINING THE SIX CHARACTER SENTINEL SPECIFIED ON THE @COL CONTROL CARD. THIS SENTINEL BEGINS IN COLUMN ONE OF THE CARD. IF NO SENTINEL IS SPECIFIED, THE STANDARD SENTINEL '@ENDCL' IS ASSUMED.

CONSOLE MESSAGES

THE TIMEOUT MESSAGE

C/U IOARB TIMEOUT ABDEG

IS THE RESULT OF THE 1108 NOT RECEIVING A REQUESTED CARD IMAGE FROM THE 1004. SEE SECTION ON THE ARBITRARY DEVICE HANDLER (10.5) FOR THE ALLOWABLE RESPONSES AND THEIR MEANINGS.

THE TIMEOUT WILL ONLY OCCUR IF THE 1004 IS NOT ON (POWER OFF) OR IN A STOP CONDITION (THE STOP LIGHT LIT). THE ADDITION OF THE EXTERNAL INTERRUPT PROVIDES FOR AN INTERRUPT RETURNED ON READ JAMS, EMPTY HOPPER, MISFEEDS ETC. IN THESE CASES THE MESSAGE

CRX INTLK ET

WILL APPEAR. THE SYMBIONT WILL CONTINUE TO REQUEST THE CARD FROM THE 1004. IF ANY ACTION OTHER THAN RECOVERY IS DESIRED THE 'SM' MESSAGE MUST BE USED BEFORE RECOVERY IS COMPLETE. THE OPTIONS ARE:

E-END OF INPUT
T-TERMINATE CURRENT FILE

THE MESSAGE

CRX ILCHAR   AE

WILL APPEAR IF AN ILLEGAL CHARACTER WAS DETECTED ON AN ATTEMPTED READ. THE SYMBIONT WILL AWAIT AN ANSWER TO THIS MESSAGE BEFORE ATTEMPTING TO RESUME OPERATIONS. THE THE OPERATOR OPTIONS ARE:

A- RESUME OPERATIONS
E- CLOSE THIS FILE

PRINTER

SYMBIONT NAME 'PRX'

PRINTER INITIATION
THIS SYMBIONT IS ACTIVATED WHEN A PRINT FILE IS ENTERED INTO ITS QUEUE VIA THE PRINT$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PRNCN$ CONTROL FUNCTIONS
THE LIST OF CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'H', 'L', 'M' AND 'S'.

CONSOLE MESSAGES

C/U IOARB TIMEOUT ABDEG

IS THE RESULT OF THE 1108 NOT RECEIVING A RESPONSE FROM THE LAST PRINT FUNCTION. SEE SECTION ON THE ARBITRARY DEVICE HANDLER (10.5) FOR THE ALLOWABLE RESPONSES AND THEIR MEANINGS.

PRX PRINT ERROR RT
PRX PRINT ABNML RT

THESE MESSAGES ARE THE RESULT OF AN EXTERNAL INTERRUPT RETURNED ON AN ATTEMPT TO PRINT ON A 1004. THE PRINTER ERROR MESSAGE INDICATES THE 'OUT OF PAPER' OR 'RUN-AWAY PAPER' CONDITIONS, WHILE PRINTER ABNORMAL INDICATES THAT THE PRINT CARRIAGE IS OUT OR THE CHANGE RIBBON POSITION HAS BEEN REACHED. THE SYMBIONT WILL CONTINUE TO ATTEMPT RECOVERY UNTIL THE CONDITION IS CORRECTED AT WHICH TIME NORMAL OPERATION WILL CONTINUE. IF ANY ACTION OTHER THAN RECOVERY IS REQUIRED THE 'SM' MESSAGE MUST BE USED PRIOR TO CLEARING THE ERROR CONDITION. THE OPTIONS USED ARE
R-REPRINT XX PAGES
T-TERMINATE CURRENT PRINT FILE

CARD PUNCH

SYMBIONT NAME 'CPX'

PUNCH INITIATION
THIS SYMBIONT IS ACTIVATED WHEN A CARD PUNCH FILE IS ENTERED INTO ITS QUEUE VIA THE PUNCH$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PCHCN$ CONTROL FUNCTIONS
THE CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'C' AND 'S'.

CONTROL MESSAGES

C/U IOARB TIMEOUT ABDEG

THIS IS THE RESULT OF THE 1108 NOT RECEIVING A RESPONSE FROM THE LAST PUNCH FUNCTION. SEE THE SECTION ON THE ARBITRARY DEVICE HANDLER (10.5) FOR THE ALLOWABLE OPTIONS AND THEIR MEANINGS.

```
CPX PUNCH B JAM     RT
CPX PUNCH ABNML     RT
CPX PUNCH ERROR     RT
```

ONE OF THE ABOVE MESSAGES WILL APPEAR AS THE RESULT OF AN ERROR CONDITION ON AN ATTEMPT TO PUNCH. THE 'B JAM' MESSAGE OCCURS WHENEVER A B-JAM IS DETECTED. THE 'ERROR' MESSAGE INDICATES AN A-JAM OR AN INTERLOCK CONDITION, AND THE 'ABNORMAL' MESSAGE INDICATES STACKER FULL, STACKER JAM, FULL CHIP BOX OR EMPTY HOPPER CONDITIONS. THE SYMBIONT WILL CONTINUE TO SEND THE PUNCH FUNCTION UNTIL RECOVERY IS MADE. IF ANY OTHER ACTION IS REQUIRED, THE 'SM' MESSAGE MUST BE USED BEFORE CONDITIONING THE 1004. THE OPTIONS ARE

R-REPUNCH XX CARDS
T-TERMINATE CURRENT PUNCH FILE

```
CPX PUNCH CHECK     RT
```

THIS MESSAGE APPEARS ONLY AFTER SIX (6) ATTEMPTS HAVE BEEN MADE AT RECOVERY WITHOUT SUCCESS. THE SYMBIONT WILL ATTEMPT RECOVERY. IF ANY ACTION OTHER THAN RECOVERY IS DESIRED THE 'SM' MESSAGE IS USED WITH THE SAME OPTIONS AS FOR THE OTHER PUNCH ERRORS.

PAPER TAPE READ

SYMBIONT NAME   'PTRX' FOR ALL LEVELS

THE TWO MODES OF READING PAPER TAPE ARE TRANSLATE MODE AND BINARY MODE. IN THE TRANSLATE MODE A STANDARD CODE WILL BE ASSUMED AND TRANSLATION WILL BE FOR THIS CODE. IN THE BINARY MODE NO TRANSLATION WILL BE DONE BY THE SYMBIONT. IMAGES WILL BE PASSED AS THEY OCCUR FOR THE WORKER PROGRAM TO TRANSLATE IN ANY CODE DESIRED. ALL READS INITIATED FROM THE 1004 ARE ASSUMED TO BE IN THE TRANSLATE MODE.

ALL FOUR LEVELS OF PAPER TAPE READING ARE PROVIDED. FIVE AND SIX LEVEL TAPE IS READ IN THE 80 COLUMN MODE, AND SEVEN AND EIGHT LEVEL TAPE IS READ IN THE CODE IMAGE MODE. FOR SEVEN AND EIGHT LEVEL, TAPE THE BITS WILL BE TRANSPOSED BY THE SYMBIONT SO THAT THEY OCCUR IN THE SAME ORDER IN THE BUFFER AS ON TAPE.

TO INITIATE PAPER TAPE READING FROM THE 1004, THE PROBE ROUTINE AND AN ALTERATION SWITCH IS USED. EVERY 15 SECONDS THE PROBE ROUTINE CHECKS FOR A SET ALTERATION SWITCH. IF THE SWITCH IS SET, THE PROBE ROUTINE WILL INITIATE THE PAPER TAPE READ SYMBIONT FOR THE MODE SET AT SYSTEM GENERATION AS STANDARD. FOR ANY OTHER MODE A @COL STATEMENT MUST BE USED. THE SWITCH USED AND ITS MEANING IS

ALT 2- INITIATE PAPER TAPE READ

TO INITIATE PAPER TAPE READING FROM A CARD INPUT STREAM, THE @COL CONTROL STATEMENT IS USED. ON ALL PAPER TAPE READS TRAILER AND LEADER ARE OMITTED AS

DATA. DELETE CODES ARE ONLY RECOGNIZED IN TRANSLATE MODE.

CONSOLE MESSAGES

THE TIME OUT MESSAGE

```
C/U IOARB TIMEOUT   ABDEG
```

IS THE RESULT OF THE 1108 NOT RECEIVING THE REQUESTED PAPER TAPE IMAGE FROM THE 1004. SEE THE SECTION ON THE ARBITRARY DEVICE HANDLER (10.5) FOR THE ALLOWABLE OPTIONS AND THEIR MEANINGS.

PAPER TAPE PUNCH

SYMBIONT NAME 'PTPX' FOR ALL LEVELS

THIS SYMBIONT IS ACTIVATED WHENEVER A PAPER TAPE PUNCH FILE IS ENTERED INTO ITS QUEUE. TWO MODES OF PUNCHING ARE PROVIDED, (1)TRANSLATE MODE AND (2) BINARY MODE. IN BINARY MODE THE DATA IS PASSED TO THE PUNCH AS IT OCCURS WITH NO TRANSLATION. IN THE TRANSLATE MODE, THE DATA IS ASSUMED TO BE FIELD-DATA AND IS CONVERTED TO A STANDARD CODE BEFORE PUNCHING.
ALL FOUR LEVELS OF PAPER TAPE PUNCHING ARE PROVIDED FOR. FIVE AND SIX LEVEL TAPES ARE PUNCHED IN THE 80 COLUMN MODE AND SEVEN AND EIGHT LEVEL TAPES ARE PUNCHED IN THE CODE IMAGE MODE.

CONSOLE MESSAGES

```
C/U IOARB TIMEOUT   ABDEG
```

THIS IS THE RESULT OF THE 1108 NOT RECEIVING A RESPONSE FROM THE PREVIOUS PAPER TAPE PUNCH FUNCTION. SEE THE SECTION ON THE ARBITRARY DEVICE HANDLER (10.5) FOR THE ALLOWABLE OPTIONS AND THEIR MEANINGS.

9.7. REMOTE 1004-BATCH OPERATION

SYMBIONT INITIATION

THE REMOTE BATCH MODE IS CONDUCTED IN A MANNER SIMILAR TO ON-SITE SYMBIONT OPERATIONS, ONCE THE REMOTE LINE CONNECTION HAS BEEN ESTABLISHED. INITIATION OF A SYMBIONT OPERATION COMMENCES WHEN THE SITE 'ID' AND 'READY' COMMAND ARE RECEIVED FROM THE REMOTE STATION.

CONTROL OF THE REMOTE RUNS SUBMITTED TO THE SYSTEM IS MAINTAINED WITH THE REMOTE OPERATION. VARIOUS CONTROL COMMANDS ARE AVAILABLE WITH ALTERATION SWITCH SETTINGS RELAYED TO THE 1108. WITHIN THE AVAILABLE COMMANDS IS THE CAPABILITY OF THE REMOTE OPERATOR TO SIGNAL THE 1108 OPERATOR TO SWITCH FROM DATA TO VOICE COMMUNICATION. THE REMOTE OPERATOR CAN THAN DIRECT THE ON-SITE OPERATOR TO INFORM THE SYSTEM OF ERROR RECOVERY PROCEDURES, OR ANY OTHER ACTION TO BE TAKEN BY THE SYMBIONT, VIA UNSOLICITED CONSOLE MESSAGES.

TO PROPERLY INITIATE THE 1004 THE FOLLOWING STEPS SHOULD BE TAKEN

1. AT LEAST ONE CARD SHOULD BE IN THE INPUT HOPPER. THIS SHOULD EITHER BE A @RUN CONTROL CARD OR A BLANK CARD TO OCCUPY THE READ WAIT STATION. THIS INSURES A CONTINUOUS OPERATION WHEN READING IS TO BEGIN.
2. DEPRESS THE CLEAR, START, FEED AND RUN SWITCHES.

3. DEPRESS ALTERATION SWITCH 1 THEN SWITCH 4. THE ABOVE THREE STEPS CAN EITHER BE TAKEN BEFORE OR AFTER THE COMMUNICATION LINE HAS BEEN SWITCHED TO THE DATA MODE. THE SYSTEM THEN REQUESTS THE SITE ID. IF DETERMINED AS A VALID ID THE PRINTOUT 'READY' IS SENT TO THE PRINTER. THE ABSENCE OF THE 'READY' PRINTOUT FOR A PREDETERMINED INTERVAL (ABOUT ONE MINUTE) INDICATES THAT EITHER NO ID OR AN INVALID ID WAS RECEIVED BY THE SYSTEM. THE DATA-SETS SHOULD BE SWITCHED TO VOICE BY THE OPERATORS TO DETERMINE FURTHER ACTION. IF NO ID WAS RECEIVED BY THE SYSTEM THE LINE CONNECTION MAY BE FAULTY AND RE-DIALING MAY BE NECESSARY.

A REMOTE RUN IS ENTERED INTO THE SCHEDULE QUEUE WHEN THE @FIN CARD IS RECEIVED FROM THE 1004. THE ENTIRE RUN STREAM IS DIRECTED TO MASS STORAGE EXCEPT WHEN A @FILE CARD IS ENCOUNTERED. IF THE DESIGNATED REPOSITORY FOR THE DATA FILE IS MAGNETIC TAPE THE INPUT TRANSMISSION WILL BE DISCONTINUED UNTIL A TAPE HAS BEEN LOADED BY ON-SITE PERSONNEL.

INITIATING A SYMBIONT OPERATION FROM THE ON-SITE LOCATION TO A REMOTE SITE IS ACCOMPLISHED BY RUN FILE EXECUTION THROUGH DEVICE ASSOCIATION, OR BY A @SYM CONTROL STATEMENT. THE OUTPUT SYMBIONT DIRECTING THE FILE TRANSMISSION INITIATES THE DIALING PROCEDURES TO THE REMOTE SITE THROUGH THE COMMUNICATIONS HANDLER.

SCHEDULING OF A SYMBIONT OPERATION TO TRANSMIT FILES TO A REMOTE SITE IS PROVIDED WITH THE @SYM CONTROL STATEMENT. FILES SENT TO REMOTE SITES ARE PROCESSED INDIVIDUALLY BY DEVICE, BUT ONE DEVICE ORIENTED SYMBIONT IS CAPABLE OF PROCESSING N FILES BEING SENT TO N LIKE REMOTE DEVICES. THUS THE ABILITY EXISTS TO SEND A SINGLE CATALOGUED MASS STORED FILE TO N REMOTE SITES AS A CONCURRENT OPERATION WITH UNIQUE CONTROL OF EACH DEVICE. TRANSMISSION OF A SINGLE TAPE FILE TO N REMOTE SITES IS HANDLED AS A SERIAL OPERATION BY THE SYMBIONT.

ALTERATION SWITCH COMMAND DEFINITIONS

THE 1004 ALTERATION SWITCHES ARE USED TO DEFINE COMMANDS AVAILABLE TO THE REMOTE OPERATOR. THE SETTING OF THESE SWITCHES ARE INTERPRETTED BY THE 1004 AND RELAYED TO THE 1108 'REMOTE' SYMBIONT. ALTERATION SWITCH 4 IS THE 'EXECUTE' COMMAND WHICH INITIATES THE INTERROGATION OF THE REMAINING THREE SWITCHES. THUS ANY COMBINATION OF ALTERATION SWITCHES 1,2 AND 3 WHICH DESIGNATES THE COMMAND TO BE TAKEN MUST BE SET PRIOR TO THE 'EXECUTE' SWITCH.

ALTERATION SWITCH COMMANDS

SWITCHES	COMMAND	ACTION
1	READY	INITIAL 'READY' - NOTIFIES THE 1108 SYSTEM THAT THE REMOTE 1004 IS 'READY' FOR OPERATION AND THAT A NEW USER IS TAKING CONTROL OF THE CHANNEL. THE SYMBIONT BEGINS TO REFERENCE THE 1004 AND THEREAFTER MAINTAINS COMMUNICATIONS BETWEEN THE TWO SITES. THE SYMBIONT ASSUMES THAT THE 1004 SITE IS READY TO RECEIVE PRINT FILES AND PUNCH FILES AT THIS TIME. 'READY'AFTER HALT (SEE ACTION OF 'HALT' COMMANDS) - NOTIFIES THE SYMBIONT THAT COMMUNICATIONS ARE TO BE RESUMED AND THAT THE INPUT/OUTPUT OPERATIONS WHICH WERE IN PROGRESS AT THE TIME OF THE 'HALT' ARE TO BE CONTINUED.
2	READ	NOTIFIES SYMBIONT TO READ ONE OR MORE RUN FILES AND SUBMIT THEM FOR EXECUTION. EACH RUN FILE IS HEADED BY A RUN CONTROL CARD. THE LAST FILE TO BE READ ON THIS COMMAND MUST BE TERMINATED BY A STOP CARD (PLUS ADDITIONAL CARD FOR THE WAIT STATION). THE ADDITIONAL CARD COULD BE THE RUN CONTROL CARD OF THE FIRST FILE OF THE NEXT GROUP.

3	HALT	NOTIFIES SYMBIONTS TO HALT ALL COMMUNICATIONS ON THE 1004 CHANNEL. WHEN THE 'READY'COMMAND IS EXECUTED, ALL OPERATIONS WILL RESUME.
1,2	HALT, GO VOICE	SAME AS 3 (HALT), EXCEPT THE 1108 OPERATOR IS NOTIFIED TO PLACE THE DATA LINE IN THE 'TALK' MODE FOR VOICE COMMUNICATIONS WITH THE 1004 OPERATOR. AFTER BOTH OPERATORS SWITCH BACK TO THE 'DATA' MODE, THE 'READY' COMMAND IS USED TO RESUME OPERATIONS.
1,3	ABORT PRINT	NOTIFIES THE PRINT SYMBIONT THAT THE REMAINDER OF THE FILE CURRENTLY BEING PRINTED ON THE 1004 IS TO BE IGNORED AND THAT PRINTING IS TO BEGIN WITH THE NEXT PRINT FILE, IF PRESENT.
2,3	ABORT PUNCH	NOTIFIES THE CARD PUNCH SYMBIONT THAT THE REMAINDER OF THE FILE CURRENTLY BEING PUNCHED ON THE 1004 IS TO BE IGNORED AND THAT PUNCHING IS TO BEGIN WITH THE NEXT PUNCH FILE, IF PRESENT.
1,2,3	OFF LINE	THIS COMMAND IS USED WHEN THE 1004 OPERATOR IS READY TO RELINQUISH THE CHANNEL. THIS COMMAND IS NORMALLY EXECUTED FOR TE/TE OF THE REMOTE OPERATION. WHEN ENCOUNTERED THE SYMBIONT PREPARES FOR TERMINATION OF FURTHER COMMUNICATION. 1) IF INPUT IS BEING ACCEPTED FROM THE 1004, WHEN AN OFF LINE COMMAND IS RECEIVED TRANSMISSION IS CONTINUED UNTIL THE @@ CARD IS DETECTED. WHEN THE OUTPUT BECOMES AVAILABLE FOR THIS RUN, THE SYMBIONT WILL INITIATE DIALING PROCEDURES TO THE SITE OF RUN SUBMISSION. 2) IF OUTPUT IS IN PROGRESS WHEN THE 'OFF LINE' COMMAND IS RECEIVED, THE PRINT AND/OR PUNCH FILE CURRENTLY INVOLVED IN THE TRANSMISSION IS TO BE COMPLETED PRIOR TO HANGING UP. ARRANGEMENTS MUST BE MADE BETWEEN THE 1004 AND 1108 OPERATORS FOR THE DISPOSITION OF ANY OUTPUT FILE REMAINING IN THE QUEUE DESTINED FOR THE REMOTE SITE. THE 'SR'MESSAGE OPTION 'X', IS AVAILABLE FOR THOSE QUEUED OUTPUT FILES WHICH ARE TO BE DELETED FROM TRANSMISSION TO SITE 'ID'.

TERMINATION CARDS

THE TERMINATION CARDS ARE THE STOP CARD, IN WHICH COLUMNS 1 AND 2 MUST CONTAIN A 7-8 PUNCH, FOLLOWED BY AT LEAST ONE ADDITIONAL CARD TO OCCUPY THE 1004 READ WAIT STATION. PRESENCE OF THE STOP CARD TERMINATES THE READ OPERATION. THE 'READ' COMMAND MUST BE EXECUTED FOR ANY FURTHER READING. THE ADDITIONAL CARD SHOULD EITHER BE A'BLANK CARD OR A @RUN CARD OF A SUCCEEDING RUN.

GENERAL INFORMATION

IN ORDER TO PREVENT UNNECESSARY 'TIME-OUTS'AT THE 1108, THE 1004 STOP BUTTON SHOULD NEVER BE DEPRESSED EXCEPT UPON INITIAL CARD LOADING OR DURING THE HALT PERIOD.

IF READING IS TO OCCUR, THE OPERATOR MUST ALWAYS PROVIDE A CARD FOR THE WAIT STATION.

IF A CARD JAM OCCURS, REMOVE THE UNREAD CARDS (INCLUDING THE CARD IN THE WAIT STATION), REPAIR THE DECK, PLACE THE CARDS IN THE HOPPER, AND DEPRESS FEED AND RUN.

A DELAY AT THE 1004 WILL CAUSE THE 1108 TO 'TIME-OUT' AND GIVE THE 1108 OPERATOR THE OPPORTUNITY TO REINITIATE THE FUNCTION. THE 1004 OPERATOR IS GIVEN AT LEAST 5 MINUTES TO MAKE REPAIR (FIX CARD JAMS, PUT IN PAPER, ETC.).

TWO REMOTE SYSTEMS ARE PROVIDED. 'RMS1' (REMOTE SYSTEM #1) IS TO BE USED WITH VOICE GRADE LINES (2400 BPS) AND UTILIZES THE COMPRESSION/DECOMPRESSION TECHNIQUES DEVELOPED FOR THE 1107 REMOTE 1004. ONLY 80 COLUMN READING AND PUNCHING PLUS PRINTING IS PROVIDED WITH THIS SYSTEM.

TO TAKE ADVANTAGE OF THE INCREASED SPEEDS PROVIDED BY A TELPAK*A SYSTEM (40800 BPS) A NEW SYSTEM CALLED 'RMS2' IS PROVIDED. WITH THIS SYSTEM THERE IS NO COMPRESSION OR BUFFERING, BUT RATHER A ONE-TO-ONE RELATIONSHIP EXISTS BETWEEN REQUESTS AND CARDS READ OR PUNCHED, LINES PRINTED, ETC. 80 COLUMN, COLUMN BINARY AND PAPER TAPE READING/PUNCHING AND PRINTING CAPABILITIES ARE PROVIDED WITH 'RMS2'.

1108 OPERATOR ASSISTANCE

THE MSG CONTROL CARD IS USED TO TYPE A MESSAGE TO THE 1108 OPERATOR. A MSG CARD OR GROUP OF MSG CARDS CAN BE USED BY THE REMOTE PROGRAMMER OR OPERATOR TO REQUEST ASSISTANCE FROM THE 1108 OPERATOR. THE MESSAGES ARE PRINTED ON BOTH THE CONSOLE/PRINTER AND THE REMOTE PRINTER.

ON-SITE CONTROL

CERTAIN MALFUNCTION RECOVERY PROCEDURES REQUIRE INFORMATION TO BE PASSED ON TO THE SYMBIONT IN ORDER TO ALTER ITS DIRECT INPUT OR OUTPUT OF REMOTE FILES. THESE PROCEDURES ARE GOVERNED BY VOICE COMMUNICATIONS FROM THE 1004 OPERATOR TO THE 1108 OPERATOR. THE 1108 OPERATOR THEN TYPES IN AN UNSOLICITED MESSAGE DEFINING THE REQUIRED ACTION TO BE TAKEN. THESE MESSAGES ARE AVAILABLE TO ANY ACTIVE REMOTE SITE.

```
.SR ID A SN
```

SR- IS THE DEFINITION FOR ALL REMOTE SYMBIONT MESSAGES. THIS INFORMS THE CONSOLE ROUTINE TO CHANNEL THIS MESSAGE TO SYMBIONT REMOTE CONTROL ROUTINE.

ID- IS THE SIX CHARACTER IDENTIFICATION OF THE SITE TO WHICH THE MESSAGE IS DIRECTED.

SN- THE NAME OF SYMBIONT TO WHICH THE MESSAGE IS BEING DIRECTED

A- IS THE ACTION TO BE TAKEN BY THE SYMBIONT. THE FIELD DEFINITION OF THE LEGAL CODES ARE

H- HALT REMOTE OPERATION

R- REPRINT OR REPUNCH (IN THE FORM RXX)WHERE 'XX'IS THE NUMBER OF: 1) PAGES TO REPRINT OR, 2)CARDS TO BE REPUNCH. THE 'XX'VALUE HAS AN IMPLIED MINUS (-) SIGN AND HAS A MAXIMUM VALUE OF 63. TO SKIP A PORTION OF A FILE A PLUS (+) SIGN MUST PRECEDE 'XX'. E.G.

```
SR ID R+28 RMPR
```

THIS WILL BYPASS PRINTING OF THE NEXT 28 PAGES OF THE CURRENT PRINT FILE. THE CURRENT PAGE IS COMPLETED AND EXCLUDED FROM THE COUNT.

```
SR ID R5 RMCP
```

REPUNCH LAST FIVE CARDS.

X- THE FILE DELETION FUNCTION IS ONLY AVAILABLE WITH THE SM MESSAGE AND APPLIES TO ANY FILE IN THE PRINT/PUNCH OUTPUT QUEUE, REMOTE OR ON LINE OPERATIONS.

*TRADEMARK OF AMERICAN TELEPHONE AND TELEGRAPH COMPANY

CARD READER

SYMBIONT NAME 'RMCR'

READER INITIATION

TO PREPARE THE 1004 FOR CARD READING:

1. LOAD INPUT HOPPER WITH CARDS
2. ADD TERMINATION CONTROL CARDS.
3. DEPRESS ALTERATION SWITCH 2 THEN SWITCH 4.

THE 'READ' COMMAND IS RELAYED TO THE SYMBIONT CONTROL ROUTINES WHICH INDICATES THE CARD READ SYMBIONT. THE ALTERATION SWITCHES ARE INTERNALLY CLEARED BY THE 1004 PRIOR TO READING THE FIRST CARD. (THE SWITCH LIGHTS REMAIN LIT UNTIL CLEARED BY THE OPERATOR. I.E. PRIOR TO ISSUING ANOTHER COMMAND ALTS 2 AND 4 MUST BE DEPRESSED AGAIN.)

PRINTER

SYMBIONT NAME 'RMPR'

PRINTER INITIATION

THIS SYMBIONT IS ACTIVATED WHEN A PRINT FILE IS ENTERED INTO ITS QUEUE VIA THE PRINT$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PRNCN$ CONTROL FUNCTIONS

THE CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'H', 'L', 'M' AND 'S'. THE SPECIAL FORM MESSAGE IS DISPLAYED AT THE REMOTE TERMINAL AND ON-SITE CONSOLE. VOICE COMMUNICATION IS NECESSARY TO INFORM THE ON-SITE OPERATOR OF THE APPROPRIATE SYMBIONT ACTION.

ON-SITE CONSOLE MESSAGES

THE 'SR' MESSAGE FOR THE PRINT SYMBIONT

```
SR ID RXX RMPR
```

HAS THE FOLLOWING 'A'OPTIONS:

R- REPRINT OR SKIP 'XX' PAGES.

CARD PUNCH

SYMBIONT NAME 'RMCP'

PUNCH INITIATION

THIS SYMBIONT IS ACTIVATED WHEN A CARD PUNCH FILE IS ENTERED INTO ITS QUEUE VIA THE PUNCH$ ROUTINE OR THE @SYM CONTROL STATEMENT.

PCHCN$ CONTROL FUNCTIONS

THE CONTROL FUNCTIONS APPLICABLE TO THIS SYMBIONT ARE 'C' AND 'S'.

ON-SITE CONSOLE MESSAGES

THE 'SR' MESSAGE FOR THE PUNCH SYMBIONT

```
SR ID RXX RMCP
```

HAS THE FOLLOWING OPTIONS:

R- REPUNCH OR SKIP 'XX' CARDS.

9.8. REMOTE DEMAND OPERATION

9.8.1. GENERAL DEMAND TERMINAL HANDLER FEATURES

THE INPUT SYMBIONTS OPERATING IN THE DEMAND MODE ALWAYS ACCEPT INPUT FROM A REMOTE TERMINAL EXCEPT WHEN IN THE @ADD MODE. THUS, THE RATE OF INPUT TO THE SYSTEM IS SUBJECT TO THE DISCRETION OF THE REMOTE OPERATOR. INPUT FROM THE TERMINAL IS BUFFERED IN MAIN STORAGE IN THE SAME MANNER AS AN INPUT RUN STREAM FROM THE ON-SITE CARD SUBSYSTEM.

USE OF THE SYMBIONT INTERFACE ROUTINES (READ$, PRINT$, ETC.) AND THE SYMBIONT CONTROL FUNCTIONS PRNCN$ AND PCHCN$ WHILE IN THE DEMAND MODE ARE AS PREVIOUSLY DEFINED IN THIS CHAPTER.

9.8.2. INITIATION PROCEDURES

AT THE COMPLETION OF THE DIALING OPERATION, THE REMOTE OPERATOR MUST RESPOND WITH HIS SITE ID. THIS SITE ID IS VERIFIED BY THE COMMUNICATION HANDLER AND IF DIRECT SYMBIONT CONTROL OF THE TERMINAL IS SPECIFIED, THE SYMBIONT IS GIVEN CONTROL AND TRANSMITS THE MESSAGE 'READY'. DISPLAY OF THE 'READY' MESSAGE INFORMS THE OPERATOR THAT THE SYMBIONT IS IN THE RECEIVE MODE AND WAITING FOR ITS INITIAL COMMAND.

EACH SUPPORTED DEVICE HAS AN ASSOCIATED HANDLER TO PERFORM THE INITIAL INPUT INTERPRETATION OF CHARACTERS. FOR EACH DEVICE A SMALL SET OF CHARACTERS IS RESERVED TO CONTROL IMAGE FORMATION. BASIC CONTROLS ARE:

END OF IMAGE

THIS CHARACTER MARKS THE END OF AN INPUT IMAGE. IT IS NOT PLACED IN THE IMAGE, BUT DIRECTS THE SYMBIONT TO TRANSMIT THE CURRENT IMAGE AND PREPARE FOR THE NEXT INPUT.

DELETE IMAGE

RECEIPT OF THIS CONTROL CHARACTER CAUSE THE ERASING OF ANY PARTIALLY CONSTRUCTED IMAGE. THE SYMBIONT PREPARES FOR RE-TYPING OF THE IMAGE.

CHARACTER ERASE

RECEIPT OF THE CONTROL CHARACTER ERASES THE PREVIOUSLY INPUT CHARACTER ALLOWING ITS REPLACEMENT BY THE NEXT INPUT. MORE THAN ONE CHARACTER CAN BE ERASED FROM RIGHT TO LEFT BY REPEATED SUBMISSION.

INTERRUPT

INPUT OF THIS CONTROL INTERRUPTS THE PROGRAM CURRENTLY EXECUTING. IT CAN SIGNAL THE PROGRAM TO TAKE AN ALTERNATE ACTION, OR REQUEST INPUT FOR FURTHER DIRECTION.

RUN INITIATION FROM A REMOTE TERMINAL DOES NOT NECESSARILY IMPLY THAT THE CONTROL STREAM MUST BE RECEIVED FROM THAT TERMINAL. THE SYSTEM ALLOWS THE USER TO START A MASS STORED RUN FROM A REMOTE TERMINAL WITH THE 'REMOTE' SYSTEM GENERATION STATEMENT. IF THE RUN INTENDS TO USE THE TERMINAL AS A DEVICE OF THE RUN, THE LINE MUST BE ASSIGNED WITH AN @ASG STATEMENT.

9.8.3. TERMINATION OF DEMAND RUNS

TERMINATION OF A DEMAND RUN OCCURS WHEN A @FIN STATEMENT IS RECEIVED. AN END OF TRANSMISSION (EOT) OUTSIDE A RUN CAUSES THE SYMBIONT TO RELEASE THE LINE ASSOCIATED WITH THE TERMINAL TO THE COMMUNICATIONS HANDLER, WHICH THEN RESUMES MONITORING THE IDLE LINE. IF ANOTHER RUN IS TO BE INITIATED FROM THIS TERMINAL, ONLY THE SITE ID IS REQUIRED TO RE-ESTABLISH SYMBIONT CONTROL.

THE REMOTE OPERATOR SHOULD INSURE, PRIOR TO HIS HANG-UP, THAT THE SYMBIONT IS AWARE OF THE FORTHCOMING DISCONNECT. IF NOT THE SYMBIONT WILL MAINTAIN CONTROL OF THE LINE. WHEN NO TRANSMISSIONS HAVE OCCURRED FOR 15 MINUTES, THE ON-SITE OPERATOR IS INFORMED OF THE INACTIVITY WITH A TIME OUT MESSAGE. IF THE OPERATOR HAS ESTABLISHED THAT THE REMOTE TERMINAL HAS BEEN DISCONNECTED, HE MAY THEN TERMINATE THE

REMOTE OPERATION WITH THE UNSOLICITED SYMBIONT 'SR' MESSAGE.

9.8.3.1. TELETYPE* MODEL 35 ASR-KSR MACHINES

SYMBIONT NAME

TTY35 FOR TELETYPE*MODEL 35 ASR MACHINE (ASCII CODE)

INITIATION PROCEDURES

THE SITE ID MUST BE KEYED IN FROM THE TERMINAL AND IT MUST CONTAIN SIX CHARACTERS. THE INITIAL CHARACTER OF EACH SITE ID ORIGINATING FROM A TELETYPE MUST BE A 'U'. E.G 'USITE1','USITE2'. WHEN THE SITE ID HAS BEEN VERIFIED, THE MESSAGE

UNIVAC 1108 TIME/SHARING EXEC

WILL BE TRANSMITTED TO THE SITE. IF THE SITE ID RECEIVED BY THE 1108 IS EITHER INVALID OR ALREADY IN USE, THE SITE ATTEMPTING TO BE ESTABLISHED WILL NOT BE RECOGNIZED. DUE TO LINE STATIC, PARITY ERRORS, ETC., THE KEYED IN SITE ID MAY HAVE TO BE RETRANSMITTED. IF THE ABOVE MESSAGE HAS NOT BEEN RECEIVED WITHIN APPROXIMATELY 10 SECONDS(THIS RESPONSE IS GENERALLY IMMEDIATE) THE SITE ID SHOULD BE RE-TRANSMITTED. IN WHICH CASE IT MAY BE NECESSARY TO RE-SYNCHRONIZE THE SYSTEM TO YOUR SITE BY TYPING SIX CONSECUTIVE 'X' CHARACTERS FOLLOWED BY THE SITE ID.

THE SYMBIONT IS NOW READY TO RECEIVE INPUT. EACH EXECUTIVE COMMAND IS RECOGNIZED BY THE LEAD CHARACTER # .

CONTROL CHARACTERS

END IMAGE- THE END OF IMAGE CHARACTER FOR THE TTY35 IS THE 'RETURN' KEY. MAXIMUM IMAGE LENGTH IS 72 CHARACTERS.

DELETE- THE DELETE IS REPRESENTED ON THE KEYBOARD WITH THE 'QUESTION MARK' (?) CHARACTER. WHEN RECEIVED FROM THE TERMINAL, THE CURRENT IMAGE IS DISCARDED. THE SYMBIONT RESPONDS TO THE 'DELETE' WITH THE SEQUENCE 'CR-LF'.

ERASE- THE ERASE IS REPRESENTED ON THE KEYBOARD WITH THE DOUBLE QUOTE KEY. WHENEVER THE CHARACTER COUNT OF THE IMAGE IS REDUCED TO ZERO THE SYMBIONT WILL RESPOND WITH THE CHARACTER SEQUENCE 'CR-LF'.

INTERRUPT- THIS CHARACTER IS REPRESENTED ON THE TELETYPE CONSOLE WITH THE KEY LABELED 'BREAK' OR 'RTS'. THE SYMBIONT WILL SUSPEND ITS CURRENT OPERATION AND ACCEPT THE NEXT INPUT IMAGE IMMEDIATELY. THE MESSAGE

INTERRUPT LAST LINE

IS SENT TO THE TERMINAL IN RECOGNITION OF THIS SPECIAL CHARACTER. WHEN THE 'CR' IS RECEIVED, THE LINE OF OUTPUT THAT WAS INTERRUPTED WILL BE RE-TRANSMITTED.

SPECIAL INTERRUPT COMMANDS

ONLY CERTAIN COMMANDS ARE ALLOWABLE ON ENTRY WITH INTERRUPT INPUT MODE. THEY ARE EITHER SYSTEM DIRECTIVES OR ARE ROUTED TO THE USER VIA HIS CONTINGENCY INTERRUPT LOCATION.

1- SYSTEM COMMANDS

*TRADEMARK OF TELETYPE CORPORATION

X- TERMINATE CURRENT USER EXECUTION.

T- TERMINATE OUTPUT TRANSMISSION. ALL PRINT IMAGES CURRENTLY BACKLOGGED IN THE SYSTEM, AND ANY PRINT DATA WHICH MAY BE GENERATED WILL BE DISCARDED BY THE TTY35 SYMBIONT.

A- ALLOW PRINTING TO CONTINUE.

2- USER COMMANDS

THE USER MAY ALSO RECEIVE INTERRUPT COMMANDS. THESE COMMANDS, WHEN RECEIVED, ARE GIVEN IMMEDIATELY TO THE USER VIA HIS CONTINGENCY INTERRUPT ROUTINE IN THE FORM OF A STATUS CODE. THE CONTINGENCY TYPE IS DEFINED WITH AN OCTAL CODE OF 10. THE FORMAT FOR SUBMITTING THE CONTINGENCY CODE FOLLOWING THE INTERRUPT CHARACTER IS

```
CXX
```

WHERE 'C' DEFINES THE CONTINGENCY IMAGE, AND THE 'XX' IS THE OCTAL VALUE TO BE GIVEN THE USER. THIS CODE IS PLACED IN THE 'ERROR CODE' POSITION OF THE CONTINGENCY STATUS WORD.

SYSTEM MESSAGES

NO RUN ACTIVE- THIS MESSAGE IS SENT TO THE TTY TERMINAL WHEN AN IMAGE IS RECEIVED FROM THE TERMINAL AND NO RUN HAS BEEN INITIATED. A @RUN STATEMENT SHOULD THEN BE SUBMITTED TO PROPERLY CONDITION THE CONVERSATIONAL ENVIRONMENT.

WAIT - THIS MESSAGE IS SENT TO THE TERMINAL WHEN INPUT IS ATTEMPTED, AND THE SYSTEM IS NOT READY TO ACCEPT THE INCOMING DATA. THE INPUT MODE IS RE-INITIATED WITH THE NEXT OUTPUT TRANSMISSION, OR BY THE MESSAGE '***READY***'.

READY - THIS MESSAGE INFORMS THE REMOTE USER THAT THE SYSTEM IS IN A CONDITION TO RECEIVE FURTHER INPUT. THIS MESSAGE IS ONLY TRANSMITTED IF THE '***WAIT***' MESSAGE HAD PREVIOUSLY BEEN SENT TO THE TERMINAL. THE '***READY***' MESSAGE IS NOT SENT TO THE TERMINAL IF OUTPUT DATA FROM THE RUN IS AVAILABLE. IN EITHER CASE THE 'WAIT' CONDITION IS REMOVED FROM THE TERMINAL

TIME OUT - IF NO ACTIVITY HAS OCURRED ON THE LINE FOR AT LEAST 5 MINUTES, THE 'TIME OUT' MESSAGE IS SENT TO THE TERMINAL. IN ADDITION, A CORRESPONDING TIME OUT MESSAGE IS DISPLAYED AT THE ON-SITE OPERATOR'S CONSOLE. THE ON-SITE OPERATOR IS THEN RESPONSIBLE TO DETERMINE IF ANYONE IS STILL USING THE TERMINAL. IF NOT, HE MAY THEN INSTRUCT THE SYSTEM TO TERMINATE THE SITE.

TERMINATION PROCEDURES

THE STANDARD TERMINATION PROCEDURE IS PERFORMED WHEN A @FIN STATEMENT HAS BEEN RECEIVED BY THE SYSTEM. THE SYMBIONT RETAINS CONTROL OF THE LINE TERMINAL UNTIL ALL OUTPUT DESTINED FOR THE SITE HAS BEEN SENT. WHEN THE OUTPUT OPERATION IS COMPLETE

THE SYMBIONT WILL SEND THE FOLLOWING CHARACTER SEQUENCE

*** LINE INACTIVE ***

THE SYMBIONT THEN RETURNS TO ITS INITIAL COMMAND MODE. EITHER ANOTHER @RUN STATEMENT OR AN 'EOT' SHOULD FOLLOW. THE TELEPHONE IS STILL AVAILABLE FOR FURTHER COMMUNICATION.

WHEN THE DEMAND RUN IS TERMINATED WITH THE END OF TRANSMISSION (EOT) KEY, OR BY THE ON-SITE OPERATOR THE LINE TERMINAL IS IMMEDIATELY RELEASED. THE RUN IS TERMINATED WITH NO INDICATION OF SUCH SENT TO THE TELETYPE. ANY INFORMATION PREVIOUSLY RECEIVED BY THE SYMBIONT AND NOT PROCESSED IS DISCARDED. LIKEWISE, ANY ACCUMULATION OF OUTPUT BY THE SYMBIONT IS ALSO LOST.

PAGE FORMATTING

THE TELETYPE PAGE IS FORMATTED AS AN 8 1/2 X 11 PAGE. THE MARGIN DEFINITION ASSUMED BY THE TRANSMITTING SYMBIONT IS 66 LINES PER-PAGE WITH A TOP MARGIN SETTING OF 5 LINES, AND THE BOTTOM MARGIN SETTING OF 4 LINES. PAGE HEADINGS ARE INTRODUCED WITH THE @HDG STATEMENT OR THE PRNCN$ CONTROL FUNCTION. THE MAXIMUM LENGTH OF THE HEADING IS 60 CHARACTERS. THE DATE AND PAGE OPTIONS ARE HONORED. IF DATE IS SPECIFIED IT IS PRINTED ON THE HEADING LINE.

PRTCN$ CONTROL FUNCTIONS

THE FUNCTIONS AVAILABLE WITH THIS SYMBIONT ARE:

H- HEADING
L- SPACE TO LOGICAL LINE
M- MARGIN SETTINGS
S- SPECIAL FORM REQUEST

9.9. SYSTEM GENERATION PARAMETERS

9.9.1. DEVICE ASSOCIATION

EACH INPUT DEVICE WHICH IS CAPABLE OF ENTERING A CONTROL STREAM INTO THE SYSTEM HAS ASSOCIATED OUTPUT DEVICES TO WHICH THE PRINT$ AND PUNCH$ FILES ARE DIRECTED. THIS INPUT/OUTPUT ASSOCIATION IS DEFINED BY THE SYSTEM AND LINKS ON-SITE INPUT DEVICES TO ON-SITE SITE OUTPUT DEVICES AND REMOTE INPUT TO REMOTE OUTPUT WITH THE SAME TERMINAL IDENTITY. ASSOCIATION OF DEVICES IS SUBJECT TO USER SPECIFIED DEFINITION WITH SYSTEM GENERATION PARAMETERS (SEE SYSTEM GENERATION STATEMENTS - ASSOCIATE SYMBIONTS), OR WITHIN ANY GIVEN RUN VIA THE @SYM CONTROL STATEMENT (SEE CHAPTER 5).

THE SYMBIONT DEFINED ASSOCIATION FOR REMOTE OPERATIONS IS DETERMINED BY EQUIPMENT TYPE AND SITE ID. INPUT RECEIVED FROM A REMOTE SITE WILL HAVE ITS OUTPUT RETURNED TO THE STATION SUBMITTING THE INPUT. IF A REMOTE INSTALLATION HAS TWO OR MORE 1004'S, EACH OF THESE SUBSYSTEMS WOULD HAVE A UNIQUE SITE ID. THEREFORE, A RUN SUBMITTED FROM ANY ONE OF THE 1004'S WOULD HAVE ITS OUTPUT RETURNED TO THAT 1004. IN ORDER FOR ANY 1004 OF THE INSTALLATION TO BE ELIGIBLE OUTPUT RECIPIENTS, THE SYSTEM DEFINED ASSOCIATION MUST BE OVERRIDDEN. THIS IS ACCOMPLISHED WITH THE 'REMOTE' SYSTEM GENERATION PARAMETER WHICH DEFINES THE EQUIPMENT LOCATED AT THE REMOTE TERMINALS (SEE SYSTEM GENERATION STATEMENTS - REMOTE TERMINALS).

THE 'REMOTE' PARAMETER IS USED TO ASSOCIATE PRINT$/PUNCH$ OUTPUT TERMINALS WITH THE RUN INITIATING TERMINAL, BY SPECIFYING SITE ID CODES IN THE PRINT$ AND PUNCH$ ASSOCIATION FIELDS. IF EITHER, OR BOTH, FIELDS ARE OMITTED, THE REMOTE IDENTITY FIELD DICTATES THE OUTPUT TERMINAL TO THE SYSTEM.

THE SYMBIONT DEFINED ASSOCIATION FOR ON-SITE OPERATIONS IS DETERMINED BY EQUIPMENT TYPE. THE SOURCES OF RUN INPUT, CARD OR PAPER TAPE, HAS FOR ITS ASSOCIATED PRINT$ OUTPUT ANY AVAILABLE PRINTER AT THE ON-SITE LOCATION. PUNCH$ OUTPUT IS DIRECTLY ASSOCIATED WITH ITS RUN'S SOURCE OF INPUT; CARDS IN-CARDS OUT TO ANY AVAILABLE PUNCH (1004 OR 300 CPM PUNCH), PAPER TAPE IN-PAPER TAPE OUT IN SAME LEVEL AS INPUT. THE 'ASSOCIATED SYMBIONTS' STATEMENT (SEE SYSTEM GENERATION STATEMENTS - ASSOCIATE SYMBIONTS) ALLOWS FOR ASSOCIATION BY DEVICE OR A CLASS OF SYMBIONT DEVICES. A CLASS OF SYMBIONT DEVICES IS DEFINED AS A SET OF PERIPHERAL EQUIPMENT WITH ONE SYMBIONT HANDLER.

9.9.2. PROBE

THE SYSTEM GENERATION PARAMETER 'PROBE' SPECIFIES THOSE ON-SITE DEVICES, OR DEVICE CLASSES, WHICH ARE TO BE PROBED FOR INCOMING RUN STREAMS. PROBING IS INITIATED WHEN THE SYSTEM IS BOOTSTRAPPED AND EVERY 15 SECONDS THEREAFTER.

NOTE: DEVICE NAMES CORRESPOND TO SYMBIONT NAMES AS DESCRIBED ABOVE.

9.9.3. MASS STORAGE REQUIREMENTS

THE MASS STORAGE AREA REQUIRED FOR PRINT$ AND PUNCH$ FILES VARIES ACCORDING TO THE RUN OUTPUT PRODUCED AND WITH THE SPEED OF THE OUTPUT DEVICES PROCESSING THESE FILES. THE TOTAL NUMBER OF PAGES OR CARDS EXPECTED AS OUTPUT FROM A RUN ARE SPECIFIED ON EACH @RUN STATEMENT. IF EITHER FIELD IS OMITTED, THE COUNT ASSIGNED WITH THE 'STANDARD PAGE(CARD) LIMIT' SYSTEM GENERATION PARAMETER IS USED AS THE PAGE OR CARD LIMITATION. THE INITIAL SYSTEM VALUES ARE 100 PAGES OF PRINTING AND THE EQUIVALENT OF 100 80 COLUMN CARD IMAGES FOR PUNCHING. TERMINATION OF A RUN WHICH EXCEEDS EITHER LIMIT IS SPECIFIED ON THE @RUN STATEMENT FOR EACH RUN, OR WITH THE SYSTEM GENERATION PARAMETER 'TERMINATE RUN WHEN P FILE EXCEEDED' FOR ALL RUNS. THE INITIAL SYSTEM ASSUMES THE 'CONTINUE RUN WHEN EXCEEDED' CONDITION.

CONSOLE MESSAGES

IF THE PRINT AREA OF A RUN IS EXCEEDED, THE OPERATOR IS NOTIFIED WITH THE MESSAGE.

```
RUNID MAX PAGES
```

IF THE TERMINATE OPTION IS IN EFFECT THE RUN IS ABORTED AT THIS TIME. IF THE TERMINATE OPTION IS NOT SPECIFIED, THE RUN WILL CONTINUE.

IF THE PUNCH AREA OF A RUN IS EXCEEDED, THE OPERATOR IS NOTIFIED WITH THE MESSAGE

```
RUNID MAX CARDS
```

THE SAME ACTION IS TAKEN FOR THIS CONDITION AS FOR MAX PAGES.

9.10. REMOTE DCT-2000 BATCH OPERATIONS.

BATCH OPERATIONS FROM A REMOTE DCT-2000 IS SIMILAR TO REMOTE 1004 BATCH OPERATION WITH ONE MAJOR EXCEPTION: THE DCT-2000 WILL PERMIT ONLY ONE OPERATION AT A TIME, AND SO CONCURRENT READ/PRINT/PUNCH IS NOT POSSIBLE WITH THE REMOTE DCT-2000.

9.10.1. STANDARD SWITCH SETTINGS

THE FOLLOWING SWITCHES ON THE DCT-2000 CONTROL PANEL SHOULD BE PLACED IN THE INDICATED POSITION BEFORE INITIALIZING THE DCT-2000. ALL SWITCHES NOT MENTIONED SHOULD BE IN THE 'UP' OR 'OFF' POSITION.

```
MAINTENANCE PANEL
    SHORT BLOCK ---ON
    PARITY CHECK --ON
    TEST MODE -----OFF

TRANSMIT CONTROL SECTION
    BLOCK LENGTH --80
    SELECT TRANSMIT UNIT --READER
    SELECT REMOTE RECEIVE UNIT --OFF

RECEIVE CONTROL
    BLOCK LENGTH --128
    SELECT RECEIVE UNIT --REMOTE
```

9.10.2. DCT-2000 SPECIAL CONTROL CARDS

SIX SPECIAL CONTROL CONTROL CARDS ARE PROVIDED TO ALTER THE OPERATION OF THE DCT-2000 SYMBIONTS. THESE CARDS ARE HANDLED BY THE DCT-2000 SYMBIONTS AND ARE NEVER PASSED TO THE SYSTEM OR INCLUDED IN THE RUN STREAM. IN ALL CASES THE CHARACTERS USED ARE PUNCHED IN COLUMNS 1-6 ON THE CARD.

1. *XXXXX - SITEID CARD WHERE 'XXXXX' IS THE FIVE CHARACTER SITE IDENTIFIER WHICH IS UNIQUE FOR EACH SITE, AND THE ASTERISK (*) WHICH MUST APPEAR IN COLUMN ONE(1) AND IS THE DCT-2000 IDENTIFIER. THE SITEID CARD IS USED TO INITIALIZE A REMOTE DCT-2000 SITE.

2. @@TERM - TERMINATION CARD WHICH IS USED TO TERMINATE AN ACTIVE DCT-2000 SITE. THE DCT-2000 SITE IS TERMINATED IMMEDIATELY, AND NO FURTHER ACTIVITY WILL OCCUR WITHOUT RE-INITIALIZING THE SITE.

3. @@ABRT - USED TO ABORT THE CURRENT OUTPUT FILE. OUTPUT WILL CONTINUE WITH THE NEXT FILE, IF ANY EXIST, OR THE NEXT OPERATION WILL BEGIN.

4. @@READ - CAUSES SUSPENSION OF THE CURRENT OUTPUT OPERATION AND INITIATES THE READ SYMBIONT. THE SUSPENDED OPERATION WILL CONTINUE WHEN THE READ OPERATION TERMINATES.

5. @@HGOV - SUSPENDS OPERATIONS AND DISPLAYS THE FOLLOWING MESSAGE ON THE 1108 OPERATOR'S CONSOLE.
   'SITEID HALT GO VOICE'
   THIS INDICATES THAT THE DCT-2000 OPERATOR REQUIRES VOICE COMMUNICATIONS WITH THE 1108 OPERATOR.

6. @@REDY - THIS CARD IS USED IN COMBINATION WITH THE

'@@HGOV' CARD. AFTER DISPLAYING THE 'HALT-GO-VOICE' MESSAGE ON THE CONSOLE, THE DCT-2000 SYMBIONT WILL REQUEST CARD INPUT. THE ONLY CARD RECOGNIZED AT THIS TIME IS THE '@@REDY', AND ANY OTHER CARD(S) WILL BE IGNORED. UPON RECEPTION OF THE '@@REDY' CARD OPERATIONS WILL RESUME.

9.10.3. REMOTE DCT-2000 INITIALIZATION

BEFORE SWITCHING TO 'DATA' ON THE DATA-PHONE, THE DCT-2000 MUST BE CONDITIONED AS FOLLOWS:

1. POWER--------- ON
2. SYSTEM MODE -- ATTENDED, ON-LINE

BEFORE THE 1108 SYSTEM WILL RECOGNIZE A REMOTE DCT-2000 SITE, THE UNIQUE SITE-ID CARD MUST BE SENT BY THE DCT-2000 TO THE 1108 FOR VALIDATION. THIS MAY BE ACCOMPLISHED BY EITHER OF TWO METHODS, AS FOLLOWS:

1. AFTER SWITCHING TO DATA THE SITE-ID CARD IS PLACED INTO THE INPUT HOPPER AND THE SWITCHES 'GENERAL CLEAR', 'CARD FEED', 'CARD CLEAR', AND 'TRANSMIT' ARE DEPRESSED ONCE AND IN ORDER. THE SITE ID CARD WILL BE READ AND TRANSMITTED TO THE 1108. AFTER THE SITE ID CARD HAS BEEN READ THE DCT-2000 OPERATOR MUST EITHER SWITCH TO 'UNATTENDED' MODE, OR MANUALLY SEND AN EOT SEQUENCE BY DEPRESSING THE 'SEND EOT' LEVER SWITCH IN THE TRANSMIT CONTROL SECTION. SINCE THE OPERATOR WILL SWITCH TO'UNATTENDED'IN EITHER CASE,(NORMAL OPERATING MODE IS UNATTENDED UNLESS OTHERWISE SPECIFIED) THE FIRST METHOD IS RECOMMENDED. WHEN THE SITE ID CARD IS VALIDATED, THE MESSAGE:
   '1108 READY'
   WILL BE DISPLAYED ON THE DCT-2000 PRINTER, AND THE MESSAGE:
   'DCT SITEID ACTV'
   WILL BE DISPLAYED ON THE 1108 OPERATOR'S CONSOLE. THE DCT-2000 IS NOW READY FOR FURTHER OPERATIONS. IF ANY OUTPUT FILES ARE QUEUED TO THIS DCT-2000 SITE, OUTPUT WILL BEGIN IMMEDIATELY AFTER THE SITE BECOMES ACTIVE.

2. IF AN INPUT DECK IS TO BE READ IN AT THIS TIME, THE SITE ID CARD FOLLOWED BY ONE(1) BLANK CARD IS PLACED AT THE BEGINNING OF THE INPUT DECK AND THE ENTIRE DECK IS PLACED INTO THE INPUT HOPPER. THE SWITCHES 'GENERAL CLEAR', 'CARD FEED', 'CARD CLEAR', AND 'TRANSMIT' ARE DEPRESSED ONCE AND IN ORDER. THE SITE ID CARD WILL BE READ IMMEDIATELY, AND, IF VALIDATED, THE INPUT DECK WILL BE READ AFTER A SHORT PAUSE. IT IS IMPERATIVE THAT THE OPERATOR SWITCH TO 'UNATTENDED' MODE BEFORE THE LAST CARD IS READ.

9.10.4. REMOTE DCT-2000 INPUT

AFTER THE DCT-2000 HAS BEEN INITIALIZED AND HAS BEEN SWITCHED TO THE 'UNATTENDED' MODE, AN INPUT DECK MAY BE TRANSMITTED TO THE 1108. IF PRINT IS CURRENTLY IN PROGRESS, THE READ WILL WAIT UNTIL THE PRINT IS COMPLETED. HOWEVER, THE DCT-2000 MAY BE CONDITIONED TO TRANSMIT AT ANY TIME (EXCEPT WHEN PUNCHING) AS FOLLOWS:

1. PLACE INPUT DECK INTO THE INPUT HOPPER
2. DEPRESS THE 'CARD FEED', 'CARD CLEAR', AND 'TRANSMIT' SWITCHES ONCE AND IN ORDER.

NOTE - NEVER DEPRESS 'GENERAL CLEAR' WHILE IN THE 'UNATTENDED' MODE

AN INPUT DECK IS DEFINED AS ONE OR MORE @RUN DECKS TERMINATED BY A @FIN CARD. IF THE LAST CARD IN ANY GIVEN INPUT DECK IS NOT A @FIN CARD, THE MESSAGE:
'NO FIN CARD'
WILL BE DISPLAYED ON THE DCT-2000 PRINTER. TO RECOVER, A @FIN CARD IS PLACED INTO THE INPUT HOPPER AND STEP #2 ABOVE IS EXECUTED.

9.10.5. REMOTE DCT-2000 OUTPUT

9.10.5.1. REMOTE DCT-2000 PRINT

NO DCT-2000 OPERATOR INTERVENTION IS REQUIRED TO PRINT, OTHER THAN PROVIDING SUFFICIENT PAPER AND DEPRESSING THE PRINTER 'CLEAR' SWITCH BEFORE INITIALIZING.

9.10.5.2. REMOTE DCT-2000 PUNCH

BEFORE PUNCHING BEGINS, THE MESSAGE:
'CONDITION DCT-2000 FOR PUNCHING' WILL BE DISPLAYED ON THE DCT-2000 PRINTER. AT THIS TIME THE DCT-2000 OPERATOR MUST EXECUTE THE STEPS OUTLINED BELOW, BEFORE ANY PUNCHING CAN OCCUR.

1. REMOVE DATA CARDS FROM THE INPUT HOPPER, IF ANY, AND CLEAR ANY CARDS FROM THE READ STATIONS BY USE OF THE 'CARD FEED' SWITCH.
2. INSERT A DECK OF BLANK CARDS INTO THE INPUT HOPPER.
3. DEPRESS THE 'CARD CLEAR' SWITCH ONCE.
4. PLACE THE 'READ/PUNCH' SWITCH IN THE 'PUNCH' POSITION.
5. DEPRESS THE 'CARD FEED' AND 'CARD CLEAR' SWITCHES ONCE AND IN ORDER.
6. PLACE THE 'BLOCK LENGTH' SWITCH IN THE 'RECEIVE-CONTROL' SECTION IN THE 80 POSITION.

PUNCHING WILL BEGIN IMMEDIATELY AND ALL PUNCH FILES CURRENTLY QUEUED WILL BE PUNCHED. WHEN THE QUEUE FOR THIS DCT-2000 SITE HAS BEEN EXHAUSTED, THE MESSAGE:
'DCT-2000 PUNCH TERM'
WILL BE DISPLAYED ON THE DCT-2000 PRINTER. AT THIS TIME THE DCT-2000 OPERATOR SHOULD RETURN THE'READ/PUNCH' SWITCH TO THE 'READ' POSITION AND SET THE 'BLOCK LENGTH' SWITCH IN THE 'RECEIVE
CONTROL' SECTION TO THE 128 POSITION.

9.10.6. DCT-2000 BEL SEQUENCE

THE 'SIGNAL REMOTE' SWITCH IN THE 'SYSTEM CONTROL' SECTION, IS USED TO INTERRUPT

THE CURRENT OPERATION IN ORDER TO SEND ONE OF THE SPECIAL DCT-2000 CONTROL CARDS TO THE 1108. IT IS ONLY NECESSARY TO USE THIS METHOD DURING OUTPUT SINCE THE SPECIAL CONTROL CARDS MAY BE INCLUDED AS PART OF AN INPUT DECK, OR READ ALONE AS NORMAL INPUT.

TO SEND A 'BEL' SEQUENCE THE FOLLOWING STEPS ARE EXECUTED BY THE DCT-2000 OPERATOR.

1. SWITCH TO THE 'ATTENDED' MODE OF OPERATION
2. PLACE THE 'RUN/STOP' SWITCH IN THE 'SYSTEM CONTROL' SECTION IN THE 'STOP' POSITION. THE DCT-2000 WILL COME TO AN ORDERLY HALT (THE 'RECEIVE' OR 'TRANSMIT' INDICATOR WILL GO OUT).
3. DEPRESS THE 'SIGNAL REMOTE' SWITCH IN THE 'SYSTEM CONTROL' SECTION.
4. PLACE THE SPECIAL CONTROL CARD(S) INTO THE INPUT HOPPER
5. PLACE THE 'RUN/STOP' SWITCH IN THE 'RUN' POSITION AND DEPRESS THE 'GENERAL CLEAR' SWITCH.
6. SWITCH TO 'UNATTENDED' MODE OF OPERATION
7. DEPRESS THE 'CARD FEED', 'CARD CLEAR', AND 'TRANSMIT' SWITCHES ONCE AND IN ORDER.

THE SPECIAL CONTROL CARD WILL BE READ AND THE INDICATED ACTION TAKEN. (SEE 9.10.2 FOR A DESCRIPTION OF THE SPECIAL CONTROL CARDS)

10. INPUT/OUTPUT DEVICE HANDLERS

10.1. I/O CONTROL

10.1.1. GENERAL

INPUT/OUTPUT OPERATIONS ARE CONTROLLED BY MEANS OF A CENTRAL ROUTINE. IT MAY BE THOUGHT OF AS THE MASTER DEVICE HANDLER ACCEPTING AND QUEUEING REQUESTS AND INTERRUPTS AND GIVING CONTROL OF THEM TO THE I/O DEVICE HANDLER, WHEN APPROPRIATE. PARTIAL DUAL CHANNELS, FULL DUAL CHANNELS, DUAL-COMPUTER CHANNELS, AND ESI CHANNELS REQUIRE SPECIAL TREATMENT, PARTIAL-DUAL AND FULL-DUAL . A SPECIAL MECHANISM IS USED TO SERVE AS A COMMON ROUTINE FOR TREATMENT OF WAIT-TYPE ER'S, AND INTERRUPT-REQUEST-TYPE ER'S. OTHER SUBROUTINES COMMON TO ALL THE HANDLERS ARE CONTAINED WITHIN CENTRAL CONTROL, SUCH AS THE CONVERSION OF A USER'S ACCESS WORD FROM RELATIVE TO ABSOLUTE AND CHECKING IT AGAINST HIS ASSIGNED AREA OF CORE.

REFERENCE TO I/O CONTROL IS DEPENDENT UPON THE CLASS OF EQUIPMENT BEING REFERRED:

COMMUNICATIONS DEVICES,
MAGNETIC DRUM OR TAPE, OR
LOW-SPEED ON-SITE DEVICES.

THE COMMUNICATIONS DEVICES ARE REFERENCED VIA A SPECIAL HANDLER DESCRIBED IN SECTION 6 BELOW. THE SYMBIONTS CONTROL THE LOW-SPEED ON-SITE DEVICES VIA THE ARBITRARY DEVICE HANDLER. THIS HANDLER IS ALSO AVAILABLE TO OTHER ROUTINES SUCH AS MAINTENANCE ROUTINES AND SPECIAL HANDLING OF PERIPHERAL DEVICES. FOR REFERENCING TAPE AND DRUM UNITS THE STANDARD BASIC LEVEL CONSISTS OF THE PACKET MODE USING AN EXECUTIVE RETURN INSTRUCTION AND A0 LOADED WITH THE PACKET ADDRESS AS FOLLOWS:

```
LA    A0,PKT ADDR
ER    ENTRANCE TAG
```

THE ENTRANCE TAG CAN BE ONE OF THE FOLLOWING

IO$	INITIATE THE REQUESTED OPERATION AND RETURN CONTROL IN-LINE IMMEDIATELY. PROGRAM ACTIVITY CONTINUES WITH THE ASSUMPTION THAT A WAIT ON COMPLETION IS CODED AT A SUBSEQUENT POINT FOR PROPER SYNCHRONIZATION.
IOW$	INITIATE THE REQUESTED OPERATION AND RETURN CONTROL AFTER THE OPERATION IS COMPLETED. NO FURTHER SYNCHRONIZATION IS NEEDED. THIS ENTRANCE SAVES THE TIME REQUIRED TO STORE AND RESTORE ICR IF FURTHER OPERATION IS DEPENDENT UPON THE I/O COMPLETION.
IOXI$	INITIATE THE REQUESTED OPERATION AND EXIT. UPON COMPLETION OF THE OPERATION CONTROL IS GIVEN TO THE INTERRUPT ROUTINE DEFINED BELOW. THIS ENTRANCE CAN BE USED ONLY IF AN INTERRUPT ROUTINE ADDRESS IS SPECIFIED. THE EXIT AND RESTART AS AN INTERRUPT ROUTINE INCREASES THE COMPLETION PRIORITY AND SAVES THE TIME NEEDED TO STORE AND RESTORE THE CONTROL REGISTER SET.
IOI$	SAME AS IO$ WITH THE ADDITION OF AN INTERRUPT ROUTINE INITIATED AT COMPLETION OF THE I/O REQUEST.
IOWI$	SAME AS IOW$ WITH THE ADDITION OF THE COMPLETION

INTERRUPT ROUTINE.

IN MANY RESPECTS THE INTERRUPT ACTIVITY IS THE SAME AS ALL OTHER REGISTERED ACTIVITIES USING THE FORK FUNCTION. IT DIFFERS IN THE FOLLOWING AREAS:

THE PRIORITY OF THE ACTIVITY IS RAISED TO THE HIGHEST POSSIBLE LEVEL WITHIN THE PROGRAM CLASS OF THIS WORKER, I.E. FOR A BATCH WORKER PROGRAM THESE I/O COMPLETION ACTIVITIES WILL RECEIVE CONTROL BEFORE ANY OTHER BATCH PROGRAM ACTIVITY.

THE INTERRUPT ROUTINE IS NOT INTERRUPTABLE IN FAVOR OF ANY OTHER SIMILAR ACTIVITY OF THE SAME PROGRAM. ALL ARE QUEUED IN A FIRST-IN-FIRST-OUT LIST OF ALL PROGRAMS WITHOUT REGARD TO PRIORITY WITHIN THE CLASS.

ANY EXEC SERVICE REQUEST REMOVES THE INTERRUPT ACTIVITY FROM THE HIGH PRIORITY LIST AND PUTS IT BACK WITHIN THE WORKER PROGRAM.

THE CR SUBSET IN THE INTERRUPT ROUTINE IS LIMITED TO B11, A0 THROUGH A5, AND R1 THROUGH R3. THE A0 REGISTER IS PASSED ON TO THE INTERRUPT ROUTINE FROM THE CONTENTS AT THE REFERENCE TO I/O CONTROL. IF THE SUPPRESS RECOVERY MODE IS SET, THE A1 REGISTER IS LOADED WITH THE STATUS WORD FROM THE SUBSYSTEM EXTERNAL INTERRUPT.

IN THE ABSENCE OF ANY OTHER EXEC SERVICE REQUEST THE NORMAL PROGRAM STATUS CAN BE RESTORED BY USING THE UNLCK$ FUNCTION (SEE EXECUTIVE REQUEST SECTION).

EVERY ATTEMPT IS MADE IN THE MULTIPROGRAMMING ENVIRONMENT TO PROVIDE PROPER SWITCHING ALLOWING IMMEDIATE ACCESS TO THE AMOUNT OF COMPUTATION REQUIRED TO INITIATE ANOTHER I/O OPERATION FOLLOWING ANY OTHER I/O COMPLETION. THE DIFFICULTY LIES IN EFFECTIVELY PREVENTING ABUSE OF THE HIGH PRIORITY PROVISIONS MADE FOR THE INTERRUPT ACTIVITIES. THE INTENT IS TO LIMIT THE FACILITY AVAILABLE TO NO MORE THAN THAT NECESSARY TO REQUEST ANOTHER I/O OPERATION HAVING CHECKED THE STATUS OF THE LAST.

10.1.2. QUEUEING AND UNIT CONTROL

THE REFERENCES TO I/O CONTROL RESULT IN ENTERING THE HANDLER CONTROLLING THE DEVICE REFERENCED. THE HANDLER IN TURN CONSIDERS THE REQUEST AND QUEUES IT FOR THE PARTICULAR SUBSYSTEM. WHEN THE SUBSYSTEM BECOMES FREE AN ENTRY IS REMOVED FROM THE SUBSYSTEM QUEUE AND THE HANDLER IS ENTERED AT THE APPROPRIATE POINT. QUEUING IS BYPASSED IF THE SUBSYSTEM IS INITIALLY NOT BUSY.

THE CHANNEL REQUEST QUEUE AND INTERRUPT QUEUE CONTAIN INFORMATION TO DIRECT THE ATTENTION OF THE DEVICE HANDLERS TO THE UNIT (AND WHERE REQUIRED, TO THE FILE) WITH WHICH THE REQUEST OR INTERRUPT IS ASSOCIATED.

AN AID TO THE USER FOR DETECTION OF A LOOP IN A NONDEBUGGED PROGRAM WHICH INCLUDES A REFERENCE TO I/O IS A NEGATIVE TEST ON THE WORD OF THE PACKET CONTAINING THE STATUS CODE (THE PACKET IS DEFINED IN THE NEXT SECTION). WHEN AN I/O REQUEST IS MADE BY THE USER THE EXEC SETS THE STATUS WORD NEGATIVE TO INDICATE AN IN-PROGRESS STATE. BEFORE SETTING THE WORD NEGATIVE A CHECK IS MADE TO SEE IF IT IS ALREADY NEGATIVE, INDICATING A POSSIBLE LOOP, IN WHICH CASE A UNIQUE STATUS CODE IS PLACED IN THE PACKET AND CONTROL IS GIVEN TO THE 'ERR MODE' ROUTINE OF THE USER IF ONE EXISTS. FOR ALL COMPLETIONS OF PACKETS A POSITIVE VALUE IS ALWAYS PLACED IN THE STATUS WORD HENCE NO HOUSEKEEPING IS NECESSARY BY THE USER AND AN INITIAL NEGATIVE VALUE CAN BE CORRECTLY INTERPRETED AS A SOFTWARE LOGIC ERROR.

EFFICIENT UTILIZATION OF ALL DRUM TYPES INCLUDING FASTRAND DICTATES THAT SERVICING REQUESTS FOR A GIVEN FILE WILL NOT BE RESTRICTED TO THE ORDER OF SUBMISSION. THIS ALLOWS A SPEEDUP OF SERVICING BY INCLUSION OF THE 'SEND ANGULAR ADDRESS' FUNCTION AND

THE FACT THAT FILES MAY BE ALLOCATED ACROSS UNITS AND HENCE ACROSS ACCESS PATHS. TESTING EACH PACKET IS NECESSARY TO INSURE COMPLETION, AND DO NOT ASSUME COMPLETION BY TESTING A SUBSEQUENT PACKET.

FOR ALL REFERENCES TO I/O CONTROL VIA NONSYNCHRONOUS ER'S FOR ALL TYPES OF DEVICES, IT IS NECESSARY TO TEST THE STATUS CODE FOR COMPLETION OF I/O BEFORE EXITING FROM AN ACTIVITY. UPON ACTIVITY EXIT, EITHER NORMAL OR ABNORMAL, LISTED I/O REQUESTS MAY BE DISREGARDED WITHOUT SERVICING THE REQUEST.

10.1.3. INPUT/OUTPUT PACKET

PRIOR TO EXECUTING THE ER INSTRUCTION TO REFERENCE THE TAPE OR DRUM HANDLER, THE A0 REGISTER MUST BE LOADED WITH THE ADDRESS OF AN INPUT/OUTPUT PACKET. THIS PACKET IS THE FIRST PART OF THE FILE CONTROL TABLE IF THE BLOCK BUFFERING OR ITEM HANDLING ROUTINES ARE USED. THE LENGTH OF THE REQUEST PACKET CAN VARY FROM FOUR TO EIGHT WORDS DEPENDING UPON THE OPERATION DESIRED. THE FORMAT OF THE PACKET IS AS FOLLOWS:

```
     35        29        23        17        11        05       00
   ------------------------------------------------------------------
01 :                 INTERNAL FILE NAME (WORD 1)                    :
   ------------------------------------------------------------------
02 :                 INTERNAL FILE NAME (WORD 2)                    :
   ------------------------------------------------------------------
03 :USED BY EXEC        :INT ACT ID: INTERRUPT ACTIVITY START       :
   ------------------------------------------------------------------
04 : STATUS   :FUNCTION :  AFC    :      SUBSTATUS FIELD            :
   ------------------------------------------------------------------
05 :                            ACCESS WORD                         :
   ------------------------------------------------------------------
06 :                    :        DRUM ADDRESS                       :
   ------------------------------------------------------------------
07 :                          SEARCH IDENTIFIER                     :
   ------------------------------------------------------------------
08 :                    :     SEARCH FIND DRUM ADDRESS              :
   ------------------------------------------------------------------
```

WORD 1 AND WORD 2- THE INTERNAL FILE NAME USED IN ALL REFERENCES TO THE FILE. THIS NAME IS EITHER THE SAME AS SOME EXTERNAL FILE NAME OF THE @ASG STATEMENT OR IS ATTACHED TO AN EXTERNAL FILE NAME BY A @USE STATEMENT.

WORD 3 TI-USED BY THE EXEC TO ASSOCIATE THE INTERNAL FILE NAME TO AN EXEC TABLE LOCATION.

S3-THE NUMERIC IDENTITY (1-35) USED TO IDENTIFY THE INTERRUPT ACTIVITY IF SYNCHRONIZATION IS INTENDED WITH SOME OTHER ACTIVITY.

H2-INTERRUPT ACTIVITY STARTING ADDRESS. THIS IS THE LOCATION AT WHICH THE WORKER RECEIVES CONTROL UPON OCCURRENCE OF AN INTERRUPT SIGNIFYING COMPLETION OF THE I/O OPERATION.

WORD 4 S1-THE STATUS OF THE LAST FUNCTION PERFORMED.

S2-THE CODE DENOTING THE FUNCTION TO BE PERFORMED.

S3-AFC- FOR MAGNETIC TAPE FILES IF THE CHARACTER COUNT IS NOT A MULTIPLE OF 6(FOR 7-CHANNELS PER FRAME DRIVES)OR A MULTIPLE OF 9 (FOR 9-CHANNELS PER FRAME) THIS SIXTH OF A WORD CONTAINS THE NUMBER OF CHARACTERS IN THE LAST WORD READ (FIRST WORD OF THE BUFFER FOR A READ BACKWARD). THIS USED FIELD IS IN CONJUNTION WITH A STATUS CODE OF 04. IF THE ACCESS WORD DOES NOT HAVE A WORD COUNT LARGE ENOUGH TO ALLOW TRANSFER OF THE ENTIRE BLOCK AND A STATUS CODE OF 04 IS RETURNED, THIS FIELD IS SET TO ZERO.(FOR IVC UNITS IF THE ACCESS WORD GOES TO ZERO, IT IS INDETERMINATE WHETHER ALL WORDS WERE READ. IF ALL WORDS OF THE BLOCK ARE READ, AS DETERMINED BY THE USER, THEN THE COUNT OF FRAMES OF DATA IN THE LAST WORD READ IS STORED IN THE LOWER SIXTH OF THE LAST DATA WORD.) FOR 9-CHANNEL TAPES, THE COUNT IS THE NUMBER OF 8-BIT BYTES ASSEMBLED AND TRANSFERRED TO THE COMPUTER IN THE LAST 2-WORD SEQUENCE I.E. A VALUE OF 1 INDICATES AN ODD NUMBER OF WORDS WITH ONE 8-BIT BYTE ASSEMBLED IN THE FINAL WORD AND THE REMAINDER OF THE WORD PADDED WITH ZEROES . A VALUE OF 5 INDICATES AN EVEN NUMBER OF WORDS WITH 4 DATA BITS IN THE LAST WORD WHICH ARE THE LEAST SIGNIFICANT HALF OF THE 8-BIT BYTE WITH THE MOST SIGNIFICANT 4 BITS IN THE PRECEDING WORD.

H2-AUXILIARY STATUS INFORMATION. FOR ANY FUNCTION INVOLVING INPUT TRANSFER, THIS FIELD WILL CONTAIN THE EXACT NUMBER OF WORDS READ. FOR MAGNETIC TAPE OR THE END OF A DRUM FILE, THIS NUMBER MAY DIFFER FROM THE

ACCESS WORD.

WORD 5 THE FIFTH WORD OF THE PACKET IS AN I/O ACCESS WORD SPECIFIED IN THE FORMAT DEFINED IN THE UNIVAC 1108 MULTI-PROCESSOR SYSTEM DESCRIPTION, UP-4046 REV.1. THAT IS, BITS 35-34 ARE THE INCREMENT-DECREMENT DESIGNATOR, BITS 33-18 CONTAIN THE NUMBER OF WORDS TO TRANSFER, AND BITS17-00 CONTAIN THE ADDRESS AT WHICH TRANSFER IS TO BEGIN. FOR GW$,SCR$,AND SCRB$ FUNCTIONS, THIS WORD CONTAINS THE NUMBER OF ACCESS WORDS AND THE ADDRESS AT WHICH THE STRING OF ACCESS WORDS BEGIN.

WORD 6 FOR MAGNETIC DRUM FILES, THIS WORD CONTAINS THE MASS STORAGE ADDRESS AT WHICH THE DESCRIBED I/O OPERATION IS TO START. THIS ADDRESS IS RELATIVE TO THE START OF THE MASS STORAGE FILE; THE HANDLER PROVIDES FOR DETERMINING THE ABSOLUTE POSITION. FOR FASTRAND FILES (OR SIMULATED FASTRAND ON OTHER TYPE DRUM) THE ADDRESS IS THE START OF A SECTOR AND CONSECUTIVE ADDRESSES ARE 28 WORDS APART.

WORD 7 THIS WORD IS THE IDENTIFIER WORD FOR SEARCH OPERATIONS.

WORD 8 THE FIND ADDRESS FOR A DRUM SEARCH IS RETURNED IN THIS WORD. THE ADDRESS IS RELATIVE TO THE START OF THE FILE.

WHENEVER THE INPUT/OUTPUT FUNCTION IS REQUESTED WITH AN INTERRUPT COMPLETION ACTIVITY, WORD 3 OF THE PACKET MUST BE CODED AS OUTLINED. THIS WORD DESCRIBES THE LINKAGE TO THE WORKER PROGRAM FOR HIGH PRIORITY PACKET PROCESSING. THE PRIORITY IS SET TO LEVEL 1 WITHIN THE PROGRAM TYPE (SEE SUPERVISOR CHAPTER). THE INTERRUPT ACTIVITY IS LIMITED TO THE MINOR SET OF CONTROL REGISTERS. THE INTERRUPT ACTIVITY STARTING ADDRESS MUST BE A NONZERO VALUE. IF NO SYNCHRONIZATION IS INTENDED THE NUMERIC IDENTITY FOR THE INTERRUPT ACTIVITY IS NOT NEEDED.

THE SYSTEM PROCEDURES AVAILABLE TO GENERATE I/O REQUEST PACKETS FOR DRUM AND TAPE ARE:

```
I$OD     U,F,INT ADDR,INT ID   COUNT,ADDR,INC  DRUM ADDR, SENT
I$OT     U,F,INT ADDR,INT ID   COUNT,ADDR,INC
```

WHERE:

U-IS THE SYMBOLIC FILE NAME ASSOCIATED WITH THE UNIT BEING REFERENCED
F-IS THE SYMBOLIC OR OCTAL CODE IDENTIFYING THE FUNCTION (SEE TABLE 10-1)
INT ADDR-IS THE INTERRUPT ROUTINE STARTING ADDRESS
INT ID-IS THE INTERRUPT ROUTINE IDENTITY (1-35). IF OMITTED THIS FIELD IS SET TO ZERO.
COUNT,ADDR,INC-SPECIFIES THE CORE AREA TO TRANSFER TO OR FROM. THE INC FIELD CAN BE OMITTED IF INCREMENTATION IS DESIRED, GIVEN THE SYMBOL D FOR DECREMENTATION OR N FOR NO INCREMENTING OR DECREMENTING
DRUM ADDR, SENT-IDENTIFY THE STARTING POINT AND, IF A SEARCH OPERATION, THE SEARCH SENTINEL.

THE TAG ON THE PROCEDURE LINE IS ALLOCATED TO THE FIRST WORD OF THE FILE NAME. THE I$OD PROCEDURE GENERATES SIX OR EIGHT WORDS DEPENDING UPON THE PRESENCE OF THE SENT FIELD. THE I$OT PROCEDURE GENERATES FIVE WORDS REGARDLESS OF THE FIELDS CODED.

10.1.4. PROGRAM-I/O SYNCHRONIZATION

THE ABILITY OF A WORKER PROGRAM TO SYNCHRONIZE WITH THE COMPLETION OF AN I/O OPERATION IS AVAILABLE THROUGH ENTRANCE TO THE EXECUTIVE VIA AN ER INSTRUCTION WITH

AN OPERAND OF WAIT$ OR WANY$.

THE ENTRY WAIT$ WAITS FOR COMPLETION OF A PARTICULAR I/O REQUEST. THE WANY$ ENTRY SYNCHRONIZES WITH ANY I/O REQUEST COMPLETION FOR THE ACTIVITY. THE TEST MADE WITHIN THE EXEC ON A WANY$ REQUEST IS TO DETERMINE IF ANY I/O REQUEST HAS BEEN COMPLETED FOR THE ACTIVITY SINCE THE LAST TIME THE ACTIVITY WAS PLACED IN A WAIT CONDITION BY A PREVIOUS WAIT$ OR WANY$ REQUEST.

WHENEVER A PROGRAM SUBMITS A REQUEST TO EXEC FOR AN I/O OPERATION, THE EXEC SETS THE FOURTH WORD OF THE PACKET NEGATIVE. THIS WORD WILL REMAIN NEGATIVE UNTIL THE REQUEST IS COMPLETELY SERVICED AND CAN BE USED BY THE PROGRAM TO DETERMINE WHETHER OR NOT A REFERENCE TO WAIT$ OR WANY$ IS REQUIRED.

THE LINKAGE TO WAIT$ MUST BE PRECEDED BY A TEST POSITIVE INSTRUCTION ON WORD 4 OF THE PACKET:

```
  TP   PKT ADDR+3
  ER   WAIT$
```

THE NECESSITY OF EXEC PERFORMING THE SAME TEST INSTRUCTION A SECOND TIME TO DETERMINE COMPLETION OF THE REQUEST BETWEEN THE TP AND ER REQUIRES THAT THE H AND I DESIGNATORS OF THE INSTRUCTION BE SET TO ZERO. THE PACKET ADDRESS IS THE SPECIFIC REQUEST WAITED FOR AT WAIT$.

THE REFERENCE TO WANY$:

```
  ER   WANY$
```

NEED NOT BE PRECEDED BY ANY TEST INSTRUCTION. THE ONLY REQUIREMENT IS THAT AT LEAST ONE I/O REQUEST MUST HAVE BEEN SUBMITTED SINCE THE LAST REFERENCE TO WAIT$ OR WANY$. THIS ASSISTS THE EXEC IN DETECTING LOOPS IN UNDEBUGGED PROGRAMS.

ACTIVITIES WHICH ATTEMPT TO RELEASE CONTROL TO WAIT$ OR WANY$ WITHOUT AN OUTSTANDING REQUEST OR WITHOUT A LEGAL TEST POSITIVE (WAIT$ ONLY) CALL WILL EITHER BE TERMINATED OR WILL REGAIN CONTROL AT THE ERROR MODE RETURN. THE ERROR TYPE FOR THIS TERMINATION IS 04 AND THE ERROR CODE IS 01 (NO OUTSTANDING REQUESTS)OR 02(ILLEGAL TEST). UPON COMPLETION OF AN I/O REQUEST ONLY THE SUBMITTING ACTIVITY IS CHECKED TO BE IN A WAIT CONDITION, THEREFORE IT IS NOT ADVISABLE FOR ONE ACTIVITY TO GO TO WAIT$ OR WANY$ TO WAIT ON A REQUEST SUBMITTED BY A DIFFERENT ACTIVITY.

10.1.4.1. STATUS CODES

UPON COMPLETION OF AN I/O REQUEST, A STATUS CODE IS STORED INTO S1 OF THE FOURTH WORD OF THE REQUEST PACKET DENOTING THE CONDITIONS OF THE COMPLETION. ALL CODES FROM 20 TO 37 CAUSE THE PROGRAM TO BE TERMINATED AND ARE HENCE NOT OBSERVED BY THE EXECUTING PROGRAM UNLESS AN ERROR TERMINATION ACTIVITY IS CALLED (PREVIOUS REFERENCE TO IERR$). IF CONTROL IS GIVEN TO THE ERROR TERMINATION ACTIVITY, ANY INTERRUPT ACTIVITIES SPECIFIED BY THE REQUEST ARE DELETED. THE EXCEPTION TO THIS RULE IS IOXI$ IN WHICH CASE THE INTERRUPT ACTIVITY ITSELF IS GIVEN CONTROL AT THE ERROR TERMINATION ACTIVITY ENTRANCE.

OCTAL CODE	DEFINITION
00	THE REQUEST HAS BEEN COMPLETED NORMALLY. IF DATA TRANSFER IS INVOLVED, THE COUNT IS GIVEN IN H2 OF WORD 4.
01	END-OF-FILE BLOCK DETECTED ON MAGNETIC TAPE, AN ANSWER OF 'E' TO AN I/O ERROR MESSAGE, A BLOCK READ DRUM FUNCTION WAS TRUNCATED BY ENCOUNTERING AN END-OF-BLOCK WORD, OR A FIND WAS MADE ON A MASS STORAGE DEVICE SEARCH AND THEN THE END OF THE ASSIGNED AREA (OR END-OF-BLOCK SENTINEL FOR BLOCK SEARCH READ) WAS ENCOUNTERED BEFORE THE SPECIFIED WORD COUNT WAS TRANSFERRED. THE COUNT ACTUALLY TRANSFERRED IS SPECIFIED IN H2 OF WORD 4 OF THE PACKET. STATUS 00 IS RETURNED IF TRANSFER IS COMPLETED AFTER THE SEARCH

OR IF NO TRANSFER IS INVOLVED. WORD 8 GIVES THE RELATIVE DRUM ADDRESS AT WHICH THE FIND WAS MADE.

02 END-OF-TAPE MARK ENCOUNTERED ON MAGNETIC TAPE ON A READ BACKWARD FROM LOAD POINT OR ON A WRITE. NO TRANSFER TAKES PLACE FOR THE READ BACKWARD. THE WRITE IS DONE IN THE NORMAL MANNER. SUBSEQUENT WRITES ARE PERFORMED IN THE SAME FASHION AND, BARRING OTHER PROBLEMS, WILL RESULT IN RETURNING THE SAME STATUS CODE.

03 NO 'FIND' WAS MADE ON A MASS STORAGE DEVICE SEARCH. THE SEARCH WAS TERMINATED BY AN END-OF-BLOCK, END-OF-TRACK, END-OF-POSITION, OR EXPIRATION OF SUFFICIENT TIME TO PASS OVER THE ENTIRE AREA OF CONCERN DEPENDING UPON THE PHYSICAL DEVICE AND TYPE OF SEARCH.

04 A NONINTEGRAL BLOCK WAS READ FROM MAGNETIC TAPE. THE NUMBER OF DATA CHARACTERS ACCEPTED FROM THE LAST WORD IS INDICATED BY S3 OF WORD 4 OF THE PACKET AND IS EXPLAINED IN DETAIL IN SECTION 10.1.3

05 AN ATTEMPT WAS MADE TO INITIATE A MASS STORAGE SEARCH OR READ FROM AN AREA WHICH IS WHOLLY OR PARTIALLY UNASSIGNED. IF THE STARTING ADDRESS IS LEGAL THE READ IS TRUNCATED AS REFLECTED BY THE WORD COUNT IN THE SUBSTATUS FIELD.

10 THE AREA OF THE FASTRAND FILE BEING UNLOCKED BY THIS WRITE OR UNLOCK REQUEST TIMED OUT IN THE LOCKING LIST OR A SUBSEQUENT REQUEST BY THE SAME ACTIVITY HAD A PACKET FORMAT ERROR DETECTED BETWEEN THE TIME OF SUBMITTING THE REQUEST AND THE TIME OF SERVICING. OTHER REQUESTS BY OTHER ACTIVITIES FOR THE AREA MAY HAVE BEEN HONORED IN THE INTERIM. IF THE FUNCTION IS WRITE, THE TRANSFER IS NOT PERFORMED.

11 A NONRECOVERABLE ERROR HAS OCCURRED AND EITHER THE SUPPRESS RECOVERY MODE IS SET FOR MAGNETIC TAPE OR AN ANSWER OF 'G' WAS GIVEN TO AN ERROR MESSAGE. IF THE SUPPRESS RECOVERY MODE IS SET, THE EI STATUS CODE IS STORED IN A_1 OF THE INTERRUPT ACTIVITY CONTROL REGISTER SET.

12 A READ, OR WRITE ERROR ON MAGNETIC TAPE HAS RESULTED IN LOSS OF POSITION ON THE UNIT. THIS CODE IS RETURNED FOR ALL OUTSTANDING REQUESTS AT THE TIME THE ANSWER OF 'B' WAS ENTERED IN RESPONSE TO THE I/O ERROR MESSAGE. ANY SUBSEQUENT REQUESTS WILL BE HONORED BUT THE LOST POSITION IS MAINTAINED AND NO FURTHER PROGRAM CHECKPOINTS WILL BE VALID.

13 THE PERIPHERAL UNIT WAS DECLARED DOWN EITHER BY AN UNSOLICITED OPERATION KEYIN OR IN RESPONSE TO AN ERROR MESSAGE TYPED AFTER THE NORMAL RECOVERY FAILED TO RESOLVE A MALFUNCTION.

17 (EXEC REFERENCE ONLY) AN ATTEMPT WAS MADE TO REFERENCE A FILE FOR WHICH NO ASSIGMENT HAS BEEN MADE. THE FILE DESCRIPTION AREA EXAMINED FOR THE NAME IS EITHER THE EXECUTIVE FILES DESCRIPTION OR A RUN ASSOCIATED AREA AS DEFINED BY THE REQUESTING EXECUTIVE ROUTINE.

20 SOME FORM OF WRITE OR A FUNCTION CAUSING AREA RELEASE WAS ATTEMPTED ON A FILE ASSIGNED IN THE READ ONLY MODE, OR A FORM OF READ WAS ATTEMPTED ON A FILE IN THE WRITE ONLY

MODE.

21 AN ATTEMPT WAS MADE TO REFERENCE A FILE NAME FOR WHICH NO ASSIGNMENT HAS BEEN MADE.

22 AN ATTEMPT WAS MADE TO REFERENCE AN UNASSIGNED AREA OF A DRUM FORMAT MASS STORAGE FILE OR PAST THE MAXIMUM ASSIGNMENT FOR A FASTRAND FORMAT FILE.

23 THE ADDRESS SPECIFIED IN THE A0 REGISTER IS NOT WITHIN THE PROGRAM LIMITS OR DEFINES A PACKET SPLIT BETWEEN THE INSTRUCTION AND DATA BANKS OF THE PROGRAM.

24 THE FUNCTION CODE IS NOT DEFINED FOR THE ASSIGNED EQUIPMENT TYPE. THIS CODE ALSO COVERS NONCOMPATIBLE FIELDS ON A SET MODE REQUEST.

25 THE I/O ACCESS WORD REFERS TO A BUFFER WHICH IS WHOLLY OR PARTIALLY OUTSIDE OF THE PROGRAM AREA OR SPLIT BETWEEN THE INSTRUCTION AND DATA BANK OF THE PROGRAM. FOR GW$, SCR$,AND SCRB$ FUNCTIONS THIS ERROR CODE IS GIVEN IF THE NUMBER OF ACCESS WORDS IS 0 OR MORE THAN 50 OR IF THE TOTAL WORD COUNT IS MORE THAN 65K.

26 ILLEGAL INTERRUPT ROUTINE STARTING ADDRESS.

27 AN I/O REQUEST WAS MADE WITH THE STATUS WORD OF THE REQUEST PACKET SET NEGATIVE INDICATING A POSSIBLE PROGRAM LOOP.

030 THE INTERRUPT ACTIVITY SPECIFIED IS GREATER THAN 35 OR IS ALREADY IN USE.

031 A MAGNETIC TAPE OPERATION WAS ISSUED WITH USER RECOVERY SPECIFIED AND AN INTERRUPT ACTIVITY WAS NOT SPECIFIED.

40 THE REQUEST IS EITHER IN THE PROCESS OF BEING EXECUTED OR IS LISTED ON THE REQUEST QUEUE FOR THE PARTICULAR CHANNEL.

10.1.5. ERROR MESSAGES

FOR EACH I/O FUNCTION ON MAGNETIC TAPE OR DRUM, THE EXEC HAS A STANDARD RECOVERY PROCEDURE FOR EACH POSSIBLE MALFUNCTION WHICH MAY OCCUR. IF AN ABNORMAL STATUS IS RETURNED FOR AN I/O FUNCTION AND THE USER HAS NOT SUPPRESSED RECOVERY FOR THE FUNCTION (SEE SECTION 2 BELOW), THE EXEC WILL INITIATE THE STANDARD PROCEDURE FOR THE PARTICULAR FUNCTION TO ATTEMPT RECOVERY. IN THE EVENT THAT RECOVERY IS NOT SUCCESSFUL AN ERROR MESSAGE IS DISPLAYED ON THE CONSOLE IN THE FOLLOWING FORMAT:

```
C/U    FUNC ERROR RESPONSE
```

WHERE

C/U IS FILLED IN WITH THE CHANNEL AND UNIT OR PHYSICAL ADDRESS AT WHICH THE ERROR OCCURRED.

FUNC IS THE MNEMONIC OF THE FUNCTION BEING PERFORMED I.E. READF, READB, WRITE, WREOF, ETC.

ERROR IS AN INDICATION OF THE TYPE OF ERROR I.E. PARITY, ABNFC FOR ABNORMAL FRAME COUNT, INTLK FOR INTERLOCK, ETC.

RESPONSE IS ONE OR MORE LETTERS WHICH THE OPERATOR MAY ENTER AS A RESPONSE TO THE MESSAGE AS FOLLOWS:

A-TRY ISSUING THE FUNCTION AGAIN WITH STANDARD RECOVERY
B-RETURN BAD POSITION CODE OF 12 TO THE PACKET
D-DECLARE DEVICE DOWN AND RETURN A CODE OF 13 TO THE

```
  PACKET.  THE DOWNED UNIT IS NOT CONSIDERED FOR REUSE
  UNTIL BROUGHT BACK INTO THE AVAILABLE POOL BY AN 'UP'
  UNSOLICITED MESSAGE.
E-TREAT AS AN END-OF-FILE AND RETURN A CODE OF 01 TO THE
  PACKET.
G-UNRECOVERABLE ERROR, POSITION GOOD, RETURN A CODE OF 11
  TO THE PACKET.
```

TABLE 10-1 OCTAL AND MNEMONIC I/O CODES
(DEFINED IN SYS$*RLIB$)

FUNCTION	OCTAL	SYMBOL
WRITE	10	W$
WRITE END OF FILE	11	WEF$
CONTINGENCY WRITE	12	CW$
SKIP WRITE	13	SW$
ABSOLUTE WRITE (SYSTEM ONLY)	14	ABW$
GATHER WRITE	15	GW$
ACQUIRE FASTRAND	17	ACQ$
READ	20	R$
READ BACKWARD	21	RB$
READ AND RELEASE	22	RR$
RELEASE	23	REL$
BLOCK READ DRUM	24	BRD$
READ AND LOCK	25	RDL$
UNLOCK	26	UNL$
ABSOLUTE READ	27	ABR$
TRACK SEARCH ALL WORDS	30	TSA$
TRACK SEARCH FIRST WORD	31	TSF$
POSITION SEARCH ALL WORDS	32	PSA$
POSITION SEARCH FIRST WORD	33	PSF$
SEARCH DRUM	34	SD$
BLOCK SEARCH DRUM	35	BSD$
SEARCH READ DRUM	36	SRD$
BLOCK SEARCH READ DRUM	37	BSRD$
REWIND	40	REW$
REWIND WITH INTERLOCK	41	REWI$
SET MODE	42	SM$
SCATTER READ	43	SCR$
SCATTER READ BACKWARD	44	SCRB$
MOVE FORWARD	50	MF$
MOVE BACKWARD	51	MB$

10.2. MAGNETIC TAPE HANDLER

10.2.1. EQUIPMENT

THE 1108 EXECUTIVE SYSTEM PROVIDES SUPPORT FOR THE FOLLOWING MAGNETIC TAPE DEVICES:

UNISERVO VIC/VIIIC
UNISERVO IVC
UNISERVO IIIA
UNISERVO IIA

TABLE 10-2 LISTS THE FUNCTIONS AVAILABLE ON THE VARIOUS KINDS OF TAPE DEVICES.

10.2.2. OPERATION

THE VARIOUS TAPE FUNCTIONS ARE CONTROLLED BY A ROUTINE ALWAYS PRESENT IN CORE. RECOVERY SEQUENCES ARE STORED ON MAGNETIC DRUM AND ARE RETRIEVED ONLY AS NEEDED.

THE CURRENT POSITION OF EACH TAPE IS KEPT IN TERMS OF A BLOCK COUNT. THIS INFORMATION IS MADE AVAILABLE TO THE ERROR LOGGING, CHECKPOINT, AND ENDING LABEL ROUTINES.

NO PROVISION IS MADE WITHIN THE SYSTEM FOR AUTOMATIC TREATMENT OF MIXED PARITY AND MIXED DENSITY TAPE FILES.

UTILIZATION OF THE CONTINGENCY WRITE AND SKIP WRITE FUNCTIONS ARE AUTOMATICALLY PROVIDED FOR BY THE HANDLER AND UNLESS THE USER PROVIDES HIS OWN ERROR RECOVERY, THESE FUNCTIONS SHOULD BE OF NO CONCERN TO THE USER.

THE DEFINITION OF THE VARIOUS TAPE FUNCTIONS IS AS FOLLOWS:

FUNCTION — DEFINITION

WRITE

STARTING AT THE ADDRESS IN H2 OF WORD 5 OF THE REQUEST PACKET, TRANSFER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD5 TO FORM A SINGLE BLOCK ON MAGNETIC TAPE. TRANSFER IS ACCOMPLISHED ACCORDING TO THE STANDARD MODES OR THE REQUESTED MODES I,E. PARITY, DENSITY, ETC. NORMAL COMPLETION RESULTS WHEN ALL WORDS HAVE BEEN TRANSFERRED EXCEPT FOR 7-TRACK FORMAT EVEN PARITY WHERE A CHARACTER OF ZERO, AFTER TRANSLATION IF REQUESTED, WILL CONCLUDE THE REQUEST FOR MORE DATA BY THE SUBSYSTEM.

WRITE END OF FILE

WRITE A SENTINEL ON MAGNETIC TAPE WHICH,WHEN READ,WILL RESULT IN AN END OF FILE STATUS BEING RETURNED TO THE PROGRAM.

CONTINGENCY WRITE

WRITE ZEROES IN EVEN CHANNELS ONLY FOR 2.5 INCHES OF TAPE TO ALLOW WRITING AFTER READING FORWARD(IIIA ONLY). THIS FUNCTION IS AUTOMATICALLY PROVIDED BY THE SYSTEM AND SHOULD BE OF NO CONCERN TO THE USER UNLESS THE SUPPRESS RECOVERY MODE IS EMPLOYED.

SKIP WRITE

ERASE THREE INCHES OF TAPE THEN THE SAME AS A WRITE FUNCTION. THIS FUNCTION IS AUTOMATICALLY PROVIDED IN THE SYSTEM FOR WRITE PARITY RECOVERY. THE ONLY USER NEED WOULD BE IN THE SUPPRESS RECOVERY MODE OR IF AN EXTENDED INTERBLOCK GAP IS NEEDED ON COMPATIBLE TAPE TYPES.

GATHER WRITE

WRITE A SINGLE BLOCK ON MAGNETIC TAPE SPECIFIED BY A STRING OF ACCESS WORDS. THE NUMBER OF ACCESS WORDS IS SPECIFIED IN H1 OF WORD 5 AND THE STARTING ADDRESS OF THE STRING IS SPECIFIED IN H2 OF WORD 5.

READ FORWARD

INITIATE TAPE MOTION IN THE FORWARD DIRECTION AND TRANSFER THE WORDS READ INTO THE AREA DEFINED BY WORD 5 OF THE PACKET. TRANSFER IS NORMALLY CONCLUDED

	BY EITHER ENCOUNTERING THE END OF BLOCK OR TRANSFERRING THE NUMBER OF WORDS REQUESTED.
READ BACKWARD	SAME AS READ FORWARD EXCEPT OPPOSITE DIRECTION.
MOVE FORWARD	SAME AS READ FORWARD EXCEPT NO TRANSFER TAKES PLACE HENCE NO ACCESS WORD IS REQUIRED.
MOVE BACKWARD	BACKSPACE THE TAPE ONE BLOCK.
REWIND	REPOSITION THE TAPE AT THE 'LOAD POINT'. THIS IS THE POINT AT WHICH A READ FORWARD READS THE FIRST BLOCK ON TAPE AND A READ BACKWARDS REPORTS AN END OF TAPE STATUS.
REWIND WITH INTERLOCK	REPOSITION THE TAPE TO 'UNLOAD POINT' AND LOCK THE UNIT AGAINST FURTHER FUNCTIONS.
SCATTER READ FORWARD	SAME AS READ FORWARD EXCEPT THE WORDS READ ARE TRANSFERRED INTO AREAS SPECIFIED BY A STRING OF ACCESS WORDS DEFINED BY WORD 5.
SCATTER READ BACKWARD	SAME AS SCATTER READ FORWARD EXCEPT OPPOSITE MOTION DIRECTION.

10.2.3. THE SET MODE REQUEST

FOR THE SET MODE FUNCTION, THE I/O ACCESS WORD SHOULD BE SET TO POINT TO A ONE-WORD BUFFER WHICH DEFINES THE MODES TO BE SET. THE BUFFER WORD IS FORMATTED AS FIVE 2-BIT FIELDS STARTING IN THE MOST SIGNIFICANT BIT FOLLOWED BY AN 8-BIT AND 18-BIT FIELD (FORM 2,2,2,2,2,8,18). THE 8-BIT FIELD SHOULD BE CODED AS ZERO. THE VALUES IN THE 2-BIT FIELDS ARE INTERPRETED AS FOLLOWS:

FIELD 1-DENSITY
0=NO CHANGE
1=LOW
2=MEDIUM
3=HIGH

FIELD 2-PARITY
0=NO CHANGE
1=ODD (BINARY)
2=EVEN (BCD)

FIELD 3-TRANSLATE
0=NO CHANGE
1=SET CHARACTER TRANSLATE MODE
2=DISCONTINUE TRANSLATION

FIELD 4-ALLOW NOISE
0=NO CHANGE
1=SET THE NOISE CONSTANT TO THE NUMBER OF CHARACTERS IN THE 18-BIT FIELD (H2)

FIELD 5-SUPPRESS RECOVERY

0=NO CHANGE
1=RETURN EXTERNAL INTERRUPT STATUS CODE TO THE WORKER PROGRAM IN CASE OF MALFUNCTION WITHOUT ATTEMPTING RECOVERY.
2=DISCONTINUE SUPPRESS RECOVERY MODE

IN LIEU OF SETTING MODES BY THE WORKER PROGRAM, THE STANDARD MODES ESTABLISHED BY THE EXEC AT INITIALIZATION AND RE-ESTABLISHED WHENEVER A TAPE IS RELEASED ARE:

HIGH DENSITY
ODD PARITY
NO CHARACTER TRANSLATION
18-CHARACTER NOISE CONSTANT
STANDARD RECOVERY

IN ADDITION TO THE SERVICE ENTRANCE, THE PARITY AND DENSITY MODES CAN BE SET BY THE OPTIONS ON THE ASG CARD.

THE MAGNETIC TAPE HANDLER IS SET WITH A STANDARD 'NOISE CONSTANT'. THIS VALUE IS INTERPRETED BY THE HANDLER TO MEAN THAT ON READING ANY BLOCK SHORTER THAN THIS VALUE IF A PARITY ERROR IS DETECTED THE BLOCK WILL BE CONSIDERED AS INTERBLOCK-GAP NOISE AND WILL BE PASSED OVER. ANY RECORDING DETECTED WHILE WRITING WHICH IS LESS THAT THE NOISE CONSTANT WILL BE PASSED OVER IF IT IS DETERMINED THAT THE BLOCK CAN BE BYPASSED AS 'NOISE' WHEN READING WITH THE SAME NOISE CONSTANT. BY USE OF THE SET MODE FUNCTION, THE USER CAN SET THE NOISE CONSTANT TO WHATEVER VALUE IS DESIRED. IT IS ASSUMED BY THE HANDLER THAT IF THE NOISE CONSTANT IS CHANGED FOR WRITING A TAPE, THE SAME CONSTANT WILL BE USED FOR READING. THIS ASSUMPTION IS IMPORTANT IF AN INCREASE IN THE CONSTANT IS MADE FOR WRITING, AS BLOCKS MAY BE LEFT ON THE TAPE WHICH ARE UNREADABLE WITH THE STANDARD NOISE CONSTANT. IF 'NOISE BLOCKS' ARE READ FROM TAPE OR LEFT ON TAPE WHEN WRITING, AT THE TIME THE TAPE IS REWOUND A MESSAGE OF THE FORM:

```
C/U NOISE W-XX R-YY
```

IS DISPLAYED ON THE CONSOLE WHERE XX IS THE NUMBER OF REMAINING BLOCKS WHEN WRITING AND YY IS THE NUMBER OF BLOCKS BYPASSED WHEN READING. ALSO WHENEVER NOISE IS LEFT THE COUNT ENTERED IN THE SYSTEM LOG IS INCREMENTED. THE COUNT OF NOISE BLOCKS BYPASSED UPON READING IS ALSO RECORDED IN THE LOG. IF A TAPE IS READ AT A DENSITY OTHER THAN THE RECORDING DENSITY, IT MAY RESULT IN HAVING A LEGAL DATA BLOCK OF GREATER LENGTH THAN THE NOISE CONSTANT APPEAR TO BE A BLOCK OF NOISE. THE NOISE CONSTANT CAN BE CHANGED BY THE SET MODE ENTRANCE ONLY WHEN THE TAPE IS IN THE REWOUND POSITION. THIS IS NEEDED TO INSURE PROPER POSITIONING FOR RESTART.

TRANSLATION ON IVC, VIC, OR VIIIC CHANNELS IS ASSUMED ACCORDING TO THE WAY IN WHICH THE PLUGBOARD IS WIRED IF THE OPTIONAL BOARD IS IN THE CONFIGURATION. (TABLE 10-3 SHOWS THE STANDARD MANNER IN WHICH THE PLUGBOARD IS WIRED.) FOR THIS CASE, THE HANDLER DOES NO MORE THAN SET THE PROPER BIT IN THE FUNCTION WORD. IF THE CONFIGURATION IS WITHOUT THE PLUGBOARD, THE STANDARD TRANSLATION BY THE SOFTWARE ON THE ABOVE MENTIONED TAPE TYPES IS FROM BCD TO FIELDATA ON READING AND THE REVERSE ON WRITING (THE SAME AS HARDWARE STANDARD IN TABLE 10-3). HENCE THE PARITY MODE SHOULD BE SET TO EVEN IF THE TRANSLATE MODE IS TO BE SET USING THE SOFTWARE. TRANSLATION ON THE VIC/VIIIC UNITS IS ALLOWABLE ON 7-CHANNEL OPERATION ONLY. CHARACTER TRANSLATION BY THE SOFTWARE WILL ADD APPROXIMATELY 40 MICROSECONDS PER WORD TO THE EXECUTION OF THE PROGRAM. SOFTWARE TRANSLATION IS REMOVED AT SYSTEM GENERATION TIME IF HARDWARE TRANSLATION IS AVAILABLE ON ANY CHANNEL. THE USER MAY ALSO SPECIFY REMOVAL OF SOFTWARE TRANSLATION AT GENERATION IF HARDWARE TRANSLATION IS NOT INSTALLED AND NO TRANSLATION IS ANTICIPATED. IF HARDWARE TRANSLATION IS NOT AVAILABLE ON ALL CHANNELS AND THE MODE IS TO BE SET, IT IS IMPORTANT THAT THE ASG CARD SHOW THIS REQUIREMENT.

IF THE SUPPRESS RECOVERY MODE IS SET, THE TAPE HANDLER WILL NOT ANALYZE THE INTERRUPT BUT INSTEAD THE USER PROGRAM IS GIVEN CONTROL WITH A STATUS CODE OF 11 (SUPPRESS RECOVERY SET), AND THE STATUS WORD IN THE A1 REGISTER OF THE INTERRUPT ACTIVITY CONTROL REGISTER SET. THE SUBSTATUS FIELD REFLECT THE FINAL STATE OF THE INPUT OR OUTPUT ACCESS WORD.

IF THE MODES SPECIFIED IN THE SERVICE REQUEST ARE NOT COMPATIBLE, A STATUS CODE OF 21 IS STORED IN THE PACKET AND CONTROL IS GIVEN TO THE USER'S ERROR TERMINATION ROUTINE IF SPECIFIED. NONCONFLICTING MODES STARTING WITH FIELD-1 (DENSITY) WILL BE SET AS SPECIFIED IN THE REQUEST.

10.2.4. STANDARD RECOVERY

RECOVERY BY THE HANDLER CONSISTS OF AN OPTIMUM NUMBER OF REREADS AT VARIOUS GAINS (OR WHATEVER IS APPROPRIATE FOR THE TYPE OF UNIT) OR A NUMBER OF BACKSPACE/SKIP-WRITE CYCLES BASED ON THE SIZE OF THE BLOCK AND THE TAPE DENSITY. AN ERROR COUNT IS KEPT FOR EACH TAPE. SYSTEM LOG ENTRIES ARE MADE IF THE NUMBER OF RECOVERIES EXCEEDS AN ACCEPTABLE THRESHOLD.

IN THE EVENT THAT THE RECOVERY PROVIDED BY THE SYSTEM FAILS TO RESOLVE THE ERROR CONDITION, ONE OF THE FOLLOWING MESSAGES IS DISPLAYED ON THE CONSOLE:

MESSAGE	RESPONSES	COMMENTS
C/U INTRLK FUNC	A,B,D,E	THE REQUESTED UNIT IS IN AN INOPERABLE CONDITION. THIS CAN RESULT FROM FAILURE TO INSERT A WRITE ENABLE RING, POWER OFF, UNIT ON LOCAL, ETC.
C/U HASH FUNC	B,D	FOR VI/VIIIC UNITS THIS ERROR RESULTS FROM READING INFORMATION IN THE INTERBLOCK GAP WHEN ATTEMPTING A W, SW, OR WEF FUNCTION. LOST POSITION IS FLAGGED FOR THE UNIT AND NO FURTHER RECOVERY ATTEMPT CAN BE RELIED UPON.
C/U PARITY-1 FUNC	A,B,D,G,E	RECOVERY TO PARITY ON THE FUNCTION WAS UNSUCCESSFUL. AN ANSWER OF 'A' WILL CAUSE THE HANDLER TO REPOSITION AND ATTEMPT TO PERFORM THE FUNCTION AGAIN. THE TAPE IS POSITIONED SUCH THAT IF THE FUNCTION IS A READ ANOTHER READ OPERATION IN THE SAME DIRECTION WILL READ THE NEXT BLOCK IN THAT DIRECTION. THIS WOULD BE THE CASE FOR A READ AFTER ANSWERING 'G' TO THIS MESSAGE.
C/U PARITY-2 FUNC	A,B,D	PARITY OCCURRED ON THE FIRST WRITE ATTEMPT FROM LOAD POINT. NO RECOVERY HAS BEEN ATTEMPTED TO ALLOW THE OPERATOR TO MOUNT A NEW REEL IF DESIRED.
C/U PARITY-3 FUNC	B,D	PARITY RECOVERY BY THE HANDLER HAS BEEN UNSUCCESSFUL. FOR WRITE FUNCTIONS THE RECOVERY WHICH EITHER RESULTED IN A BAD BLOCK BEING LEFT ON THE TAPE OR LOSS OF POSITION. FOR READ OPERATIONS THE NUMBER OF WORDS SPECIED IN THE INPUT/OUTPUT PACKET WAS LESS THAN THE NOISE CONSTANT.

C/U UNDFNC FUNC	A,D	THE FUNC FIELD (NUMERIC FOR THIS CODE) CONTAINS A VALUE NOT DEFINED FOR THE SUBSYSTEM. IF THE FUNCTION IS IN THE REPERTOIRE OF THE TAPE TYPE AND IS NOT A FORM OF WRITE ON A READ-ONLY CHANNEL, THE FAULT IS IN THE SUBSYSTEM; OTHERWISE THE ERROR IS EXTERNAL TO THE TAPE SUBSYSTEM.
C/U UNEXP-XX FUNC	A,D,G	THE INTERRUPT STATUS CODE XX IS NOT EXPECTED FOR THE GIVEN FUNCTION.

10.2.5. CONSIDERATIONS FOR IV/VI/VIIIC TAPE TYPES

THE FOLLOWING DETAILS MUST BE TAKEN INTO ACCOUNT WHEN USING THE TAPE HANDLER FOR THE COMPATIBLE TAPE TYPES:

1. READ BACKWARD LIMITATIONS

   THE READ BACKWARD FUNCTION ON THE VI/VIIIC UNITS SHOULD NOT BE USED IF THE TAPE TO BE READ HAS BEEN RECORDED ON OTHER THAN A VIC OR VIIIC UNIT. IT IS NECESSARY THAT THE RECORDING PRODUCE A 'STATICALLY DESKEWED LONGITUDINAL CHECK FRAME' TO PREVENT THE READ BACKWARD FUNCTION FROM INTERPRETING THE CHECK FRAME AS DATA FRAMES.

   IF A BLOCK IS RECORDED IN 7-TRACK FORMAT WITH A BLOCK LENGTH GREATER THAN 5 FRAMES AND NOT A MULTIPLE OF 6, A READ BACKWARD PRODUCES A DIFFERENT FORMAT THAN A READ FORWARD OF THE SAME BLOCK. FOR EXAMPLE,IF THE BLOCK LENGTH IS 7 FRAMES A READ FORWARD RESULTS IN ASSEMBLING FRAMES 1-6 AS THE FIRST WORD AND FRAME 7 AS THE SECOND AND A READ BACKWARD RESULTS IN ASSEMBLING FRAMES 2-7 AS THE FIRST WORD AND FRAME 1 AS THE SECOND.

   THE SAME TYPE OF BUFFER VARIATION EXISTS FOR A READ BACKWARD ON A 9-TRACK UNIT IF THE WRITE BUFFER LENGTH IS NOT A MULTIPLE OF 2 WORDS (9-FRAMES). A ONE WORD WRITE ON A 9-TRACK UNIT RESULTS IN FIVE FRAMES BEING RECORDED WITH THE FIFTH FRAME CONTAINING 4-BITS OF ZERO PADDING. A READ BACKWARD RESULTS IN THE 4-BITS OF PADDING APPEARING AS THE LEAST SIGNIFICANT 4 BITS OF THE FIRST WORD ASSEMBLED. FURTHERMORE, REGARDLESS OF THE DIRECTION OF READING, IF A BLOCK IS WRITTEN ON A 9-TRACK FORMAT UNIT WITH AN ODD WORD COUNT IN THE ACCESS WORD, ONE MORE WORD IS MADE AVAILABLE AS INPUT THAN WAS SENT OUT TO BE WRITTEN

2. WRITE CONSIDERATIONS

   IF THE USER ATTEMPTS TO WRITE END OF FILE SENTINELS ON 7-TRACK UNITS BY DOING AN EVEN PARITY WRITE WITH TRUNCATION CAUSED BY A ZERO CHARACTER, AND IF THE TRANSLATE MODE IS SET,IT IS ESSENTIAL THAT THE FIRST TWO CHARACTERS OF THE BUFFER TRANSLATE TO 1700 OCTAL TO CAUSE AN END OF FILE STATUS WHEN READ.

   SINCE HARDWARE TRANSLATION MAY BE AVAILABLE AND THE USER HAS THE ABILITY TO VARY THE TRANSLATION, CARE MUST BE EXERCISED TO PREVENT UNWANTED TRANSLATION OF A CHARACTER TO ZERO WHICH CAUSES TRUNCATING A WRITE TRANSFER WHEN WRITING IN THE EVEN PARITY MODE. IF SOFTWARE CONVERSION IS

USED FOR WRITE OPERATIONS THE WORDS ARE CONVERTED IN THE BUFFER BEFORE THE WRITE OPERATION IS PERFORMED. IF A BLOCK IS WRITTEN
WITH LESS CHARACTERS THAN THE NOISE CONSTANT, THE RISK EXISTS OF BYPASSING THE DATA BLOCK AS NOISE WHEN READING. ALSO, A ZERO AS THE FIRST CHARACTER RESULTS IN AN ERRONEOUS BLOCK COUNT. ON A ZERO CHARACTER COUNT, UP TO THREE WORDS LEAVE THE COMPUTER AND ARE CONSIDERED TO HAVE BEEN WRITTEN AS REFLECTED IN THE COUNT IN THE SUBSTATUS FIELD OF THE REQUEST PACKET.

THE RECOVERY PROCEDURE FOR A PARITY ERROR OR CERTAIN TAPE HASH ERROR ON A WRITE OPERATION MAY UTILIZE TWO FEET OF TAPE OR TWICE THE LENGTH OF THE BLOCK WHICHEVER IS LARGER. HENCE, IF BLOCKS ARE TO BE RECORDED WHICH ARE LONGER THAN TWO FEET (OR LESS DEPENDING UPON WHETHER AN ENDING INTERRUPT ACTIVITY SUBMITS THE NEXT REQUEST OR IF REQUESTS ARE QUEUED AHEAD BY I/O CONTROL), IT IS RECOMMENDED THAT TAPES BE USED WHICH HAVE THE END OF TAPE WARNING MARKER PLACED FURTHER FROM THE END OF TAPE. THE NORMAL PLACEMENT IS 14 FEET FROM THE END OF TAPE AND IT IS RECOMMENDED THAT AT LEAST 10 FEET OF TAPE REMAIN ON THE SUPPLY SIDE OF THE WRITE HEAD TO INSURE THAT THE TAPE IS NOT PULLED OFF THE SUPPLY REEL.

3. MOVE CONSIDERATIONS

THE MOVE FORWARD AND MOVE BACKWARD FUNCTIONS ARE INCLUDED TO COVER THE CONDITION WHERE THE ONLY CONCERN IS POSITION. PARITY ERRORS ARE NOT REPORTED AND ARE ONLY EXAMINED TO DETERMINE NOISE BLOCKS. FOR THE IVC UNITS THE PARITY STATUS IS NOT RETURNED FOR THE BACKSPACE BLOCK FUNCTION, THEREFORE THE MOVE BACKWARD IS NOT RECOMMENDED ON THE IVC IF NOISE IS A PROBLEM AS LOST POSITION MAY RESULT.

10.2.6. MULTIPLE-CHANNEL OPERATION

THE MAGNETIC TAPE HANDLER IS CAPABLE OF A SIMULTANEOUS OPERATION ON ANY NUMBER OF CHANNELS INVOLVING ANY MIXTURE OF TAPE DEVICE TYPES.

THE FULL DUAL CHANNEL ON UNISERVO VIC/VIIIC WILL BE SUPPORTED WITHOUT USER COGNIZANCE.

TABLE 10-2 TAPE FUNCTIONS

FUNCTION	CODE	PKT	VIC/VIIIC	IIIC/IVC	IIIA	IIA
	OCT	LEN				
READ FORWARD	20	5	*	*	*	*
READ BACKWARD	21	5	*	I	*	*
SCATTER READ	43	5	*	*	I	I
SCATTER READ BKWD	44	5	*	I	I	I
MOVE FORWARD	50	4	*	*	*	*
MOVE BACKWARD	51	4	*	*	*	*
WRITE	10	5	*	*	*	*
WRITE END OF FILE	11	4	*	*	*	I
CONTINGENCY WRITE	12	4	I	I	*	I
SKIP-WRITE	13	4	*	*	I	I
GATHER WRITE	15	5	*	*	I	I
REWIND	40	4	*	*	*	*
REWIND WITH INTLK	41	4	*	*	*	*
SET MODE	42	5	*	*	*	*
HIGH DENSITY			*	*	*	*
MEDIUM DENSITY			*	*	I	I
LOW DENSITY			*	*	*	*
ODD PARITY			*	*	I	I
EVEN PARITY			*	*	I	I
TRANSLATE			*	*	*	*
ALLOW NOISE			*	*	*	*
SUPPRESS RECOVERY			*	*	*	*

CODE *-AVAILABLE
I-ILLEGAL FUNCTION, CAUSES TERMINATION

TABLE 10-3 STANDARD TAPE TRANSLATION

TAPE TO PROCESSOR				PROCESSOR TO TAPE			
TAPE CODE	CPU CODE	TAPE CODE	CPU CODE	CPU CODE	TAPE CODE	CPU CODE	TAPE CODE
00	46	40	41	00	17	40	74
01	61	41	17	01	75	41	40
02	62	42	20	02	55	42	60
05	63	43	21	03	77	43	76
04	64	44	22	04	57	44	13
05	65	45	23	05	20	45	16
06	66	46	24	06	61	46	00
07	67	47	25	07	62	47	53
10	70	50	26	10	63	50	54
11	71	51	27	11	64	51	34
12	60	52	55	12	65	52	35
13	44	53	47	13	66	53	15
14	72	54	50	14	67	54	72
15	53	55	02	15	70	55	52
16	45	56	73	16	71	56	33
17	00	57	04	17	41	57	36
20	05	60	42	20	42	60	12
21	74	61	06	21	43	61	01
22	30	62	07	22	44	62	02
23	31	63	10	26	45	63	03
24	32	64	11	24	46	64	04
25	33	65	12	25	47	65	05
26	34	66	13	26	50	66	06
27	35	67	14	27	51	67	07
30	36	70	15	30	22	70	10
31	37	71	16	31	23	71	11
32	77	72	54	32	24	72	14
33	56	73	75	33	25	73	56
34	51	74	40	34	26	74	21
35	52	75	01	35	27	75	73
36	57	76	43	36	30	76	37
37	76	77	03	37	31	77	32

10.3. MAGNETIC DRUM HANDLER

10.3.1. EQUIPMENT

THE MAGNETIC DRUM HANDLER PROVIDES FOR CONTROL OVER OPERATIONS ON THE FH432 AND FH880 DRUM UNITS. EXTENSIONS WILL BE MADE TO THE HANDLER AT A FUTURE DATE TO INCORPORATE THE FH1782 AND A SUBSYSTEM CONSISTING OF FH432 AND FH1782 UNITS ON THE SAME SINGLE OR DUAL CHANNEL.

10.3.2. OPERATION

TWO GENERAL MODES OF DRUM OPERATION ARE PROVIDED WITHIN THE DRUM HANDLER. THE FIRST IS DRUM SIMULATION OF FASTRAND WHICH ALLOWS EXECUTION OF A PROGRAM WITH FILES DESIGNED FOR FASTRAND ALLOCATION ALLOCATED TO THE SECTION OF THE 'FLYING HEAD' DRUM STORAGE SPECIFIED AS SIMULATED FASTRAND. THE SECOND MODE IS DRUM AS A RANDOM STORAGE DEVICE ALLOCATED IN FIXED ASSIGNMENTS WITH ONE WORD GRANULARITY. THE INTERPRETATION OF FUNCTION CODES FOR SIMULATED FASTRAND ARE DEFINED IN THE FASTRAND HANDLER SECTION. FOR DRUM FORMAT THE FUNCTIONS ARE DEFINED AS FOLLOWS:

FUNCTION — DEFINITION

WRITE

STARTING AT THE CORE ADDRESS SPECIFIED IN H2 OF WORD 5 TRANSFER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5 TO THE DRUM AREA STARTING AT THE RELATIVE ADDRESS IN WORD 6 OF THE PACKET.

GATHER WRITE

TRANSFER THE NUMBER OF WORDS SPECIFIED BY A STRING OF ACCESS WORDS SPECIFIED BY WORD 5 FROM THE AREAS SPECIFIED BY THESE ACCESS WORDS TO THE DRUM AREA STARTING AT THE RELATIVE ADDRESS IN WORD 6. THE NUMBER OF ACCESS WORDS IS SPECIEIED IN H1 OF WORD 5 AND THE ADDRESS OF THE ACCESS WORDS IS SPECIFIED IN H2 OF WORD 5.

READ

STARTING AT THE RELATIVE DRUM ADDRESS IN WORD 6 OF THE REQUEST PACKET TRANSFER THE NUMBER OF WORDS IN H1 OF WORD 5 INTO THE AREA STARTING AT THE ADDRESS IN H2 OF WORD 5. NORMAL COMPLETION(STATUS 00) INDICATES THE SPECIFIED NUMBER OF WORDS HAVE BEEN TRANSFERRED TO CORE FROM DRUM.

SCATTER READ

STARTING AT THE RELATIVE DRUM ADDRESS IN WORD 6 OF THE PACKET TRANSFER THE NUMBER OF WORDS SPECIFIED BY A STRING OF ACCESS WORDS DEFINED BY WORD 5 TO THE AREAS SPECIFIED BY THESE ACCESS WORDS. THE NUMBER OF ACCESS WORDS IS SPECIFIED IN H1 OF WORD 5 AND THE ADDRESS OF THE ACCESS WORDS IS SPECIFIED IN H2 OF WORD 5.

BLOCK READ

STARTING AT THE RELATIVE DRUM ADDRESS IN WORD 6 OF THE REQUEST PACKET TRANSFER WORDS FROM DRUM TO CORE AT THE ADDRESS IN H2 OF WORD 5 UNTIL EITHER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5 HAS BEEN READ OR UNTIL THE END-OF-BLOCK SENTINEL (A WORD OF ALL ONES) IS READ. ENCOUNTERING A SENTINEL IS NOTED BY AN 01 STATUS CODE AND THE SENTINEL WORD IS TRANSFERRED AS THE FINAL WORD IN THE BUFFER. THE SUBSTATUS FIELD INDICATES THE NUMBER OF WORDS READ. IF COMPLETION IS BY END OF BLOCK AND THE BUFFER LENGTH IS SUCH THAT ANOTHER WORD CAN BE ACCEPTED, THE OVERFLOW WORD (THE WORD ON DRUM FOLLOWING THE SENTINEL) IS STORED IN THE BUFFER FOLLOWING (PRECEDING IF DECREMENTATION) THE SENTINEL WORD WITH THE UPPER 6 BITS SET TO 04.

SEARCH

STARTING AT THE RELATIVE DRUM ADDRESS IN WORD 6 COMPARE ALL WORDS ON DRUM UNTIL EITHER A A MATCH EQUAL IS MADE WITH WORD 7 OF THE PACKET OR UNTIL THE REMAINDER OF THE FILE HAS BEEN TESTED. IF A 'FIND' IS MADE (STATUS 00) THE RELATIVE ADDRESS OF THE FIND IS STORED IN WORD 8 OF THE PACKET.

BLOCK SEARCH

SAME AS A SEARCH WITH THE ADDED CONDITION THAT READING AN END-OF-BLOCK SENTINEL WORD TERMINATES THE SEARCH WITH A NO FIND, 03, STATUS CODE.

SEARCH READ

STARTING AT THE RELATIVE DRUM ADDRESS IN WORD 6 COMPARE ALL WORDS ON DRUM UNTIL EITHER A COMPARE EQUAL IS MADE WITH WORD 7 OF THE PACKET OR UNTIL ALL REMAINING WORDS OF THE FILE ARE TESTED. IF A 'FIND' IS MADE, STORE THE RELATIVE ADDRESS OF THE FIND WORD IN WORD 8 OF THE PACKET AND TRANSFER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5 INTO THE CORE AREA STARTING AT THE ADDRESS IN H2 OF WORD 5. TRUNCATE THE READ CYCLE IF THE END OF ASSIGNMENT PRECEDES THE COUNT IN H1 OF WORD 5.

BLOCK SEARCH READ

STARTING AT THE RELATIVE DRUM

ADDRESS IN WORD 6 OF THE PACKET COMPARE EQUAL BETWEEN THE DRUM WORDS AND WORD 7 OF THE PACKET. NO FIND IS DENOTED BY ENCOUNTERING AN END-OF-BLOCK SENTINEL OR THE END OF THE ASSIGNED AREA FOR THE FILE. UPON A FIND, STORE THE RELATIVE ADDRESS OF THE FIND WORD IN WORD 8 OF THE PACKET AND TRANSFER WORDS AS FOR THE BLOCK READ WITH TRUNCATING FOR END-OF-BLOCK SENTINEL OR END OF ASSIGNED AREA (END OF ASSIGNED AREA IF THE FINAL WORD IS NOT THE EOB SENTINEL). STORING OF THE OVERFLOW WORD FOLLOWS THE SAME CRITERIA AS THE BLOCK READ FUNCTION.

THESE FUNCTIONS ARE PERFORMED ON AREAS RESERVED THROUGH THE USE OF THE ASG CONTROL CARD. THESE ASSIGNMENTS ARE FIXED IN LENGTH, HENCE AN ATTEMPT TO READ, WRITE, OR INITIATE A SEARCH PAST THE END OF THE ASSIGNED AREA RESULTS IN AN ERROR CONDITION. IF ANY PART OF A READ AFTER SEARCH FIND IS OUTSIDE OF THE ASSIGNMENT, THE REQUEST IS TRUNCATED. A WRITE REQUEST MUST BE TOTALLY WITHIN THE ASSIGNED AREA.

SEARCH FUNCTIONS ARE TERMINATED BY THE SOFTWARE AFTER A TIME INTERVAL HAS ELAPSED WHICH WOULD ALLOW PASSING OVER THE AREA OF CONCERN WITHOUT RECEIVING AN INTERRUPT INDICATING A FIND. THE AREA OF CONCERN WOULD BE THE END OF THE ASSIGNMENT FOR NORMAL DRUM OR THE END OF TRACK OR POSITION EQUIVALENT IF SIMULATED FASTRAND. THUS THE LENGTH OF TIME THE SUBSYSTEM IS 'TIED UP' FOR A SEARCH IS NEARLY THE SAME AS A READ OR WRITE OF THE SAME LENGTH AREA. THE HANDLER INSURES THAT A SEARCH FIND IS WITHIN THE ASSIGNED AREA BEFORE READING THUS GUARANTYING FILE PRIVACY. IF A READ AFTER SEARCH MUST BE TRUNCATED, A STATUS CODE OF 01 IS RETURNED TO THE PROGRAM.

10.3.3. RECOVERY

RECOVERY OF ERRONEOUS DRUM OPERATIONS CONSISTS OF REPEATED ATTEMPTS TO PERFORM THE OPERATION. THE RECOVERY PROCEDURES RESIDE PERMANENTLY IN CORE WITH THE REST OF THE HANDLER. IF RECOVERY FAILS ONE OF THE FOLLOWING MESSAGES IS DISPLAYED ON THE CONSOLE. DIFFERENTIATION OF UNIT NUMBER AND ADDRESS IS MADE BY THE NUMBER OF CHARACTERS; ONE OR TWO FOR UNIT NUMBER AND ALWAYS SEVEN FOR AN ADDRESS. THE ADDRESS DISPLAYED IS AN OCTAL VALUE SHOWING THE POINT AT WHICH TRANSFER WAS TERMINATED BY AN EXTERNAL INTERRUPT. FOR A WRITE FUNCTION DEPENDING UPON DRUM TYPE THE CONTROL UNIT MAY HAVE BEEN BUFFERING UP TO TWO WORDS WHEN THE ERROR OCCURRED.

MESSAGE		RESPONSES	COMMENTS
C/A	PARITY-06 FUNC	A,D,G	PARITY FOR OTHER THAN READ FUNCTION.
C/A	PARITY-07 FUNC	A,D,G	PARITY ON WORD FOLLOWING EOB SENTINEL.
C/A	PARITY-64 FUNC	A,D,G	PARITY ERROR DURING READ FUNCTION.
C/A	WFAULT-14 FUNC	A,D,G	SELECTION OF MORE THAN ONE WRITE HEAD. ALSO OVERTEMP ON FH880 FOR ANY FUNCTION. NO RECOVERY IS ATTEMPTED FOR THIS ERROR.
C/U	CSCHAR-30 FUNC	A,D,G	FH880 ONLY. CHANNEL SYNC

				CHARACTER COUNTER OVERFLOW.
C/U	CUSEQ-60	FUNC	A,D,G	FH880 ONLY. CHARACTER TIMING AND WORD MARK OUT OF SYNC.
C/U	CUCHAR-70	FUNC	A,D,G	FH880 ONLY. DELAY IN CHAR TRANSFER DURING A WRITE.
C/A	UNDADD-54	FUNC	A,D,G	NON-EXISTENT ADDRESS, ADDRESS OF AN INOPERABLE DRUM UNIT OR ADDRESS PROTECTED FROM WRITING BY THE LOCKOUT SWITCH.
C/U	UNDFNC-50	FUNC	A,D,G	THE FUNC FIELD FOR THIS ERROR CONTAINS A NUMERIC VALUE. IF THIS VALUE IS A LEGAL DRUM FUNCTION, THE FAILURE IS MOST LIKELY IN THE DRUM CONTROL LOGIC (EXCEPTION IN DROPPED BITS);OTHERWISE AN INTERNAL ERROR (EITHER HARDWARE OR SOFTWARE) CAUSED THE INCORRECT VALUE TO BE USED AS THE FUNCTION CODE.
C/U	UNEXP-XX	FUNC	A,D,G	THE INTERRUPT STATUS CODE XX IS NOT EXPECTED FOR THE GIVEN FUNCTION.

10.3.4. MULTIPLE-CHANNEL OPERATION

THE FULL-DUAL OPTION AVAILABLE ON THE HYBRID SYSTEM WILL BE SUPPORTED WITHOUT USER INCONVENIENCE. THE DRUM HANDLER WILL BE CAPABLE OF SIMULTANEOUS OPERATION ON ANY NUMBER OF CHANNELS INVOLVING A COMBINATION OF DRUM TYPES

10.4. FASTRAND HANDLER

10.4.1. GENERAL

ALTHOUGH THE SYSTEM WILL FUNCTION WITHOUT A PHYSICAL FASTRAND UNIT, AT LEAST SOME PORTION OF MAGNETIC DRUM MUST BE SET ASIDE TO SIMULATE FASTRAND IN ITS ABSENCE. THE MINIMUM FASTRAND FORMAT IS SPACE FOR SYMBIONT INPUT AND OUTPUT FILES, SYSTEM PROCESSOR DATA AREA, PROGRAM FILE STORAGE, AND OTHER SYSTEM FUNCTIONS. SPACE ON FASTRAND IS ASSIGNED IN GRANULES OF 64 OR 4096 SECTORS. A FILE CONSISTING OF MORE THAN ONE GRANULE MAY BE CONSIDERED CONTIGUOUS BY THE PROGRAMMER BECAUSE THE FASTRAND HANDLER WILL TAKE CARE OF THE PROCESSING THAT MUST OCCUR WHENEVER A GRANULE BOUNDARY IS PASSED. THE FASTRAND HANDLERS WORK IN CLOSE CONJUNCTION WITH THE FILE SUPERVISOR IN ORDER TO CONVERT THE RELATIVE SECTOR ADDRESSES SUPPLIED BY THE USER PROGRAM INTO PHYSICAL-CHANNEL, UNIT, POSITION, AND SECTOR ADDRESSES.

AN ATTEMPT TO READ FROM AN AREA OF A FILE WHICH IS NOT ENTIRELY ASSIGNED RESULTS IN A STATUS CODE OF 05 BEING RETURNED TO THE PACKET. IF THE AREA STARTS WITHIN THE ASSIGNMENT AND RUNS BEYOND, THE SUBSTATUS COUNT REFLECTS THE PART ASSIGNED. IF GRANULES HAVE BEEN RELEASED CAUSING VOIDS WITHIN THE FILE, A REQUEST COULD SHOW A LEGAL START AND ENDING ADDRESS BUT A VOID WITHIN AND WOULD RESULT IN THE 05 STATUS WITH ONLY THE FIRST PART READ. WRITING INTO AN UNASSIGNED AREA OF A FASTRAND FILE WILL CAUSE SPACE TO BE ASSIGNED TO THAT PORTION OF THE FILE. THE AUTOMATIC EXPANSION ON A WRITE FUNCTION CAN BE NEGATED BY THE MAXIMUM ASSIGNMENT FIELD ON THE ASG STATEMENT IN WHICH CASE A STATUS CODE OF 22 IS RETURNED TO THE PACKET.

10.4.2. EQUIPMENT

THE FOLLOWING EQUIPMENT OPTIONS ARE SUPPORTED BY THE FASTRAND HANDLER:

MODEL II FASTRAND
FASTBAND
WRITE LOCKOUT

THE WRITE LOCKOUT FEATURE IS SUPPORTED BY ALLOWING THE USER TO CREATE FILES ON FASTRAND AT ABSOLUTE LOCATIONS AT SYSTEM GENERATION TIME. THUS THE 'LOCKED OUT' AREAS CAN BE SET ASIDE AND LOADED AT ANY TIME BY THE USER. THE HANDLER MAKES NO ASSUMPTIONS CONCERNING THE POSITION OF THE MANUAL WRITE LOCKOUT SWITCHES AND WILL INITIATE WRITE FUNCTIONS AT ANY ADDRESS WITHIN THE CONFIGURATION UNLESS THE FILE IS IN THE READ ONLY MODE. IF THE MANUAL SWITCHES ARE TO BE SET, IT IS THE USER RESPONSIBILITY TO INSURE THAT PART OF A LOCKED OUT AREA IS NOT LEFT AVAILABLE TO BE ALLOCATED BY EXEC TO A PROGRAM WHICH WILL ISSUE WRITE FUNCTIONS.

10.4.3. OPERATION

THE FASTRAND HANDLER ACCEPTS THE FOLLOWING FUNCTIONS:

FUNCTION — DEFINITION

READ — STARTING AT THE RELATIVE SECTOR ADDRESS SPECIFIED IN WORD 6 OF THE REQUEST PACKET, TRANSFER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5 INTO THE CORE AREA STARTING AT THE ADDRESS IN H2 OF WORD 5. READING ALWAYS STARTS AT A SECTOR BOUNDARY BUT MAY END ANYWHERE.

SCATTER READ — STARTING AT THE RELATIVE SECTOR ADDRESS SPECIFIED IN WORD 6 OF THE REQUEST PACKET, TRANSFER THE NUMBER OF WORDS SPECIFIED BY A STRING OF ACCESS WORDS SPECIFIED BY WORD 5 INTO THE CORE AREAS SPECIFIED BY THESE ACCESS WORDS. THE NUMBER OF ACCESS WORDS IS SPECIFIED IN H1 OF WORD 5 AND THE ADDRESS OF THE ACCESS WORDS IS SPECIFIED IN H2 OF WORD 5.

READ AND RELEASE — SAME AS READ WITH THE ADDITIONAL CONDITION THAT AFTER THE READ HAS BEEN PERFORMED ALL GRANULES WITH ANY PART WITHIN THE SET OF ADDRESSES DESCRIBED BY THE PACKET ARE RELEASED TO THE AVAILABLE MASS STORAGE POOL. FOR PREVIOUSLY CATALOGUED FILES, THIS FUNCTION IS LEGAL ONLY IF THE FILE IS ASSIGNED WITH EXCLUSIVE USE.

RELEASE — SAME AS READ AND RELEASE EXCEPT NO READING IS PERFORMED. THIS FUNCTION ALSO REQUIRES EXCLUSIVE USE FOR CATALOGUED FILES.

READ AND LOCK — PERFORM THE READ OPERATION AS

SPECIFIED ABOVE AND ALSO IMPOSE A LOGICAL LOCK TO BE PLACED ON THE AREA READ WHICH PREVENTS ACCESS TO THE PART OF THE FILE DEFINED BY THE ACCESS WORD AND RELATIVE STARTING ADDRESS BY OTHER RUNS UNTIL SUCH TIME AS THE LOCKING PROGRAM UNLOCKS THE AREA. REMOVAL OF THIS EXCLUSIVE USE OF A BLOCK IS ACCOMPLISHED BY WRITING INTO ANY PART OF THE BLOCK, ISSUING AN UNLOCK REQUEST AS DEFINED BELOW, OR BY TERMINATING THE PROGRAM.

UNLOCK

REMOVE ANY LOGICAL LOCKS IMPOSED ON OTHER ACTIVITIES BY READ AND LOCK REQUESTS SUBMITTED BY THIS ACTIVITY FOR THE AREA OF THE FILE SPECIFIED BY THE ADDRESS AND LENGTH OF THE PACKET FOR THIS REQUEST. LOCKS ARE MAINTAINED BY BLOCK AND UNLOCKING ANY PART OF A BLOCK UNLOCKS THE ENTIRE BLOCK. ALSO ONE UNLOCK REQUEST CAN UNLOCK SEVERAL BLOCKS.

WRITE

STARTING AT THE RELATIVE SECTOR ADDRESS SPECIFIED IN WORD 6 OF THE PACKET, TRANSFER THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5 FROM THE CORE AREA STARTING AT THE ADDRESS IN H2 OF WORD 5 TO FASTRAND. IF THE COUNT IS NOT A MULTIPLE OF 28, WRITE ZEROES INTO THE REMAINDER OF THE LAST SECTOR (ZERO PADDING IS NOT SIMULATED ON DRUM AND HENCE THE PARTIAL SECTOR IS NOT CHANGED). IF THE AREA BEING WRITTEN INTO IS NOT CURRENTLY ASSIGNED, EXPANSION OF THE FILE IS AUTOMATIC UP TO THE MAXIMUM FROM THE ASG STATEMENT. THE WRITE OPERATION ALSO REMOVES ANY LOCKS ON THE AREA WRITTEN IN THE SAME MANNER AS THE UNLOCK OPERATION DESCRIBED ABOVE.

GATHER WRITE

SAME AS WRITE EXCEPT WORD 5 SPECIFIES A STRING OF ACCESS WORDS EACH SPECIFYING A WORD COUNT AND A CORE AREA.

ACQUIRE FASTRAND

STARTING AT THE RELATIVE SECTOR ADDRESS SPECIFIED IN WORD 6 OF THE PACKET, THE FILE IS EXPANDED BY THE NUMBER OF GRANULES REQUIRED TO HOLD THE NUMBER OF WORDS SPECIFIED IN H1 OF WORD 5. THIS ALLOWS EXPANSION OF A FILE WITHOUT WRITING INTO IT. EXPANSION OF THE FILE IS AUTO-

MATIC UP TO THE MAXIMUM FROM THE ASG STATEMENT.

TRACK SEARCH ALL WORDS

STARTING AT THE RELATIVE SECTOR ADDRESS IN WORD 6 OF THE PACKET COMPARE EACH WORD ON FASTRAND WITH THE IDENTIFIER IN WORD 7 OF THE PACKET UNTIL EITHER AN EQUAL COMPARE IS MADE OR THE END OF THE TRACK IS ENCOUNTERED (SECTOR ADDRESS IS THE NEXT MULTIPLE OF 100 OCTAL). IF A COMPARE EQUAL IS FOUND, STORE THE RELATIVE SECTOR ADDRESS OF THE SECTOR IN WHICH THE FIND IS MADE IN WORD 8 OF THE PACKET AND READ AS MANY WORDS AS IS SPECIFIED IN H1 OF WORD 5 (OR TO THE END OF THE ASSIGNMENT WHICHEVER IS SMALLER) STARTING WITH THE BEGINNING OF THE SECTOR IN WHICH THE FIND WAS MADE. IF NO COMPARE EQUAL IS MADE BEFORE END OF TRACK, RETURN A NO FIND STATUS CODE.

TRACK SEARCH FIRST WORD

SAME AS TRACK SEARCH ALL WORDS EXCEPT A COMPARISON IS MADE ONLY ON THE FIRST WORD OF EACH SECTOR.

POSITION SEARCH ALL WORDS

SAME AS TRACK SEARCH ALL WORDS EXCEPT COMPARISONS ARE MADE UNTIL A SECTOR ADDRESS WHICH IS A MULTIPLE OF 10000 OCTAL IS REACHED.

POSITION SEARCH FIRST WORD

SAME AS POSITION SEARCH ALL WORDS EXCEPT COMPARISONS ARE MADE ONLY ON THE FIRST WORD OF EACH SECTOR.

SINCE THE EXCLUSIVE USE OF FILES BY BLOCK (AS DEFINED BY THE ADDRESS AND ACCESS WORD) INVOLVES AN INTERACTION BETWEEN ACTIVITY, IT BEHOOVES THE USERS TO INSURE THAT PROPER ORDER IS MAINTAINED IN SUBMITTING REQUESTS TO PREVENT TWO ACTIVITIES FROM LOCKING AGAINST EACH OTHER. AS AN AID IN DETECTING THIS INTERLOCK CONDITION, THE I/O CONTROL CHECKS THE LENGTH OF TIME THAT AN ACTIVITY LEAVES A LOCK ON AN ITEM. IF AN ITEM IS LOCKED BY ANY ONE ACTIVITY FOR OVER 12 MINUTES, AT THE TIME OF THE UNLOCK SEQUENCE (EITHER A WRITE OR UNLOCK FUNCTION) A STATUS CODE OF 10 IS RETURNED TO THE PACKET INDICATING THAT EXCLUSIVE USE HAD TIMED OUT AND HENCE BEEN REMOVED. REMOVING EXCLUSIVE USE BY THIS MEANS ALLOWS THE LOCKED ACTIVITIES TO PROGRESS IN THE NORMAL MANNER AND THE LOCKING ACTIVITY NO LONGER INTERFERES. IF THE UNLOCK OPERATION IS THE RESULT OF A WRITE REQUEST, THE WRITE FUNCTION IS NOT PERFORMED IF THE 10 STATUS CODE IS RETURNED. THE 10 STATUS CODE IS ALSO OBSERVED FOR ALL LOCKED AREAS WHICH MUST BE UNLOCKED IF A PACKET FORMAT ERROR IS DETECTED ON A SUBSEQUENT REQUEST WHEN TAKEN OFF OF THE CHANNEL LIST. THIS RESULTS FROM A CHANGE IN THE PACKET BY THE WORKER PROGRAM WHILE THE REQUEST IS LISTED AND AFTER ANY LOCK HAS BEEN IMPOSED FOR THE REQUEST IN ERROR.

DURING NORMAL OPERATION THE FASTRAND HANDLER WILL PREPOSITION THE VARIOUS UNITS TO KEEP ACCESS TIME TO A MINIMUM. FOR THIS REASON, THE POSITION FUNCTION IS NOT NEEDED IN THE USER'S REPERTOIRE.

THE POSITION SEARCHES ARE LEGAL ONLY IF THE GRANULARITY IS POSITION. THE TRACK SEARCHES ARE AVAILABLE FOR BOTH GRANULARITIES.

10.4.4. RECOVERY

RECOVERY SEQUENCES FOR THE FASTRAND HANDLER ARE KEPT ON MAGNETIC DRUM. ERRORS SIGNALED BY EXTERNAL INTERRUPTS WILL TRIGGER THE APPROPRIATE ATTEMPTS. THIS INCLUDES DATA RECONSTRUCTION IN CASE OF A PHASE SHIFT ERROR. IF RECOVERY FAILS, ONE OF THE FOLLOWING MESSAGES IS DISPLAYED ON THE CONSOLE. THE ADDRESS DISPLAYED IS THE ADDRESS RETURNED WITH THE STATUS CODE IN THE STATUS WORD ASSOCIATED WITH THE EXTERNAL INTERRUPT. THIS SECTOR ADDRESS MAY DIFFER FROM THE SECTOR IN ERROR BY UP TO TWO SECTORS BECAUSE OF HARDWARE BUFFERING. THE PHASE CHECK ERROR IS DISPLAYED ONLY IF THE DATA RECONSTRUCTION ROUTINE COULD NOT DETERMINE THE CORRECT FORMAT OF THE SECTOR: ONLY IF EITHER A BIT IS MISSED ON READING RESULTING IN AN IMPROPER SHIFT SENTINEL OR IF THE DISTANCE FROM POINT OF ERROR TO POINT OF DETECTION IS MORE THAN 11 BITS.

MESSAGE	RESPONSES	COMMENTS
C/A PHASE-06 FUNC	A,G	SINGLE PHASE CHECK ERROR IN A SECTOR NOT RECONSTRUCTABLE.
C/A NONREC-07 FUNC	A,G	MORE THAN ONE PHASE ERROR IN THE ADDRESSED SECTOR.
C/A UNDADD-14 FUNC	A,G	THE SPECIFIED ADDRESS COULD NOT BE LOCATED WITHIN 150 MS. IF THE ADDRESS IS WITHIN THE RANGE OF THE SUBSYSTEM, THEN A PROBLEM EXISTS IN THE SUBSYSTEM; OTHERWISE THE POINT OF ERROR IS HARDWARE OR SOFTWARE EXTERNAL TO THE FASTRAND SUBSYSTEM.
C/A SECLEN-34 FUNC	A,G	CHARACTER COUNT AND SECTOR MARK SIGNAL NOT COMPATIBLE.
C/A UNDFNC-50 FUNC	A,G	THE FUNC FIELD (NUMERIC FOR THIS CODE) CONTAINS A VALUE NOT DEFINE FOR FASTRAND. IF THE FUNCTION IS IN THE REPERTOIRE THE SUBSYSTEM IS AT FAULT;OTHERWISE THE ERROR IS EXTERNAL TO THE FASTRAND SUBSYSTEM.
C/A INTRLK-54 FUNC	A,G	NON-EXISTENT OR INOPERABLE ADDRESS REFERENCED. WRITE LOCKOUT SWITCH INHIBITING REFERENCE, POWER DOWN, ETC.
C/A WRITER-60 FUNC	A,G	LOSS OF ON-TRACK SIGNAL.
C/U UNEXP-XX FUNC	A,G	THE INTERRUPT STATUS CODE XX IS NOT EXPECTED FOR THE GIVEN FUNCTION.

10.5. ARBITRARY DEVICE HANDLER

10.5.1. GENERAL

THE ARBITRARY DEVICE HANDLER IS PROVIDED TO HANDLE:

1. SUPPORT OF SPECIAL I/O DEVICES

2. SUPPORT OF STANDARD DEVICES IN AN UNUSUAL MANNER
3. SUPPORT OF MAINTENANCE ROUTINES

THE NEED FOR FILE PRIVACY AND PROTECTION OF FILES BY THE EXE, DICTATES THAT ON MAGNETIC DRUM SUBSYSTEMS ONLY THE EXEC HANDLERS AND THE MAINTENANCE ROUTINES CAN BE ALLOWED DIRECT REFERENCE TO I/O INSTRUCTIONS.

10.5.2. OPERATION

THE ARBITRARY DEVICE HANDLER IS ENTERED BY A LINKAGE TO THE EXEC OF THE FORM:

```
LA    A0, PKT ADDR
ER    IOARB$ OR IOAXI$
```

WHERE PKT ADDR SPECIFIES THE LOCATION OF A FILE CONTROL TABLE OUTLINING THE OPERATION TO BE PERFORMED. THE ENTRANCE TAG IOARB$ CAUSES IMMEDIATE RETURN OF CONTROL. THE IOAXI$ ENTRANCE CAUSES THE REFERENCING ACTIVITY TO EXIT WITH CONTROL COMING BACK TO THE PROGRAM AT THE APPROPRIATE INTERRUPT ACTIVITY SPECIFIED IN THE REQUEST PACKET.

FOR IOARB$ CONTROL IS RETURNED IN LINE AS SOON AS THE REQUEST IS EITHER LISTED OR THE OPERATIONS HAVE BEEN INITIATED. PERFORMANCE OF THE OPERATIONS MAY OR MAY NOT BE COMPLETE AT THAT TIME.

THE FORMAT OF THE ARBITRARY DEVICE HANDLER FILE CONTROL TABLE IS AS FOLLOWS:

```
     35        29        23        17        11        05      00
   --------------------------------------------------------------
01 :                  INTERNAL FILE NAME (WORD 1)               :
   --------------------------------------------------------------
02 :                  INTERNAL FILE NAME (WORD 2)               :
   --------------------------------------------------------------
03 :USED BY EXEC      :ACT ID    : EI INTERRUPT ACTIVITY ADDRESS:
   --------------------------------------------------------------
04 :                  :ACT ID    :MONITOR INTERRUPT ACTIVITY ADD:
   --------------------------------------------------------------
05 : STATUS  :TIME OUT:TIME IND :  FUNCTION STRING              :
   --------------------------------------------------------------
06 :              INITIAL ACCESS WORD 1                         :
   --------------------------------------------------------------
07 : FINAL WORD COUNT 1          : REL TIME 1 (200 MICROSEC INCR:
   --------------------------------------------------------------
   :                                                            :
   :                                                            :
   :                                                            :
   --------------------------------------------------------------
   :              INITIAL ACCESS WORD N                         :
   --------------------------------------------------------------
   : FINAL WORD COUNT N          : REL TIME N                   :
   --------------------------------------------------------------
```

WORD 1 AND WORD 2 THE INTERNAL FILE NAME BY WHICH THE FILE IS REFERENCED. THIS IS EITHER THE SAME NAME AS THE EXTERNAL NAME SPECIFIED IN AN ASG STATEMENT OR IS EQUATED TO AN EXTERNAL NAME BY A USE STATEMENT.

WORD 3 T1-USED BY THE SYSTEM.
S3-NUMERIC IDENTITY (1-35) GIVEN TO THE EXTERNAL INTERRUPT ACTIVITY. THIS FIELD CAN BE LEFT AS ZERO AS FOR THE TAPE OR DRUM HANDLER PACKET IF NO SYNCHRONIZATION IS TO BE DONE WITH THIS ACTIVITY.
H2-ADDRESS AT WHICH CONTROL IS TO BE GIVEN UPON OCCURRENCE OF AN EXTERNAL INTERRUPT.

WORD 4 THIS WORD IS SIMILAR TO WORD 3 EXCEPT FOR A MONITOR INTERRUPT CAUSING INITIATION RATHER THAN AN EXTERNAL INTERRUPT

WORD 5 S1-STATUS CODE INDICATING THE DISPOSITION OF THE REQUEST.
S2-THE NUMBER OF SIX SECOND INTERVALS THE CHANNEL SHOULD BE TIMED BEFORE THE LACK OF A MONITOR OR EXTERNAL INTERRUPT IS TO BE CONSIDERED AN ERROR. THE VALUE 1 CORRESPONDS TO 6 SECONDS, 2 TO 12 SECONDS, ETC.
S3-INDICATES THE DISPOSITION OF A TIME OUT CONDITION. IF THIS FIELD IS NOT ZERO AND AN OPERATION IS LEFT OUTSTANDING ON A CHANNEL FOR A TIME IN EXCESS OF THE VALUE OF S2 OF THIS WORD, A UNIQUE STATUS CODE IS RETURNED TO THE PACKET. IF THE FIELD IS ZERO A TIME OUT MESSAGE IS DISPLAYED ON THE CONSOLE AND THE RESPONSE IS RETURNED.
H2-A STRING OF OCTAL DIGITS DEFINING THE OPERATIONS (SEE BELOW)

WORD6,8,...,N-1 THE INITIAL ACCESS WORDS TO BE USED TO CONTROL THE CHANNEL.

WORD 7,9....,N UPPER HALF=FINAL WORD COUNT AS CONTAINED IN ACCESS CONTROL REGISTER
LOWER HALF=RELATIVE TIME BETWEEN EXECUTION OF THE CORRESPONDING OPERATION IN THE

STRING AND THE EXECUTION OF THE NEXT OPERATION OR THE OCCURENCE OF A INTERRUPT. THE TIME IS GIVEN IN 200 MICRO SECONDS INCREMENTS.

THE INITIATION STRING IN H2 OF WORD 5 CONSISTS OF A GROUP OF 3-BIT BYTES INTERPRETED FROM LEFT TO RIGHT (BITS 17-15 IS THE FIRST BYTE). THE ASSIGNED CODES ARE:

0	END OF STRING
1	INITIATE FUNCTION MODE WITHOUT MONITOR (LFC)
2	INITIATE FUNCTION MODE WITH MONITOR (LFCM)
3	INITIATE OUTPUT MODE WITHOUT MONITOR (LOC)
4	INITIATE OUTPUT MODE WITH MONITOR (LOCM)
5	INITIATE INPUT MODE WITHOUT MONITOR (LIC)
6	INITIATE INPUT MODE WITH MONITOR (LICM)

STARTING AT THE LEFT OF THE STRING, THE OPERATIONS ARE CARRIED OUT AS DIRECTED. AS THE INITIATION STRING IS INTERPRETED, SUCCEEDING PAIRS OF ACCESS WORDS ARE REFERENCED. THE FINAL ACCESS WORD OF THE PRECEDING OPERATION IS UPDATED AND THE INITIAL ACCESS WORD FOR THE CURRENT OPERATION IS LOADED. AT MOST, 6 MODES CAN BE SPECIFIED IN THE INITIATION STRING. AS A PRACTICAL LIMIT THE COMBINED LENGTH OF ALL EXTERNAL FUNCTION BUFFERS IS SET AT 9, EXCEEDING THIS COUNT IS CONSIDERED TO BE A PROGRAM LOGIC ERROR AND CAUSES REFERENCE TO THE ERROR MODE RETURN POINT WITH, A STATUS CODE OF 21. AS AN EXAMPLE OF STRING INTERPRETATION, IF AN INPUT OPERATION IS TO BE PERFORMED WITH TERMINATION BY AN EXTERNAL INTERRUPT, THE INITIATION STRING COULD BE 210000 OCTAL WITH TWO SETS OF ACCESS WORD. THE FIRST OPERATION BY THE ARBITRARY DEVICE HANDLER WOULD BE TO LOAD INPUT CHANNEL ASSIGNED TO THE FILE NAME SPECIFIED IN THE PACKET USING THE ACCESS WORD IN WORD 6. THIS WOULD BE FOLLOWED BY A LOAD FUNCTION CHANNEL USING THE ACCESS WORD IN WORD 8 TO LOCATE THE FUNCTION WORD. UPON OCCURRENCE OF AN EXTERNAL INTERRUPT, THE FINAL ACCESS WORD COUNT AND THE RELATIVE WORD 7 (UPPER AND LOWER HALF RESP.) AND THE FINAL VALUES FOR LFC IN WORD 9

THE USER CAN SPECIFY THE ABOVE LISTED INSTRUCTIONS IN ANY DESIRED ORDER TO PERFORM A PARTICULAR I/O OPERATION. WHENEVER A MONITORED INSTRUCTION IS ENCOUNTERED, THE ARBITRARY DEVICE HANDLER HALTS FURTHER INTERPRETATION OF THE STRING UNTIL THE PARTICULAR MONITOR OCCURS. THE WORKER MUST MAKE CERTAIN THAT THE PROPER INSTRUCTIONS ARE MONITORED TO INSURE THAT THE RESPECTIVE ACCESS WORDS DO NOT GET OVERLAYED I.E. IF TWO SUCCESSIVE OPERATIONS ARE INITIATE OUTPUT TRANSFER, THE FIRST ONE SHOULD BE WITH MONITOR UNLESS THE TIME BETWEEN I/O INSTRUCTION EXECUTIONS ALLOWS FOR TRANSFER OF ALL WORDS OF THE FIRST OUTPUT BUFFER. TO DETERMINE WHETHER OR NOT AN ACCESS WORD HAS SUFFICIENT TIME TO COUNT DOWN BETWEEN INITIATION OF OPERATIONS BY THE ARBITRARY DEVICE HANDLER AND HENCE POSSIBLY ALLOW OPERATING AT TIMES WITHOUT A MONITOR, THE MINIMUM TIME BETWEEN EXECUTION OF THE I/O INSTRUCTIONS BY THE ARBITRARY DEVICE HANDLER IS AT LEAST 10 MICROSECONDS (THIS VARIES UPWARD DEPENDENT UPON OPERATION, OVERLAPPING DATA TRANSFERS, ETC). FOR SUCH SEQUENCES AS A FUNCTION TRANSFER OF A SINGLE WORD EF BUFFER FOLLOWED BY AN OUTPUT TRANSFER THIS IS SUFFICIENT TIME FOR THE FUNCTION TRANSFER TO BE COMPLETED BEFORE OUTPUT TRANSFER IS INITIATED WITHOUT THE NECESSITY OF MONITORING THE FUNCTION TRANSFER.

THE APPEARANCE OF MONITORED MODES DOES NOT NECESSARILY INDICATE THE NEED FOR A MONITOR COMPLETION ACTIVITY (SPECIFIED IN WORD 4) AS THE ARBITRARY DEVICE HANDLER INTERPRETS INTERMEDIATE MONITOR MODES. A MONITOR ACTIVITY IS REQUIRED IF EITHER 1) THE LAST MODE IN A STRING IS WITH MONITOR OR 2) THE LAST MODE IS NOT MONITORED AND NO EXTERNAL INTERRUPT IS EXPECTED TO SIGNAL CONCLUSION OF THE MODE ESTABLISHED AS A RESULT OF THE FINAL MODE. IF ANY MONITORED MODES PRECEDE THE FINAL MODE, WHETHER OR NOT A WAIT FOR EXTERNAL INTERRUPT SHOULD BE DONE AFTER THE FINAL I/O INSTRUCTION IS EXECUTED IS DETERMINED BY A NONZERO VALUE IN H2 OF WORD 4. FOR EXAMPLE, AN INPUT DRUM OPERATION IS NORMALLY TERMINATED WITHOUT INTERRUPT HENCE THE SEQUENCE LFC, LICM, LFC IS USED AND A MONITOR INTERRUPT ACTIVITY IS SPECIFIED AND EXECUTED WITHOUT WAITING AFTER SENDING OUT THE SECOND FUNCTION FOLLOWING THE INPUT MONITOR INTERRUPT; WHEREAS AN OUTPUT DRUM OPERATION IS NORMALLY TERMINATED WITH INTERRUPT, HENCE THE SEQUENCE LFC, LOCM, LFC MAY BE USED WITHOUT A MONITOR INTERRUPT ACTIVITY IN WHICH CASE AN EXTERNAL INTERRUPT IS WAITED FOR AFTER SENDING OUT THE SECOND FUNCTION.

REGARDLESS OF THE MANNER IN WHICH THE ARBITRARY DEVICE HANDLER GIVES CONTROL TO THE INTERRUPT ACTIVITY, IN ALL CASES, THE INPUT AND OUTPUT ACTIVE STATES ARE CLEARED ON THE PARTICULAR CHANNEL BY EXECUTION OF THE DISCONNECT INPUT IN CHANNEL (DIC) AND DISCONNECT OUTPUT IN CHANNEL (DOC) BEFORE CONTROL IS GIVEN TO THE INTERRUPT ACTIVITY.

WHEN A FUNCTION MODE IS CALLED FOR, THE HANDLER WILL INSERT THE PROPER UNIT DESIGNATOR OR ADD THE PROPER BASE ADDRESS TO THE RELATIVE ADDRESS OF THE FUNCTION WORD. ALSO, AT THAT TIME, IF THE CHANNEL CONTAINS EQUIPMENT SHARED BY OTHER ASSIGNMENTS IT MAY BE NECESSARY TO PERFORM CERTAIN ERROR CHECKING TO PREVENT LEAVING THE CHANNEL IN AN INDETERMINATE STATE AND TO PREVENT INTRUSION UPON OTHER ASSIGNMENT PRIVACY. NONSTANDARD SPECIAL I/O DEVICES ARE ASSIGNED BY CHANNEL AND THE ARBITRARY DEVICE HANDLER MAKES NO MODIFICATIONS TO THE FUNCTION WORDS FOR THESE DEVICES.

THE FUNCTION BUFFER FOR MAGNETIC TAPE OR MASS STORAGE CHANNELS IS LIMITED TO A WORD COUNT OF ONE WORD EXCEPT FOR SEARCH FUNCTIONS IN WHICH CASE A SECOND WORD, THE IDENTIFIER, AND FOR IIIA TAPES A THIRD WORD, THE MASK, ARE ALLOWED. FOR OTHER THAN THESE CASES IN A MULTIPLE WORD EF BUFFER, EACH WORD IS MODIFIED BY THE UNIT DESIGNATION AND SUBJECTED TO THE PARTICULAR TESTS BASED ON EQUIPMENT TYPE.

WORD 3 AND WORD 4 OF THE PACKET MAY BE USED TO SPECIFY INTERRUPT ACTIVITIES ONE OF WHICH WILL BE EXECUTEC WHEN THE CORRESPONDING INTERRUPT OCCURS. WORD 3 SPECIFIES THE ACTIVITY TO BE EXECUTED IN CASE OF AN EXTERNAL INTERRUPT. THE LOWER HALF OF THE WORD GIVES THE ACTIVITY STARTING ADDRESS AND S3 IS SET TO THE ACTIVITY IDENTITY IF SYNCHRONIZATION IS NECESSARY, THE CR SAVE AND PRIORITY ARE ASSUMED B11 THROUGH A5 AND R1 THROUGH R3 AND TOP PRIORITY RESPECTIVELY. AN EI ACTIVITY MUST ALWAYS BE GIVEN WHETHER A MONITOR INTERRUPT IS TO BE USED OR NOT. THE MONITOR ACTIVITY IS DEFINED IN THE FOURTH WORD IN THE SAME FORMAT AS THE EI ACTIVITY. IF BOTH A MONITOR AND AN EXTERNAL INTERRUPT OCCUR, THE EXTERNAL INTERRUPT ACTIVITY IS GIVEN CONTROL AND OCCURRENCE OF THE MONITOR INTERRUPT CAN BE DETERMINED BY EXAMINING THE ACCESS WORD. WHEN CONTROL IS GIVEN TO THE INTERRUPT ACTIVITY. A0 IS LOADED WITH THE PACKET ADDRESS AND FOR THE EI ACTIVITY A1 CONTAINS THE EXTERNAL INTERRUPT STATUS WORD.

UPON COMPLETION OF AN I/O OPERATION BY THE ARBITRARY DEVICE HANDLER, A STATUS CODE IS STORED IN S1 OF WORD 5 OF THE REQUEST PACKET DENOTING THE CONDITIONS OF THE COMPLETION. ALL CODES FROM 20 TO 37 CAUSE THE ACTIVITY TO BE TERMINATED AND HENCE ARE NOT OBSERVED BY THE EXECUTING PROGRAM UNLESS AN ERROR TERMINATION ACTIVITY IS CALLED FOR (PREVIOUS REFERENCE TO IERR$). IF ENTRANCE TO THE ARBITRARY DEVICE HANDLER IS VIA IOARB$, THE REFERENCING ACTIVITY RE-ENTRY IS SET TO THE TERMINATION POINT AND NO INTERRUPT ACTIVITIES SPECIFIED IN THE REQUEST PACKET ARE INITIATED. FOR ENTRY VIA IOAXI$, THE REFERENCING ACTIVITY HAS TERMINATED AND HENCE THE USER CAN CONSIDER THE RE-ENTRY TO THE ERROR TERMINATION POINT AS BEING MADE BY THE INTERRUPT ACTIVITY OF THE PACKET EXCEPT FOR THE SWITCHING PRIORITY BEING THAT OF THE PARENT RATHER THAN THE HIGH LEVEL NORMALLY GIVEN TO INTERRUPT ACTIVITIES. THE STATUS CODES ARE AS FOLLOWS:

OCTAL
CODE DEFINITION

00 REQUEST HAS BEEN COMPLETED AND AN INTERRUPT OCCURRED. NORMAL COMPLETION IS TO.BE DETERMINED BY THE PROGRAM.

01 THE SPECIFIED TIME INTERVAL HAS EXPIRED WITHOUT AN INTERRUPT OCCURRING. THE OPERATOR RESPONSE TO THE TIME OUT MESSAGE IS IN THE A1 REGISTER IN FIELDATA (A,B,D,E, OR G) IF THE TIME IND FIELD (S3 OF WORD 5) IS ZERO; OTHERWISE THE A1 REGISTER IS SET TO ZERO AND NO TIME OUT MESSAGE IS DISPLAYED.

02 THE SPECIFIED UNIT WAS PREVIOUSLY DECLARED DOWN EITHER IN RESPONSE TO A TIME OUT (STATUS CODE 01 ABOVE) OR AS A RESULT OF AN UNSOLICITED KEYIN.

20 A VALUE OF ZERO SPECIFIED FOR TIME OUT.

21 EF BUFFER OF OTHER THAN ONE WORD ON A MAGNETIC TAPE OR MASS STORAGE CHANNEL AND NOT A SEARCH OPERATION OR MORE THAN A TOTAL OF 9 EF WORDS IN ALL EF BUFFERS COMBINED.

22 THE ARBITRARY DEVICE HANDLER CAN NOT BE USED FOR THE DEFINED DEVICE OR NO ASSIGNMENT HAS BEEN MADE FOR THE DEVICE.

23 THE ADDRESS SPECIFIED IN THE A0 REGISTER IS NOT WITHIN PROGRAM LIMITS OR DEFINES A PACKET SPLIT BETWEEN THE INSTRUCTION AND DATA BANKS OF THE PROGRAM.

24 THE BITS IN THE INITIATION STRING CONTAIN AN ILLEGAL VALUE.

25 AN I/O ACCESS WORD REFERS TO A BUFFER WHICH IS WHOLLY OR IN PART OUTSIDE OF THE PROGRAM AREA OR SPLIT BETWEEN THE INSTRUCTION AND DATA BANK OF THE PROGRAM. FOR MAINTENANCE ROUTINES THIS APPLIES TO A DRUM ADDRESS PLUS ACCESS WORD COUNT BEING OUTSIDE OF THE ASSIGNMENT.

26 ILLEGAL STARTING ADDRESS GIVEN FOR AN INTERRUPT ACTIVITY. THIS ALSO COVERS THE CASE OF THE LAST MODE BEING WITH MONITOR BUT NO MONITOR INTERRUPT ACTIVITY SPECIFIED.

27 A REQUEST WAS MADE TO THE ARBITRARY DEVICE HANDLER WITH THE STATUS WORD (WORD 5) INITIALLY NEGATIVE INDICATING A POSSIBLE PROGRAM LOOP.

40 THE REQUEST IS EITHER LISTED OR IN THE PROCESS OF BEING SERVICED.

THE STATUS CODE OF 40 MAY STILL EXIST WHEN CONTROL IS RETURNED AT THE LINE FOLLOWING THE REFERENCE TO IOARB$. HENCE THE PACKET CAN BE TESTED AND REFERENCE MADE TO WAIT$ OR WANY$ IN THE SAME MANNER AS OTHER I/O PACKETS. THE 40 CODE IS REPLACED UPON INTERRUPT OR TIME OUT. THE TEST INSTRUCTION SHOULD REFERENCE THE PACKET ADDRESS PLUS 4.

THE MAXIMUM VALUE TO BE USED FOR THE TIME OUT FOR THE ARBITRARY DEVICE HANDLER MAY BE DICTATED AT SYSTEM GENERATION TIME BY CHANNEL. THIS CAN PROTECT REAL-TIME ACCESS FROM EXTENDED OPERATIONS SUCH AN LENGTHY SEARCHES ON MAGNETIC TAPE. IF A CHANNEL TIMES OUT, CONTROL WILL BE GIVEN TO THE EXTERNAL INTERRUPT ACTIVITY WITH A STATUS CODE OF 01 IN THE PACKET. IF THE TIME INDICATOR (S3 OF WORD 5) IS SET TO ZERO, CONTROL IS NOT SWITCHED TO THIS ROUTINE UNTIL THE OPERATOR RESPONDS TO THE MESSAGE

```
    C/U IOARB TIMEOUT ABDEG
```

THE SINGLE CHARACTER RESPONSE IS IN THE A1 REGISTER.

THE EXEC DOES NOT PROVIDE FOR LOGGING OF REFERENCES MADE TO DEVICES VIA THE ARBITRARY DEVICE HANDLER. THE USER HAS THE MEANS OF ENTERING INFORMATION IN THE LOG FOR THIS MODE OF OPERATION.

REQUESTS FOR THE ARBITRARY DEVICE HANDLER ARE TAKEN FROM A CHANNEL QUEUE WHICH MAY OR MAY NOT HAVE INTERMIXED REQUESTS FOR THE OTHER DEVICE HANDLERS. CONTROL OF THE CHANNEL IS GIVEN TO THE ARBITRARY DEVICE HANDLER FOR THE LENGTH OF TIME NECESSARY TO ISSUE ALL MODES IN THE STRING AND RECEIVE REQUESTED INTERRUPTS ON THE CHANNEL. UPON OCCURRENCE OF ALL INTERRUPTS REQUESTED, THE PROPER INTERRUPT ACTIVITY IS REGISTERED AND THE CHANNEL QUEUE IS EXAMINED FOR THE NEXT REQUEST. HENCE IF MORE THAN ONE ASSIGNMENT IS MADE ON A CHANNEL, NO KNOWLEDGE OF THE STATE OF THE CHANNEL CAN BE CARRIED OVER FROM ONE REQUEST TO THE ARBITRARY DEVICE HANDLER TO THE NEXT.

10.6. COMMUNICATIONS MULTIPLEXOR HANDLER

10.6.1. GENERAL

THE COMMUNICATIONS MULTIPLEXOR HANDLER IS DESIGNED TO PRESENT A COMMON FOCAL POINT BETWEEN THE MULTITUDE OF AVAILABLE REMOTE TERMINAL DEVICES AND THE PROGRAMS TO BE EXECUTED ON THE UNIVAC 1108. THE DIVERSITY OF AVAILABLE HARDWARE DICTATES A GENERAL ROUTINE UPON WHICH THE VARIANCES OF EACH APPLICATION CAN BE BUILT.

THE WORKER PROGRAM TO WHICH COMMUNICATIONS DEVICES ARE ASSIGNED MUST BE OPERATED AS A REAL-TIME PROGRAM. THIS IS A RESULT OF THE HIGH PRIORITY WHICH MUST BE GIVEN TO COMMUNICATIONS INTERRUPT PROCESSING. BATCH PROGRAMS ARE ABLE TO UTILIZE THE LINE TERMINAL DEVICES THROUGH THE SYSTEM FURNISHED SYMBIONTS RATHER THAN DIRECT ASSIGNMENT. EACH WORKER PROGRAM TO WHICH COMMUNICATION DEVICES ARE ASSIGNED MUST REGISTER AN ERROR ROUTINE WITH THE EXECUTIVE SYSTEM SO THAT THE WORKER PROGRAM MAY BE PROPERLY NOTIFIED CONCERNING OPERATING CONTINGENCIES. THE ERROR ROUTINE IS TO BE REGISTERED VIA ER IERR$ FOR THE 'ERR MODE' ENTRY DESCRIBED UNDER THE SUPERVISOR, INTERRUPT PROCESSING, PROGRAM CONTINGENCY. FAILURE OF THE WORKER PROGRAM TO REGISTER AN ERROR ROUTINE WILL CAUSE THE WORKER PROGRAM TO BE TERMINATED AS DESCRIBED SHOULD AN OPERATING CONTINGENCY OCCUR.

10.6.2. EQUIPMENT

COMMUNICATIONS DEVICES MAY BE ATTACHED TO 1108 CHANNELS THROUGH THREE TYPES OF SUBSYSTEMS:

COMMUNICATIONS TERMINAL, SYNCHRONOUS (CTS)
WORD TERMINAL, SYNCHRONOUS (WTS)
COMMUNICATIONS TERMINAL MODULE CONTROLLER (CTMC)

EACH OF THESE TYPES OF SUBSYSTEMS CAN INTERFACE WITH VARIOUS SPEED DEVICES WITH DIFFERENT BITS PER CHARACTER AND NUMEROUS OTHER FEATURES WHICH PERMIT THESE SUBSYSTEMS TO BE TAILORED FOR THE APPLICATION.

THE CTS AND WTS SUBSYSTEMS OPERATE IN THE INTERNALLY SPECIFIED INDEX (ISI) MODE WITH ONLY ONE REMOTE TERMINAL CONNECTED TO A CPU I/O CHANNEL AT ANY ONE TIME. THE COMMUNICATIONS HANDLER SUPPORTS ONLY SECOND LEVEL OPERATION FOR THE CTS AND WTS SUBSYSTEMS. THE MAIN CONCERN IN THE DESIGN OF THESE DEVICES IS INTERFACING THE UNIVAC 1004 AS A REMOTE CARD PROCESSOR OR IN COMMUNICATING WITH ANOTHER CPU AT ANOTHER SITE. THE BASIC DIFFERENCE BETWEEN THESE TWO DEVICES IS THE ABILITY TO ASSEMBLE CHARACTERS INTO COMPUTER WORD ON INPUT TO THE COMPUTER AND DISASSEMBLE ON OUTPUT BY THE WTS. THE ASSEMBLY PROCESS PACKS SIX 6-BIT CHARACTERS INTO A 36 BIT WORD WITH THE FIRST CHARACTER INTO BITS 35-30 THE NEXT INTO 29-24 ETC. THE WTS ALSO GENERATES THE START OF MESSAGE, END OF MESSAGE AND MESSAGE PARITY CHARACTERS. SUCCESSFUL OPERATION OF THE CTS SUBSYSTEM REQUIRES THAT THE END OF OUTPUT EXTERNAL INTERRUPT BE DISABLED.

THE CTMC LINE TERMINALS OPERATE IN THE EXTERNALLY SPECIFIED INDEX (ESI) MODE I.E. EACH TRANSFER IS ACCOMPANIED BY AN ADDRESS WHICH IDENTIFIES THE EXTERNAL LINE TO OR FROM WHICH THE TRANSFER IS DIRECTED AND EACH ADDRESS HAS A DISTINCT I/O ACCESS WORD ASSOCIATION. THESE SUBSYSTEMS MULTIPLEX UP TO 64 LINE TERMINALS ON A SINGLE I/O CHANNEL. THESE SUBSYSTEMS CAN IN TURN BE MULTIPLEXED THROUGH A SCANNER/SELECTOR. THE COMMUNICATIONS HANDLER SUPPORTS BOTH FIRST AND SECOND LEVEL OPERATIONS FOR CTMC LINE TERMINALS. FOR EACH DATA TRANSFER IN ESI MODE, AN ESI ADDRESS ACCOMPANIES THE DATA TO IDENTIFY THE APPROPRIATE ACCESS CONTROL WORD (ACW) TO BE USED. THE SYSTEM BASE ADDRESS FOR ALL ESI ACW'S IS 002000 FOR MAIN MEMORY IN ORDER FOR THE SYSTEM TO UTILIZE THE MEMORY OVERLAP FEATURE. A GROUP OF 64 CONSECUTIVE ADDRESSES (1 FOR EACH LINE TERMINAL) IS SET ASIDE FOR EACH CTMC DEFINED AT SYSTEM GENERATION TIME. THE BASE ADDRESS FOR EACH INDIVIDUAL CTMC IS IN INCREMENTS OF 64 ARRANGED ACCORDING TO CHANNEL PRIORITY STARTING AT THE SYSTEM BASE ADDRESS. THUS A SYSTEM WITH TWO CTMC'S ON CHANNELS 4 AND 5 WOULD USE 002000 AS THE CTMC BASE VALUE FOR THE CTMC ON CHANNEL 4 AND 002100 WOULD BE USED AS THE CTMC BASE VALUE FOR THE CTMC ON CHANNEL 5.

THE COMMUNICATIONS HANDLER IS WRITTEN ON THE ASSUMPTION THAT THE USER PROGRAM IS WRITTEN TO INTERFACE WITH A PARTICULAR TYPE HARDWARE AND IN THIS RESPECT ASSUMES THAT THE BUFFERS ARE ORGANIZED ACCORDINGLY, SUCH AS PACKING CHARACTERS INTO WORDS FOR WTS, SETTING UP FOR CODE CONVERSION, PROVIDING START, END, AND PARITY CHARACTERS FOR CTS, ETC. THE AMOUNT OF TIME AVAILABLE TO PROCESS BUFFERS DICTATES THAT EXEC ACTION BE KEPT TO A MINIMUM AND HENCE DISALLOWS CHARACTER MANIPULATION. FOR OTHER THAN BUFFER FORMAT AND TIME CONSIDERATIONS, THE USER NEED NOT BE AWARE OF WHICH HARDWARE ARRANGEMENT IS BEING EMPLOYED. THE SYSTEM SYMBIONT ROUTINES AND DEMAND MODE FUNCTIONS SUPPORT DEVICES ON ALL THREE SUBSYSTEM TYPES.

10.6.3. ASSIGNING LT DEVICES

AT SYSTEM GENERATION TIME EACH CHANNEL MUST BE COMPLETELY DEFINED. FOR COMMUNICATIONS DEVICES THIS INCLUDES SPECIFYING THE SUBSYSTEM TYPE AND TYPE OF LINE TERMINAL DEVICE CONNECTED TO EACH REMOTE LINE (BITS PER CHARACTER, SPEED, FIXED OR COMMON CARRIER LINE, ETC). AT THAT TIME VARIOUS DEVICES CONNECTED TO A SINGLE LINE AND PROGRAMMED AS A UNIT (ONE INPUT, ONE OUTPUT AND/OR ONE DIALING UNIT), ARE GIVEN AN LT GROUP IDENTITY. THIS IDENTITY IS USED IN ASSIGNING COMMUNICATIONS DEVICES.

THE ARBITRARY UNIT/CHANNEL ASSIGNMENT FORMAT OF THE ASG CARD IS USED TO ASSIGN THESE DEVICES. THIS IS 'NAME,TYPE' WHERE 'NAME' IS AN EXTERNAL SYMBOL TO BE RELATED TO A INTERNAL SYMBOL BY THE USE COMMAND OR THE INTERNAL SYMBOL ITSELF IN WHICH CASE NO USE IS REQUIRED AND 'TYPE' IS THE LT GROUP IDENTITY DEFINING THE EQUIPMENT. ALL GROUPS WITH THE SAME TYPE DEVICES ARE GIVEN THE SAME GROUP IDENTITY.

A THOROUGH DESCRIPTION OF CHANNEL DEFINITION IS GIVEN IN THE CHAPTER ON SYSTEM SETUP.

10.6.4. THE LT CONTROL TABLE

THE USER CONTROLS THE OPERATIONS ON EACH LT GROUP THROUGH A TABLE CONSTRUCTED IN HIS PROGRAM. EACH LT GROUP IS CONTROLLED BY A SINGLE LT CONTROL TABLE FROM THE INITIALIZE TO THE TERMINATE REQUEST (SEE NEXT SECTION). A NEW TABLE CAN BE USED ON A SUBSEQUENT INITIALIZE AFTER TERMINATE. THE LT TABLE IS SUBDIVIDED INTO THREE PARTS, OUTPUT(WORDS 2-5), INPUT(WORDS 6-9), AND (WORDS 10-11). ANY PARTS CAN BE OMITTED SIMPLY BY NOT CODING, BUT OMISSION OF A PART MUST BE ACCOUNTED FOR BY A BLANK AREA. THE ORGANIZATION OF THE LT TABLE IS AS FOLLOWS:

```
      35        29        23        17        11        05        00
   ----------------------------------------------------------------
00 :                   INTERNAL FILE NAME (WORD 0)                 :
   ----------------------------------------------------------------
01 :                   INTERNAL FILE NAME (WORD 1)                 :
   ----------------------------------------------------------------
02 :OUT STAT :EXEC USE :OUT USAGE: OUT COMPLETE ACTIVITY           :
   ----------------------------------------------------------------
03 : OUT CHAR COUNT OR 0 FOR POOL;OUT BUFFER OR POOL START         :
   ----------------------------------------------------------------
04 : END OF OUT BACKUP QUEUE     :START OF OUT BACKUP QUEUE        :
   ----------------------------------------------------------------
05 : PARTIAL BUFFER CHAR COUNT   : BUFFER TRANSFER TIME            :
   ----------------------------------------------------------------
06 :IN STAT  :EOI ACT  :IN USAGE : IN COMPLETE ACTIVITY            :
   ----------------------------------------------------------------
07 : IN CHAR COUNT OR 0 FOR POOL : IN BUFFER OR POOL START         :
   ----------------------------------------------------------------
08 :END OF IN BACKUP QUEUE       :START OF IN BACKUP QUEUE         :
   ----------------------------------------------------------------
09 : PARTIAL COUNT OR SWAP START : BUFFER TRANSFER TIME            :
   ----------------------------------------------------------------
10 :DIAL STAT:          :DIAL USAG: DIAL COMPLETE ACTIVITY         :
   ----------------------------------------------------------------
11 :                         DIAL ACCESS WORD                      :
   ----------------------------------------------------------------
```

WORD 0 AND 1 INTERNAL FILE NAME-THE IDENTITY USED TO REFER TO THE LT GROUP. THIS NAME IS RELATED TO THE ASG CONTROL FUNCTION BY THE @USE CONTROL STATEMENT IF IT IS NOT THE SAME AS THE EXTERNAL NAME.

WORD 2 S1-OUTPUT STATUS--AN OCTAL CODE DENOTING THE COMPLETION STATUS OF THE LAST BUFFER TRANSFERRED TO THE REMOTE TERMINAL. FOR MULTIPLE BUFFER MODE, THIS CODE IS STORED IN S3 OF WORD 1 OF EACH BUFFER. THE VALUES IN THIS FIELD ARE:

00-THE NUMBER OF CHARACTERS SPECIFIED IN H1 OF WORD 3 OR, FOR BUFFER POOL OPERATION, IN THE UPPER THIRD OF THE FIRST WORD OF THE BUFFER HAS BEEN TRANSFERRED TO THE REMOTE TERMINAL AND, IF SECOND LEVEL SUPPORT, A NORMAL ACKNOWLEDGE WAS RETURNED.

01-OUTPUT WAS TERMINATED DUE TO AN EXTERNAL INTERRUPT. THE NUMBER OF CHARACTERS TRANSFERRED ARE IN WORD 5 OR T1 OF WORD 1 OF BUFFER IF POOL MODE.

05-OUTPUT WAS TERMINATED BEFORE THE SPECIFIED NUMBER OF CHARACTERS WAS TRANSFERRED BY ENCOUNTERING AN END-OF-OUTPUT CHARACTER. TERMINATION IS DETECTED BY A TIME OUT VIA THE HANDLER'S TIMER. THE NUMBER OF CHARACTERS TRANSFERRED IS IN H1 OF WORD 5. OUTPUT HAS BEEN TURNED OFF. THIS MODE OF OPERATION CAN BE USED TO AVOID OUTPUT MONITOR INTERRUPTS ON THE CTMC BY SPECIFYING A FULL BUFFER, SETTING THE EOT BIT BEFORE THE END OF BUFFER AND ALLOWING THE BUFFER TO TIME OUT.

02-THE LINE HAS BEEN DECLARED DOWN BY THE OPERATOR AS A RESULT OF A TIME OUT ON OTHER THAN AN END-OF-OUTPUT CHARACTER OR INABILITY TO ACKNOWLEDGE A TRANSMISSION. H1 OF WORD 5 CONTAINS THE NUMBER OF CHARACTERS TRANSMITTED FROM THIS BUFFER, IF ANY. NO FURTHER ACTION IS TAKEN ON QUEUED BUFFERS.

20-SAME AS 00 EXCEPT THE END OF THE BUFFER QUEUE HAS BEEN REACHED AND OUTPUT HAS BEEN TURNED OFF. A

SUBSEQUENT BUFFER MAY HAVE BEEN ADDED TO THE QUEUE AFTER THE HANDLER MADE THE CHECK IN WHICH CASE IT IS NECESSARY FOR THE WORKER TO RESTART OUTPUT.

25-SAME AS 20 EXCEPT THE OUTPUT BUFFER HAS BEEN TIMED OUT.

S2-RESERVED FOR EXCLUSIVE EXECUTIVE USE TO LINK EACH LT CONTROL TABLE TO ITS CORRESPONDING PROGRAM CONTROL TABLE (PCT) ITEM.

S3-OUTPUT-COMPLETION-ACTIVITY USAGE-DENOTES THE ACTION REQUIRED TO START UP THE OUTPUT COMPLETION ACTIVITY:

00-NO ACTIVITY TO BE INITIATED.

01-GIVE CONTROL TO THE COMPLETION ACTIVITY UPON COMPLETION OF AN OUTPUT BUFFER ONLY IF THE ACTIVITY IS NOT ALREADY REGISTERED ONCE. THE HANDLER WILL DETECT EXITING FROM THIS ACTIVITY AFTER THE CHECK AND WILL RESTART IT.

02-GIVE CONTROL TO THE OUTPUT COMPLETION ACTIVITY ONLY IF THE OUTPUT BACKUP QUEUE IS EXHAUSTED OR IF A NONZERO STATUS IS RETURNED FOR A BUFFER.

H2-OUTPUT COMPLETION ACTIVITY-THIS FIELD CONTAINS THE ADDRESS OF A ROUTINE TO BE GIVEN CONTROL UPON COMPLETION OF AN OUTPUT BUFFER TRANSFER WITHIN THE CONDITIONS OF S3 OF THIS WORD. CONTROL IS GIVEN TO THIS ROUTINE WITH A0 SET TO THE ADDRESS OF THE LT TABLE CONTROLLING THE DEVICE AND A1 SET TO THE ADDRESS OF THE BUFFER JUST TRANSFERRED.

WORD 3 OUTPUT BUFFER CONTROL- THIS WORD DEFINES THE SINGLE OUTPUT BUFFER OR OUTPUT BUFFER POOL. H1 CONTAINS THE NUMBER OF CHARACTERS IF A SINGLE BUFFER MODE OR ZERO IF THE POOL MODE. THE BUFFER ADDRESS OR BUFFER POOL CONTROL WORD ADDRESS IS GIVEN IN H2. FOR CTMC, OUTPUT CHARACTERS WILL BE TRANSMITTED IN ASCENDING ORDER WITHIN A WORD STARTING AT THE LOWEST PORTION OF THE WORD. FOR WTS, OUTPUT CHARACTERS WILL BE TRANSMITTED IN DESCENDING ORDER WITHIN A WORD STARTING AT THE HIGHEST PORTION OF THE WORD. CTS OUTPUT CHARACTERS ARE ONE CHARACTER PER WORD RIGHT JUSTIFIED. BOTH SINGLE MODE AND POOL MODE INDIVIDUAL BUFFER SIZES ARE LIMITED TO 4095 CHARACTERS.

WORD 4 FOR BUFFER POOL OPERATION THIS WORD IS SET TO THE ADDRESS OF THE BUFFER AT THE END(H1) AND START (H2) OF THE QUEUE OF BUFFERS ALREADY FILLED FOR OUTPUT ON THE LINE TERMINAL DEVICE. THE HANDLER UPDATES ONLY THE START FIELD, AND THEN ONLY IF NO COMPLETION ROUTINE IS INVOLVED.

WORD 5 H1-THE NUMBER OF CHARACTERS TRANSFERRED AS OUTPUT IF OUTPUT OF A BUFFER IS COMPLETED BEFORE THE SPECIFIED COUNT IS TRANSFERRED.

H2-THE NUMBER OF BASIC TIME INTERVALS TO BE USED AS THE MAXIMUM TIME BETWEEN BUFFERS. IF A BUFFER DOES NOT TRANSFER IN THIS TIME INTERVAL, A FAULT IS SUSPECTED UNLESS THE PREVIOUS CHARACTER DENOTED END-OF-OUTPUT.

THE MINIMUM PERMISSIBLE VALUE IS THE NUMBER OF CHARACTERS IN THE BUFFER X NUMBER OF BITS/CHARACTER / LINE SPEED(BITS/SECOND). IF A TIME VALUE OF ZERO OR A VALUE LESS THAN THE PERMISSIBLE MINIMUM FOR THE LT GROUP AS DEFINED AT SYSTEM GENERATION TIME IS SPECIFIED, NO TIMING CHECK WILL BE PERFORMED BY THE COMMUNICATIONS HANDLER. IT IS NOT THE INTENT OF THE COMMUNICATIONS HANDLER TO PERFORM EXTENSIVE BUFFER TIMING CHECKS BUT MERELY TO PROVIDE A MEANS OF DETECTING A STALLED OR INACTIVE REMOTE CONDITION. ANY EXTENSIVE TIMING CHECKING WOULD MERELY BE EXCESSIVE OVERHEAD

REDUCING SYSTEM THROUGHPUT WHILE BEING OF NO APPRECIABLE VALUE TO COMMUNICATIONS HANDLER USERS. THE REAL TIME CLOCK SHOULD BE USED IF ANY CRITICAL BUFFER TIMING IS DESIRED. THE BASIC TIME INTERVAL USED BY THE COMMUNICATIONS HANDLER IS 600 MILLISECONDS BECAUSE THE DAYCLOCK INTERRUPTS AT THIS TIME INTERVAL, THE AUTOMATIC CALLING UNIT (ACU) DIAL TIME IS APPROXIMATELY 600 MILLISECONDS PER DIGIT, AND SOME MEDIUM SPEED MODEMS REQUIRE A CERTAIN AMOUNT OF IDLE TIME BETWEEN TRANSMISSIONS WHICH CAN BE MEASURED IN 600 MILLISECOND INCREMENTS.

WORD 6 S1-INPUT STATUS- AN OCTAL CODE DENOTING THE COMPLETION STATUS OF THE LAST BUFFER TRANSFERRED FROM THE REMOTE TERMINAL. FOR MULTIPLE BUFFER MODE, THIS CODE IS STORED IN S3 OF WORD 1 OF EACH BUFFER. THE VALUES IN THIS FIELD ARE:

00-THE NUMBER OF CHARACTERS SPECIFIED IN H1 OF WORD 7 OR, FOR BUFFER POOL OPERATION, THE UPPER THIRD OF THE FIRST WORD OF THE BUFFER HAS BEEN TRANSFERRED FROM THE REMOTE TERMINAL AND, IF SECOND LEVEL SUPPORT, A NORMAL ACKNOWLEDGE WAS RETURNED.

01-INPUT WAS TERMINATED DUE TO AN EXTERNAL INTERRUPT. THE NUMBER OF CHARACTERS TRANSFERRED ARE IN H1 OF WORD 9 OR T1 OF WORD 1 OF BUFFER IF POOL MODE.

05-INPUT WAS TERMINATED BEFORE THE SPECIFIED NUMBER OF CHARACTERS WAS TRANSFERRED BY ENCOUNTERING AN END-OF-INPUT CHARACTER. TERMINATION IS DETECTED BY TIME OUT VIA THE HANDLER'S TIMER. NUMBER OF CHARACTERS TRANSFERRED IS IN H1 OF WORD 9. ACTION IS ACCORDING TO THE EOI ACT FIELD.

02-THE LINE HAS BEEN DECLARED DOWN BY THE OPERATOR AS A RESULT OF A TIME OUT ON OTHER THAN AN END-OF-INPUT CHARACTER. H1 OF WORD 9 OR WORD 1 ,T1 OF BUFFER WILL CONTAIN THE NUMBER OF CHARACTERS TRANSMITTED, IF ANY. NO FURTHER INPUT IS ACCEPTED.

03-INPUT WAS TERMINATED BEFORE SPECIFIED NUMBER OF CHARACTERS HAVE BEEN TRANSFERRED. DUE TO INPUT REQUEST(CMI$) BEFORE PREVIOUS REQUEST WAS COMPLETED BY THE HANDLER. THIS APPLIES TO POOL MODE ONLY, AND WORD 1, T1 WILL CONTAIN THE PARTIAL CHARACTER COUNT, IF ANY. FURTHER ACTION IS ACCORDING TO THE EOI ACT FIELD. THIS FEATURE PROVIDES THE USER THE CAPABILITY OF TERMINATING POOL MODE INPUT IF DESIRED.

04-A PARITY ERROR HAS OCCURRED FOR THE LAST INPUT OPERATION.

20-SAME AS 00 EXCEPT THE END OF THE BUFFER POOL HAS BEEN REACHED AND INPUT HAS BEEN TURNED OFF.

21-SAME AS 01 EXCEPT THE END OF THE BUFFER POOL HAS BEEN REACHED.

23-SAME AS 03 EXCEPT THAT END OF INPUT BUFFER POOL HAS BEEN REACHED.

24-SAME AS 04 EXCEPT THAT THE END OF THE INPUT BUFFER POOL HAS BEEN REACHED.

25-SAME AS 05 EXCEPT THAT END OF INPUT BUFFER POOL HAS BEEN REACHED.

S2-END-OF-INPUT ACTION-DENOTES THE ACTION TO BE TAKEN BY THE HANDLER WHEN AN END OF INPUT IS RECEIVED FOR MULTIPLE BUFFER MODE:

00-TURN INPUT OFF.

01-REINITIATE INPUT WITH THE NEXT BUFFER FROM THE POOL.

S3-INPUT -COMPLETION-ACTIVITY USAGE-DENOTES THE ACTION REQUIRED TO START UP THE INPUT COMPLETION ACTIVITY:

- 00-NO ACTIVITY TO BE INITIATED
- 01-GIVE CONTROL TO THE COMPLETION ACTIVITY UPON COMPLETION OF AN INPUT BUFFER ONLY IF THE ACTIVITY IS NOT ALREADY REGISTERED ONCE. THE HANDLER WILL DETECT EXITING FROM THIS ACTIVITY AFTER THE CHECK AND WILL RESTART IT.

H2-INPUT COMPLETION ACTIVITY-THIS FIELD CONTAINS THE ADDRESS OF A ROUTINE TO BE GIVEN CONTROL UPON COMPLETION OF AN INPUT BUFFER TRANSFER WITHIN THE CONDITIONS OF S3 OF THIS WORD. CONTROL IS GIVEN TO THIS ROUTINE WITH A0 SET TO THE ADDRESS OF THE LT TABLE CONTROLLING THE DEVICE AND A1 SET TO THE ADDRESS OF THE BUFFER JUST TRANSFERRED.

WORD 7 INPUT BUFFER CONTROL-SAME AS WORD 3 EXCEPT FOR INPUT RATHER THAN OUTPUT. FOR CTMC INPUT CHARACTERS WILL BE RECEIVED IN ASCENDING ORDER WITHIN A WORD STARTING AT THE LOWEST PORTION OF THE WORD. FOR WTS INPUT CHARACTERS WILL BE RECEIVED IN DESCENDING ORDER WITHIN A WORD STARTING AT THE HIGHEST PORTION OF THE WORD. CTS INPUT CHARACTERS WILL BE ONE CHARACTER PER WORD RIGHT JUSTIFIED.

WORD 8 INPUT QUEUE CONTROL-SAME FORMAT AS WORD 4 EXCEPT FOR INPUT RATHER THAN OUTPUT. FOR POOL MODE THE WORKER PROGRAM IS EXPECTED TO UPDATE H2 OF WORD 8 WITH THE LINK PORTION OF EACH BUFFER AFTER THAT BUFFER HAS BEEN PROCESSED. THE WORKER PROGRAM CAN DETERMINE THE END OF THE INPUT BACKUP QUEUE BY UPDATING H2 OF WORD 8 UNTIL EITHER A VALUE OF ZERO IS ENCOUNTERED OR IT MATCHES H1 OF THE SAME WORD.

WORD9 SAME AS WORD 5 EXCEPT FOR INPUT RATHER THAN OUTPUT. H1 MUST BE ZERO WHEN POOL MODE INPUT IS TO BE USED UNLESS DUAL POOL MODE IS DESIRED IN WHICH CASE H1 SPECIFIES A POOL CONTROL WORD ADDRESS DIFFERENT FROM THE POOL ADDRESS SPECIFIED IN H2 OF WORD 7. THE PRIMARY USE FOR THE DUAL POOL INPUT MODE IS FOR POLLING OPERATIONS WHEREBY A SMALL INPUT BUFFER CAN BE INITIALLY SET-UP WITH AN IMMEDIATE SWITCH TO A POOL OF LARGER BUFFERS WHEN THE POLL RESPONSE HAS BEEN RECEIVED THEREBY PERMITTING LARGER BUFFER AREAS TO BE USED FOR THE INPUT DATA STREAM INITIATED BY THE POLLING OPERATION.

WORD10 S1-DIAL STATUS-GIVES THE COMPLETION STATUS OF THE LAST DIAL OPERATION:

- 01-SUCCESSFUL
- 02-UNSUCCESSFUL
- 03-LEASED LINE ASSIGNED
- 40-IN PROCESS

S3-DIAL USAGE-THE ACTION TO TAKE UPON COMPLETION OF THE DIAL OPERATION:

- 00-NO DIAL COMPLETION ACTIVITY
- 01-GIVE CONTROL TO THE DIAL COMPLETION ROUTINE UPON INDICATION THAT THE DIAL OPERATION HAS BEEN COMPLETED.

H2-DIAL COMPLETE ACTIVITY-THE ADDRESS TO WHICH CONTROL IS TO GO UPON COMPLETION OF THE DIAL OPERATION.

WORD11 DIAL ACCESS WORD-THE COUNT OF CHARACTERS IN H1 AND THE BUFFER ADDRESS IN H2 AT WHICH THE NUMBER TO BE DIALED IS STORED IN BCD FORMAT. FOR CTMC, DIAL CHARACTERS WILL BE TRANSMITTED IN ASCENDING ORDER WITHIN A WORD STARTING AT THE LOWEST PORTION OF THE WORD.

10.7. COMMUNICATION OPERATIONS

THE COMMUNICATION HANDLER SUPPORTS TWO LEVELS OF OPERATION. THE FIRST LEVEL CONSISTS OF A BUFFER HANDLING MODE IN WHICH THE HANDLER SUPERVISES TRANSMITTING AND RECEIVING MESSAGES ON A BUFFER BY BUFFER BASIS WITH NO ASSUMPTION CONCERNING THE CONTENT OF EACH BUFFER. THE SECOND LEVEL SUPPORT ASSUMES A SYSTEM DEFINED FORMAT ON DEVICES CAPABLE OF ACKNOWLEDGING TRANSMISSION.

ALL OPERATIONS ON A LINE TERMINAL GROUP MUST REFER TO THE GROUP THROUGH A SINGLE LT TABLE (IN CONTRAST TO OTHER INPUT/OUTPUT DEVICES WHICH MAY HAVE ANY NUMBER OF ACTIVE PACKETS).

10.7.1.1. FIRST LEVEL SUPPORT

10.7.2. FIRST LEVEL SUPPORT

THE AVAILABLE OPERATIONS AT FIRST LEVEL AND CORRESPONDING SYSTEM REFERENCES ARE:

INITIALIZE	CMS$
TERMINATE	CMT$
DIAL	CMD$
INPUT	CMI$
OUTPUT	CMO$
HANGUP	CMH$

EACH OF THESE REFERENCES ARE MADE WITH THE A0 REGISTER LOADED WITH THE ADDRESS OF THE LT TABLE DEFINING THE LINE TERMINAL GROUP. THE REFERENCE IS MADE BY AN ER INSTRUCTION WITH THE ADDRESS FIELD CONTAINING THE APPROPRIATE LABEL.

10.8. COMMUNICATION INITIALIZATION

THE INITIALIZE AND TERMINATE OPERATIONS MAY REFER TO MORE THAN ONE LINE TERMINAL GROUP. THE UPPER HALF OF THE A0 REGISTER MUST BE LOADED WITH THE NUMBER MINUS ONE OF LINE TERMINAL GROUPS TO BE CONSIDERED. THE LT TABLES FOR THESE GROUPS MUST START AT THE ADDRESS IN A0 LOWER HALF AND BE IN 12 CONSECUTIVE ADDRESSED LOCATIONS FROM THAT POINT.

THE INITIALIZE REQUEST ASSOCIATES THE INTERNAL NAME TO A @USE OR @ASG COMMAND AND SETS UP THE CM HANDLER ACCORDING TO THE LT TABLE FIELD.

WHEN AN EXECUTIVE REQUEST IS MADE TO CMS$, THE USER PROGRAM MUST BE IN THE REAL-TIME MODE AND REGISTER A0 MUST CONTAIN NUMBER OF LTS MINUS 1 IN THE LOWER HALF AND THE LOCATION OF THE FIRST LT IN THE LOWER HALF. THE CALLING SEQUENCE IS AS FOLLOWS:

```
          L         A0(LT COUNT, LT)
          ER        CMS$
```

```
        ----------------------------------------------------
A0      :     LTT COUNT MINUS ONE :        FIRST LTT ADDRESS :
        ----------------------------------------------------
```

LTT COUNT MINUS ONE (A0-H1) SPECIFIES THE NUMBER MINUS ONE OF THE CONTIGUOUSLY LOCATED LINE TERMINAL TABLES WHICH ARE TO BE INITIALIZED AND MUST NOT EXCEED THE NUMBER MINUS ONE OF COMMUNICATION LINE TERMINAL GROUPS CURRENTLY ASSIGNED TO THE USER.

FIRST LTT ADDRESS (A0-H2) SPECIFIES THE ADDRESS OF THE FIRST OF THE CONTIGUOUSLY

LOCATED LINE TERMINAL TABLES WHICH ARE TO BE INITIALIZED AND MUST SPECIFY THE ADDRESS OF A LINE TERMINAL TABLE IN THE USER'S DATA-BANK WHOSE FORMAT FOR AN EXECUTIVE REQUEST TO CMS$ IS AS FOLLOWS:

```
      -------------------------------------------------------
 00  :                                                       :  N
     :                                   INTERNAL GROUP NAME :
 01  :                                                       :  N
     :-------------------------------------------------------:
 02  :         :         : USAGE :        COMPLETION ADDRESS :  O
     :---------------------------:---------------------------:
 03  :                      MODE :                 POOL NAME :  O
     :---------------------------:---------------------------:
 04  :                           :                           :  O
     :---------------------------:---------------------------:
 05  :                           :                           :  O
     :---------------------------:---------------------------:
 06  :         : ACTION :  USAGE :        COMPLETION ADDRESS :  I
     :---------------------------:---------------------------:
 07  :                      MODE :                 POOL NAME :  I
     :---------------------------:---------------------------:
 08  :                           :                           :  I
     :---------------------------:---------------------------:
 09  :            DUAL POOL NAME :                           :  I
     :-------------------------------------------------------:
 10  :                                                       :  D
     :                                                       :
 11  :                                                       :  D
      -------------------------------------------------------
```

INTERNAL GROUP NAME (WORD 00-WW AND WORD 01-WW) SPECIFIES THE COMMUNICATION LINE TERMINAL (CLT) GROUP TO BE INITIALIZED AND MUST SPECIFY A CLT GROUP CURRENTLY ASSIGNED TO THE USER AND NOT CURRENTLY INITIALIZED. IF EITHER THE OUTPUT MODE (WORD 03-H1) OR THE OUTPUT POOL NAME (WORD 03-H2) IS NONZERO, THE INTERNAL GROUP NAME MUST SPECIFY A CLT GROUP WITH OUTPUT CAPABILITY AND WITH AN OUTPUT PATH AVAILABLE. IF THE OUTPUT MODE AND THE OUTPUT POOL NAME ARE BOTH ZERO, THE CLT GROUP SPECIFIED WILL NOT BE INITIALIZED FOR OUTPUT. LIKEWISE, IF EITHER THE INPUT MODE (WORD 07-H1) OR THE INPUT POOL NAME (WORD 07-H2) IS NONZERO, THE INTERNAL GROUP NAME MUST SPECIFY A CLT GROUP WITH INPUT CAPABILITY AND WITH AN INPUT PATH AVAILABLE. IF THE INPUT MODE AND THE INPUT POOL NAME ARE BOTH ZERO, THE CLT GROUP SPECIFIED WILL NOT BE INITIALIZED FOR INPUT. CTS AND WTS SUBSYSTEMS MUST BE INITIALIZED FOR SINGLE-MODE OUTPUT AND SINGLE-MODE INPUT. CTMC SUBSYSTEMS MAY BE INITIALIZED FOR SINGLE-MODE OR POOL-MODE OUTPUT AND/OR SINGLE-MODE OR POOLMODE OR DUAL POOL-MODE INPUT.

10.8.1. OUTPUT INITIALIZATION

IF THE SPECIFIED CLT GROUP IS TO BE INITIALIZED FOR OUTPUT, EITHER THE OUTPUT MODE OR THE OUTPUT POOL NAME MUST BE NONZERO, AND THE FOLLOWING PARAMETERS MUST BE PROVIDED:

OUTPUT USAGE (WORD 02-S3) SPECIFIES THE USAGE OF AN OUTPUT COMPLETION ACTIVITY. FOR SINGLE-MODE OUTPUT, THE OUTPUT USAGE CODE MUST BE EITHER ZERO OR ONE. FOR POOL-MODE OUTPUT, THE OUTPUT USAGE CODE MUST BE EITHER ZERO, ONE, OR TWO. ZERO SPECIFIES THAT NO OUTPUT COMPLETION ACTIVITY IS TO BE ACTIVATED. ONE SPECIFIES THAT AN OUTPUT COMPLETION ACTIVITY IS TO BE ACTIVATED EACH TIME THE OUTPUT OF A BUFFER IS COMPLETED (SUCCESSFULLY OR UNSUCCESSFULLY). TWO SPECIFIES THAT AN OUTPUT COMPLETION

ACTIVITY IS TO BE ACTIVATED EACH TIME THE OUTPUT OF THE LAST BUFFER IN AN OUTPUT QUEUE FOR POOL-MODE OUTPUT IS SUCCESSFULLY COMPLETED OR THE OUTPUT OF ANY BUFFER IN THE QUEUE IS UNSUCCESSFULLY COMPLETED.

OUTPUT COMPLETION ADDRESS (WORD 02-H2) SPECIFIES THE STARTING ADDRESS OF THE OUTPUT COMPLETION ACTIVITY. IF THE OUTPUT USAGE CODE IS NONZERO, THE OUTPUT COMPLETION ADDRESS MUST SPECIFY THE ADDRESS OF AN ACTIVITY WITHIN THE BOUNDS OF THE USER'S PROGRAM.

OUTPUT MODE (WORD 03-H1) SPECIFIES THE BUFFERING MODE FOR OUTPUT. ANY NONZERO VALUE SPECIFIES SINGLE-MODE OUTPUT. ZERO SPECIFIES POOL-MODE OUTPUT (IF THE OUTPUT POOL NAME IS NONZERO). IF THE OUTPUT MODE AND THE OUTPUT POOL NAME ARE BOTH ZERO, THE SPECIFIED CLT GROUP WILL NOT BE INITIALIZED FOR OUTPUT.

OUTPUT POOL NAME (WORD 03-H2) SPECIFIES THE POOL TO BE USED FOR POOL-MODE OUTPUT. IF THE OUTPUT MODE IS ZERO, THE OUTPUT POOL NAME MUST EITHER BE ZERO OR SPECIFY A POOL OF OPEN-CHAINED BUFFERS CURRENTLY ASSIGNED TO THE USER.

10.8.2. INPUT INITIALIZATION

IF THE SPECIFIED CLT GROUP IS TO BE INITIALIZED FOR INPUT, EITHER THE INPUT MODE OR THE INPUT POOL NAME MUST BE NONZERO, AND THE FOLLOWING PARAMETERS MUST BE PROVIDED:

INPUT ACTION (WORD 06-S2) SPECIFIES THE ACTION TO BE TAKEN BY THE COMMUNICATION HANDLER WHEN THE INPUT TO A POOL-MODE BUFFER IS COMPLETED. IF POOL-MODE INPUT IS SPECIFIED, THE INPUT ACTION CODE MUST BE EITHER ZERO OR ONE. ZERO SPECIFIES THAT INPUT IS TO BE TERMINATED WHEN THE INPUT TO A BUFFER IS COMPLETED. ONE SPECIFIES THAT INPUT IS TO CONTINUE, USING THE NEXT BUFFER (IF THERE IS ONE) IN THE INPUT POOL.

INPUT USAGE (WORD 06-S3) SPECIFIES THE USAGE OF AN INPUT COMPLETION ACTIVITY AND MUST BE EITHER ZERO OR ONE. ZERO SPECIFIES THAT NO INPUT COMPLETION ACTIVITY IS TO BE ACTIVATED. OOE SPECIFIES THAT AN INPUT COMPLETION ACTIVITY IS TO BE ACTIVATED EACH TIME THE INPUT TO A BUFFER IS COMPLETED (SUCCESSFULLY OR UNSUCCESSFULLY).

INPUT COMPLETION ADDRESS (WORD 06-H2) SPECIFIES THE STARTING ADDRESS OF THE INPUT COMPLETION ACTIVITY. IF THE INPUT USAGE CODE IS ONE, THE INPUT COMPLETION ADDRESS MUST SPECIFY THE ADDRESS OF AN ACTIVITY WITHIN THE BOUNDS OF THE USER'S PROGRAM.

INPUT MODE (WORD 07-H1) SPECIFIES THE BUFFERING MODE FOR INPUT. ANY NONZERO VALUE SPECIFIES SINGLE-MODE INPUT. ZERO SPECIFIES POOL-MODE INPUT (IF THE INPUT POOL NAME IS NONZERO). IF THE INPUT MODE AND THE INPUT POOL NAME ARE BOTH ZERO, THE SPECIFIED CLT GROUP WILL NOT BE INITIALIZED FOR INPUT.

INPUT POOL NAME (WORD 07-H2) SPECIFIES THE POOL TO BE USED FOR POOL-MODE INPUT. IF THE INPUT MODE IS ZERO, THE INPUT POOL NAME MUST EITHER BE ZERO OR SPECIFY A POOL CURRENTLY ASSIGNED TO THE USER.

INPUT DUAL POOL NAME (WORD 09-H1) SPECIFIES THE POOL TO BE USED FOR DUAL POOL-MODE INPUT. IF POOL-MODE INPUT IS SPECIFIED, THE DUAL POOL NAME MUST EITHER BE ZERO OR SPECIFY A POOL CURRENTLY ASSIGNED TO THE USER AND DIFFERENT FROM THE INPUT POOL. IF THE DUAL POOL NAME IS ZERO, THE SPECIFIED CLT GROUP WILL NOT BE INITIALIZED FOR DUAL POOL-MODE INPUT.

10.8.3. INITIALIZATION ERRORS

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMS$ ARE AS FOLLOWS:

01 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

02 LINE TERMINAL TABLE SPECIFIED TO BE INITIALIZED IS NOT WITHIN THE BOUNDS OF THE USER'S DATA-BANK.

03 INTERNAL GROUP NAME SPECIFIES A COMMUNICATION LINE TERMINAL GROUP CURRENTLY INITIALIZED.

06 INTERNAL GROUP NAME DOES NOT SPECIFY A COMMUNICATION LINE TERMINAL GROUP CURRENTLY ASSIGNED TO THE USER.

27 OUTPUT MODE AND INPUT MODE DO NOT BOTH SPECIFY SINGLE-MODE BUFFERING FOR A CTS SUBSYSTEM OR A WTS SUBSYSTEM.

53 ADDRESS OF A LINE TERMINAL TABLE TO BE INITIALIZED IS IN THE USER'S INSTRUCTION-BANK.

74 THE OUTPUT MODE AND THE OUTPUT POOL NAME ARE BOTH ZERO, AND THE INPUT MODE AND THE INPUT POOL NAME ARE BOTH ZERO.

10.8.3.1. OUTPUT ERRORS

IF A COMMUNICATION LINE TERMINAL GROUP IS TO BE INITIALIZED FOR INPUT ONLY, THE FOLLOWING OCTAL ERROR CODES AND ERROR CONDITIONS ARE NOT APPLICABLE.

14 OUTPUT USAGE CODE SPECIFIED IS EITHER A VALUE OTHER THAN ZERO OR ONE FOR SINGLE-MODE OUTPUT OR A VALUE OTHER THAN ZERO, ONE, OR TWO FOR POOL-MODE OUTPUT.

15 OUTPUT POOL NAME DOES NOT SPECIFY A POOL OF OPEN-CHAINED BUFFERS CURRENTLY ASSIGNED TO THE USER.

40 INTERNAL GROUP NAME DOES NOT SPECIFY A COMMUNICATION LINE TERMINAL GROUP WITH OUTPUT CAPABILITY.

45 OUTPUT COMPLETION ADDRESS SPECIFIED IS NOT WITHIN THE BOUNDS OF THE USER'S PROGRAM.

76 INTERNAL GROUP NAME SPECIFIES A COMMUNICATION LINE TERMINAL GROUP FOR WHICH AN OUTPUT PATH IS NOT AVAILABLE.

10.8.3.2. INPUT ERRORS

IF A COMMUNICATION LINE TERMINAL GROUP IS TO BE INITIALIZED FOR OUTPUT ONLY, THE FOLLOWING OCTAL ERROR CODES AND ERROR CONDITIONS ARE NOT APPLICABLE:

05 INPUT USAGE CODE SPECIFIED IS A VALUE OTHER THAN ZERO OR ONE.

07 INPUT ACTION CODE SPECIFIED FOR POOL-MODE INPUT IS A VALUE OTHER THAN ZERO OR ONE.

11 INPUT POOL NAME OR DUAL POOL NAME DOES NOT SPECIFY A POOL CURRENTLY ASSIGNED TO THE USER, OR THE INPUT POOL NAME AND THE DUAL POOL NAME SPECIFY THE SAME POOL.

13 INPUT COMPLETION ADDRESS SPECIFIED IS NOT WITHIN THE BOUNDS

OF THE USER'S PROGRAM.

41 INTERNAL GROUP NAME DOES NOT SPECIFY A COMMUNICATION LINE TERMINAL GROUP WITH INPUT CAPABILITY.

77 INTERNAL GROUP NAME SPECIFIES A COMMUNICATION LINE TERMINAL GROUP FOR WHICH AN INPUT PATH IS NOT AVAILABLE.

WHEN AN EXECUTIVE REQUEST IS MADE TO CMS$, THE LINE TERMINAL TABLES SPECIFIED BY THE USER WILL BE INITIALIZED IN TURN BEGINNING WITH THE LINE TERMINAL TABLE SPECIFIED IN THE LOWER HALF OF REGISTER A0. THUS, WHEN AN ERROR CONDITION IS DETECTED AND CONTROL RETURNED TO THE USER'S ERROR CONTINGENCY ROUTINE, ALL OF THE LINE TERMINAL TABLES, PRECEDING THE ONE WHICH WAS BEING INITIALIZED WHEN THE ERROR CONDITION WAS DETECTED, WILL HAVE BEEN INITIALIZED. THE USER MAY ELECT TO USE THE INITIALIZED LINE TERMINAL TABLES OR TERMINATE THEM BY MAKING AN EXECUTIVE REQUEST TO CMT$.

10.9. DIALING

THE DIAL OPERATION CAUSES INITIATION OF THE BUFFER SPECIFIED IN WORD 11 OF THE LT TABLE ADDRESSED BY A0. THE TELEPHONE NUMBER TO BE DIALED MUST BE IN BCD FORMAT. UPON DETERMINING THAT THE DIAL OPERATION HAS BEEN COMPLETED, THE USER WILL GET CONTROL AT THE DIAL COMPLETION ACTIVITY STARTING POINT , IF SUCH A ROUTINE EXISTS, WITH THE DIAL STATUS IN S1 OF WORD 11. INPUT AND OUTPUT CAN BE INITIATED BEFORE A DIAL OPERATION IS COMPLETED IN WHICH CASE NO CHARACTER TIMING IS DONE BEFORE THE FIRST CHARACTER TRANSFER OR THE DIAL TIMES OUT. FOR THE CASE OF INITIATING OUTPUT, THE OUTPUT LINE TERMINAL REQUIRES AN ENABLE FROM THE DIAL UNIT AND HENCE TRANSFER STARTS IMMEDIATELY UPON DIAL COMPLETION.

IF AN AUTOMATIC CALLING UNIT DOES NOT EXIST FOR AN LT GROUP, THE DIAL OPERATION WILL RESULT IN A MESSAGE BEING DISPLAYED ON THE CONSOLE IN THE FORMAT:

DIAL NUMBER CC/LL

WHERE CC/LL IS THE CHANNEL AND LINE NUMBER TO BE CONNECTED. THE OPERATOR MUST RESPOND WITH AN N OR Y TO INDICATE COMPLETION. A REQUEST FOR OUTPUT FOR THE MANUAL CASE IS NOT HONORED UNTIL A Y RESPONSE IS RECEIVED.

A DIAL REQUEST ON A LEASED-LINE LT GROUP IS GIVEN A UNIQUE STATUS CODE AND ANY SUBSEQUENT INPUT OR OUTPUT WILL BE HONORED.

THE DIAL COMPLETION ACTIVITY OPERATES AS A HIGH PRIORITY INTERRUPT ROUTINE AND IS ALLOWED ONLY A LIMITED TIME PERIOD TO PERFORM ITS ANALYSIS OF THE INTERRUPT. WHEN THIS ROUTINE IS ENTERED, A0 WILL CONTAIN THE LOCATION OF THE APPROPRIATE LT TABLE.

MUST BE IN THE REAL-TIME MODE AND REQISTER A0 MUST CONTAIN THE LOCATION OF THE LINE TERMINAL TABLE. THE CALLING SEQUENCE IS AS FOLLOWS:

```
                    L,U        A0,LTT
                    ER         CMD$
```

INTERNAL GROUP NAME (WORD 00-WW AND WORD 01-WW) SPECIFIES THE COMMUNICATION LINE TERMINAL (CLT) GROUP TO ORIGINATE THE CALL AND MUST SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED, AND THE SPECIFIED CLT GROUP MUST HAVE DIAL AND HANGUP CAPABILITY.

DIAL USAGE (WORD 10-S3) SPECIFIES THE USAGE OF A DIAL COMPLETION ACTIVITY AND MUST BE EITHER ZERO OR ONE. ZERO SPECIFIES THAT NO DIAL COMPLETION ACTIVITY IS TO BE ACTIVATED. ONE SPECIFIES THAT A DIAL COMPLETION ACTIVITY IS TO BE ACTIVATED WHEN THE CALL IS ANSWERED (OR ABANDONED).

DIAL COMPLETION ADDRESS (WORD 10-H2) SPECIFIES THE STARTING ADDRESS OF THE DIAL

COMPLETION ACTIVITY. IF THE DIAL USAGE CODE IS ONE, THE DIAL COMPLETION ADDRESS MUST SPECIFY THE ADDRESS OF AN ACTIVITY WITHIN THE BOUNDS OF THE USER'S PROGRAM.

DIAL CHARACTER COUNT (WORD 11-H1) SPECIFIES THE NUMBER OF BINARY CODED DECIMAL (BCD) DIAL DIGITS IN THE DIAL BUFFER AND MUST BE A VALUE BETWEEN ONE AND SIXTEEN AND MUST NOT BE A VALUE THAT WOULD CAUSE THE DIAL BUFFER TO EXTEND BEYOND THE UPPER BOUND OF THE USER'S DATA-BANK.

DIAL BUFFER ADDRESS (WORD 11-H2) SPECIFIES THE STARTING ADDRESS OF THE DIAL BUFFER AND MUST BE IN THE USER'S DATA-BANK AND MUST NOT BE IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

10.9.1.1. DIAL ERRORS

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMD$ ARE AS FOLLOWS:

54 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

55 DIAL USAGE CODE SPECIFIED IS A VALUE OTHER THAN ZERO OR ONE.

56 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED, OR THE INTERNAL GROUP NAME DOES NOT SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED.

57 DIAL COMPLETION ADDRESS SPECIFIED IS NOT WITHIN THE BOUNDS OF THE USER'S PROGRAM.

71 DIAL .HARACTER COUNT SPECIFIED IS NOT A VALUE BETWEEN ONE AND SIXTEEN.

64 DIAL BUFFER SPECIFIED IS NOT WITHIN BOUNDS OF THE USER'S DATA-BANK.

62 CHARACTERS IN THE DIAL BUFFER SPECIFIED ARE NOT BINARY CODED DECIMAL (BCD).

67 INTERNAL GROUP NAME DOES NOT SPECIFY A CLT GROUP WITH DIAL AND HANG-UP CAPABILITY.

10.10. INPUT

WHEN AN EXECUTIVE REQUEST IS MADE TO CMI$, THE USER PROGRAM MUST BE IN THE REAL TIME MODE AND REGISTER A0 MUST CONTAIN THE LOCATION OF THE LINE TERMINAL TABLE. THE CALLING SEQUENCE IS AS FOLLOWS:

```
                    L,U        A0,LTT
                    ER         CMI$
```

10.10.1. SINGLE MODE

IF AN EXECUTIVE REQUEST TO CMI$ IS FOR SINGLE-MODE INPUT, THE LINE TERMINAL TABLE MUST BE CURRENTLY INITIALIZED FOR SINGLE-MODE INPUT, AND CONVERSELY, IF THE LINE TERMINAL TABLE IS CURRENTLY INITIALIZED FOR SINGLE-MODE INPUT, AN EXECUTIVE REQUEST TO CMI$ MUST BE FOR SINGLE-MODE INPUT, AND THE FOLLOWING TWO PARAMETERS MUST BE PROVIDED.

INPUT CHARACTER COUNT (WORD 07-H1) SPECIFIES THE NUMBER OF INPUT CHARACTERS FOR SINGLE-MODE INPUT AND MUST BE GREATER THAN ZERO, MUST NOT EXCEED THE SYSTEM MAXIMUM, AND MUST NOT BE A VALUE THAT WOULD CAUSE THE INPUT BUFFER TO EXTEND BEYOND THE UPPER BOUND OF THE USER'S DATA-BANK.

INPUT BUFFER ADDRESS (WORD 07-H2) SPECIFIES THE STARTING ADDRESS OF THE INPUT BUFFER FOR SINGLE-MODE INPUT AND MUST BE IN THE USER'S DATA-BANK AND MUST NOT BE IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

INPUT ON LINE TERMINALS IS IN ONE OF TWO MODES; SINGLE AND MULTIPLE BUFFER. EITHER OF THESE MODES MAY BE SYNCHRONOUS OR ASYNCHRONOUS.

IN THE SINGLE BUFFER MODE, AN INPUT REQUEST CAUSES INITIATION OF INPUT WITH THE ACCESS WORD FROM WORD 7 OF THE LT TABLE. COMPLETION OF INPUT IS EXPECTED TO BE DENOTED BY AN EXTERNAL INTERRUPT RESULTING FROM DETECTION OF AN EOM SIGNAL FOR WTS OR CTS DEVICES OR AS A RESULT OF A TIME OUT OR INPUT MONITOR INTERRUPT FOR THE CTMC. AFTER THE END OF INPUT IS DETECTED THE INPUT LT IS TURNED OFF. THE COMPLETION STATUS OF THE INPUT BUFFER IS STORED IN S1 OF WORD 6. THE NUMBER OF CHARACTERS ACCEPTED AS INPUT IS STORED IN H1 OF WORD 9. FOR SYNCHRONOUS LT DEVICES OPERATING ON THE CTMC, SINCE AN EXTERNAL INTERRUPT DOES NOT OCCUR ON A UNIQUE CHARACTER, IF A MESSAGE IS RECEIVED WHICH IS SHORTER THAN THE BUFFER THERE MAY BE SYNC CHARACTERS IN THE BUFFER FOLLOWING THE LAST DATA CHARACTER. THE SINGLE BUFFER MODE IS ANTICIPATED FOR USAGE WHERE THE INPUT MESSAGES ARE EITHER FIXED LENGTH OR A MAXIMUM LENGTH CAN BE SPECIFIED.

10.10.2. POOL MODE

IF AN EXECUTIVE REQUEST TO CMI$ IS FOR POOL-MODE INPUT, THE LINE TERMINAL TABLE MUST BE CURRENTLY INITIALIZED FOR POOL-MODE INPUT, AND CONVERSELY, IF THE LINE TERMINAL TABLE IS CURRENTLY INITIALIZED FOR POOL-MODE INPUT, AN EXECUTIVE REQUEST TO CMI$ MUST BE FOR POOL-MODE INPUT, AND THERE MUST BE AT LEAST ONE BUFFER CURRENTLY AVAILABLE IN THE INPUT POOL.

IF AN EXECUTIVE REQUEST TO CMI$ IS FOR DUAL POOL-MODE INPUT, THE LINE TERMINAL TABLE MUST BE CURRENTLY INITIALIZED FOR DUAL POOL-MODE INPUT, AND CONVERSELY, IF THE LINE TERMINAL TABLE IS CURRENTLY INITIALIZED FOR DUAL POOL-MODE INPUT, AN EXECUTIVE REQUEST TO CMI$ MUST BE FOR DUAL POOL-MODE INPUT, AND THERE MUST BE AT LEAST ONE BUFFER CURRENTLY AVAILABLE IN THE DUAL POOL.

IN THE MULTIPLE BUFFER INPUT MODE, TWO METHODS (NORMAL POOL MODE AND DUAL POOL MODE) OF POOL USAGE ARE PROVIDED WITH H2 OF WORD 7 POINTING TO A BUFFER POOL CONTROL WORD WHICH LOCATES A POOL OF CHAINED BUFFERS. THE SINGLE/MULTIPLE MODE IS DETERMINED BY THE CONTENT OF H1 OF WORD 7 OF THE LT TABLE (ZERO FOR MULTIPLE). THE DUAL POOL MODE OF INPUT DIFFERS ONLY FROM THE NORMAL POOL MODE OF INPUT IN THAT THE FIRST BUFFER TO BE USED IS DESIGNATED BY H1 OF WORD 9 AND ALL SUCCEEDING BUFFERS TO BE USED ARE DESIGNATED BY H2 OF WORD 7. INPUT IN THE MULTIPLE MODE IS INITIATED BY SETTING UP AN I/O ACCESS WORD WITH A COUNT EQUAL TO THE NUMBER OF CHARACTERS IN A FULL BUFFER AND AN ADDRESS OF THE SECOND WORD OF THE FIRST AVAILABLE BUFFER IN THE POOL (WORD 1 IS A LINKING WORD). INPUT IS WITH MONITOR. AS EACH BUFFER IS FILLED, IT IS ADDED TO THE BACKUP QUEUE (WORD 8 OF LT TABLE), THE FIRST WORD OF THE BUFFER IS LOADED WITH THE COMPLETION STATUS (S3 OF WORD 1) AND THE NUMBER OF CHARACTERS TRANSFERRED INTO THE BUFFER (T1 OF WORD 1). DEPENDING UPON THE INPUT USAGE FIELD (S3 OF WORD 6), THE INPUT COMPLETION ROUTINE IS ACTIVATED. THE END OF A MESSAGE IS DETECTED BY AN EXTERNAL INTERRUPT (CTMC, CTS, OR WTS) OR A TIME OUT (CTMC) FOR THE SINGLE BUFFER MODE. A MONITOR INTERRUPT CAUSES BUFFER SWITCHING WITH INPUT INITIATED USING THE NEXT BUFFER IN THE POOL SPECIFIED BY H2 OF WORD 7.

UPON OCCURRENCE OF AN END-OF-MESSAGE INDICATION, FOR THE MULTIPLE BUFFER MODE, S2 OF WORD 6 IS TESTED TO INDICATE WHETHER TO SET UP INPUT INTO ANOTHER BUFFER (NONZERO VALUE) OR DISCARD FURTHER INPUT UNTIL THE NEXT INPUT REQUEST.

THE COMMUNICATIONS BUFFER POOL HANDLING ROUTINES ARE DESCRIBED IN THE CHAPTER ON FILE REFERENCING UNDER THE FILE CONTROL SYSTEM. THE CM HANDLER PROVIDES FOR USING A COMMON POOL AMONG SEVERAL LT GROUPS IF THE APPLICATION WILL ALLOW. ON INPUT THE

HANDLER REMOVES BUFFERS FROM THE POOL, AND THE WORKER PROGRAM MUST RETURN THEM AFTER PROCESSING.

THE INPUT COMPLETION ROUTINE IS ACTIVATED WITH A0 SET TO THE STARTING ADDRESS OF THE LT TABLE. THIS ROUTINE IS GIVEN CONTROL AS A HIGH PRIORITY INTERRUPT PROCESSING ROUTINE AND HENCE IS ALLOWED MINIMUM TIME FOR ANALSYSIS OF THE INTERRUPT. COMPLETION ROUTINE TIMING CONSIDERATION ARE OUTLINED IN THE NEXT SECTION.

10.10.3. INPUT ERRORS

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMI$ ARE AS FOLLOWS:

12 INPUT POOL DOES NOT HAVE ANY BUFFERS CURRENTLY AVAILABLE FOR POOL-MODE INPUT, OR DUAL POOL DOES NOT HAVE ANY BUFFERS CURRENTLY AVAILABLE FOR DUAL POOL-MODE INPUT.

21 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED FOR INPUT.

23 CHARACTER COUNT SPECIFIED FOR SINGLE-MODE INPUT IS EQUAL TO ZERO OR IS GREATER THAN THE SYSTEM MAXIMUM.

26 INPUT BUFFER SPECIFIED FOR SINGLE-MODE INPUT IS NOT WITHIN THE BOUNDS OF THE USER'S DATA-BANK.

42 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

46 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED, OR THE INTERNAL GROUP NAME DOES NOT SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED.

50 INPUT BUFFER SPECIFIED FOR SINGLE-MODE INPUT IS IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

10.11. OUTPUT

OUTPUT ON LINE TERMINALS IS IN ONE OF TWO MODES: SINGLE AND MULTIPLE BUFFER. EITHER OF THESE MODES MAY BE SYNCHRONOUS OR ASYNCHRONOUS. THE FOLLOWING DISCUSSES THE TWO TYPE OF MODES, OPERATING PROCEDURES, AND ERROR CONDITIONS THAT MAY RESULTS.

10.11.1. SINGLE MODE

THE SINGLE BUFFER MODE IS DENOTED BY A NONZERO VALUE IN H1 OF WORD 3. SINGLE BUFFER MODE OUTPUT IS EXPECTED TO COMPLETE AS A RESULT OF AN EXTERNAL INTERRUPT FOR WTS OR CTS SUBSYSTEMS OR AS A RESULT OF A MONITOR INTERRUPT FOR THE CTMC. THE USER CAN AVOID THE INTERRUPT PROCESSING FOR OUTPUT BY SPECIFYING A CHARACTER COUNT GREATER THAN THE CHARACTER POSITION CONTAINING THE EOT INDICATION FOR THE CTMC IN WHICH CASE THE DEVICE WILL TIME OUT WITH A STATUS CODE OF 5 IN S1 OF WORD 2 OF THE LT TABLE. THE TIME OUT METHOD IS NOT ALLOWED ON THE WTS OR CTS BECAUSE OF THE NECESSITY OF TRANSMITTING END OF MESSAGE AND MESSAGE PARITY CHARACTERS. THE NUMBER OF CHARACTERS TRANSFERRED IN THE SINGLE BUFFER MODE IS STORED IN H1 OF WORD 5 BEFORE THE OUTPUT COMPLETION ROUTINE IS ACTIVATED, IF SUCH A ROUTINE IS SPECIFIED.

IF AN EXECUTIVE REQUEST TO CMO$ IS FOR SINGLE-MODE OUTPUT, THE LINE TERMINAL TABLE MUST BE CURRENTLY INITIALIZED FOR SINGLE-MODE OUTPUT, AND CONVERSELY, IF THE LINE TERMINAL TABLE IS CURRENTLY INITIALIZED FOR SINGLE-MODE OUTPUT, AN EXECUTIVE REQUEST

TO CMO$ MUST BE FOR SINGLE-MODE OUTPUT, AND THE FOLLOWING TWO PARAMETERS MUST BE PROVIDED.

OUTPUT CHARACTER COUNT (WORD 03-H1) SPECIFIES THE NUMBER OF OUTPUT CHARACTERS FOR SINGLE-MODE OUTPUT AND MUST BE GREATER THAN ZERO, MUST NOT EXCEED THE SYSTEM MAXIMUM, AND MUST NOT BE A VALUE THAT WOULD CAUSE THE OUTPUT BUFFER TO EXTEND BEYOND THE UPPER BOUND OF THE USER'S DATA-BANK.

OUTPUT BUFFER ADDRESS (WORD 03-H2) SPECIFIES THE STARTING ADDRESS OF THE OUTPUT BUFFER FOR SINGLE-MODE OUTPUT AND MUST BE IN THE USER'S DATA-BANK AND MUST NOT BE IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

10.11.2. POOL MODE

IF AN EXECUTIVE REQUEST TO CMO$ IS FOR POOL-MODE OUTPUT, THE LINE TERMINAL TABLE MUST BE CURRENTLY INITIALIZED FOR POOL-MODE OUTPUT, AND CONVERSELY, IF THE LINE TERMINAL TABLE IS CURRENTLY INITIALIZED FOR POOL-MODE OUTPUT, AN EXECUTIVE REQUEST TO CMO$ MUST BE FOR POOL-MODE OUTPUT, AND THE FOLLOWING TWO PARAMETERS MUST BE PROVIDED.

START OF BACKUP QUEUE (WORD 04-H2) SPECIFIES THE ADDRESS OF THE FIRST BUFFER IN THE OUTPUT QUEUE FOR POOL-MODE OUTPUT AND MUST SPECIFY A BUFFER CURRENTLY REMOVED FROM THE OUTPUT POOL AND NOT CURRENTLY IN ANOTHER OUTPUT QUEUE.

END OF BACKUP QUEUE (WORD 04-H1) SPECIFIES THE ADDRESS OF THE LAST BUFFER IN THE OUTPUT QUEUE FOR POOL-MODE OUTPUT AND MUST SPECIFY A BUFFER CURRENTLY REMOVED FROM THE OUTPUT POOL AND NOT CURRENTLY IN ANOTHER OUTPUT QUEUE.

THE OUTPUT QUEUE MAY CONTAIN ANY NUMBER OF BUFFERS BETWEEN ONE AND THE TOTAL NUMBER OF BUFFERS CURRENTLY REMOVED FROM THE OUTPUT POOL AND NOT CURRENTLY IN ANOTHER OUTPUT QUEUE AND WHOSE FORMAT IS AS FOLLOWS:

```
     -------------------------------------------------------
00   : CHARACTER COUNT :          :       NEXT BUFFER ADDRESS :
     :-----------------------------------------------------:
     :                                                     :
     /                                                     /
     \                 (OUTPUT CHARACTERS)                 \
     /                                                     /
     :                                                     :
     -------------------------------------------------------
```

OUTPUT CHARACTER COUNT (WORD 00-T1) SPECIFIES THE NUMBER OF CHARACTERS TO BE OUTPUT FROM THE BUFFER AND MUST BE GREATER THAN ZERO AND MUST NOT EXCEED THE CAPACITY OF THE BUFFER.

NEXT BUFFER ADDRESS (WORD 00-H2) SPECIFIES THE ADDRESS OF THE NEXT BUFFER IN THE OUTPUT QUEUE AND, EXCEPT FOR THE LAST BUFFER IN THE OUTPUT QUEUE, MUST SPECIFY A BUFFER CURRENTLY REMOVED FROM THE OUTPUT POOL AND NOT CURRENTLY IN ANOTHER OUTPUT QUEUE. WORD 00-H2 OF THE LAST BUFFER IN THE OUTPUT QUEUE MAY CONTAIN ANYTHING.

IN THE MULTIPLE BUFFER OUTPUT MODE, EACH BUFFER IS TRANSFERRED WITH MONITOR, AND UPON OCCURRENCE OF AN INTERRUPT, THE NEXT BUFFER IN THE BACKUP QUEUE IS INITIATED. AS EACH BUFFER IS REMOVED FROM THE QUEUE, THE CM HANDLER UPDATES THE START OF OUTPUT BACKUP QUEUE FIELD (H2 OF WORD 4). THE WORKER PROGRAM ADDS TO THE QUEUE BY UPDATING H1 OF WORD 4. AS EACH BUFFER IS EMPTYED, THE HANDLER STORES THE COMPLETION STATUS IN S3 OF WORD 1 OF THE BUFFER AND THE NUMBER OF CHARACTERS TRANSFERRED IN T1 OF WORD 1 AND, DEPENDING UPON THE CONTENTS OF THE OUTPUT USAGE FIELD, ACTIVATES THE OUTPUT COMPLETION ROUTINE. THE CM HANDLER EXAMINES THE START OF BACKUP QUEUE FIELD (H2 OF WORD 5) WHEN THE OUTPUT IS FIRST INITIATED AND THEN WORKS FROM THE LINK FIELD (H2 OF WORD 1 OF THE BUFFER) TO DETERMINE THE END OF THE CHAIN. THE WORKER PROGRAM MUST ADD THE COMPLETED BUFFERS BACK TO THE AVAILABLE POOL.

WHEN THE END OF THE BACKUP QUEUE IS REACHED, THE HANDLER WILL TURN OFF OUTPUT AND RETURN A STATUS CODE OF 20 TO DENOTE THE 'CAUGHT UP' CONDITION. IF THE WORKER PROGRAM SUBMITS A NEW BUFFER AFTER THE HANDLER MAKES THE CHECK, THIS STATUS CAN BE USED BY THE WORKER AS AN INDICATION TO MAKE ANOTHER REFERENCE TO CMO$. THE START OF BACKUP QUEUE MUST BE RESET FOR THIS REFERENCE.

THE OUTPUT COMPLETION ACTIVITY, IF ANY, IS ACTIVATED WITH A0 CONTAINING THE ASSOCIATE LT TABLE ADDRESS, AND A1 CONTAINING THE BUFFER OR USER'S AREA LOCATION FROM WITH OUTPUT WAS INITIATED. THIS ROUTINE IS GIVEN CONTROL AS A HIGH PRIORITY INTERRUPT ROUTINE AND HENCE IS ALLOWED MINIMUM TIME IN ANALYSIS OF THE INTERRUPT. COMPLETE ROUTINE TIMING CONSIDERATION ARE OUTLINED IN THE NEXT SECTION.

10.11.3. OUTPUT ERRORS

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMO$ ARE AS FOLLOWS:

03 ADDRESS OF A BUFFER IN THE OUTPUT QUEUE FOR POOL-MODE OUTPUT DOES NOT SPECIFY A BUFFER CURRENTLY REMOVED FROM THE OUTPUT POOL OR SPECIFIES A BUFFER CURRENTLY IN ANOTHER OUTPUT QUEUE.

04 CHARACTER COUNT OF A BUFFER IN THE OUTPUT QUEUE FOR POOL-MODE OUTPUT IS EQUAL TO ZERO OR IS GREATER THAN THE CAPACITY OF THE BUFFER.

23 CHARACTER COUNT SPECIFIED FOR SINGLE-MODE OUTPUT IS EQUAL TO ZERO OR IS GREATER THAN THE SYSTEM MAXIMUM.

31 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED FOR OUTPUT.

34 OUTPUT BUFFER SPECIFIED FOR SINGLE-MODE OUTPUT IS NOT WITHIN THE BOUNDS OF THE USER'S DATA-BANK.

42 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

46 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED, OR THE INTERNAL GROUP NAME DOES NOT SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED.

51 OUTPUT BUFFER SPECIFIED FOR SINGLE-MODE OUTPUT IS IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

10.11.3.1. SECOND LEVEL SUPPORT

SECOND LEVEL SUPPORT IS AN EXTENSION OF THE INTERFACE USED FOR REMOTE CPU TRANSMISSIONS WHICH INCLUDES THE UNIVAC 1004 CARD PROCESSOR. THIS LEVEL USES THE SYSTEM REFERENCE ER CMSA$ MEANING SEND AND ACKNOWLEDGE, IN ADDITION TO THE INITIALIZE CMS$, TERMINATE CMT$, DIAL CMD$, AND HANGUP CMH$ DESCRIBED IN FIRST LEVEL. THESE REFERENCES MUST BE MADE VIA AN ER INSTRUCTION WITH THE A0 REGISTER LOADED WITH THE LT TABLE STARTING ADDRESS FOR THE GROUP CONCERNED.

FOR SECOND LEVEL OPERATIONS THE EXECUTIVE REQUEST CMSA$ IS THE NORMAL SYSTEM REFERENCE, BUT THE FIRST LEVEL REFERENCES, CMI$ AND CMO$, MAY ALSO BE USED IF APPLICABLE FOR A USERS APPLICATION. THE CALLING SEQUENCE FOR THE CMSA$ REQUEST IS AS FOLLOWS:

```
          L,U        A0,LTT
          ER         CMSA$
```

10.11.3.2. BUFFER FORMATS

OUTPUT CHARACTER COUNT (WORD 03-H1) SPECIFIES THE NUMBER OF CHARACTERS TO BE OUTPUT FROM THE OUTPUT BUFFER AND MUST BE GREATER THAN ZERO, MUST NOT EXCEED THE SYSTEM MAXIMUM, AND MUST NOT BE A VALUE THAT WOULD CAUSE THE OUTPUT BUFFER TO EXTEND BEYOND THE UPPER BOUND OF THE USER'S DATA-BANK.

OUTPUT BUFFER ADDRESS (WORD 03-H2) SPECIFIES THE STARTING ADDRESS OF THE OUTPUT BUFFER AND MUST BE IN THE USER'S DATA-BANK AND MUST NOT BE IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

INPUT CHARACTER COUNT (WORD 07-H1) SPECIFIES THE NUMBER OF CHARACTERS TO BE INPUT TO THE INPUT BUFFER AND MUST BE GREATER THAN ZERO, MUST NOT EXCEED THE SYSTEM MAXIMUM, AND MUST NOT BE A VALUE THAT WOULD CAUSE THE INPUT BUFFER TO EXTEND BEYOND THE UPPER BOUND OF THE USER'S DATA-BANK.

INPUT BUFFER ADDRESS (WORD 07-H2) SPECIFIES THE STARTING ADDRESS OF THE INPUT BUFFER AND MUST BE IN THE USER'S DATA-BANK AND MUST NOT BE IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

INPUT BUFFER TIME INTERVALS (WORD 09-H2) SPECIFIES THE NUMBER OF 600-MILLISECOND TIME INTERVALS TO BE ALLOWED FOR THE NORMAL COMPLETION OF INPUT TO THE INPUT BUFFER.

THIS SUPPORT ASSUMES SEVEN BIT CHARACTERS(SIX DATA PLUS ONE PARITY)WITH THE FOLLOWING MESSAGE FORMAT:

SOM - START OF MESSAGE
CONTROL CHARACTER
DATA
EOM - END OF MESSAGE
MPC - MESSAGE PARITY CHARACTER
EOB - END OF BUFFER

THE VARIATIONS THAT MAY APPEAR IN THE WORKER PROGRAM BUFFER DEPENDS UPON THE TYPE OF SUBSYSTEM. FOR THE CTS, ALL OF THESE CHARACTERS MUST BE PRESENT IN OUTPUT BUFFERS AND ALL EXCEPT THE SOM AND EOB ARE TRANSMITTED TO THE INPUT BUFFER. THE WTS COMPUTES MESSAGE PARITY AND GENERATES THE SOM AND EOM HENCE ONLY THE CONTROL AND DATA CHARACTERS ARE NEEDED FOR OUTPUT AND OBSERVED FOR INPUT. FOR CTMC DEVICES ALL OF THE SPECIFIED CHARACTERS MUST BE PRESENT FOR OUTPUT AND ALL EXCEPT EOB WILL APPEAR IN THE INPUT BUFFER. IN ADDITION FOR CTMC DEVICES, A MINIMUM OF TWO SYNC CHARACTERS MUST PRECEDE THE SOM CHARACTER FOR OUTPUT. THE SYNC CHARACTERS WILL NOT APPEAR IN THE INPUT BUFFER. THE FOLLOWING ILLUSTRATES THE PREVIOUS EXPLANATIONS OF SECOND LEVEL WORKER PROGRAM BUFFER VARIATIONS FOR THE DIFFERENT SUBSYSTEMS OF WTS, CTS, AND CTMC.

CTMC OUTPUT BUFFER	LINE SEQUENCE & SOURCE	CTMC INPUT
SYNC (MINIMUM IS 2 SYNC)	SYNC (BUFFER)	SOM
SYNC (MAX AS PROGRAMMED)	SYNC (BUFFER)	CONTROL CHAR
SOM	SOM (BUFFER)	DATA
CONTROL CHARACTER	CONTROL CHAR (BUFFER)	EOM
DATA	DATA (BUFFER)	MPC
EOM	EOM (BUFFER)	
MPC	MPC (BUFFER)	
EOB		

CTS OUTPUT BUFFER	LINE SEQUENCE & SOURCE	CTS INPUT
SOM	SYNC (CTS--MIN=2)	CONTROL CHAR
CONTROL CHARACTER	SYNC (MAX=CTS OPTION)	DATA
DATA	SOM (BUFFER)	EOM
EOM	CONTROL CHAR (BUFFER)	MPC
MPC	DATA (BUFFER)	
EOB	EOM (BUFFER)	
	MPC (BUFFER)	

WTS OUTPUT BUFFER	LINE SEQUENCE & SOURCE	WTS INPUT
CONTROL CHARACTER	SYNC (WTS--MIN=2)	CONTROL CHAR
DATA	SYNC (MAX=WTS OPTION)	DATA
	SOM (WTS)	
	CONTROL CHAR (BUFFER)	
	DATA (BUFFER)	
	EOM (WTS)	
	MPC (WTS)	

THE CONTROL CHARACTER INDICATES BOTH THE OPERATION TO BE PERFORMED AND THE SUCCESS OF A PRECEEDING TRANSMISSION. THE TOGGLING OF THE LOW ORDER BIT POSITION IS USED TO INDICATE A SUCCESSFUL OPERATION WHILE A RETRANSMISSION IS INDICATED BY THE RECEIPT OF A CONTROL CHARACTER IDENTICAL TO THAT OF A PRECEEDING TRANSMISSION.

THE ACKNOWLEDGE MESSAGE IS IN THE FORMAT PREVIOUSLY DEFINED WITH ZERO DATA CHARACTERS. THIS DICTATES THAT ALL MESSAGES EXCEPT ACKNOWLEDGE MUST BE TWO CHARACTERS (ONE CONTROL CHARACTER PLUS ONE DATA CHARACTER) OR MORE IN LENGTH.

EACH MESSAGE TRANSFERRED BY SECOND LEVEL MUST BE CONTAINED WITHIN A SINGLE BUFFER. COMPLETE OUTPUT MESSAGES MAY BE CHAINED, IN WHICH CASE A REQUEST TO CMSA$ IS NECESSARY ONLY TO START THE CHAIN. SINCE FINAL CHECK ON INPUT MUST BE MADE BY THE WORKER BEFORE ANOTHER MESSAGE CAN BE RECEIVED, CHAINED INPUT IS NOT POSSIBLE AND EACH INPUT MESSAGE MUST BE REQUESTED.

A REQUEST TO CMSA$ CAUSES THE CM HANDLER TO SET UP OUTPUT OF THE MESSAGE ON THE LINE TERMINAL AND AFTER TRANSMISSION, TO SET UP INPUT TO RECEIVE THE ACKNOWLEDGE. IF THE ACKNOWLEDGE HAS ILLEGAL PARITY OR SPECIFICALLY REQUESTS RETRANSMISSION, THE SAME BUFFER WILL BE SENT AGAIN. IF THE ACKNOWLEDGE IS FAVORABLE, A STATUS CODE IS STORED IN S1 OF WORD 3 (SINGLE BUFFER MODE) OR IN S3 OF THE FIRST WORD OF THE BUFFER IF A POOL IS USED, AND THE NEXT MESSAGE IN THE CHAIN, IF ONE EXISTS, IS SENT. TRANSMISSION COMPLETION CAUSES INITIATION OF THE OUTPUT COMPLETION ROUTINE UNDER THE SAME CIRCUMSTANCES AS DESCRIBED FOR FIRST LEVEL SUPPORT. IF NO ACKNOWLEDGE IS

RECEIVED IN RESPONSE TO A TRANSMISSION OR IF THE ACKNOWLEDGE IS UNSUCCESSFUL AFTER 5 RETRIES, THE ON-SITE OPERATOR IS NOTIFIED BY ONE OF THE MESSAGES:

C/U SEND TIMEOUT
C/U SEND PARITY

THE TIME OUT RESULTS FROM NO ACKNOWLEDGEMENT RETURNED WITHIN 5 MINUTES. THIS TIME IS ALLOWED FOR REMOTE OPERATOR MANUAL INTERVENTION; CHANGING PAPER, ETC. IN ADDITION TO THE TYPEOUT, ALL FAILURES ARE WRITTEN IN THE SYSTEM LOG. AT TERMINATION OF THE COMMUNICATIONS LINKAGE(REFERENCE TO CMT$),THE NUMBER OF MESSAGES SENT AND RECEIVED AND THE NUMBER OF RETRANSMISSIONS REQUIRED IS ALSO LOGGED. THE OPERATOR CAN RESPOND TO THESE MESSAGES TO EITHER TRY TRANSMITTING AGAIN (LETTER A) OR DECLARE THE LINE DOWN (LETTER D). A DOWN RESPONSE RESULTS IN A STATUS CODE OF 02 TO BE RETURNED. NO FURTHER ACTION IS TAKEN ON QUEUED BUFFERS. IF A SUBSEQUENT REQUEST IS MADE WITH THE LINE DOWN, THE 02 STATUS CODE IS RETURNED WITHOUT TAKING ANY OTHER ACTION ON THE BUFFER.

THE CMSA$ REQUEST CAN ALSO BE USED TO SEND AN ACKNOWLEDGE MESSAGE AND SET UP TO RECEIVE A MESSAGE ON THE INPUT LINE TERMINAL ASSIGNED TO THE ADDRESSED LT TABLE. EXCEPT FOR CTMC CHANNELS, THE HARDWARE PROVIDES FOR CHARACTER PARITY CHECKING FOR THE INPUT MESSAGE. UPON DETECTION OF A PARITY ERROR, A REQUEST FOR RETRANSMISSION IS SENT. THUS THE MESSAGE HAS PASSED PARITY (EXCEPT ON CTMC) WHEN THE WORKER PROGRAM RECEIVES THE MESSAGE. THE WORKER PROGRAM IS EXPECTED TO VERIFY MESSAGE PARITY FOR AN INPUT MESSAGE RECEIVED VIA THE CTMC SINCE THE WORKER PROGRAM WILL PROBABLY PERFORM SOME TYPE OF DATA PACKING ON THE INPUT CTMC MESSAGE WHICH IS THE MOST CONVENIENT POINT FROM A SYSTEM STANDPOINT TO PERFORM THE MESSAGE PARITY VERIFICATION. UPON COMPLETION OF INPUT, THE INPUT COMPLETION ROUTINE IS ACTIVATED, DEPENDENT UPON THE INPUT USAGE FIELD VALUE, AND THE INPUT LINE TERMINAL IS TURNED OFF UNTIL THE NEXT REQUEST. THE USER PROGRAM MAY SUBJECT THE INPUT MESSAGE TO WHATEVER CHECKS DESIRED BEFORE ACKNOWLEDGING. REGARDLESS OF THE FACT THAT ONLY ONE MESSAGE IS RECEIVED AT A TIME, THE BUFFER POOL TECHNIQUE CAN BE USED FOR INPUT AS WELL AS OUTPUT.

SECOND LEVEL OPERATION IS PROVIDED IN EITHER HALF OR FULL DUPLEX OPERATION. FOR FULL DUPLEX OPERATION IT IS NECESSARY THAT THE REMOTE SUBSYSTEM RETURNS THE ACKNOWLEDGE MESSAGE AS A SEPARATE ENTITY.

10.11.4. ERROR CODES

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMSA$ ARE AS FOLLOWS:

21 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED FOR INPUT.

23 OUTPUT CHARACTER COUNT SPECIFIED OR INPUT CHARACTER COUNT SPECIFIED IS EQUAL TO ZERO OR EXCEEDS THE SYSTEM MAXIMUM.

25 INPUT CHARACTER COUNT SPECIFIED FOR A WTS SUBSYSTEM IS NOT AN INTEGRAL MULTIPLE OF.SIX CHARACTERS.

26 INPUT BUFFER SPECIFIED IS NOT WITHIN THE BOUNDS OF THE USER'S DATA-BANK.

31 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED FOR OUTPUT.

34 OUTPUT BUFFER SPECIFIED IS NOT WITHIN THE BOUNDS OF THE USER'S DATA-BANK.

42 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

46 LINE TERMINAL TABLE ADDRESS DOES NOT SPECIFY A LINE TERMI-

NAL TABLE CURRENTLY INITIALIZED, OR THE INTERNAL GROUP NAME DOES NOT SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED.

50 INPUT BUFFER SPECIFIED IS IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

51 OUTPUT BUFFER SPECIFIED IS IN THE SAME STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS.

10.12. HANG-UP

THE EXECUTIVE REQUEST CMH$ CAUSES RELEASE OF THE CURRENT REMOTE CONNECTION. AT THE TIME THE HANG-UP REQUEST IS MADE, THE WORKER PROGRAM SHOULD INSURE THAT OUTPUT HAS BEEN COMPLETED AND ANY INPUT WHICH MAY OCCUR IS OF NO CONCERN. THE HANGUP REQUEST DISREGARDS THE CURRENT LINE ACTIVITY AND ISSUES A REMOTE RELEASE TO THE DIAL AND INPUT CLTS, ANY FURTHER ACTIVITY AFTER THE HANGUP MUST BE PRECEDED BY A DIAL REQUEST TO ACTIVATE THE TERMINAL.

IF NO AUTOMATIC DIALING EXIST FOR A LT GROUP, A HANGUP REQUEST WILL CAUSE THE FOLLOWING MESSAGE TO BE DISPLAYED ON THE CONSOLE:

```
HANGUP    CC/UU
```

NO RESPONSE IS REQUIRED BY THE OPERATOR FOR THIS MESSAGE. A HANGUP REQUEST FOR A LEASED-LINE LT GROUP IS IGNDRED EXCEPT CAUSING QUEUED OUTPUT OR SUBSEQUENT INPUT TO BE IGNORED.

WHEN AN EXECUTIVE REQUEST IS MADE TO CMH$ THE USER MUST BE IN THE REAL-TIME MODE AND REGISTER A0 MUST CONTAIN THE ADDRESS OF THE LINE TERMINAL TABLE. THE CALLING SEQUENCE IS AS FOLLOWS:

```
L,U       A0,LTT
ER        CMH$
```

10.13. COMMUNICATION TERMINATION

THE FUNCTION OF THE CMT$ EXECUTIVE REQUEST IS TO DEACTIVATE THE INPUT AND/OR OUTPUT TERMINALS AND PERFORM VARIOUS HOUSEKEEPING ASSOCIATED WITH A LT GROUP. THE REQUEST WILL CAUSE A RELEASE FUNCTION TO BE SENT TO THE INPUT CLT ASSOCIATED WITH THE LTT. THE ASSIGNMENT FOR THE DEVICE IS NOT RELEASED FROM THE USER'S PROGRAM AND CAN BE REINITIALIZED VIA A CMS$ REQUEST. A COMMUNICATION DEVICE IS NOT RELEASED UNTIL THE DEVICE IS RELEASED VIA A @FREE OR WHEN THE PROGRAM TERMINATES.

10.13.1. TERMINATION ERRORS

OCTAL ERROR CODES AND THE ERROR CONDITIONS WHICH WILL BE DETECTED WHEN AN EXECUTIVE REQUEST IS MADE TO CMT$ ARE AS FOLLOWS:

16 ADDRESS OF A LINE TERMINAL TABLE DOES NOT SPECIFY A LINE TERMINAL TABLE CURRENTLY INITIALIZED, OR THE INTERNAL GROUP NAME DOES NOT SPECIFY THE CLT GROUP FOR WHICH THE LINE TERMINAL TABLE WAS INITIALIZED.

17 USER PROGRAM MAKING THE EXECUTIVE REQUEST IS NOT IN THE REAL-TIME MODE.

10.14. ROUTE$

THE 1108 COMMUNICATION HANDLER PROVIDES THE USER THE OPPORTUNITY TO DYNAMICALLY ALTER THE PRIMARY PATHS OF COMMUNICATION LINE TERMINAL GROUPS(LTG) BY INITIATING THE EXECUTIVE REQUEST ROUTE$.

EACH PRIMARY LTG IS DEFINED AT SYSTEM GENERATION TIME AND PROVIDES THE NECESSARY INFORMATION FOR THE COMMUNICATION HANDLER AND FACILITY INVENTORY TO MAKE ASSIGNMENTS VIA THE ASG CONTROL CARD. THE PRIMARY LTG MAY DEFINE UP TO THREE TERMINALS-INPUT, OUTPUT, AND/OR DIAL. THE PRIMARY LTG PARAMETERS ARE MAINTAINED IN THE ELEMENT TODRUM FROM 21 PARAMETERS AS SPECIFIED IN CHAPTER 18 OF THE PROGRAMMERS REFERENCE MANUAL.

AS THE PRIMARY LTG WILL NORMALLY DEFINES THREE CLT'S AS A GROUP, INPUT, OUTPUT, AND DIAL, AN AN ALTERNATE LTG CONFIGURATION MAY DEFINE ONE, TWO, OR THREE CLT'S. AN ALTERNATE LTG MAY BE ANOTHER ASSIGNABLE PRIMARY LTG OR UNASSIGNABLE EXCEPT BY THE ROUTE$ EXECUTIVE REQUEST. IF A PRIMARY LTG IS ROUTED TO ANOTHER PRIMARY LTG, BOTH INPUT AND OUTPUT CLT'S MUST BE ROUTED OR THE FIRST PRIMARY WILL REMAIN IN AN ASSIGNED STATE AND CAN NOT BE ASSIGNED AGAIN VIA EITHER AN ASG CONTROL CARD OR A ROUTE$ REQUEST. ONCE A PRIMARY LTG HAS BEEN ROUTED, ALL DIAL AND/OR HANG UP OPERATIONS WILL BE INITIATED USING THE NEW LTG. HOWEVER, IF THE PRIMARY LTG WAS ONLY PARTIALLY ROUTED, INPUT OR OUTPUT, BUT NOT BOTH, THE COMMUNICATION HANDLER WILL PROVIDE DIALING SPECIFIED BY THE OUTPUT CLT ALTERNATE ONLY. IT WILL BE THE RESPONSIBILITY OF THE USER TO PERFORM MANUAL DIALING FOR THE LTG NOT DIALED BY THE HANDLER. IF BOTH INPUT AND OUTPUT CLTS OF A PRIMARY LTG ARE ROUTED, THE PRIMARY LTG WILL BE HUNG UP BY THE HANDLER.

10.14.1. ROUTE$ CALLING SEQUENCE

THE FOLLOWING DEFINES THE FORMAT FOR THE EXECUTIVE REQUEST ROUTE$:

```
          L          A0,(MODE,LTADDR)
          L          A1,POINTER

     WHERE ;
               MODE=1-REQUEST FOR INPUT ALTERNATE CLT
                    2-REQUEST FOR OUTPUT ALTERNATE CLT.
                    3-REQUEST FOR BOTH INPUT AND OUTPUT CLTS.
             LTADDR=USER'S LINE TERMINAL (LT) TABLE ADDRESS.
            POINTER=SPECIFY AN ALTERNATE OF THE
                    PRIMARY LTG AS DEFINED VIA
                    THE ACTMC CONFIGURATION
                    CARD, WHERE 1,2,3,...,N, SPECIFY
                    THE LOGICAL ALTERNATE.
```

10.14.2. ROUTING PROCEDURES

BEFORE A ROUTE$ REQUEST CAN BE MADE TO THE COMMUNICATION HANDLER, THE USER'S LT TABLE MUST BE INITIALIZED USING THE EXECUTIVE REQUEST CMS$. ONCE THE LT TABLE HAS BEEN INITIALIZED, THE ROUTE$ REQUEST MAY BE REFERENCED AS FREQUENTLY AS DESIRED. ONCE A PRIMARY LTG HAS BEEN ROUTED, THE PRIMARY LTG IS NOT AVAILABLE FOR THIS ASSIGNMENT. EITHER A FREE REQUEST MUST BE INITIATED OR THE PRIMARY LTG MAY BE AN ALTERNATE OF IT'S ALTERNATES AND THE ROUTE$ REQUEST MAY BE INITIATED TO REESTABLISH THE ORIGINAL ASSIGNMENT. IF THE PRIMARY LTG BEING ROUTED HAS AN IDLE/LINE MONITOR STATE AND BOTH INPUT AND OUTPUT ARE ROUTED THE PRIMARY LTG WILL BE REESTABLISHED IN THE IDLE/LINE MONITOR STATE.

10.14.3. ROUTE$ ERROR CODES

THE FOLLOWING ERROR CODES WILL BE RETURNED IN THE ROUTE$ WILL SUPERCEDE THOSE SPECIFIED IN THE 1108 PROGRAMMERS REFERENCE MANUAL. THE ROUTE$ ERROR CODES WILL BE

RETURNED IN THE USER'S ERROR CONTINGENCY PACKET IN S2 WITH S1 SET EQUAL TO 06 WHEN AN ERROR IS DETECTED BY THE COMMUNICATION HANDLER.

20 A ROUTE$ REQUEST WITH AN INVALID PROGRAM TYPE. PROGRAM TYPE MUST BE REAL TIME.
21 A ROUTE$ REQUEST WITH AN INVALID LT LOCATION.
22 A ROUTE$ REQUEST WITH NO FACILITIES ASSIGNED FOR THE SPECIFIED LT.
23 A ROUTE$ REQUEST AND USERS LT HAS NOT BEEN INITIALIZED. A CMS$ REQUEST MUST BE INITIATE BEFORE A ROUTE$ REQUEST WILL BE HONORED.
24 A ROUTE$ REQUEST WITH A MODE CODE GREATER THAN 3 OR EQUAL TO ZERO.
25 A ROUTE$ REQUEST WITH AN INVALID POINTER WHEN REQUESTING AN ALTERNATE PATH. THE POINTER SPECIFIED IN A1 IS GREATER THAN THE NUMBER OF ALTERNATES SPECIFIED AT SYSTEMS GERERATION TIME OR ZERO.
26 A ROUTE$ REQUEST REQUESTING AN OUTPUT ALTERNATE, BUT OUTPUT NOT PERMITTED FOR THE ALTERNATE.
27 A ROUTE$ REQUEST REQUESTING AN INPUT ALTERNATE, BUT INPUT NOT PERMITTED FOR THE ALTERNATE.
30 A ROUTE$ REQUEST FOR AN ALTERNATE PATH ALREADY ASSIGNED.
31 A ROUTE$ REQUEST WITH ALTERNATE DRUM PACKET ADDRESS SPECIFIED AT SYSTEMS GENERATION TIME EQUAL TO ZERO. INDICATES AN ERROR IN SYSTEM GENERATION PARAMETER ALTPKT.
32 A ROUTE$ REQUEST WITH ALTERNATE'S DRUM ADDRESS IN DRUM PACKET EQUAL ZERO.

10.15. COMPLETION ACTIVITIES

COMPLETION ACTIVITIES ARE INITIALIZED BY THE COMMUNICATION HANDLER WHEN THE EXECUTIVE REQUEST CMS$ IS REFERENCED. COMPLETION ACTIVITIES, NORMALLY REFERRED TO AS ESI ACTIVITIES, ARE ACTIVITIES THAT ARE GIVEN CONTROL ON THE OCCURANCE OF AN ESI INTERRUPT OR ISI INTERRUPT IF CTS OR WTS SUBSYSTEMS. THE USERS CAN SPECIFY AN ESI ACTIVITY FOR EACH CLT, INPUT AND/OR OUTPUT, BY SPECIFYING IN THE LTT A USEAGE CODE AND A COMPLETION LOCATION FOR EACH RESPECTIVE MODE OF OPERATION. AN ESI ACTIVITY IS CONTROLLED BY THE CM VIA A 4 WORD PACKET, REFERRED TO AS ESI CONTROL PACKET, WHICH IS CREATED IN THE USER'S PCT. WHEN AN ESI ACTIVITY IS GIVEN CONTROL THE USER'S A0 WILL CONTAIN THE RESPECTIVELY LTT ADDRESS AND A1 WILL CONTAIN THE USER'S BUFFER LOCATION IF POOL MODE OPERATION OR DATA AREA IF SINGLE MODE OPERATION. THE COMPLETE SET OF USER'S REGISTERS ARE AVAILABLE IN AN ESI ACTIVITY, BUT ARE NEITHER PASSED OR RESTORED ONCE THE ACTIVITY HAS RELEASED CONTROL; NOR ARE THEY PASSED BETWEEN REAL-TIME AND ESI ACTIVITIES.

10.15.1. ESI ER REQUESTS

THE ESI ACTIVITY MAY RELEASE CONTROL VIA 4 EXECUTIVE REQUESTS: EXIT$, ACT$, CADD$, AND ADACT$. REFERENCE TO ANY OTHER REQUEST WILL RESULT IN AN ESI CONTINGENCY CONDITION. THE FOLLOWING DISCUSSESS THE APPLICATION AND REQUIRED FORMATS OF THE ABOVE REQUESTS.

10.15.1.1. ESI EXIT$

THE EXECUTIVE REQUEST EXIT$ FROM AN ESI ACTIVITY PROVIDES THE NORMAL EXIT$ FOR THAT COMPLETION ACTIVITY.

10.15.1.2. ESI ACT$

THE EXECUTIVE REQUEST ACT$ PROVIDES AN EXIT FROM AN ESI ACTIVITY AND ACTIVATES AN

ACTIVITY PREVIOUSLY NAMED USING THE EXECUTIVE REQUEST NAME$ IN THE REAL TIME PROGRAM. THE CALLING SEQUENCE OF THE ACT$ REQUEST IS AS FOLLOWS:

```
                    L     A0,NAME
                    ER    ACT$
```

WHERE;
WHERE NAME IS THE CONTENTS OF A0 RETURNED FROM THE EXECUTIVE REQUEST NAME$.

10.15.1.3. ESI CADD$

THE EXECUTIVE REQUEST CADD$ PROVIDES AN EXIT FROM THE ESI ACTIVITY AND RELEASE OF THE BUFFERS IN THE 2 WORD PACKET SPECIFIED BY THE USER'S A0, THE CALLING SEQUENCE IS AS FOLLOWS:

```
                   L      A0,PKTADDR
                   ER     CADD$
```

WHERE;

PKTADD IS THE LOCATION OF A TWO WORD PACKET IN THE FORMAT SPECIFIED IN THE PROGRAMMERS REFERENCE MANUAL.

10.15.1.4. ESI ADACT$

THE EXECUTIVE REQUEST ADACT$ PROVIDES AN EXIT FROM AN ESI ACTIVITY, RELEASES THE BUFFERS PROVIDED IN THE TWO WORD PACKET SPECIFIED BY THE USER,S A0 AND ACTIVATES THE ACTIVITY SPECIFIED BY THE NAME IN THE USER'S A1 THE CALLING SEQUENCE OF THE ADACT$ REQUEST IS AS FOLLOWS:

```
              L      A1,NAME
              L      A0,PKTADDR
              ER     ADACT$
```

WHERE;

NAME-DEFINES THE NAME PASSED IN A0 FROM THE NAME$ EXECUTIVE REQUEST.

PKTADDR-DEFINES THE ADDRESS OF A TWO WORD CADD$ PACKET IN THE FORMAT SPECIFIED IN THE PROGRAMMER'S REFERENCE.

10.16. IDLE LINE MONITOR

AT SYSTEM GENERATION THE INPUT LINE TERMINAL DEVICES CAN BE GIVEN AN UNASSIGNED STATUS OF EITHER OFF OR STANDBY. THE STANDBY STATUS CAUSES INPUT TO BE ENABLED ON THE DEVICES WHEN NOT ASSIGNED TO A PROGRAM. UPON RECEIPT OF A PARTICULAR IDENTIFYING CHARACTER STRING, EITHER AN APPROPRIATE SYMBIONT OR A NAMED RUN IS INITIATED. THE RUN FILE MUST ALREADY HAVE BEEN READ INTO THE OPERATING ENVIRONMENT. IF UNATTENDED ANSWERING IS PROVIDED ON THE INPUT DEVICE AND THE SUBSYSTEM IS CAPABLE OF CAUSING EXTERNAL INTERRUPTS, THEN NO INPUT BUFFER IS SET UP UNTIL THE OCCURRENCE OF A DIAL SUCCESSFUL INTERRUPT. CORE STORAGE REQUIREMENTS FOR THE IDLE LINE MONITOR FEATURE ARE APPROXIMATELY 85 PERMANENT LOCATIONS. THESE LOCATIONS HOLD SUCH THINGS AS INTERNAL TABLES, SWITCH LIST ENTRIES, AND ACTIVITY SAVE AREAS PLUS EIGHT LOCATIONS FOR EACH INPUT LINE ACTUALLY IN A STANDBY CONDITION.

10.17. TIMING CONSIDERATIONS

WITHIN THE REALM OF COMMUNICATIONS EQUIPMENT HANDLING THERE ARE THREE LEVELS OF ACTIVITY WHICH MUST BE TAKEN INTO ACCOUNT TO DETERMINE THE COMMUNICATION ACTIVITY WHICH CAN BE ALLOWED IN CONJUNCTION WITH OTHER I/O ACTIVITY, REAL-TIME CLOCK ACTIVITY

AND CPU USAGE. THESE LEVELS ARE:

1. INTERRUPT RESPONSE
2. BUFFER PROCESSING
3. INFORMATION ANALYSIS

10.17.1. INTERRUPT LEVEL

THE FIRST LEVEL, INTERRUPT RESPONSE, OCCURS WITHIN THE CM HANDLER AND CONSISTS OF SETTING UP THE NEXT BUFFER FOR THE INTERRUPTING LINE TERMINAL, QUEUING THE INTERRUPT, AND REINITIATING THE INPUT OR OUTPUT MODE. THE NECESSARY CRITERIA TO BE MET BY THE HANDLER IS, 1) INSURING THAT THE MODE BE RESET WITHIN THE CHARACTER AVAILABILITY OF THE FASTEST LINE TERMINAL DEVICE ON THE CHANNEL(196 MICROSECONDS FOR 40,800 CPS AT 8 BIT CHARACTERS), AND 2) INSURING THAT EACH INTERRUPT IS PROCESSED IN A TIME INTERVAL SUCH THAT ALL ACTIVE LINES COULD BE READY TO INTERRUPT AT THE SAME TIME AND NO INFORMATION IS LOST ON ANY LINE. IF SINGLE BUFFER MODE IS EMPLOYED, THEN THE SECOND CASE IS OF NO CONCERN FOR THE GIVEN LINE(THE INTERRUPT TERMINATES ACTIVITY ON THAT LINE)BUT MUST BE CONSIDERED FOR LOWER PRIORITY MULTIPLE BUFFER LINES. THE CONFIGURATION SHOULD BE ARRANGED WITH THE HIGHEST SPEED LINE TERMINALS WHICH MAY BE USED IN THE MULTIPLE BUFFER MODE IN THE HIGHEST PRIORITY INTERRUPT POSITION (LOWEST ESI CHANNEL NUMBER AND HIGHEST PRIORITY MULTIPLEXOR POSITION). THEN TO INSURE NO LOSS OF INFORMATION FOR EACH LINE TERMINAL, HIGHER PRIORITY INTERRUPTS PLUS THE SINGLE INTERRUPT FOR THIS LINE TERMINAL MUST BE HANDLED WITHIN THE CHARACTER AVAILABILITY TIME OF THAT LINE. THE COUNT OF HIGHER PRIORITY INTERRUPTS MUST INCLUDE:

1. ONE FOR EACH ESI EXTERNAL INTERRUPT WHICH CAN OCCUR IN THE CHARACTER AVAILABILITY TIME.
2. ONE FOR EACH HALF DUPLEX I/O LINE TERMINAL PAIR OF HIGHER PRIORITY
3. ONE FOR EACH SIMPLEX LINE OF HIGHER PRIORITY.
4. TWO FOR EACH HIGHER PRIORITY FULL DUPLEX PAIR
5. ONE FOR THE LINE OF CONCERN

IF ANY BUFFER OF A HIGHER PRIORITY LINE CAN FILL ONCE OR MORE WITHIN THE CHARACTER AVAILABILITY TIME, ONE MUST BE ADDED FOR EACH OCCURRENCE. THE CHARACTER AVAILABILITY DIVIDED BY 40 MICROSECONDS SHOULD BE GREATER THEN THE NUMBER COMPUTED. THIS VALUE TAKES INTO ACCOUNT THE INTERRUPT PROCESSING INSTRUCTIONS PLUS DATA TRANSFERS. FOR INSTANCE WITH THE CONFIGURATION OF:

10 FULL DUPLEX 1004'S AT 4800 BPS(CA=1.25MS)
250 HALF DUPLEX KSR 35 TELETYPE*MACHINES AT 100 WPM(CA=20MS)

THE INTERRUPT COUNT FOR THE LOWEST PRIORITY 1004 IS 19 (2 FOR EACH OF THE 9 HIGHER PRIORITY FULL DUPLEX PAIR PLUS 1 FOR THE LINE OF CONCERN) AND THE LOWEST PRIORITY TELETYPE* MACHINE IS 270 WHICH ARE LESS THAN THE LIMITS OF 1250/40=31 AND 20000/40=500 RESPECTIVELY, HENCE THERE WOULD BE NO LOSS OF INFORMATION IF THE ENTIRE SYSTEM WAS OPERATED SIMULTANEOUSLY IN A MULTIPLE BUFFER MODE(REMOTE 1004 UNDER EXEC CONTROL WILL NORMALLY BE A SINGLE BUFFER OPERATION). IT SHOULD BE NOTED THAT CTS AND WTS EQUIPMENT IS OPERATED ON ISI MODE CHANNELS AND HENCE ARE LOWER INTERRUPT PRIORITY THAN ALL ESI CHANNELS. ALSO THE WTS INCORPORATES CHARACTER ASSEMBLY INTO FULL WORDS AND HENCE MULTIPLIES THE CHARACTER AVAILABILITY TIME.

10.17.2. BUFFER LEVEL

BUFFER PROCESSING IS A WORKER FUNCTION TO BE PERFORMED FROM THE INPUT AND OUTPUT COMPLETION ACTIVITIES. THESE ACTIVITIES ARE GIVEN PRIORITY, WITHIN THE DISPATCHING ALGORITHM IMMEDIATELY BELOW INTERRUPT HANDLING. THE MOST CRITICAL BUFFER PROCESSING ROUTINES ARE EXPECTED TO BE THOSE THAT HANDLE MULTIPLE BUFFER MODE LINE TERMINALS. THESE MUST OPERATE WITHIN THE CONSTRAINT THAT EACH BUFFER MUST BE PROCESSED WITHIN THE TIME IT TAKES TO FILL THE NEXT BUFFER IN THE CHAIN(UNLESS THE POOL IS OF SUFFICIENT LENGTH TO CONTAIN AN ENTIRE MESSAGE FOR EACH TERMINAL CONCERNED), AND EACH BUFFER PROCESSING ROUTINE MUST SHARE THE CPU SUCH THAT ALL ROUTINES HAVE A CHANCE TO

*TRADEMARK OF TELETYPE CORPORATION

PROCESS THEIR CORRESPONDING BUFFERS. THE NEED TO SHARE THE CPU AMONG THE PROCESSING ROUTINES DICTATES THAT A TIME INCREMENT MUST BE CHOSEN SUCH THAT SWITCHING FROM ONE COMPLETION TO ANOTHER IS BASED ON THE REAL-TIME CLOCK. THIS TIME INCREMENT IS SET BY A SYSTEM GENERATION PARAMETER(IF OMITTED NO MAXIMUM EXISTS). THE BASIS FOR DETERMINING THIS INTERVAL IS:

> EACH BUFFER MUST BE PROCESSED IN THE TIME INTERVAL, TD, TO FILL THE LARGEST (IN TERMS OF TIME) BUFFER, TMAX, DIVIDED BY THE NUMBER OF BUFFERS WHICH CAN FILL IN THE MAXIMUM INTERVAL. A GIVEN BUFFER CAN FILL TMAX/TI TIMES WHERE TI IS THE TIME REQUIRED TO FILL A BUFFER (NUMBER OF CHARACTERS TIMES THE CHARACTER TRANSFER RATE). TMAX IS THE LARGEST VALUE OF TI IN THE SYSTEM. HENCE THE TIME LIMIT, TD, IS GIVEN BY THE FORMULA:

$$TD=\frac{TMAX}{\frac{TMAX}{T1}+\frac{TMAX}{T2}+\frac{TMAX}{T3}+...+\frac{TMAX}{TN}}$$

$$=\frac{1}{\frac{1}{T1}+\frac{1}{T2}+\frac{1}{T3}+...+\frac{1}{TN}}$$

$$=\frac{T1T2T3T4...TN}{T1T3...TN+T1T2T4...TN+...+T2T3...TN}$$

THE UPPER AND LOWER LIMITS OF TD ARE DETERMINED BY T1=T2=T3= ...TN=TMIN AND TI=TMIN WITH ALL OTHER T1...TN CONSIDERABLY LARGER WHICH GIVES:

$$\frac{TMIN}{N} \leq TD < TMIN$$

THE APPROXIMATION IS REASONABLE THAT ALL DEVICES WILL BE BUFFERED SUCH THAT THE TIME TO FILL EACH IS NEARLY EQUAL, HENCE TD=TMIN/N CAN BE USED. THE VALUE N INCLUDES:

1 FOR EACH SIMPLEX INPUT OR OUTPUT
1 FOR EACH HALF DUPLEX PAIR
2 FOR EACH FULL DUPLEX PAIR

WHERE ONLY MULTIPLE BUFFER MODE LINES ARE CONSIDERED. TD CAN NOW BE CHANGED AS BUFFERS ARE LENGTHENED OR SHORTENED. IN THE INTERVAL TD DURING WHICH THE BUFFERS MUST BE PROCESSED, THE ONLY TIME WHICH THE COMPLETION ACTIVITY WILL NOT HAVE CONTROL IS DURING CYCLES TAKEN FOR DATA TRANSFERS AND TIME NEEDED TO QUEUE INTERRUPTS. IN THE WORST CASE ALL ACTIVE COMMUNICATIONS LINES AND ALL STANDARD LINES MAY INTERRUPT. HENCE WITH:

TD=10 MILLISECONDS
50 ACTIVE LINE LINES AT 2400 BPS AT 7 BITS/CHAR
1 FH432(240000 WORDS/SEC)
1 FASTRAND CHANNEL(25,150 WPS)
1 VIII C TAPE CHANNEL(16,000 WPS)

IN THE TD INTERVAL FOR THE EXAMPLE GIVEN ABOVE, THERE COULD BE (240+25.15+16)(1000)(0.01)=2811 DATA TRANSFERS AT 0.75 MICROSECONDS PER TRANSFER=2.1 MILLISECONDS. FOR HIGH SPEED CHANNELS, 50(10 MS)(1.5 MICROSEC)/2.91 MS = 0.25 MS FOR COMMUNICATIONS LINE TERMINAL DATA TRANSFERS AND (53 INTERRUPTS)(25 MICROSECONDS PER INTERRUPT)=1.33 MILLISECONDS FOR INTERRUPTS WHICH LEAVES 10-(2.1+0.25+1.33)=6.32 MILLISECONDS AS THE MINIMUM TIME THE PROGRAM CAN COUNT ON FOR EXECUTING INSTRUCTIONS. SINCE ONLY MULTIPLE BUFFER MODE IS CRITICAL, THE CM HANDLER WILL ENTER INTERRUPT

COMPLETION ACTIVITIES INTO ONE OF TWO QUEUES CONTROLLED BY THE DISPATCHER:

MULTIPLE BUFFER COMPLETION ACTIVITIES
SINGLE BUFFER COMPLETION ACTIVITIES

THE NATURE OF MULTIPLE BUFFER MODE PROCESSING IS SUCH THAT IT IS MORE TIME CRITICAL THAN SINGLE BUFFER MODE PROCESSING AND WILL BE HANDLED BY THE DISPATCHER IN A MANNER SUCH THAT SINGLE BUFFER MODE ACTIVITIES WILL BE GIVEN CONTROL ONLY WHEN THE MULTIPLE BUFFER MODE ACTIVITY QUEUE HAS BEEN EXHAUSTED.

10.17.3. ANALYSIS LEVEL

THE THIRD LEVEL OF ACTIVITY TO BE CONSIDERED, INFORMATION ANALYSIS, IS ALSO A WORKER FUNCTION. THIS ACTIVITY IS NORMALLY EXPECTED TO BE REAL-TIME WITH A USER DETERMINED RESPONSE TIME. THIS IS HANDLED BY THE EXECUTIVE THROUGH THE STANDARD DISPATCHING ALGORITHM.

10.17.4. ERROR CODES FOR LT CONTINGENCIES

EACH WORKER PROGRAM TO WHICH COMMUNICATION DEVICES ARE ASSIGNED IS EXPECTED TO HAVE AN ERROR HANDLING ROUTINE REGISTERED WITH THE EXECUTIVE SYSTEM. THIS ERROR HANDLING ROUTINE WILL BE THE COMMUNICATIONS HANDLER'S MEANS OF NOTIFYING THE USER OF SYSTEM AND LT GROUP ERROR CONDITIONS WHICH DO NOT APPLY TO THE TRANSMISSION OR RECEPTION OF A PARTICULAR BUFFER. THE WORKER PROGRAM WILL BE NOTIFIED IN THE MANNER AND FORMAT AS DEFINED UNDER PROGRAM CONTINGENCY 'ERR MODE' CONDITION. FOLLOWING IS A LIST OF THE POSSIBLE ERROR CODES AND THEIR MEANING WHEN THAT CODE IS STORED IN THE S2 PORTION OF THE FIRST WORD OF THE ERROR ROUTINE.

CODE	MEANING	ACTION
1	AN ER CMS$ REFERENCE HAS BEEN MADE FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEAD-LINE BATCH, OR BATCH ACTIVITIES.	
2	AN ER CMS$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 WHICH IS BEYOND THE LIMITS OF THE PROGRAM.	
3	THE INITIALIZATION REQUEST SPECIFIES A LT TABLE WHICH IS CURRENTLY IN USE OR HAS BEEN INITIALIZED BUT HAS NEVER BEEN TERMINATED.	
5	AN INVALID VALUE WAS SPECIFIED FOR THE INPUT COMPLETION ACTIVITY USAGE CODE WITHIN THE REQUESTING PACKET FOR AN ER CMS$.	
6	THE FILE NAME FOR AN INITIALIZATION REQUEST DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE (PCT).	THIS PROBABLY INDICATES THE FILE HAS NOT BEEN ASSIGNED.
7	THE END OF INPUT ACTIVITY CODE IN THE PACKET FOR AN INITIALIZATION REQUEST IS NOT VALID.	
10	AN ER CMI$ OR CMO$ REFERENCE AND LTG	

HAS BEEN TERMINATED DUE TO ESI CONTINGENCY ERROR OR NO ESI CONTINGENCY SPECIFIED FOR AN ESI CONTINGENCY.

11 THE CONTROL GROUP ADDRESS SPECIFIED IN A PACKET TO BE USED FOR POOL MODE OF OPERATION IS INVALID.

12 POOL MODE OF OPERATION HAS BEEN SPECIFIED, BUT NO BUFFERS PRESENTLY EXIST IN THE POOL SPECIFIED AS THE ONE TO BE USED.

THE POOL MAY BE TEMPORARILY EXHAUSTED, TOO SMALL OF A POOL WAS ESTABLISHED, OR THE REAL-TIME PROGRAM MAY HAVE FAILED TO RETURN TO THE POOL THE BUFFERS WHICH HAD BEEN REMOVED PREVIOUSLY BY THE COMMUNICATIONS HANDLER. IF THE SITUATION CAN BE CORRECTED DYNAMICALLY, THE PACKET MUST BE RESUBMITTED.

13 THE ADDRESS WITHIN THE PACKET SPECIFIED AS THE INPUT COMPLETION ACTIVITY WAS NOT VALID.

14 THE OUTPUT USAGE CODE SPECIFIED WITHIN THE PACKET FOR AN INITIALIZATION REQUEST IS NOT VALID.

15 AN ER CMS$ REFERENCE SPECIFYING OUTPUT POOL MODE WITH INVALID POOL ID OR CLOSED MODE.

16 AN ER CMT$ REFERENCE HAS SPECIFIED A LT GROUP WHICH HAS BEEN TERMINATED PREVIOUSLY OR HAS NEVER BEEN INITIALIZED.

17 AN ER CMT$ REFERENCE HAS BEEN MADE FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.

20 AN ER CMT$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 BEYOND THE BOUNDS OF THE PROGRAM.

21 A REQUEST FOR AN INPUT OPERATION HAS SPECIFIED A FILE NAME FOR WHICH INPUT IS NOT PERMISSABLE (NO INPUT FACILITY FOR THE LT GROUP WAS DEFINED AT SYSTEMS GENERATION TIME).

22 AN INVALID VALUE WAS SPECIFIED FOR THE INPUT COMPLETION USAGE CODE WITHIN THE REQUESTING PACKET FOR AN ER CMI$.

23 THE NUMBER OF CHARACTERS SPECIFIED FOR A SINGLE MODE INPUT OR OUTPUT OPERATION EXCEEDS THE MAXIMUM PERMISSABLE VALUE ESTABLISHED AT SYSTEMS GENERATION TIME.

24 AN ADDRESS BEYOND THE BOUNDS OF THE PROGRAM WAS SPECIFIED IN H2 OF A0 FOR A CMI$, CMO$, OR CMSA$ REQUEST.

25 A NON-MULTIPLE OF 6 WAS SPECIFIED AS THE CHARACTER COUNT FOR AN INPUT OPERATION ON THE WTS SUBSYSTEM.

26 AN INPUT OPERATION HAS SPECIFIED A CHARACTER COUNT AND BUFFER STARTING ADDRESS SUCH THAT A PORTION OF THE INPUT BUFFER EXISTS BEYOND THE LIMITS OF THE PROGRAM.

27 AN INPUT OR OUTPUT OPERATION HAS SPECIFIED POOL MODE OF OPERATION FOR A FILE NAME ASSOCIATED WITH EQUIPMENT WHICH USES THE ISI METHOD OF BUFFERING. BUFFER SWAPPING FOR ISI DEVICES IS NOT PERMITTED BECAUSE OF THE DESIGN OF THE HARDWARE WHICH OPERATES ON DATA BY BLOCKS RATHER THAN AS A CONTINUOUS INPUT STREAM.

30 AN CMI$, CMO$, OR CMSA$ REQUEST AND SPECIFIED LTT HAS INVALID FILE NAME.

31 A REQUEST FOR AN OUTPUT OPERATION HAS SPECIFIED A FILE NAME FOR WHICH OUTPUT IS NOT PERMISSABLE (NO OUTPUT FACILITY FOR THE LT GROUP WAS DEFINED AT SYSTEMS GENERATION TIME).

32 AN INVALID VALUE WAS SPECIFIED FOR THE OUTPUT COMPLETION USAGE CODE WITHIN THE REQUESTING PACKET FOR AN ER CMO$.

34 AN OUTPUT OPERATION HAS SPECIFIED A CHARACTER COUNT AND A BUFFER STARTING ADDRESS SUCH THAT A PORTION OF THE OUTPUT BUFFER EXISTS BEYOND THE LIMITS OF THE PROGRAM.

36 POOL AREA IS ILLEGAL BECAUSE: (1) POOL SIZE EQUALS ZERO WORDS, (2) POOL AREA IS SPLIT BETWEEN USER PROGRAM'S D-BANK AND I-BANK, OR (3) POOL AREA IS BEYOND BOUNDS OF USER PROGRAM'S AREA.

37 THE CORE AREA SPECIFIED BY AN ER CPOOL$ IS LOCATED IN THE I-BANK. POOL BUFFERING USING THE I-BANK IS NOT PERMITTED SINCE A PROGRAM IS SUPPOSED TO BE SO POSITIONED IN CORE SUCH THAT THE HARDWARE MEMORY OVER-

THE REFERENCE TO CPOOL$ MUST SPECIFY A CORE AREA ASSOCIATED WITH ITS D-BANK.

	LAP FEATURE FOR ESI DATA TRANSFERS WILL BE UTILIZED IN ORDER TO REDUCE THE NUMBER OF CORE MEMORY ACCESSES FROM 3 TO 2 FOR EACH DATA TRANSFER.	
40	BUFFER SIZE IS ILLEGAL BECAUSE: (1) CHARACTER COUNT EQUALS ZERO OR (2) CHARACTER COUNT EXCEEDS SYSTEM'S SPECIFIED MAXIMUM.	
41	POOL AREA IS NOT LARGE ENOUGH FOR: (1) A MINIMUM OF ONE BUFFER FOR AN OPEN CHAIN OR (2) USER PROGRAM'S SPECIFIED NUMBER OF BUFFERS FOR A CLOSED CHAIN.	
42	AN ER TO THE COMMUNICATIONS HANDLER HAS BEEN EXECUTED FROM THE WRONG TYPE OF CODING. USE OF THE COMMUNICATIONS HANDLER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.	
43	THE SINGLE MODE OF BUFFERING HAS SPECIFIED A COMMUNICATIONS BUFFER WHICH IS IN THE I-BANK. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE I-BANK BECAUSE REAL-TIME PROGRAMS ARE PROPERLY POSITIONED IN CORE SO THAT THE D-BANK CAN UTILIZE THE MEMORY OVERLAP FEATURE FOR EACH CHARACTER TRANSFER.	
45	THE ADDRESS WITHIN THE PACKET SPECIFIED AS THE OUTPUT COMPLETION ACTIVITY WAS NOT VALID FOR AN ER CMS$.	
46	AN ER HAS BEEN MADE REQUESTING USE OF A LT GROUP BY THE COMMUNICATIONS HANDLER BEFORE THAT LT GROUP HAS BEEN INITIALIZED.	A LT GROUP MUST BE INITIALIZED VIA AN ER CMS$ BEFORE ANY INPUT OR OUTPUT CAN BE PERFORMED.
47	AN ER CMO$ HAS DESIGNATED THAT POOL MODE SHOULD BE USED FOR OUTPUT, BUT THE BUFFER POOL CONTROL WORD SPECIFIED IN THE LT TABLE IS INVALID.	
50	AN ER CMI$ FOR SINGLE BUFFER MODE HAS SPECIFIED AN INPUT BUFFER AREA WHICH CONFLICTS WITH THE MAIN STORAGE MODULE LOCATIONS OF THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED.	THE REAL-TIME PROGRAM MUST BE PROPERLY POSITIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.
51	AN ER CMO$ FOR SINGLE BUFFER MODE HAS SPECIFIED AN OUTPUT BUFFER AREA WHICH CONFLICTS WITH THE MAIN STOR-	THE REAL-TIME PROGRAM MUST BE PROPERLY POSI-

	AGE MODULE LOCATIONS OF THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED.	TIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.
52	THE CORE AREA SPECIFIED BY AN ER CPOOL$ IS LOCATED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED.	THE REFERENCE TO CPOOL$ MUST BE PROPERLY POSITIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.
53	AN ER CMS$ HAS SPECIFIED A LT TABLE WHICH EXISTS IN THE REAL-TIME PROGRAM'S I-BANK. REAL-TIME PROGRAMS ARE EXPECTED TO BE SOPHISTICATED ENOUGH TO UTILIZE THE HARDWARE FEATURE OF MAIN STORAGE OVERLAP BY BEING DIVIDED INTO APPROPRIATE I-BANK AND D-BANK PORTIONS.	THE LT TABLE MUST BE PLACED BY THE REAL-TIME PROGRAM IN ITS D-BANK.
54	A CMD$ OR CMH$ REFERENCE HAS SPECIFIED FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEAD-LINE BATCH, OR BATCH ACTIVITIES.	
55	AN ER CMD$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 WHICH IS BEYOND THE LIMITS OF THE PROGRAM.	
56	A CMD$ OR CMH$ REFERENCE HAS SPECIFIED A LT GROUP WHICH HAS BEEN TERMINATED PREVIOUSLY OR HAS NEVER BEEN INITIALIZED.	
57	THE ADDRESS WITHIN.THE PACKET SPECIFIED AS THE DIAL COMPLETION ACTIVITY ADDRESS WAS NOT VALID FOR AN ER CMD$.	
60	ESI CONTINGENCY ERROR FOR CONTINGENCY TYPES 1-5 FROM ESI COMPLETION ACTIVITY	
61	ESI CONTINGENCY ERROR FOR ACTS OR ADACT$ FROM ESI COMPLETION ACTIVITY.	
62	ESI CONTINGENCY ERROR FOR CADD$ OR ADACT$ FROM ESI COMPLETION ACTIVITY.	
63	ESI CONTINGENCY ERROR FOR ILLEGAL ER FROM ESI COMPLETION ACTIVITY.	
64	AN ER CMD$ REFERENCE HAS SPECIFIED A DIAL ACCESS CONTROL WORD WHOSE CHARACTER COUNT AND BUFFER STARTING ADDRESS ARE SUCH THAT A PORTION OF THE BUFFER AREA EXISTS BEYOND THE LIMITS OF THE PROGRAM.	
67	A CMD$ OR CMH$ CALL SPECIFYING A FILE NAME THIS SHOULD BE	

FOR WHICH NO DIALING OR HAND-UP CAPABILITY EXISTS.

CONSIDERED A NORMAL RESPONSE FOR A HANG-UP OPERATION ON A LEASED LINE.

65 AN ER CMT$ REFERENCE HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OR THE USER'S PROGRAM CONTROL TABLE.

THIS INDICATES THAT THE ER WAS NOT EXECUTED IN A PROPER SEQUENCE OR THE FILE HAS ALREADY BEEN RELEASED.

66 A CMD$ OR CMH$ REQUEST HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE.

THIS INDICATES THAT THE ER WAS NOT EXECUTED IN A PROPER SEQUENCE OR THE FILE HAS ALREADY BEEN RELEASED.

74 A CMS$ REFERENCE AND USER'S LT SPECIFIES NEITHER INPUT OR OUTPUT MODES.

76 A CMS$ REFERENCE WITH NO OUTPUT CPU PATH OR HAS BEEN DOWNED BY THE SYSTEM.

77 A CMS$ REFERENCE WITH NO INPUT CPU PATH OR HAS BEEN DOWNED BY THE SYSTEM.

THE FOLLOWING ERROR CODES WILL BE RETURNED TO THE USER'S ERROR PACKET IN S2 WITH S1 SET EQUAL TO 6.

3 A CMO$ REQUEST FOR OUTPUT POOL MODE AND USER BUFFER ADDRESS IS INVALID

4 A CMO$ REQUEST FOR OUTPUT POOL MODE AND BUFFER CHARACTER COUNT IS INVALID.

5 A CADD$ OR CGET$ REQUEST AND SPECIFIED BUFFER COUNT TO ADD OR GET IS ZERO.

6 A CADD$ OR CGET$ REQUEST WITH INVALID POOL ID OR CLOSED CHAIN POOL.

20 A ROUTE$ REQUEST WITH INVALID PROGRAM TYPE.

21 A ROUTE$ REQUEST HAS SPECIFIED AN ADDRESS IN H2 OF A0 WHICH IS BEYOND THE LIMITS OF THE PROGRAM.

22 A ROUTES REQUEST HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE.

23 A CGET$ REQUEST WITH SPECFIED BUFFER COUNT GREATER THAN NUMBER OF BUFFERS AVAILABLE

24 A ROUTE$ REQUEST WITH INVALID MODE PARAMETER.

25 A CADD$ REQUEST WITH INVALID BUFFER SPECIFIED.

26 A ROUTE$ REQUEST SPECIFYING OUTPUT, BUT OUTPUT NOT DEFINED FOR THE ALTERNATE ROUTE.

27 A ROUTE$ REQUEST SPECIFYING INPUT, BUT OUTPUT NOT DEFINED FOR THE ALTERNATE ROUTE.

30 A ROUTE$ REFERENCE REQUESTING AN ALTERNATE PATH WHICH HAS ALREADY BEEN

ASSIGNED.
31 A ROUTE$ REFERENCE WITH ALTERNATE DRUM LOCATION IN PRIMARY SITE TABLE ZERO. INDICATES IMPROPER SYSTEMS GENERATION.
32 A ROUTE$ REFERENCE AND ALTERNATE SITE TABLE LOCATION IS ZERO IN THE ALTERNATE DRUM PACKET. INDICATES IMPROPER SYSTEMS GENERATION.
33 A ROUTE$ REFERENCE HAS SPECIFIED A LT TABLE THAT HAS NOT BEEN INITIALIZED.
34 A ROUTE$ REFERENCE WITH INVALID POINTER PARAMETER.
35 A ROUTE$ REFERENCE FOR AN INPUT ALTERNATE WHICH HAS BEEN DOWNED OR NO CPU PATH IS AVAILABLE.
36 A ROUTE$ REFERENCE FOR AN OUTPUT ALTERNATE WHICH HAS BEEN DOWNED OR NO CPU PATH IS AVAILABLE.
11 A CMD$ REFERENCE WHILE A PREVIOUS DIAL REQUEST IS IN PROGRESS OR A DIAL REQUEST FOR A CTS OR WTS SUBSYSTEM.

10.18. REAL-TIME SYSTEM CONSIDERATION

10.18.1. PROGRAM LOCATION

THE NATURE OF A REAL-TIME PROGRAM CAUSES THE EXECUTIVE SYSTEM TO GIVE IT CONSIDERATIONS DIFFERENT FROM THOSE OF DEMAND AND BATCH RUNS. A REAL-TIME PROGRAM IS NEVER SWAPPED OUT OF CORE STORAGE NOR RELOCATED WITHIN CORE STORAGE BECAUSE IT MUST BE AVAILABLE AT ANY TIME FOR IMMEDIATE USE. THUS IT IS EXPECTED THAT THE USER WILL PROPERLY POSITON AND LOCATE ALL REAL-TIME PROGRAMS IN SUCH A MANNER SO AS TO MOST EFFICIENTLY UTILIZE THE SYSTEM GIVING CONSIDERATION TO SUCH THINGS AS PROGRAM CONTROL TABLE EXPANSION AND HARDWARE MEMORY OVERLAP.

10.18.2. EXECUTION

THE EXECUTIVE DOES NOT TIME SHARE (SLICE) BETWEEN REAL-TIME ACTIVITIES AT THE SAME PRIORITY LEVEL BUT PERMITS A REAL-TIME ACTIVITY TO CONTINUE, LIMITED TO WITHIN A REASONABLE LENGTH OF TIME, UNTIL IT VOLUNTARILY RELEASES CONTROL. THUS AT A REAL-TIME LEVEL THERE MAY BE ANY NUMBER OF ACTIVITES IN A CONDITION WAITING FOR CONTROL BUT ONLY ONE REAL-TIME ACTIVITY PER CPU ACTUALLY SUSPENDED BY THE EXEC. EXEC SUSPENDED ACTIVITIES AT A SINGLE PRIORITY LEVEL. FOR REAL-TIME PROGRAMS, THE EXECUTIVE WILL ALWAYS RETURN CONTROL TO THE SUSPENDED ROUTINE, AND THAT REAL-TIME ACTIVITY WILL BE PERMITTED TO CONTINUE UNTIL IT VOLUNTARILY RELEASES CONTROL BEFORE ANY OTHER ACTIVITES AT THAT SAME REAL-TIME LEVEL ARE GIVEN CONTROL. HOWEVER, ALL ACTIVITIES OF A REAL-TIME PROGRAM ARE TIMED WITH A MAXIMUM PERMISSABLE TIME VALUE ALLOWING THE SYSTEM TO DETECT ERRORS AND EXCESSIVE LOOPS WHICH MAY DEVELOP IN AN UNDEBUGGED PROGRAM.

10.18.3. BUFFER OPERATIONS

ALTHOUGH THE SYSTEM PROVIDES BOTH SINGLE BUFFER MODE AND POOL BUFFER MODE CAPABILITIES, IT IS EXPECTED THAT THE USER WILL EMPLOY THE METHOD MOST ADVANTAGEOUS FOR THE APPLICATION. THE SINGLE BUFFER MODE OF OPERATION IS THE MOST EFFICIENT METHOD AS THERE IS NO EXEC OVERHEAD INVOLVED WITH CONTROLS FOR THE POOL BECAUSE CONTROLS, IF ANY EXIST, ARE THE RESPONSIBILITY OF THE USER. THE SINGLE BUFFER MODE

WILL USE LESS CORE STORAGE PER BUFFER BECAUSE THE POOL MODE REQUIRES ADDITIONAL STORAGE AREA IN BOTH THE WORKER PROGRAM AND WITHIN THE EXEC FOR THE POOL CONTROL INFORMATION. THIS AMOUNTS TO THREE WORDS FOR EACH BUFFER IN THE POOL AS ONE WORD IS MAINTAINED WITH PARAMETER INFORMATION IN THE USER'S AREA AND TWO WORDS IN THE EXEC AREA ARE USED FOR LINKING AND QUEUEING PURPOSES BY THE COMMUNICATIONS HANDLER. THE TYPE OF TRANSMISSION IS PERHAPS THE BEST GUIDE AS TO WHICH SHOULD BE USED.

10.18.3.1. TRANSMISSION TYPES

TYPES OF TRANSMISSIONS MAY BE CLASSIFIED AS FIXED IN LENGTH, VARIABLE IN LENGTH, OR INDETERMINATE IN LENGTH. AN EXAMPLE OF A FIXED LENGTH TRANSMISSION WOULD BE THE USE OF A 1004 AS THE REMOTE TERMINAL OR A SIMILAR PRINTING DEVICE WHICH WOULD EMPLOY A PRINT LINE IMAGE OF, SAY, 80 OR 132 CHARACTERS. ALSO THE POLL MESSAGE (NOT TO BE CONFUSED WITH A POLL RESPONSE) FOR A POLLED NETWORK IS GENERALLY OF A FIXED LENGTH. THE SINGLE BUFFER MODE OF OPERATION SHOULD ALWAYS BE USED FOR FIXED LENGTH TRANSMISSIONS.

A VARIABLE LENGTH MESSAGE IS A TRANSMISSION WHOSE MAXIMUM LENGTH CAN BE PREDICTED. AN EXAMPLE OF A VARIABLE LENGTH MESSAGE IS A TRANSMISSION FOR A CRT OPERATING AS A REMOTE INQUIRY AND DISPLAY DEVICE WHERE THE REMOTE CRT USER MAY ENTER ANY QUANTITY OF INFORMATION UP TO THE MAXIMUM SIZE OF THE CRT SCREEN. IF THE AMOUNT OF CORE STORAGE AVAILABLE FOR USE AS BUFFERS FOR VARIABLE LENGTH MESSAGES WAS EXTREMELY LIMITED, IT WOULD BE BETTER TO USE A POOL OF BUFFERS SINCE NOT ALL MESSAGES WOULD BE IN PROGRESS AT ONCE AND SO LESS BUFFER AREA WOULD BE NEEDED THAN IF SINGLE BUFFER MODE WAS USED AND BUFFERS WERE PERMANENTLY ASSIGNED. EITHER SINGLE MODE OR POOL MODE COULD BE USED FOR A VARIABLE LENGTH TRANSMISSION WITH THE MOST ADVANTAGEOUS METHOD BEING CHOSEN AS THE APPLICATION DICTATES.

A TRANSMISSION WHICH IS INDETERMINATE IN LENGTH SHOULD ALWAYS BE PROCESSED USING THE POOL MODE OF OPERATION. A TRANSMISSION OF INDETERMINATE LENGTH IS MOST FREQUENTLY ENCOUNTERED IN MESSAGE SWITCHING APPLICATIONS WHERE THE LENGTH OF AN INPUT MESSAGE IS UNDER THE JURISDICTION OF THE REMOTE STATION, AND THE SYSTEM MUST USE SEGMENTATION TO ACCEPT, PROCESS, STORE, AND FORWARD THAT ENTIRE MESSAGE.

10.18.3.2. CORE AVAILABITY

ANOTHER FACTOR WHICH LARGELY AFFECTS WHETHER SINGLE MODE OR POOL MODE OF BUFFERING IS TO BE USED IS THE QUANTITY OF CORE STORAGE AVAILABLE FOR BUFFER AREAS. THE LACK OF ADEQUATE BUFFERING AREAS DICTATES THAT THE POOL METHOD SHOULD BE USED SO THAT THE BUFFER AREA MAY BE SHARED BY NUMEROUS LT GROUPS. SUFFICIENT BUFFER AREA MAY PERMIT THE USE OF SINGLE MODE BUFFERING OR EVEN THE EXTREME CASE OF A CLOSED POOL OF BUFFERS FOR EACH LT GROUP. HOWEVER, SUCH AN EXTREME CASE HAS AN ADDED RESTRAINT THAT EACH BUFFER OF INFORMATION MUST HAVE SUFFICIENT STAGING AREA AND ADEQUATE MASS STORAGE TRANSFER TIME SO THAT NO DATA IS LOST FOR AN OVERLOAD SITUATION. THE OPEN CHAIN POOL METHOD INSURES THAT A BUFFER WILL NEVER BE REUSED BY THE SYSTEM UNTIL SO INSTRUCTED BY THE USER. IF THE POOL MODE OF BUFFERING IS CHOSEN, CONSIDERATION MUST BE GIVEN TO THE SIZE OF THE CORE STORAGE AREA TO BE USED FOR BUFFERING. IF ADEQUATE AREA IS AVAILABLE, THE DESIRED SIZE CAN BE SET ASIDE FOR THE BUFFER POOL. HOWEVER, SUCH IS GENERALLY THE EXCEPTION RATHER THAN THE RULE.

10.18.3.3. POOL SIZE

THE OPTIMUM SIZE FOR THE BUFFER POOL IS ACTUALLY DETERMINED BY THE APPLICATION BUT INFLUENCED BY THE WORK LOAD OF THE SYSTEM. IF THE APPLICATION IS SUCH THAT THE LOSS OF ANY AMOUNT OF DATA CANNOT BE TOLERATED, THERE IS NO CHOICE BUT TO FIX THE SIZE OF THE BUFFER POOL AT SOME MAXIMUM TO ADEQUATELY HANDLE THE PEAK LOAD. SYSTEMS WITH SUCH STRINGENT REQUIREMENTS DO EXIST, BUT MORE GENERALLY THE REAL-TIME PROGRAM HAS SOME DEGREE OF CONTROL OVER THE REMOTE STATIONS. FOR EXAMPLE, THE POLLING OPERATIONS CAN BE REDUCED IF AN EXCESS WORK LOAD IS ENCOUNTERED, OR THE REMOTE STATION CAN BE INSTRUCTED TO RETRANSMIT IF ANY PORTION OF A TRANSMISSION IS LOST JUST AS THOUGH A PARITY ERROR HAD OCCURRED.

IT IS MOST ADVANTAGEOUS FOR THE REAL-TIME PROGRAM TO DYNAMICALLY ADJUST THE SIZE OF THE BUFFER POOL TO CORRESPOND TO THE WORK LOAD PROVIDING THAT ADDITIONAL CORE STORAGE CAN ALWAYS BE ACQUIRED BY THE REAL-TIME PROGRAM WHEN IT IS NEEDED. FOR

EXAMPLE, LITTLE REAL-TIME ACTIVITY MIGHT OCCUR FROM 11 PM TO 6 AM AFTER WHICH TIME THE WORK LOAD MIGHT INCREASE TO ITS PEAK AT NOON. FOR THIS CASE THE SIZE OF THE BUFFER POOL WOULD BE AT ITS MAXIMUM DURING THE DAYTIME AND MINIMUM AT NIGHT SO THAT THE EXCESS CORE STORAGE AREA COULD BE UTILIZED BY OTHER PORTIONS OF THE SYSTEM IF THE AREA WAS NOT NEEDED BY THE REAL-TIME PROGRAM FOR OTHER USES SUCH AS A STATISTICAL SUMMARY OR UPDATING OF REAL-TIME DATA FILES. IT SHOULD BE KEPT IN MIND THAT THE RELEASE OF CORE STORAGE IN INCREMENTS OF 512 WORDS IS NECESSARY FOR THE AREA TO BE USED BY OTHER PORTIONS OF THE SYSTEM. THE NUMBER OF SIMULTANEOUS USER'S OF THE BUFFER POOL ALSO CAN BE USED TO DETERMINE THE SIZE OF THE BUFFER POOL. THERE MAY BE CASES WHERE THE SIZE CAN BE FIXED AT SOME MINIMUM AND STILL BE ADEQUATE TO HANDLE ALL TRAFFIC. AN EXAMPLE OF THIS SITUATION IS AN APPLICATION WHOSE COMMUNICATIONS NETWORK IS DESIGNED SUCH THAT IT IS NEVER LOADED BEYOND 25% OF ITS CAPACITY. SUCH NETWORK LOADING IS OFTEN EMPLOYED FOR MESSAGE SWITCHING IN ORDER TO PROVIDE A REASONABLE RESPONSE TIME BETWEEN STATIONS ON A MULTI-STATION POLLED CIRCUIT.

10.18.3.4. BUFFER SIZE

VERY CLOSELY CONNECTED WITH THE SIZE OF THE BUFFER POOL IS THE CONSIDERATION TO BE GIVEN TO DETERMINING THE PROPER SIZE FOR THE INDIVIDUAL INPUT AND OUTPUT COMMUNICATIONS BUFFER AREAS (NOT TO BE CONFUSED WITH MASS STORAGE BUFFERING AREAS). COMMUNICATIONS BUFFER SIZE IS NORMALLY FIXED AT A SINGLE ADEQUATE VALUE AND IS NOT DYNAMICALLY CHANGED OR INFLUENCED BY CHANGES IN THE SYSTEM WORK LOAD OR TIME OF DAY. THE INDIVIDUAL BUFFER SIZE IS AGAIN DETERMINED BY THE APPLICATION, BUT IT IS GREATLY INFLUENCED BY SUCH THINGS AS THE MASS STORAGE MEDIUM, STAGING AREA AND WORKING AREA SIZE, ACTUAL LINE SPEEDS OF THE COMMUNICATION NETWORK, NUMBER OF CIRCUITS IN THE NETWORK, AND DATA PACKING TECHNIQUES. WITH ALL OF THESE FACTORS TAKEN INTO ACCOUNT, TYPICAL BUFFER SIZES FOR EXISTING REAL-TIME INSTALLATIONS RANGE FROM TEN TO 100 DECIMAL CHARACTERS. THE MASS STORAGE MEDIUM IS PERHAPS THE LARGEST INFLUENCE IN THE SIZE FOR INDIVIDUAL BUFFERS DUE TO HARDWARE CHARACTERISTICS SUCH AS ACCESS TIME AND ADDRESSABILITY AS WELL AS THE SOFTWARE UTILIZATION OF MASS STORAGE IN AREAS SUCH AS THE AMOUNT AND METHOD OF SEGMENT AND MESSAGE LINKAGE, REAL-TIME DIRECTORY CONTENTS AND LOCATION, REAL-TIME REPACKING PRINCIPLES, ETC. A MASS STORAGE DEVICE WITH A HIGH ACCESS TIME DICTATES THAT A LARGER BUFFER SIZE SHOULD BE USED WHILE SMALLER BUFFERS MAY BE SUCCESSFULLY USED IF THE MASS STORAGE DEVICE HAS A VERY FAST ACCESS TIME AND ALSO A HIGH DATA TRANSFER RATE.

THE ADDRESSABILITY OF THE MASS STORAGE MEDIUM DETERMINES STAGING AREA SIZE WHICH CONTROLS THE SIZE OF INDIVIDUAL BUFFERS SINCE IT IS BEST TO ESTABLISH A BUFFER WHICH IS A FRACTION OF THE SIZE OF THE STAGING AREA. FOR EXAMPLE, A MASS STORAGE DEVICE WITH EITHER TRACK OR SECTOR ADDRESSABILITY WOULD NECESSITATE THAT THE SIZE OF THE STAGING AREA BE A MULTIPLE OF THE TRACK OR SECTOR SIZE. THE ENTIRE AREA OF SOFTWARE TECHNIQUES FOR THE PROPER USE OF MASS STORAGE IS TOO EXTENSIVE TO ELABORATE ON IN THIS DISCUSSION SINCE TOPICS COULD ONLY BE DISCUSSED IN GENERAL UNTIL A PARTICULAR MASS STORAGE DEVICE WAS SPECIFIED. IT IS EXPECTED THAT THE SIZE OF INDIVIDUAL BUFFERS MUST INCREASE WITH THE LINE SPEED (A LARGER BUFFER SIZE WOULD BE MORE SUCCESSFUL FOR A 4800 BAUD LINE THAN THE BUFFER SIZE CHOSEN FOR A 75 BAUD LINE). THE SIZE OF BUFFERS SHOULD ALSO INCREASE WITH THE NUMBER OF LINES CONTROLLED BY THE SYSTEM (NOTE USE OF THE TERM SYSTEM SINCE IT IS PROBABLE THAT SEVERAL REAL-TIME PROGRAMS MAY EXIST IN THE SYSTEM WITH THE WORK LOAD DIVIDED AMONG VARIOUS PROGRAMS ACCORDING TO TYPE OF WORK WITH COMMON LINKAGE BETWEEN PROGRAMS EXISTING ONLY IN THE FORM OF QUEUES, TABLES, FILES, ETC.) SINCE REAL-TIME PROGRAMS ARE NOT TIME SHARED OR 'SLICED' BY THE EXECUTIVE SYSTEM BUT RATHER ARE EXPECTED TO SHARE THE PROCESSOR.

THE AREA OF DATA PACKING TECHNIQUES ALSO INFLUENCES INDIVIDUAL BUFFER SIZE. DATA PACKING TECHNIQUES COVER BOTH HARDWARE CHARACTERISTICS AND SOFTWARE METHODS. HARDWARE CHARACTERISTICS INCLUDE SUCH THINGS AS WHOLE WORD, HALF WORD, OR QUARTER WORD COMMUNICATIONS BUFFERING METHODS, PARTIAL WORD ADDRESSING CAPABILITIES OF THE PROCESSOR, AND PERIPHERAL SUBSYSTEM OPERATION SUCH AS THE READING OF BYTES VIA THE 6C-8C SUBSYSTEM WHICH PACKS 9-BIT BYTES IN A DOUBLE WORD FORMAT.

SOFTWARE METHODS INCLUDE USE OF A COMMON INTERNAL CODE AND ITS RELATED PACKING. SOME INTERNAL CODE MUST BE PICKED AS THE STANDARD TO BE USED INTERNALLY BY THE REAL-TIME PROGRAM. THE USUAL BASIS FOR THE SELECTION OF THE CODE IS USING THE CODE WHICH IS MOST COMMON THROUGHOUT THE COMMUNICATIONS NETWORK WITH MINOR INFLUENCE

ASSERTED BY LINE SPEED TO AVOID EXCESSIVE CHARACTER TRANSLATION ON HIGH SPEED LINES. THIS WILL GENERALLY NOT BE FIELDATA WHICH IS USED INTERNALLY BY THE EXECUTIVE SYSTEM BUT WOULD PROBABLY BE EITHER BAUDOT OR ASCII SINCE THE MAJORITY OF REMOTE STATIONS HAVE HARDWARE DESIGN CHARACTERISTICS EMPLOYING EITHER ONE OR THE OTHER OF THESE CODES. ANY INPUT WHICH IS RECEIVED WHICH DOES NOT CONFORM TO THE STANDARD INTERNAL CODE IS TRANSLATED BEFORE BEING PROCESSED BY THE REAL-TIME PROGRAM SO TO THE REAL-TIME PROGRAM IT ALWAYS APPEARS AS SIMILAR DATA TO THE REAL-TIME-PROGRAM. ACCORDINGLY FOR OUTPUT, THE DATA IN THE STANDARD INTERNAL CODE IS TRANSLATED TO THE CODE DESIRED BY THE REMOTE STATION AS THE COMMUNICATIONS OUTPUT BUFFER AREA IS BEING FILLED.

ONCE A PARTICULAR CODE HAS BEEN SELECTED AS THE INTERNAL STANDARD FOR THE REAL-TIME PROGRAM, VARIOUS DATA PACKING METHODS BY THE SOFTWARE CAN BE BUILT AROUND THAT CODE. FOR INSTANCE, THE SELECTION OF ASCII PERMITS ONLY NINE CHARACTERS TO BE PACKED INTO A DOUBLE WORD WHILE THE USE OF BAUDOT WOULD ALLOW TEN CHARACTERS TO BE PACKED INTO A DOUBLE WORD WITH THE ADDED ADVANTAGE OF TWO BITS REMAINING FOR USE AS CONTROL INFORMATION. THE PROPER CONTROL TECHNIQUES AND THE GREATER EFFICIENCY OF PACKING USING A SMALLER CODE SUCH AS BAUDOT CAN PROVIDE A LARGER SYSTEM CAPACITY WHEN MASS STORAGE AREA IS LIMITED. THE USE OF VERTICAL PACKING RATHER THAN HORIZONTAL PACKING CAN ELIMINATE THE OUTPUT STAGING AREA SINCE THE COMMUNICATIONS OUTPUT BUFFER AREA CAN DOUBLE AS THE STAGING AREA. VERTICAL PACKING MAY ALSO BE ADVANTAGEOUS FOR CERTAIN UNIQUE HARDWARE CHARACTERISTICS.

ALSO WORTHY OF MENTION IS THE DUAL POOL METHOD AVAILABLE FOR ACCEPTING INPUT. THE PRIMARY USE FOR THE DUAL POOL INPUT MODE IS TO PROVIDE A RAPID RESPONSE VIA SOFTWARE FOR FEATURES OTHERWISE PROVIDED BY HARDWARE OPTIONS FOR POLLING OPERATIONS. THE RESPONSE FROM A POLL MESSAGE MAY BE EITHER A SHORT 'NO BUSINESS' RESPONSE OR A LENGTHY TRANSMISSION OF DATA FROM THAT PARTICULAR REMOTE STATION. IT IS MOST DESIRABLE FOR THE REAL-TIME PROGRAM TO HAVE IMMEDIATE NOTIFICATION WHEN THE FIRST PORTION OF THE POLL RESPONSE IS RECEIVED SO THAT THE NEXT POLL MESSAGE MAY BE INITIATED IF THE RESPONSE INDICATES 'NO BUSINESS'. RATHER THAN USE A TIMER, THE OCCURRENCE OF AN INPUT MONITOR INTERRUPT FOR A SMALL BUFFER CAN BE USED TO TRIGGER THE REAL-TIME PROGRAM'S ANALYSIS OF THE POLL RESPONSE. HOWEVER, IF A LENGTHY TRANSMISSION WAS INITIATED BY THE POLL MESSAGE, IT IS NOT ADVISABLE FROM A SYSTEM STANDPOINT TO CONTINUE WITH SMALL BUFFERS FOR ACCEPTING THE INPUT DATA. THE DUAL POOL INPUT MODE PROVIDES THE ABILITY TO ACCEPT A TRANSMISSION INTO A SMALL BUFFER AREA INITIALLY WHEN THE ER CMI$ IS EVALUATED WITH AN IMMEDIATE SWITCH TO A POOL OF LARGER BUFFERS FOR SUBSEQUENT PORTIONS OF THE DATA TRANSMISSION IF ANY SHOULD FOLLOW.

10.18.4. PRIORITY CATEGORIES

THE EXECUTIVE SYSTEM CLASSIFIES AN I/O REQUEST INTO ONE OF THREE CATEGORIES DEPENDING UPON THE NATURE OF THE ACTIVITY WHICH SUBMITS THE I/O REQUEST. THE THREE CATEGORIES ARE ASSIGNED PRIORITES IN THE FOLLOWING ORDER, AND ALL REQUESTS IN A CATEGORY ARE COMPLETED BEFORE ANY REQUEST IS HONORED FOR THE NEXT LOWER PRIORITY:

REAL-TIME
EXECUTIVE
DEMAND/BATCH

LOOK AHEAD TECHNIQUES ARE USED WITHIN A CATEGORY WHENEVER APPROPRIATE SO THAT THE AVERAGE EXECUTION TIME FOR I/O REQUESTS MAY BE REDUCED. SINCE LOOK AHEAD WILL CAUSE I/O REQUESTS TO BE COMPLETED IN A SEQUENCE DIFFERENT FROM THE ORDER OF SUBMISSION, THE USER SHOULD BE AWARE THAT THE WORKER PROGRAM MUST PROVIDE ITS OWN PROTECTIVE MEASURES AND TAKE THE APPROPRIATE ACTION IF IT IS CONCERNED WITH THE SEQUENCE OF EXECUTION OF I/O REQUESTS.

THE EXECUTIVE SYSTEM PROVIDES DIFFERENT METHODS TO BE USED BY REAL-TIME PROGRAMS FOR CHANGING PRIORITY LEVELS, REGISTERING ACTIVITIES, AND DISPERSING THE WORK LOAD AMONG THE PROCESSORS IN A MULTIPROCESSOR ENVIRONMENT. THE PROPER USE OF THESE EXECUTIVE FUNCTIONS SHOULD BE UNDERSTOOD SO THAT THEY ARE NOT ABUSED AND ARE PROPERLY UTILIZED SO AS TO HAVE MINIMUM SYSTEM OVERHEAD TO ACHIEVE A DESIRED GOAL. THIS DISCUSSION IS CONCERNED WITH THE EXECUTIVE REQUESTS OF RT$, NRT$, FORK$, EXIT$, AND

UNLCK$ AS WELL AS THE PRIORITY LEVELS OF THE EXECUTIVE SYSTEM AS THEY APPLY TO THE REAL-TIME PROGRAM. THE NUMEROUS PRIORITY LEVELS SHOULD BE UNDERSTOOD IN ORDER TO DETERMINE THE PROPER USE OF THE PREVIOUSLY MENTIONED EXECUTIVE REQUESTS. THE AREAS OF DEADLINE BATCH AND DEMAND HAVE A TYPE OF PRIORITY BELOW THAT OF REAL-TIME AND WILL BE CONSIDERED TO BE GROUPED WITH BATCH RUNS FOR THE PURPOSES OF THIS DISCUSSION. THE MAIN CATEGORIES OF PRIORITIES OF THE EXECUTIVE SYSTEM THEN REVERT TO THE FOLLOWING ORDER:

A) ALL INTERRUPT QUEUEING AT THE TIME OF OCCURRENCE
B) ANY ESI COMPLETION ACTIVITY IN ORDER OF OCCURRENCE
C) ALL EXECUTIVE FUNCTIONS SUCH AS DYNAMIC ALLOCATOR, DISPATCHER, I/O CONTROL, I/O HANDLERS AND CLOCK CONTROL
D) ANY REAL-TIME INTERRUPT ACTIVITY IN ORDER OF OCCURRENCE
E) REAL-TIME LEVELS 2-35
F) OTHER EXECUTIVE OPERATIONS SUCH AS LOAD$, MCORE$, LOG$, PRINT$, AND TSWAP$
G) ANY BATCH INTERRUPT ACTIVITY IN ORDER OF OCCURRENCE
H) ALL BATCH WORK

10.18.4.1. CATEGORY A

THE OCCURRENCE OF ANY WORK IN A HIGHER PRIORITY CATEGORY WILL CAUSE THE LOWER PRIORITY CATEGORY TO BE SUSPENDED UNTIL ALL HIGHER PRIORITY WORK HAS BEEN COMPLETED. ALL CATEGORIES EXCEPT A ARE CONTROLLED BY THE SOFTWARE AND AS SUCH ARE INTERRUPTABLE AND SUBJECT TO SUSPENSION. INTERRUPT QUEUEING IS THE PROCESS PERFORMED AT THE OCCURRENCE OF A HARDWARE INTERRUPT AND AMOUNTS TO APPROXIMATELY 45 MICROSECONDS OF INSTRUCTION EXECUTION TIME FOR AN ESI INTERRUPT AS DISCUSSED FURTHER IN CHAPTER 6 UNDER REAL-TIME PROCESSING, TIMING.

10.18.4.2. CATEGORY B

CATEGORY B HAS THE HIGHEST SOFTWARE PRIORITY AND IS USED FOR THE PROCESSING NECESSARY BY ESI COMPLETION ACTIVITIES. IT IS CONSIDERED TO BE REAL-TIME LEVEL 1 BY THE EXECUTIVE. ESI COMPLETION ACTIVITES ARE ALWAYS ACTIVATED WITH A LIMITED SET OF REGISTERS

IN ORDER TO DETECT PROGRAM ERRORS AND EXCESSIVE LOOPS IN THE HIGH PRIORITY CATEGORIES AVAILABLE TO REAL-TIME PROGRAMS, THE EXECUTIVE SYSTEM ALWAYS TIMES BOTH ESI COMPLETION ACTIVITIES AND REAL-TIME ACTIVITIES. THE AMOUNT OF TIME PERMITTED FOR THESE ACTIVITIES CAN BE EITHER FIXED AT THE VALUE DEFINED AT SYSTEMS GENERATION TIME OR DYNAMICALLY CONTROLLED FOR EACH ACTIVITY WITH A VALUE BASED UPON QUANTITY OF CHARACTERS TO BE PROCESSED. THE
IF A TIME OUT OCCURS FOR A REAL-TIME ACTIVITY, IT CAUSES THAT ACTIVITY TO BE ENTERED INTO THE 'ERR MODE' CONDITION.

10.18.4.3. CATEGORY C, D, AND E

CATEGORY C IS USED BY THE EXEC FOR ALL THE OPERATIONS THAT THE EXECUTIVE SYSTEM MUST PERFORM. CATEGORY D IS USED FOR REAL-TIME I/O INTERRUPT ACTIVITES AND IS ALWAYS ACTIVATED WITH A LIMITED REGISTER SET AND ONLY A LIMITED SET MAY BE USED DURING INTERRUPT PROCESSING SINCE ANY EXECUTIVE SERVICE REQUEST DROPS THE PRIORITY OF THE INTERRUPT ACTIVITY TO WITHIN THE WORKER PROGRAM. THERE IS A MAXIMUM OF 34 REAL-TIME LEVELS AVAILABLE WITHIN THE SYSTEM FOR USE BY ALL REAL-TIME PROGRAMS. THE USER IS EXPECTED TO DELEGATE THE REAL-TIME LEVELS APPROPRIATELY FOR THE APPLICATION. EITHER A LIMITED SET OR A FULL SET OF REGISTERS MAY BE USED AT ANY OF THE REAL-TIME LEVELS 2-35 WITH THE LIMITED SET ASSUMED TO EXIST UNLESS SPECIFICALLY STATED OTHERWISE BY THE REAL-TIME PROGRAM VIA AN ER FORK$.

10.18.4.4. CATEGORY G AND H

ALL OF THE CONDITIONS WHICH WERE GIVEN FOR CATEGORY D ALSO APPLY TO CATEGORY G EXCEPT THAT THE INTERRUPT ACTIVITY APPLIES TO THE BATCH TYPE OF PROGRAMS AND AS SUCH

MAY BE SUBJECT TO SWAPPING. THE BATCH ACTIVITIES AT CATEGORY H ARE UNDER THE CONTROL OF THE DYNAMIC ALLOCATOR AND ARE ALWAYS TIME SHARED AND ALSO MAY BE SWAPPED. WHETHER OR NOT THEY ARE SWAPPED DEPENDS UPON THE TYPE OF THE OTHER ACTIVITIES FOR THE PROGRAM. THE BATCH ACTIVITES OF A REAL-TIME PROGRAM WILL NOT BE SWAPPED AS LONG AS THERE EXISTS ANY ACTIVITY AT THE REAL-TIME LEVEL SINCE A 'SWAP LOCK' INDICATOR IS SET FOR THE PROGRAM AS LONG AS ANY REAL-TIME ACTIVITY OR ESI COMPLETION ACTIVITY EXISTS FOR THAT PROGRAM.

10.18.4.5. PRIORITY CONTROL

THE VARIOUS ER'S MENTIONED PREVIOUSLY CAN BE USED BY THE REAL-TIME PROGRAM TO CONTROL THE PRIORITIES AND DISTRIBUTE THE WORK LOAD IN A MULTIPROCESSOR ENVIRONMENT. THE ER RT$ IS THE MOST DESIRABLE MEANS OF CHANGING PRIORITY LEVELS WITHIN THE REAL-TIME PROGRAM. IT CAN ACCOMPLISH EITHER OR BOTH OF THE FOLLOWING:

1) CHANGE OF PRIORITY LEVEL FOR A REAL-TIME ACTIVITY
2) DISTRIBUTE THE WORK LOAD AS AN ACTIVITY FOR ANOTHER PROCESSOR IN THE MULTIPROCESSOR ENVIRONMENT

10.18.4.6. RT$ AND NRT$ ER REQUEST

THE ER RT$ MAY BE EXECUTED BY ANY TYPE OF PROGRAM (BATCH, DEMAND, OR REAL-TIME AT LEVELS 2-35, WITH A CHECK PERFORMED BY THE EXEC ONLY IF THE ACTIVITY IS UPGRADED FROM NORMAL STATUS TO REAL-TIME STATUS TO DETERMINE IF THE ACCOUNT NUMBER PERMITS REAL-TIME ACTIVITIES. IF THE CHANGE OF PRIORITY REQUEST IS TO A LEVEL SUFFICIENTLY BELOW THAT OF SEVERAL OTHER ACTIVITIES CURRENTLY ACTIVE IN THE MULTIPROCESSOR SYSTEM, THE ACTIVITY MAY BECOME ASSIGNED TO ANOTHER PROCESSOR WHEN ITS TURN COMES ON THE SWITCH LIST. THE ACTIVITY AT THE NEW PRIORITY LEVEL MAINTAINS THE SAME REGISTER SET THAT WAS IN USE AT THE ER RT$. CLOSELY ASSOCIATED WITH THE ER RT$ IS THE ER NRT$ WHICH PROVIDES THE ABILITY FOR A REAL-TIME ACTIVITY TO ENTER THE BATCH STATUS. THE LEVEL FOR A BATCH ACTIVITY IS UNDER THE JURISDICTION OF THE DYNAMIC ALLOCATOR AND AS SUCH THE PRIORITY LEVEL CANNOT BE CONTROLLED BY A BATCH ACTIVITY LIKE IT CAN BE CONTROLLED BY A REAL-TIME ACTIVITY. THE SAME REGISTER SET IN USE AT THE ER NRT$ WILL BE AVAILABLE TO THE BATCH ACTIVITY.

10.18.4.7. FORK$

THE MOST POWERFUL ER AVAILABLE TO THE REAL-TIME PROGRAM IN THE AREA OF CHANGING PRIORITY LEVELS IS THE ER FORK$ AND WITH SUCH POWER IT MAY BE EASILY ABUSED. AN ER FORK$ CAN ACCOMPLISH SIMULTANEOUSLY ANY OR ALL OF THE FOLLOWING:

1) ADJUSTMENT OF REGISTER SET FROM LIMITED TO FULL OR VICE VERSA
2) SEPARATE AND INDEPENDENT ACTIVITY REGISTRATION VIA THE EXECUTIVE SYSTEM
3) CHANGE OF PRIORITY LEVEL FOR A REAL-TIME ACTIVITY
4) DISTRIBUTE THE WORK LOAD AS AN ACTIVITY FOR ANOTHER PROCESSOR IN THE MULTIPROCESSOR ENVIRONMENT

IT SHOULD BE RECOGNIZED THAT THE LATTER TWO FUNCTIONS CAN BE PERFORMED USING AN ER RT$ SO ALL REAL-TIME PROGRAMS SHOULD REFRAIN FROM USING THE ER FORK$ TO ACCOMPLISH WHAT CAN BE DONE WITH LESS SYSTEM OVERHEAD VIA THE ER RT$. HOWEVER, IF IT IS DESIRED TO ACCOMPLISH EITHER OF THE LATTER TWO FUNCTIONS IN CONJUNCTION WITH EITHER ONE OR BOTH OF THE FIRST TWO FUNCTIONS, THEN AN ER FORK$ IS THE PROPER METHOD TO BE USED.

ONE OF THE MAIN PURPOSES OF AN ER FORK$ SHOULD BE TO ESTABLISH AN INDEPENDENT ACTIVITY. THE ADDITION OF A NEW ACTIVITY TO THE SWITCH LIST CAN BECOME AN EXTENSIVE PROCESS IF ADDITIONAL CORE AREA HAS TO BE ACQUIRED AND ALLOCATED TO THE REAL-TIME PROGRAM AND ITS PROGRAM CONTROL TABLE ADJUSTED TO PROVIDE SUFFICIENT SPACE FOR THE PASSAGE OF REGISTER CONTENTS FROM THE REQUESTING ACTIVITY TO THE NEW ACTIVITY. IF THE SYSTEM OVERHEAD IS TOO EXCESSIVE FOR SOME REAL-TIME APPLICATIONS, IT MAY BE DESIRABLE FOR THE REAL-TIME PROGRAM ITSELF TO MAINTAIN A RECORD OF OUTSTANDING WORK RELATED TO REAL-TIME ACTIVITIES RATHER THAN MAINTAIN THE WORK LOAD DISTRIBUTION VIA ACTIVITIES REGISTERED WITH THE EXECUTIVE SYSTEM.

THE ONLY WAY FOR ANY PROGRAM, REAL-TIME OR BATCH, TO CHANGE FROM THE LIMITED REGISTER SET TO THE FULL REGISTER SET OR VICE VERSA IS BY AN ER FORK$ SO THIS SHOULD ALSO BE ANOTHER USE FOR THE FORK$. IF PÓSSIBLE, THE REAL-TIME PROGRAM SHOULD ATTEMPT TO PERFORM ALL OF ITS PROCESSING USING JUST THE LIMITED SET BECAUSE THERE IS LESS SYSTEM OVERHEAD REQUIRED TO SUSPEND AN ACTIVITY IF ONLY A FEW REGISTERS HAVE TO BE SAVED AND RESTORED RATHER THAN SAVING AND RESTORING THE COMPLETE REGISTER SET. THIS IS A SIGNIFICANT FACTOR FOR OVERLOAD SITUATIONS.

NORMALLY THE ER EXIT$ IS USED AS A FINAL EXIT TO THE EXEC OR FOR THE TERMINATION OF ANY ACTIVITY AND CAUSES THE EXEC TO DELETE THE ACTIVITY. HOWEVER, THE ER EXIT$ WORKS SLIGHTLY DIFFERENT WHEN IT IS EXECUTED BY AN ACTIVITY THAT WAS ORIGINALLY INITIATED AS AN ESI COMPLETION ACTIVITY. RATHER THAN THE ACTIVITY BEING DELETED, IT IS RETURNED TO THE SWITCH LIST BUT PLACED THERE IN A WAIT CONDITION SO THAT IT IS AGAIN ELIGIBLE FOR CONTROL AS AN ESI COMPLETION ACTIVITY WHEN A COMMUNICATIONS INTERRUPT OCCURS. WHEN IT IS IN A WAIT CONDITION ON THE SWITCH LIST, IT CAN BE GIVEN CONTROL PRACTICALLY IMMEDIATELY WITHOUT THE USUAL WORK INVOLVED WITH ACQUIRING AND REPOSITIONING AN AREA USED BY THE EXEC TO HOLD INFORMATION RELATED TO THE ACTIVITY SINCE THE NEEDED AREA IS PERMANENTLY MAINTAINED FOR THE ESI COMPLETION ACTIVITY. THE ESI COMPLETION ACTIVITY WILL BE DELETED WHEN AN ER CMT$ IS EXECUTED. FOR MULTIPLE BUFFER MODE THE UPDATING OF THE LT TABLE IN THE REAL-TIME PROGRAM WILL OCCUR ONLY WHEN THE ESI COMPLETION ACTIVITY IS IN A WAIT CONDITION ON THE SWITCH LIST. THUS THE REAL-TIME PROGRAM CAN CONTROL THE UPDATING OF THE LT TABLE BY WHEN THE ER EXIT$ IS MADE FROM AN ESI COMPLETION ACTIVITY. THE REAL-TIME PROGRAM MUST USE ITS OWN DISCRETION IN DETERMINING THE MOST ADVANTAGEOUS TIME TO HAVE ITS LT TABLES UPDATED.

10.18.4.8. UNLCK$

AN INTERRUPT ACTIVITY IS INITIATED WITH A LIMITED REGISTER SET AT A PRIORITY LEVEL ABOVE ALL OTHER ACTIVITIES FOR THAT TYPE OF PROGRAM, AND THE FIRST EXEC SERVICE REQUEST WILL CAUSE ITS PRIORITY TO BE LOWERED UNDER CONTROL OF THE EXEC TO THAT OF ITS PARENT ACTIVITY. THE UNLOCK FUNCTION MAY BE USED BY THE REAL-TIME INTERRUPT ROUTINE TO REDUCE ITS PRIORITY LEVEL BY EXECUTING AN ER UNLCK$ IF THE NEW PRIORITY LEVEL IS TO BE DETERMINED BY THE EXEC. OTHERWISE THE REAL-TIME INTERRUPT ROUTINE SHOULD USE THE ER RT$ WHEN IT DESIRES A SPECIFIC PRIORITY LEVEL.

10.18.5. TEST AND SET USAGE.

SINCE TEST AND SET (TS) INSTRUCTION IS AVAILABLE IN ALL SYSTEMS (SIMULATED VIA SOFTWARE IF NOT AVAILABLE), THIS INSTRUCTION CAN ALWAYS BE USED TO PROTECT 'COMMON DATA' IN A USER PROGRAM. FOR BATCH, DEMAND AND DEADLINE PROGRAMS, THE EXECUTIVE AUTOMATICALLY RESOLVES THE CONFLICTS THAT MAY OCCUR IN A USER PROGRAM BECAUSE OF ACTIVITIES AT DIFFERENT SWITCH LIST LEVELS REFERENCING 'COMMON DATA'. THE EXECUTIVE ASSUMES TOTAL MANAGEMENT OF THE SWITCH LISTS FOR THESE TYPES OF PROGRAMS AND WILL AUTOMATICALLY DEGRADE BY ONE LEVEL, ANY ACTIVITY THAT EXPERIENCES A TS FAILURE (TS INTERRUPT). INTERRUPT ACTIVITIES ARE DEGRADED TO THE LEVEL OF THE ACTIVITY THAT CAUSED THE I/O FORK ON THE FIRST INTERRUPT, AND MARKED AS NORMAL ACTIVITIES IN RESPECT TO FUTURE TS INTERRUPTS. IT IS ASSUMED THAT THE ACTIVITY CAN BE PLACED BACK TO ITS NORMAL LEVEL ONCE AN ER INTERRUPT IS PROCESSED FOR THIS ACTIVITY. IN BRIEF, FOR PROGRAMS OTHER THAN REAL TIME, THE EXECUTIVE WILL, IN THE WORST CASE, DEGRADE ALL FAILING ACTIVITIES OF A GIVEN USER PROGRAM TO LEVEL N ON THE LIST AND THEN ROTATE CONTROL AMONG THE ACTIVITIES IN AN ATTEMPT TO REMOVE THE INTERLOCK. IF THE NUMBER OF FAILING ACTIVTITES IS EQUAL TO THE TOTAL NUMBER OF ACTIVITIES FOR THE PROGRAM AND ALL ACTIVITIES MOVE TO N-1 (FAIL THE TS),THE PROGRAM IS TERMINATED WITH A 'TS LOCKUP' ERROR CONDITION NOTED IN THE USER PRINT FILE.

FÓR REAL TIME PROGRAMS, THE EXECUTIVE DOES NOT CHANGE THE ACTIVITIES SWITCHING LEVEL ON THE TS INTERRUPT, EXCEPT FOR INTERRUPT ACTIVITIES WHICH ARE TREATED THE SAME AS BATCH (DROPPED TO THE LEVEL OF THE ACTIVITY CAUSING THE I/O FORKS). INSTEAD, THE USER ACTIVITY IS DIRECTED TO THE CONTINGENCY POINT FOR THE PARTICULAR ACTIVITY IF ONE EXISTS, OTHERWISE TO CONTINGENCY POINT FOR THE PROGRAM IF ONE EXIXTS. IF A CONTINGENCY POINT IS NOT AVAILABLE, THE ACTIVITY IS MARKED AS AVAILABLE FOR EXECUTION AS THOUGH THE TS INTERRUPT HAD NEVER OCCURRED. IT IS OBVIOUS THAT THE REAL TIME PROGRAM CAN HANG THE SYSTEM IF THE SITUATION WILL NOT AUTOMATICALLY RESOLVE ITSELF AT THE ORIGINAL SWITCHING LEVELS, OR IF THE PROPER ACTION IS NOT TAKEN IN THE

CONTINGENCY ROUTINE TO CHANGE THE SWITCHING LEVEL IN RELATION TO OTHER ACTIVITIES THAT REFERENCE THE DATA. SINCE ON CONTINGENCY THE REAL TIME ACTIVITY KNOWS ITS OWN IDENTITY AND SWITCHING LEVEL, IT WILL BE ABLE TO MAKE RT$ REFERENCE TO CHANGE ITS LEVEL IF A POTENTIAL LOCKUP EXISTS, OR SIMPLY DO LOWER PRIORITY WORK FOR SOME PERIOD OF TIME. IT SHOULD BE POINTED OUT THAT A CONTINGENCY NOTIFICATION ON A TS INTERRUPT WILL NOT OCCUR UNLESS THE ACTIVITY SPECIFICALLY REQUESTS SUCH ACTION, INDEPENDENT OF OTHER TYPES OF CONTINGENCIES.

A REAL TIME PROGRAM (ONE WITH AT LEAST ONE REAL TIME ACTIVITY) MAY HAVE ACTIVITIES ON THE BATCH SWITCH LIST. FOR ACTIVITIES OTHER THAN REAL TIME, THE TS CONTINGENCY FEATURE IS NOT AVAILABLE. IT IS ILLEGAL FOR A R/T PROGRAM TO ALLOW BOTH REAL TIME AND BATCH ACTIVITIES TO REFERENCE THE SAME 'COMMON DATA'. THE BATCH ACTIVITY MUST CAUSE ITSELF TO BE RAISED TO REAL TIME STATUS BEFORE EXECUTING A TS POINTING TO DATA 'COMMON' TO A R/T ACTIVITY. IT CAN REVERT TO A BATCH LEVEL AFTER THE NEED FOR THE 'COMMON DATA' REFERENCE HAS ENDED.

IT IS ILLEGAL FOR ESI ACTIVITIES TO REFERENCE 'COMMON DATA' THAT IS ALSO REFERENCED UNDER A TS CONDITION BY R/T ACTIVITIES. NO ACTION, OTHER THAN A RETURN OF CONTROL, IS TAKEN BY THE EXEC WHEN A TS INTERRUPT OCCURS IN AN ESI ACTIVITY.

11. OPERATOR COMMUNICATIONS

11.1. GENERAL

THE EXECUTIVE SYSTEM HAS BEEN DESIGNED FOR OPERATION WITH A MINIMUM OF OPERATOR INTERVENTION. HOWEVER, IT IS RECOGNIZED THAT SOME FUNCTIONS FREQUENTLY IN USE ARE BEYOND THE SCOPE OF THE EXECUTIVE SYSTEM, WHILE OTHERS DEMAND OPERATOR CONCURRENCE. IN ADDITION, CERTAIN INFORMATION MUST BE PRESENTED AUTOMATICALLY TO THE OPERATOR, WHILE OTHER INFORMATION MUST BE AVAILABLE TO ANSWER OPERATOR REQUESTS.

11.2. SYSTEM MESSAGES

11.2.1. MESSAGE CLASSIFICATION

INSOFAR AS OPERATOR FUNCTIONS ARE REQUIRED FOR A LARGE NUMBER OF ACTIVITIES, THE 1108 EXECUTIVE SYSTEM APPORTIONS THESE FUNCTIONS INTO FOUR CLASSES, THUS DIVIDING OPERATOR DUTIES. THE FOUR FUNCTIONAL CLASSES ARE:

SYSTEM CONTROL
REPORTS SIGNIFICANT EVENTS THAT TAKE PLACE DURING A RUN SUCH AS ABNORMAL RUN TERMINATION, CHECKPOINT DUMP, ETC., AND FOR REPORTING SYSTEM STATUS.

INPUT/OUTPUT ACTIVITY
ALL MESSAGES CONCERNED WITH ON-SITE PERIPHERALS E.G., TAPE MOUNTING DIRECTIONS, I/O ERROR CONDITIONS, PERIPHERAL EQUIPMENT STATUS, ON-SITE SYMBIONT INITIATED MESSAGES, ETC..

COMMUNICATIONS ACTIVITY
MESSAGES FOR INFORMING THE OPERATOR OF EVENTS PERTAINING TO REMOTE TERMINAL OPERATIONS. THE MESSAGES INCLUDE DIALING DIRECTORS, TERMINAL IDENTIFICATION OF SITES WITH AUTOMATIC DIALING AND ANSWERING EQUIPMENT, REMOTE SYMBIONT MESSAGES, ETC..

HARDWARE CONFIDENCE
THE HARDWARE CONFIDENCE ROUTINES USE THIS CATEGORY TO REPORT THEIR FINDINGS OF DIAGNOSTIC TESTING ON THE CPU, CORE, AND ON-SITE PERIPHERALS.

THE CATEGORIES MAY BE ASSOCIATED WITH AS MANY AS FOUR OPERATOR CONSOLES OR AS FEW AS ONE. THE CONSOLE/CATEGORY IS SPECIFIED AT SYSTEM GENERATION TIME. ALL MESSAGES REQUIRING AN OPERATOR RESPONSE MUST BE ANSWERED AT THE CONSOLE DISPLAYING THE MESSAGE. UNSOLICITED OPERATOR MESSAGES CAN BE ENTERED FROM ANY CONSOLE. THE ALLOCATION OF PHYSICAL CONSOLES WITH RESPECT TO CATEGORY MAY ALSO BE CHANGED BY AN UNSOLICITED FACILITY CONTROL KEYIN.

11.2.2. MESSAGE TYPES

ALL SYSTEM CONSOLE COMMUNICATIONS MAY BE FURTHER CATEGORIZED AS:

UNSOLICITED
TYPE (GIVES INFORMATION ONLY)
TYPE AND READ (REQUIRES OPERATOR RESPONSE)

A DESCRIPTION OF THESE TYPES FOLLOWS.

11.2.2.1. UNSOLICITED

UNSOLICITED MESSAGES ARE OPERATOR KEYINS OF THE FORM:

```
MM TEXT
```

'MM' IDENTIFIES THE MESSAGE TO THE SYSTEM. THE MESSAGE ID IS RESTRICTED TO TWO CHARACTERS. THE 'TEXT' PORTION COMPLETES THE MESSAGE WHEN MORE THAN ONE OPTION IS AVAILABLE. IF 'MM' IS UNDEFINED TO THE SYSTEM, THE WORD 'KEY ER' IS DISPLAYED FOLLOWING THE ILLEGAL CODE.

11.2.2.2. TYPE

THIS TYPE OF MESSAGE DIRECTS OR INFORMS OPERATIONS PERSONNEL OF CURRENT INTERNAL REQUESTS OR STATUS. NO OPERATOR KEYIN IS INVOLVED, BUT OPERATOR ACTION MAY BE REQUESTED TO LOAD A TAPE UNIT, CLEAR AN INTERLOCK CONDITION, ETC. THE MESSAGE MAY ORIGINATE FROM WITHIN THE EXECUTIVE SYSTEM OR USER PROGRAM AND TAKES THE FORM:

```
1. EXEC
     MMMMMM  TEXT
2. USER
     IDIDID* TEXT
```

'MMMMMM' IS THE EXECUTIVE SYSTEM IDENTIFICATION OF THE MESSAGE. 'IDIDID' IS THE RUN IDENTIFICATION OF THE USER PROGRAM INITIATING THE MESSAGE. THE ASTERISK APPEARS IN AND IDENTIFIES ALL USER MESSAGES.

11.2.2.3. TYPE AND READ

TYPE AND READ MESSAGES MAY ORIGINATE FROM WITHIN THE EXECUTIVE SYSTEM OR USER PROGRAMS. AN OPERATOR RESPONSE IS REQUIRED. THE MESSAGE TAKES THE FORM:

```
1. EXEC
     N IDIDID  TEXT
2. USER
     N IDIDID* TEXT
```

'N' IS A ONE DIGIT MESSAGE NUMBER. THE OPERATOR REPLIES TO THE MESSAGE BY FIRST TYPING IN THE MESSAGE NUMBER FOLLOWED BY HIS RESPONSE. IF 'N' DOES NOT CORRESPOND TO AN OUTSTANDING MESSAGE NUMBER, OR IF THE NUMBER OF CHARACTERS IN THE TEXT OF THE RESPONSE IS GREATER THAN THE INTERNAL BUFFER, THE WORD 'KEY ER' IS TYPED. THE OPERATOR MUST RE-INITIATE HIS REPLY. 'IDIDID' IS THE IDENTIFICATION OF THE RUN TO WHICH THE MESSAGE APPLIES. THE ASTERISK FOLLOWING THE ID FLAGS USER INITIATED MESSAGES.

11.2.3. KEYBOARD OPERATIONS-TELETYPE* MODEL 35 MACHINE

OPERATION OF THE KEYBOARD IS CONTROLLED INTERNALLY WITH THE CONSOLE ROUTINE. THE INPUT MODE ON THE CONSOLE CHANNEL IS ALWAYS ACTIVATED, EXCEPT DURING A TYPING OPERATION. OPERATOR KEYINS WILL ALWAYS BEGIN WHEN ANY KEY ON THE KEYBOARD IS DEPRESSED. THE CARRIAGE RETURN TERMINATES THE INCOMING MESSAGE. AS EACH CHARACTER IS RECEIVED FROM THE KEYBOARD, IT IS DISPLAYED FOR VISUAL VERIFICATION ON THE MONITOR PRINTER. IF THE OPERATOR WISHES TO CANCEL A PARTIALLY TYPED MESSAGE, HE MAY DO SO BY TYPING A SPECIAL 'DELETE' CODE. THIS 'DELETE' CODE IS REPRESENTED ON THE KEYBOARD BY THE LOZENGE KEY (OCTAL CODE 76). THE ENTIRE MESSAGE IS IGNORED AND LOGICALLY NOT RECEIVED.

MESSAGE NUMBERS ARE USED TO IDENTIFY THOSE MESSAGES REQUIRING AN OPERATOR KEYIN TO COMPLETE THE MESSAGE. AS SOON AS THE MESSAGE HAS BEEN RESPONDED TO BY THE OPERATOR, THE NUMBER IDENTIFYING THAT MESSAGE IS MADE AVAILABLE FOR FURTHER USE. THE LOWEST

*Trademark of TELETYPE CORPORATION

NUMBER AVAILABLE IS ALWAYS ATTACHED TO A NEW MESSAGE. WHEN THE TEN VALUES AVAILABLE FOR MESSAGE NUMBERS ARE IN USE, NO FURTHER MESSAGES OF THIS CLASS WILL BE DISPLAYED. AS MESSAGES ARE COMPLETED BY THE OPERATOR, THEIR NUMBERS ARE AGAIN MADE AVAILABLE FOR ANY MESSAGE ENTERING THE CONSOLE QUEUE. THUS THE MAXIMUM NUMBER OF OUTSTANDING MESSAGES AT ANY GIVEN TIME WILL NEVER EXCEED TEN.

EACH CHARACTER ACCEPTED AS INPUT FROM THE KEYBOARD INITIATES A 20 SECOND TIMING CYCLE IN WHICH THE NEXT CHARACTER MUST BE RECEIVED. IF THE NEXT CHARACTER IS NOT RECEIVED WITHIN THIS ALLOCATED TIME SPAN, THE SOFT PILLOW IS TYPED ON THE PRINTER. THE CARRIAGE RETURN AND DELETE CODE TERMINATE THE TIMING CYCLE.

AN END OF MESSAGE SYMBOL IS TYPED AT THE CONCLUSION OF EACH MESSAGE DISPLAYED ON THE PRINTER. THIS SYMBOL IMMEDIATELY FOLLOWS THE LAST CHARACTER OF EACH MESSAGE, AND IS REPRESENTED ON THE KEYBOARD BY THE 'STOP' CHARACTER (HEXAGON WITH AN INSCRIBED 'S', OCTAL CODE 57). FOR CONSOLE INPUT THE STOP CODE IS LOGICALLY EQUIVALENT TO THE CARRIAGE RETURN.

THE 'INTERRUPT ENABLE' BUTTON ALLOWS THE OPERATOR TO OBTAIN CONTROL OF THE CONSOLE PRINTER DURING A CONTINUOUS DISPLAY OF OUTPUT MESSAGES. AFTER THE INTERRUPT ENABLE BUTTON HAS BEEN SET, ANY KEY MAY BE DEPRESSED CAUSING THE INTERRUPT TO OCCUR. WHEN THE MESSAGE CURRENTLY BEING TYPED IS COMPLETED, THE INPUT MODE IS ACTIVATED. IF NO CHARACTER IS RECEIVED FROM THE KEYBOARD WITHIN 20 SECONDS, THE CURRENT INPUT MODE IS TERMINATED WITH THE LOZENGE MESSAGE.

11.2.4. KEYBOARD OPERATIONS - CRT DISPLAY CONSOLE

KEYBOARD OPERATION OF THE CRT DISPLAY CONSOLE IS SIMILAR TO THAT OF THE TELETYPE* CONSOLE DESCRIBED ABOVE WITH THE EXCEPTION OF THE FOLLOWING DIFFERENCES: MESSAGES, INCLUDING A PERMANENTLY DISPLAYED STATUS SUMMARY ON THE TOP TWO LINES, ARE DISPLAYED ON THE FACE OF A CATHODE RAY TUBE (CRT) UNTIL NO LONGER CURRENT AND THEN ARE ROLLED OFF ONTO A UNIVAC PAGEWRITER CONNECTED TO THE CONSOLE. CHARACTERS ENTERED AT THE KEYBOARD ARE DISPLAYED ON THE BOTTOM LINE OF THE CRT. MESSAGES MAY BE EDITED DURING ENTRY BY EMPLOYING THE INTERRUPT KEYS JUST ABOVE THE CONVENTIONAL KEYBOARD. DEPRESSING KEY 7 DELETES THE LAST CHARACTER ENTERED AND BACKSPACES ONCE. DEPRESSING KEY 6 DELETES THE ENTIRE MESSAGE. INTERRUPT KEY 8 SERVES THE SAME FUNCTION AS THE INTERRUPT ENABLE ON THE TELETYPE CONSOLE.

INTERRUPT KEY 5 FORCES ALL CONSOLE OUTPUT TO THE PAGEWRITER ONLY, BYPASSING THE CRT DISPLAY. THIS FUNCTION IS NECESSARY ONLY IN THE CASE THE CRT DISPLAY IS DOWN.

INTERRUPT KEY 4 FORCES THE ENTIRE DISPLAY INCLUDING THE STATUS SUMMARY TO ROLL OFF ONTO THE PAGEWRITER. THIS FUNCTION IS USED EITHER AT SYSTEM SHUTDOWN TIME OR WHEN A CONSOLE TEAR SHEET IS DESIRED.

INTERRUPT KEY 2 REPLACES THE BASIC OPERATIONS MESSAGE DISPLAY WITH A SYSTEM MEMORY UTILIZATION REPORT. THE BASIC DISPLAY MAY BE RETURNED BY DEPRESSING KEY 1.

11.3. SYNOPSIS OF SYSTEM MESSAGES

11.3.1. SYSTEM CONTROL MESSAGES

11.3.1.1. FACILITY CONTROL

UNSOLICITED

DN C/U (SEE FILE CONTROL CHAPTER FOR MASS STORAGE)

PERIPHERAL EQUIPMENT MAY BE SET TO AN INACTIVE STATE WHEN NECESSARY. MALFUNCTIONING UNITS MAY BE SUSPENDED FROM THE

*TRADEMARK OF TELETYPE CORPORATION

SYSTEM IN THIS MANNER.

DN C

SAME AS FOR DN C EXCEPT ALL EQUIPMENT ON THE CHANNEL MAY BE SET INACTIVE BY THIS SINGLE TYPEIN. THIS TYPEIN IS NOT APPLICABLE TO MASS STORAGE CHANNELS.

UP C/U A   (SEE FILE CONTROL CHAPTER FOR MASS STORAGE)

PERIPHERAL UNITS TEMPORARILY REMOVED FROM THE SYSTEM MAY BE SET TO AN AVAILABLE CONDITION VIA THIS METHOD. THE LETTER 'A' IS SPECIFIED FOR MASS STORAGE UNITS TO CAUSE ALL AREAS TO GO INTO THE AVAILABLE POOL. WITHOUT THE 'A' DESIGNATED AN ATTEMPT IS MADE TO RE-ESTABLISH CATALOGUED FILES.

UP C

SAME AS FOR UP C EXCEPT ALL EQUIPMENT ON THE CHANNEL IS SET AVAILABLE FOR ASSIGNMENT.

RV C/U

ANY NON DRUM OR FASTRAND UNIT MAY BE SET IN A RESERVE STATUS TO AVOID ANY FURTHER ASSIGNMENT OF THE UNIT. THIS RESERVE STATUS WILL NOT INTERFERE WITH THE UNITS OPERATION IF ASSIGNED. AT THE COMPLETION OF THE RUN(S) OR FREE STATEMENT, THE UNIT BECOMES INACTIVE. SYMBIONT CONTROLLED DEVICES BECOME INACTIVE UPON COMPLETION OF CURRENT ACTIVITY.

AC X/Y

REALLOCATES LOGICAL CONSOLE NUMBER OR FUNCTIONAL CATEGORY X TO PHYSICAL CONSOLE OR PROCESSOR NUMBER Y. THIS KEYIN MAY BE USED TO REALLOCATE CONSOLE USAGE OR TO SWITCH ACTIVITY FROM ONE CONSOLE TO ANOTHER WHEN BRINGING THE FIRST ONE DOWN.

11.3.1.2. COARSE SCHEDULER

UNSOLICITED

CS RUNID*PXDYYYYSYYYY

THIS MESSAGE INFORMS THE EXECUTIVE THAT THE PRIORITY, DEADLINE AND/OR START-TIME SHOULD BE CHANGED TO THE VALUE(S) GIVEN. THE 'P' INDICATES THAT THE PRIORITY SHOULD BE CHANGED TO THE LETTER SPECIFIED BY 'X'. THE 'D' INDICATES THAT THE DEADLINE FOR THIS RUN IS THE WALL CLOCK TIME SPECIFIED BY THE 24 HOUR CLOCK VALUE 'YYYY'. THE 'S' AND IT'S 'YYYY' INDICATE START-TIME SHOULD BE CHANGED RATHER THAN DEADLINE. THE THREE SEPARATE FIELDS ARE INDEPENDENT IN THAT ANY OR ALL CAN BE SPECIFIED AND IN ANY ORDER. (THE 24 HOUR CLOCK VALUE IS DIFFERENT FROM THAT SPECIFIED ON THE RUN STATEMENT BECAUSE THE KEY-IN PERTAINS STRICTLY TO WALL CLOCK TIME RATHER THAN OPTIONALLY TO TIME SINCE RUN SUBMISSION .)

CS HOLD RUNID

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD HOLD THE SCHEDULING (OPENING) OF RUNID. RUNID IS OPTIONAL AND IF NOT PRESENT, THE EXECUTIVE WILL HOLD THE SCHEDULING (WILL NOT OPEN) OF ALL THE RUNS IN THE RUN QUEUE.

CS ALLOW RUNID

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD ALLOW 'RUNID' TO BE SCHEDULED (OPENED). RUNID IS OPTIONAL AND IF NOT PRESENT, THE EXECUTIVE WILL ALLOW ALL

'HELD' RUNS TO BE SCHEDULED.

CS TYPE HOLD

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD TYPE (LIST) ALL RUNS CURRENTLY BEING 'HELD'. HOLD IS OPTIONAL AND IF NOT PRESENT, THE EXECUTIVE WILL TYPE ALL ACTIVE RUNIDS. ACCOMPANYING EACH ACTIVE DEMAND RUNID WILL BE ITS TERMINAL LOCATION.

CS REMOVE RUNID

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD REMOVE RUNID FROM THE RUN QUEUE.

CS LIST BACKLOG

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD LIST THE RUNS CURRENTLY IN THE RUN QUEUE. ACCOMPANYING EACH RUNID WILL BE ITS PRIORITY, ESTIMATED RUN TIME, PAGE COUNT, PUNCH COUNT, PROJECT NUMBER, ACCOUNT NUMBER, START TIME AND DEADLINE TIME, I.E., A SUMMARY OF ITS @RUN CONTROL STATEMENT.

CS LOG TEXT

THIS MESSAGE INFORMS THE EXECUTIVE THAT IT SHOULD ENTER THE 'TEXT' INTO THE SYSTEM LOG. TEXT IS LIMITED TO 132 CHARACTERS.

ST ,PRIORITY/RUN-OPTIONS NAME,SET,RUNID,ACCOUNTING,...

AN INDEPENDENT RUN WILL BE SCHEDULED AS DIRECTED BY THE KEYIN. THE ACCOUNTING FIELD IS REQUIRED AND THE REQUEST IS REJECTED IF IT IS NOT PRESENT. ALL PARAMETERS AFTER THE 'ST ' ARE IDENTICAL TO THOSE ON THE @START STATEMENT AND MUST FOLLOW THE SAME FORMAT AND DROPOUT RULES. (SEE THE SECTION ON THE @START STATEMENT IN THE CHAPTER ENTITLED 'EXECUTIVE CONTROL LANGUAGE' FOR A COMPLETE LIST AND FORMAT DESCRIPTION OF THE PARAMETERS.)

TYPE AND READ

RUNID UNDEF ACCT NNNNNNNNNNNN ARE

THIS MESSAGE INDICATES THAT AN UNDEFINED ACCOUNT NUMBER (REPRESENTED BY NN...N ABOVE) WAS SPECIFIED ON RUN-ID. THE 'A' RESPONSE INFORMS THE EXECUTIVE TO ACCEPT THE RUN WHILE THE 'R' RESPONSE WILL CAUSE REJECTION OF THE RUN. THE 'E' RESPONSE INFORMS THE EXECUTIVE TO ACCEPT THE RUN AND ENTER THE ACCOUNT NUMBER IN THE 'ALLOWABLE' ACCOUNTS TABLE.

TYPE

XXXXXX DUPLICATED. NEW ID IS YYYYYY

THIS MESSAGE INDICATES THAT A NON-UNIQUE RUN-ID (REPRESENTED BY XXXXXX) WAS SPECIFIED. THE COARSE SCHEDULER CHANGED THE RUN-ID TO YYYYYY.

11.3.1.3. DYNAMIC ALLOCATOR

UNSOLICITED

CP KK MM

CHANGE THE CPU TIME SHARING PERCENTAGES TO:

```
        KK (DEMAND MINIMUM %)
        MM (DEMAND MAXIMUM %)

    SS
        INITIATE A SYSTEM STATUS REPORT.
        THE RESPONSE WILL BE:

        LAST PERIOD USAGE; EXEC XX%, BATCH XX%,
        DEMAND XX%, IDLE XX%, RT XX%, OPENBCH YY,
        UNOPRUN YY, AVEBCH ZZ%, AVEDEM ZZ%

        WHERE- XX IS THE PERCENTAGE OF THE LAST 6 SECOND
               PERIOD USED PER ITEM
               YY IS THE NUMBER OF BATCH RUNS
               ZZ IS THE PERCENTAGE OF THE ELAPSED TIME
                  SINCE BOOTING PER ITEM.
```

THE SYSTEM STATUS REPORT IS ALWAYS DISPLAYED ON THE CRT CONSOLE AND IS FREQUENTLY UPDATED. THUS IT IS NEVER NECESSARY TO REQUEST A STATUS REPORT WITH THE SS KEYIN ON A CRT DISPLAY CONSOLE.

11.3.2. TERMINATIONS

11.3.2.1. ABORT$

THE ABORTING OF A RUN MAY BE INITIATED BY AN ER REQUEST OR BY THE OPERATOR THROUGH THE TYPEIN :

```
X   RUN IDENTITY
```

AN ABORT MESSAGE TO THE OPERATOR IS DISPLAYED WITH THE FOLLOWING FORMAT :

```
(RUN IDENTITY) - ABORT
```

11.3.2.2. ERR$

IF THE LAST ACTIVITY OF A PROGRAM ERROR TERMINATES THE FOLLOWING MESSAGE WILL BE DISPLAYED TO THE OPERATOR:

```
(RUN IDENTITY) - ERROR
```

11.3.2.3. INTERRUPT

FROM THE OPERATORS CONSOLE A PROGRAM CAN BE INTERRUPTED IF THE PROGRAM HAS ARRANGED TO ACCEPT THE CONSOLE INTERRUPT; OTHERWISE THE KEYIN IS IGNORED AND THE OPERATOR IS NOTIFIED.

THE TYPE IN IS:

```
II  RUN IDENTITY
```

11.3.2.4. CHECKPOINT/RESTART MESSAGES

```
        UNSOLICITED
        -----------

    CK OPTIONS,FILENAME,RUNID

        A CHECKPOINT WILL BE TAKEN OF THE GIVEN RUN ON THE
        SPECIFIED FILE.  THE OPTIONS CAN CONTAIN P WHICH
        STATES THAT A CHECKPOINT MESSAGE WILL BE TYPED, AND
        T WHICH STATES THAT THE RUN IS TO BE TERMINATED AFTER
        CHECKPOINT.

    RS P,PRIORITY,RUNID,ACC-NBR,CKPT#,FILENAME,REEL
```

RESTART OF THE GIVEN RUNID WILL BE INITIATED ON THE REEL, FILE, AND CHECKPOINT NUMBER, USING SPECIFIED PRIORITY AND ACC-NBR FOR RESTARTING.

ALL OF THE FOLLOWING MESSAGES WILL APPEAR ONLY IF THE P OPTION APPEARED IN THE CHECKPOINT OR RESTART PARAMETERS.

TYPE

```
CKPT# FILENAME, REEL NBR
CKPT FS
```

IF THE P OPTION WAS SPECIFIED, THIS MESSAGE WILL BE TYPED AT COMPLETION OF A CHECKPOINT.

ERROR

```
CK TP ER
CK DM ER
```

AN UNRECOVERABLE TAPE OR DRUM ERROR OCCURRED DURING A CHECKPOINT. THE RUN WILL BE TERMINATED.

```
RS TP ER
RS DM ER
```

AN UNRECOVERABLE TAPE OR DRUM ERROR OCCURRED DURING A RESTART. THE RESTART WILL NOT BE TAKEN.

```
'FILENAME' N CAT
```

THE ERROR OPTION FOR NO RELOAD WAS SPECIFIED AT THE TIME OF RESTART. THE FILE IS NO LONGER CATALOGUED. THE RESTART WILL NOT BE TAKEN.

TYPE AND READ

```
CK TP ER, RSTRT REEL#?
CK DM ER, RSTRT REEL#?
```

AN UNRECOVERABLE TAPE OR DRUM ERROR OCCURRED DURING A CHECKPOINT. THE APPROPRIATE RESPONSES ARE:

Y - RESTART ON THE LAST CHECKPOINT ON THE ABOVE REEL WILL BE INITIATED.
N - THE RUN WHICH CALLED FOR THE CHECKPOINT CAN BE TERMINATED BY THE OPERATOR.

```
CK ER, FILENAME?
```

THE FILENAME ON WHICH THE CHECKPOINT WAS TO BE TAKEN IS IN ERROR. THE ANSWERS ARE:

'FILENAME' - THE CHECKPOINT WILL BE TAKEN ON THE CORRECTED FILENAME.
N - THE CHECKPOINT WILL BE TERMINATED.

```
RS TP ER, RSTRT REEL#?
RS DM ER, RSTRT REEL#?
```

AN UNRECOVERABLE TAPE OR DRUM ERROR OCCURRED DURING A RESTART. THE APPROPRIATE RESPONSES ARE:

Y - RESTART ON THE LAST CHECKPOINT ON THE ABOVE REEL WILL BE INITIATED.
N - THE RUN WHICH CALLED FOR THE RESTART CAN BE TERMINATED.

```
CKPTXX N ON FILENAME?
```

THE CHECKPOINT CAN NOT BE FOUND. THE RESPONSES ARE:

'FILENAME' - THE CHECKPOINT NUMBER WILL BE SEARCHED ON THE TYPED FILENAME.
N - THE RESTART WILL BE TERMINATED.

```
CKPTXX N ON REEL#?
```

THE CHECKPOINT CAN NOT BE FOUND. THE RESPONSES ARE:

'REEL#' - THE OPERATOR MUST LOAD THE PROPER REEL ON WHICH THE CHECKPOINT CAN BE FOUND.
N - THE RESTART WILL BE TERMINATED.

11.3.3. INPUT/OUTPUT ACTIVITY MESSAGES

11.3.3.1. ON-SITE PERIPHERAL DEVICE HANDLER

THE CONSOLE MESSAGES FOR THE ON-SITE DEVICE HANDLERS ARE DOCUMENTED IN THE CHAPTER ENTITLED 'INPUT/OUTPUT DEVICE HANDLERS'.

11.3.3.2. ON-SITE SYMBIONT MESSAGES

UNSOLICITED

IC C/U (SEC.9.6)

INITIATE SYMBIONT PROBE CYCLE ON SPECIFIED C/U.

RN C/U RUNID1/RUNID2/...RUNIDN (SEC.9.6)

SCHEDULE SPECIFIED RUNS FROM MAGNETIC TAPE C/U.

SM C/U F (SEC.9.6.1)

ON-SITE SYMBIONT CONTROL MESSAGES. 'F' FUNCTIONS INCLUDE

A- CONTINUE OPERATIONS ON SPECIFIED CHANNEL/UNIT.
E- END OF INPUT
R- REPRINT OR REPUNCH IN THE FORM RXX.
T- TERMINATE CURRENT FILE ON SPECIFIED C/U.
X- DELETE FILE FROM OUTPUT QUEUE. C/U IS REPLACED WITH EITHER 'PR' FOR PRINT QUEUE OR 'PU' FOR PUNCH QUEUE. IN THE FORM - SM PR X RUNID/FILENAME-.

SR SITEID RS F (SEC.9.7)

REMOTE SYMBIONT CONTROL MESSAGES. 'RS' DEFINITIONS ARE :

CR- CARD READ OPERATION

CP- CARD PUNCH OPERATION
PN- PRINT OPERATION
PR- PAPER TAPE READ OPERATION
PP- PAPER TAPE PUNCH OPERATION.

'F' FUNCTIONS INCLUDE:

A- CONTINUE REMOTE OPERATION
H- HALT REMOTE OPERATION
E- END OF INPUT
R- REPRINT OR REPUNCH
T- TERMINATE CURRENT FILE FOR 'RS' OPERATION
X- DELETE ALL REMAINING FILES IN THE 'RS' QUEUE
FOR 'SITEID'.

TC C/U

TERMINATE SYMBIONT PROBE CYCLE ON SPECIFIED C/U.

I/O ERROR MESSAGES

C/U INTLK

SYMBIONT CONTROLLED DEVICES

SYMBIONT OPERATIONS ARE RESUMED WITHOUT AN OPERATOR KEYIN WHEN THE INTERLOCK CONDITION HAS BEEN CLEARED. IF ANY RECOVERY PROCEDURE IS REQUIRED, THE UNSOLICITED MESSAGE 'SM' MUST BE TYPED IN PRIOR TO REMOVING INTERLOCK. NO DIRECT RESPONSE TO THIS MESSAGE IS REQUIRED.

C/U ILCHAR AES

CARD CONTAINING AN ILLEGAL CHARACTER WAS READ. THIS CARD AND THE TWO FOLLOWING ARE THE TOP THREE CARDS IN THE STACKER.

A- CONTINUE THE INPUT OPERATION
E- END OF INPUT, RUN STREAM IS ENTERED INTO SCHEDULE.
S- TERMINATE RUN INPUT FILE WITH ILLEGAL CHARACTER
CARD AND SEARCH INPUT STREAM FOR @RUN CONTROL CARD.

C/U PNCHK AT

CARD PUNCH CHECK ERROR WAS DETECTED. THE ERROR CARD AND TWO FOLLOWING WILL BE CHANNELLED TO THE ERROR STACKER.

A- CONTINUE PUNCHING FILE. THE ERROR CARD AND TWO
FOLLOWING CARDS ARE REPUNCHED
T- TERMINATE PUNCH FILE.

C/U RUN SEARCH A

THIS MESSAGE IS TYPED ON THE CONSOLE PRINTER IF A @RUN STATEMENT IS NOT ENCOUNTERED AS THE FIRST NON-BLANK STATEMENT OF THE CONTROL STREAM OR AS THE FIRST NON-BLANK STATEMENT FOLLOWING @FIN. THE INPUT MODE IS CONTINUED UNTIL A @RUN OR @FIN IS ENCOUNTERED, AT WHICH POINT THE INPUT MODE IS SUSPENDED UNTIL THE 'A' RESPONSE IS RECEIVED FROM THE KEYBOARD. SINCE THIS IS A TYPE ONLY MESSAGE,

```
    THE 'A' RESPONSE IS ENTERED WITH THE SM
    UNSOLICITED MESSAGE.

 C/U TIMOUT FUNCTION

    THIS IS THE RESULT OF THE 1108 NOT RECEIVING A RESPONSE
    FROM THE 1004 ON THE LAST FUNCTION SUBMITTED.  THE 1108 SYMBIONT
    WILL CONTINUE TO SEND THE LAST FUNCTION TO THE 1004 IN AN
    ATTEMPT TO COMPLETE THE OPERATION.  ANY CORRECTIVE ACTION,
    OTHER THAN CONTINUATION, MUST BE SUBMITTED TO THE SYMBIONT
    VIA THE 'SM' MESSAGE.  THE 'FUNCTION' DEFINITIONS
    ARE 'PRINT', 'PUNCH' AND 'READER'.


         TYPE AND READ
         -------------

 RUNID SPFORM C/U AQ
      USER'S TEXT

    THIS MESSAGE REQUIRES THAT A SPECIAL FORM BE LOADED ON THE
    SPECIFIED C/U.  THE 'A' RESPONSE INITIATES THE OUTPUT.
    THE 'Q' RESPONSE RE-ENTERS THIS FILE INTO THE APPROPRIATE
    QUEUE.


         TYPE
         ----

    SYMBIONT FILE AREA FILLED.

    AREA CAN BE EXPANDED WITH THE SM MESSAGE OPTION 'EX'.

 RUNID PRINT EXCEEDED

 RUNID PUNCH EXCEEDED
            C/U PRINT ERROR
            C/U PRINT ABNML
            C/U PUNCH B JAM
            C/U PUNCH ERROR
            C/U PUNCH ABNML
```

11.3.4. COMMUNICATIONS ACTIVITY MESSAGES

11.3.4.1. COMMUNICATIONS DEVICE HANDLER

MESSAGES ORIGINATING IN THE COMMUNICATIONS DEVICE HANDLERS ARE DOCUMENTED IN THE CHAPTER ENTITLED 'INPUT/OUTPUT DEVICE HANDLERS'.

11.3.4.2. REMOTE SYMBIONT MESSAGES

```
         ERROR
         -----
  C/U NO RESPONSE

    THIS MESSAGE INDICATES THAT OUTPUT ON LINE U WAS ATTEMPTED
    BY THE SYSTEM, BUT NO RESPONSE WAS RECEIVED FROM THE CENTRAL
    SITE UNIT ON LINE U IN AN ALLOTTED TIME INTERVAL.  THIS
    INDICATES A POSSIBLE HARDWARE FAILURE OF A UNIT THAT WAS SUP-
    POSED TO BE ON-LINE AND OPERATIONAL.  POTENTIAL CAUSES OF THIS
    ERROR ARE LOSS OF POWER, OFF-LINE STATUS OF UNIT, CONTINUOUS
    MASTER CLEAR CONDITION PRESENT FOR UNIT, OR AN IMPROPER ESI
    BASE VALUE IF LINE U IS ON A CTMC.  THE LINE U HAS BEEN AUTO-
    MATICALLY ASSIGNED A SYSTEM DOWN STATUS SO THAT THE OPERATOR
```

MAY CORRECT THE PROBLEM.

```
     TYPE AND READ
     -------------
 C/U PARITY ERR
```

THIS MESSAGE INDICATES THAT CONSECUTIVE PARITY ERRORS WERE ENCOUNTERED ON REPEATED ATTEMPTS BY SECOND LEVEL SUPPORT TO TRANSMIT A BLOCK OF DATA TO OR FROM THE REMOTE TERMINAL PRESENTLY CONNECTED TO LINE U. SECOND LEVEL TRANSMISSIONS ARE GENERALLY PERFORMED ONLY BY SYMBIONTS.

```
   A- REPEAT THE TRANSMISSION AGAIN OF THE SAME BLOCK OF DATA
      ON LINE U FOR WHICH THE ERRORS HAD OCCURRED
   D- DECLARE THE LINE U TO BE DOWN AND NO LONGER AVAILABLE
      FOR USE BY ANY PART OF THE SYSTEM
C/U DIAL NY
```

THIS MESSAGE INDICATES THAT USE OF LINE U IS REQUESTED BY THE SYSTEM AND ITS USE REQUIRES A MANUAL DIAL CONNECTION TO N TO BE PERFORMED BY THE OPERATOR.

```
   Y- YES MEANING THE SPECIFIED DIAL CONNECTION HAS BEEN
      COMPLETED
   N- NO MEANING THE REQUESTED DIAL CONNECTION HAS NOT BEEN
      MADE

     TYPE
     ----
 C/U HANGUP
```

THIS MESSAGE INDICATES THAT USAGE OF LINE U HAS BEEN COMPLETED AND A MANUAL DISCONNECT IS TO BE PERFORMED BY THE OPERATOR.

11.3.5. HARDWARE CONFIDENCE MESSAGES

A COMPUTER MALFUNCTION MESSAGE IS DISPLAYED IF A USER PROGRAM HAS BEEN TERMINATED BECAUSE OF HARDWARE PROBLEMS AND/OR THE EXECUTIVE SYSTEM REQUIRES OPERATOR ACTION. THE FORMAT OF THIS MESSAGE IS :

```
  (MALFUNCTION) - (ERROR ADDRESS) - (STATUS) - (ACTION)
 WHERE,

  MALFUNCTION IS : CONTROL MEMORY FAILURE, I/O DATA PARITY
                   ERROR, CORE STORAGE PARITY ERROR OR POWER
                   FAILURE.

  STATUS IS :      RUN IDENTITY TERMINATED(ITEM MAY BE ABSENT)
                   EXEC DOWN              (ITEM MAY BE ABSENT)
                   CHANNELS DOWN          (ITEM MAY BE ABSENT)
 AND

  ACTION IS :      INITIALIZE SYSTEM OR MAINTENANCE REQUIRED.
```

11.3.6. INITIAL LOAD MESSAGES

THE INITIAL LOAD ROUTINE PRODUCES TWO TYPES OF CONSOLE MESSAGES. THE FIRST TYPE REQUIRES A RESPONSE AND IS DEPENDENT UPON A JUMP SWITCH BEING SET. THE SECOND TYPE IS AN ERROR MESSAGE AND REQUIRES NO RESPONSE. THE FORMAT OF THESE TWO TYPES OF MESSAGES IS GIVEN BELOW.

```
TYPE 1

               D,P  MOD
        AND
               P,T  MOD

TYPE 2
               ERR: CH CC SSSSSS DDDDDD AAAAAA
               ERR: CH CC DDDDDD AAAAAA
               ERR: DDDDDD AAAAAA
```

FOR EXPLANATION OF THESE TWO TYPES OF MESSAGES SEE THE SECTION ENTITLED INITIAL LOAD CONSOLE COMMUNICATIONS.

11.4. USER CONSOLE REQUESTS

11.4.1. SYSTEM ENTRANCE-COM$

TO REQUEST USE OF THE CONSOLE DISPLAY, REGISTER A0 MUST CONTAIN THE ADDRESS OF THE CONSOLE COMMUNICATION PACKET, FOLLOWED BY AN 'ER COM$'. THE PACKET IS IMMEDIATELY CHECKED FOR THE TYPE OF OPERATION AND FOR LEGAL ACCESS WORDS. CONTROL FROM THE CONSOLE SECTION IS EITHER RETURNED TO THE 'ERROR MODE' ACTIVITY, OR TO THE INSTRUCTION FOLLOWING THE COMMUNICATION ENTRY.

THE CALLING SEQUENCE IS:

```
        L,U  A0,PKT
        ER   COM$
        NORMAL RETURN
```

WHERE 'PKT' IS THE ADDRESS OF THE COMMUNICATION PACKET WHICH DEFINES LOCATIONS AND LENGTHS OF CONSOLE MESSAGE TEXTS. THE PACKET FORMAT IS:

```
35      30        24         18                            00
-------------------------------------------------------------
:   S   :     C    :     W    :              NCI            :
-------------------------------------------------------------
: PW    :    MCC              :     TYPE ADDRESS            :
-------------------------------------------------------------
:          ICC                :          READ ADDRESS       :
-------------------------------------------------------------
```

ONE OF TWO FUNCTIONS, 'TYPE' OR 'TYPE AND READ', IS PERFORMED BY COM$ DEPENDENT ON THE PACKET DEFINITION. THE THREE WORD PACKET MUST ALWAYS BE DEFINED, AND THE FUNCTION IS DETERMINED BY WORD THREE. IF THE UPPER HALF OF WORD THREE IS EQUAL TO ZERO, THE 'TYPE' OPERATION IS PERFORMED. IF WORD THREE IS UNEQUAL TO ZERO, THE 'TYPE AND READ' FUNCTION IS PERFORMED. PW MAY BE ENTERED AS A NUMBER FROM 1 TO 6 IN ORDER TO DIRECT HARD COPY LOGGING OF A MESSAGE TO A PARTICULAR PAGEWRITER OF A CRT DISPLAY CONSOLE IF MORE THAN ONE IS AVAILABLE. IF PW IS ZERO THE BASIC SYSTEM PAGEWRITER IS USED.

FUNCTION DEFINITIONS

1-TYPE: THIS FUNCTION ALLOWS THE PROGRAM TO REQUEST THAT A SPECIFIED NUMBER OF CHARACTERS BE TRANSFERRED TO THE TYPEWRITER OR CRT DISPLAY FROM CONTIGUOUS MEMORY LOCATIONS BEGINNING WITH 'TYPE ADDRESS'. THE MESSAGE CHARACTER COUNT (MCC) IS THE NUMBER OF CHARACTERS IN THE MESSAGE TEXT. THIS FUNCTION REQUIRES THE UPPER HALF OF THE THIRD WORD OF THE PACKET TO BE EQUAL TO ZERO. IF THE VALUE OF MCC IS ZERO, NO TYPEOUT WILL OCCUR.

2-TYPE AND READ: THIS FUNCTION ALLOWS THE PROGRAM TO REQUEST A SPECIFIED NUMBER OF CHARACTERS BE TRANSFERRED TO THE TYPEWRITER

CRT, AND THAT A SPECIFIED NUMBER OF CHARACTERS BE ACCEPTED FROM THE KEYBOARD IN REPLY. THE RUNID* OF THE PROGRAM IS DISPLAYED PRECEDING THE PROGRAM MESSAGE TO THE TYPEWRITER OR CRT DISPLAY. THE OPERATOR IS EXPECTED TO TYPE IN A REPLY TO THE MESSAGE. THE INCOMING MESSAGE IS TRANSFERRED TO 'READ ADDRESS' IN CONTIGUOUS MEMORY LOCATIONS. IF THE MAXIMUM INCOMING CHARACTER COUNT (ICC) IS EXCEEDED, THE TYPE IS OMITTED AND THE OPERATOR IS REQUESTED TO RETYPE THE MESSAGE. THE NUMBER OF CHARACTERS ACTUALLY RECEIVED AS INPUT IS STORED INTO 'NCI'. THIS FUNCTION IS DETERMINED BY THE PRESENCE OF THE 'READ ADDRESS' ENTRY. IF THE NUMBER OF CHARACTERS OUTPUT (MCC) IS ZERO, THE REQUEST WILL BE IGNORED.

THE USER MAY DIRECT THE MESSAGE TO ANY CONSOLE BY ENTERING THE APPROPRIATE CATEGORY CODE IN THE 'C' FIELD. THESE CODES ARE:

CODE	CATEGORY
00	SYSTEM CONTROL
01	INPUT/OUTPUT ACTIVITY
02	COMMUNICATIONS ACTIVITY
03	HARDWARE CONFIDENCE

IF THE CONSOLE ON WHICH THE MESSAGE IS TO BE DISPLAYED HAS MORE THAN ONE PAGEWRITER, THE PARTICULAR PAGEWRITER ON WHICH LOGGING IS TO BE DONE MAY BE SPECIFIED BY SETTING ITS NUMBER (FROM 1 THROUGH 6) IN PW. IF NO VALUE APPEARS IN PW THE MESSAGE WILL BE LOGGED ONLY ON THE SYSTEM PAGEWRITER (NUMBER 1).

THE OUTPUT CHARACTERS OF THE MESSAGE TEXT ARE OBTAINED FROM SUCCESSIVE SIXTHS OF A WORD, BEGINNING WITH THE MOST SIGNIFICANT SIXTH OF LOCATION 'TYPE ADDRESS'. SUCCEEDING WORDS ARE OBTAINED FROM CONSECUTIVELY INCREASING ADDRESSES.

THE OUTPUT MESSAGE CHARACTER COUNT (MCC) IS RESTRICTED TO A VALUE OF 50 CHARACTERS OR LESS. EACH CHARACTER IS EDITED, AND MASTER SPACES (00) ARE ELIMINATED FROM THE MESSAGE, BUT ARE INCLUDED IN THE MCC DEFINITION. EACH LINE OF THE DISPLAY IS RESTRICTED TO A MAXIMUM OF 64 CHARACTERS. IF A CARRIAGE RETURN OR STOP CODE IS NOT ENCOUNTERED AS THE 65TH CHARACTER OF A LINE OF TEXT TO BE DISPLAYED, THE CONSOLE ROUTINE WILL PERFORM CARRIAGE RETURN AND LINE FEED FUNCTIONS. THE OVERFLOW CHARACTERS WILL BE DELETED ON THE CRT DISPLAY CONSOLE.

INPUT CHARACTERS FROM THE 'TYPE AND READ' MESSAGE ARE STORED IN SUCCESSIVE SIXTHS OF A WORD, BEGINNING WITH THE MOST SIGNIFICANT SIXTH OF LOCATION 'READ ADDRESS'. SUCCEEDING WORDS ARE STORED AT CONSECUTIVELY INCREASING ADDRESSES. IF THE LAST WORD OF THE INPUT MESSAGE DOES NOT CONTAIN SIX CHARACTERS, THE REMAINDER OF THE WORD IS FILLED WITH FIELDATA SPACE CODES (05). THE INPUT CHARACTER COUNT (ICC) IS RESTRICTED TO A COUNT OF 50 CHARACTERS OR LESS. THE END OF MESSAGE SYMBOL IS NOT TRANSFERRED TO THE MESSAGE AREA.

11.4.2. PROCEDURE DEFINITIONS-C$OM AND C$OMPK

THE PROCEDURE 'C$OM' PROVIDES THE CONSOLE COMMUNICATION CALLING SEQUENCE. THE PROCEDURE DEFINITION IS:

```
C$OM  PKT
```

THIS PROCEDURE GENERATES THE TWO WORD CALLING SEQUENCE REQUIRED FOR REFERENCING THE CONSOLE HANDLER.

THE PROCEDURE 'C$OMPK GENERATES THE CONSOLE COMMUNICATION PACKET.

```
PKT  C$OMPK,C  MCC,TYPE ADDRESS,ICC,READ ADDRESS
```

'PKT' IS A PROGRAMMER DEFINED LABEL. THIS PROCEDURE GENERATES THE THREE WORD PACKET REQUIRED FOR REFERENCING THE CONSOLE HANDLER.

11.4.3. STATUS CODES

CONTROL IS RETURNED TO A USER WITH STATUS = 0 AS SOON AS THE IMAGE IS SAVED FOR A TYPE ONLY REQUEST. FOR A TYPE AND READ REQUEST, THE USER IS FORCED TO A WAIT CONDITION. THE STATUS WILL BE SET TO 00 AND THE WAIT REMOVED WHEN THE INPUT HAS BEEN RECEIVED.

EACH REQUEST SUBMITTED TO THE CONSOLE SECTION IS VALIDATED FOR LEGAL PACKET DEFINITION. ANY ERROR WHICH IS DETECTED IN THIS DEFINITION WILL TERMINATE THE REQUESTING PROGRAM, OR RETURN CONTROL TO THE SPECIFIED ERROR TERMINATION ACTIVITY. THE ERROR CODE IS STORED IN THE FIRST WORD OF THE PACKET ('S' FIELD), AND IN THE FIRST WORD OF THE TERMINATION ACTIVITY. FOR CONSOLE ERRORS, S1 OF THE ERROR STATUS WORD IS SET TO 00. UNLESS THE USER SETS ERROR RECOVERY MODE, THE ONLY VALUE OF THE STATUS HE WILL SEE IS ZERO. ALL OTHERS CAUSE IMMEDIATE TERMINATION OF THE PROGRAM.

OCTAL
CODE — DEFINITION

20 THE ADDRESS SPECIFIED IN REGISTER A0 IS NOT WITHIN THE PROGRAM LIMITS, OR DEFINES A PACKET SPLIT BETWEEN THE INSTRUCTION AND DATA BANKS OF THE PROGRAM.

22 THE I/O ACCESS WORD OF THE 'TYPE' BUFFER IS WHOLLY OR IN PART OUTSIDE OF THE PROGRAM AREA, OR SPLIT BETWEEN THE INSTRUCTION AND DATA BANK OF THE PROGRAM.

23 THE I/O ACCESS WORD OF THE 'READ' BUFFER IS WHOLLY OR IN PART OUTSIDE OF THE PROGRAM AREA, OR SPLIT BETWEEN THE INSTRUCTION AND DATA BANK OF THE PROGRAM.

24 THE MESSAGE CHARACTER COUNT FOR 'TYPE' OPERATION EXCEEDS 132 CHARACTERS.

25 THE INCOMING MESSAGE CHARACTER COUNT FOR A 'TYPE AND READ' OPERATION EXCEEDS 50 CHARACTERS.

12. FILE CONTROL SYSTEM

12.1. FILE SUPERVISION

THE FILE SUPERVISOR ROUTINES EXERCISE CENTRALIZED CONTROL OVER OPERATIONS ON ALL FILES WITHIN THE SYSTEM. DEPENDING UPON THE USAGE, THESE ROUTINES ARE LOCATED BOTH AS PART OF THE RESIDENT AND AS AUXILIARY ROUTINES STORED ON MAGNETIC DRUM. THE PRIMARY FUNCTIONS PERFORMED BY THE FILE SUPERVISOR CONSIST OF:

1. MAINTAINING A DIRECTORY OF BOTH CATALOGUED PERMANENT FILES AND RUN TEMPORARY FILES.
2. CONTROL ALLOCATION OF MASS STORAGE SPACE AS NEW FILES ARE ASSIGNED AND EXISTING FILES ARE EXPANDED.
3. PROVIDE AN INTERFACE BETWEEN THE WORKER PROGRAM AND THE MASS STORAGE DEVICE HANDLERS TO MAINTAIN THE ABSOLUTE ADDRESSES OF THE VARIOUS GRANULES OF EACH FILE.
4. INHIBIT ACCESS TO PARTICULAR AREAS OF MASS STORAGE FILES OR ASSIGNMENT OF MASS STORAGE FILES BY ALL RUNS CURRENTLY EXECUTING EXCEPT FOR THE RUN TO WHICH THE AREA OR FILE IS ASSIGNED EXCLUSIVELY.
5. PROVIDE A MEANS WHEREBY WORKER PROGRAMS CAN DETERMINE THE CURRENT ASSIGNMENT GIVEN TO A FILE NAME AND THE VARIOUS FIELDS OF THE MASTER DIRECTORY ITEM.

12.2. FILE ORGANIZATION

12.2.1. MASTER DIRECTORY

FOR EACH FILE KNOWN TO THE SYSTEM, OTHER THAN TEMPORARY FILES, AN ENTRY CONTAINING THE IDENTIFICATION AND CHARACTERISTICS OF THE FILE IS MAINTAINED BY THE SYSTEM IN A MASTER DIRECTORY OF FILES. THE PROCESS OF ENTERING A FILE INTO THE MASTER DIRECTORY IS REFERRED TO AS CATALOGUING AND IS EFFECTED BY THE ASG CONTROL STATEMENT OPTIONS. BY USE OF THE MASTER DIRECTORY THE SYSTEM REMAINS COGNIZANT OF THE USAGE OF MASS STORAGE AND MAGNETIC TAPES.

AN ENTRY EXISTS IN THE MASTER DIRECTORY CORRESPONDING TO EACH CATALOGUED FILE. THE INFORMATION CONTAINED IN EACH ENTRY INCLUDES THE FOLLOWING:

1. EXTERNAL NAME OF THE FILE INCLUDING QUALIFIERS.
2. PROJECT IDENTITY FROM THE RUN CONTROL STATEMENT.
3. ACCOUNT NUMBER FROM THE RUN CONTROL STATEMENT.
4. DATE AT WHICH THE FILE WAS CATALOGUED.
5. ACTIVITY OF THE FILE (INCLUDING DATE OF LAST REFERENCE).
6. USAGE AUTHORIZATION.
7. RECORDING MODE IF TAPE.
8. GRANULARITY AND NUMBER OF GRANULES ASSIGNED IF MASS .STORAGE.
9. NUMBER OF REELS OF TAPE AND TAPE REEL NUMBERS IF A TAPE FILE.
10. LINKAGE TO THE VARIOUS GRANULES IF A MASS STORAGE FILE.
11. F-CYCLE- ABSOLUTE AND RELATIVE.

THE EXTERNAL NAME AND QUALIFIERS ARE THOSE SPECIFIED ON THE ASG CONTROL STATEMENT. EACH OF THESE FIELDS IS TWO WORDS IN LENGTH. IF NO QUALIFIER IS GIVEN ON THE ASG STATEMENT, THEN THE PROJECT IDENTITY FROM THE RUN STATEMENT IS USED AS A QUALIFIER AND ONLY THOSE RUNS WITH THE SAME PROJECT IDENTITY CAN HAVE THIS FILE

ASSIGNED. HENCE THE EXTERNAL NAME AND QUALIFIER MUST BE UNIQUE ACROSS THE SYSTEM AND CAN BE ASSIGNED TO ANY RUN (IF NO USAGE LOCK EXISTS), BUT IF NO QUALIFIER EXISTS, THE EXTERNAL NAME DOES NOT HAVE TO BE UNIQUE ACROSS THE SYSTEM BUT ONLY WITHIN THE PROJECT.

THE ACTIVITY OF THE FILE IS MAINTAINED TO DETERMINE WHICH FILES TO ROLLOUT TO MAGNETIC TAPE IF MASS STORAGE IS NEARING THE OVERFLOW STATE.

THE MASTER DIRECTORY SHOWS THE RECORDING MODE IN EFFECT AT THE TIME A MAGNETIC TAPE IS ASSIGNED. THIS INCLUDES DENSITY, PARITY, AND NOISE CONSTANT. IF THE MODES ARE CHANGED WHILE CREATING THE FILE THE WORKER PROGRAM MUST MAKE NOTE OF THIS OCCURRENCE. IF MAGNETIC TAPE FILES ARE CATALOGUED WHICH WERE CREATED OUTSIDE OF THE SYSTEM OR WHICH HAD BEEN WRITTEN AND CATALOGUED IN SEPARATE RUNS, THE RECORDING MODE MUST BE ENTERED AS PARAMETERS AT THE TIME OF CATALOGUING IF OTHER THAN THE SYSTEM STANDARD. WHEN A CATALOGUED MAGNETIC TAPE FILE IS ASSIGNED TO A RUN, THE RECORDING MODES ARE SET TO THE CONDITIONS SAVED IN THE DIRECTORY.

THE TAPE REEL NUMBERS ARE OBTAINED FROM THE OPERATOR RESPONSE TO THE MOUNT MESSAGES AND ARE INSERTED INTO SUBSEQUENT MOUNT MESSAGES WHEN CATALOGUED FILES ARE ASSIGNED (SEE TAPE SWAPPING IN CHAPTER 10).

12.2.2. MASS STORAGE ALLOCATION

12.2.2.1. GENERAL

THE TERM 'MASS STORAGE' IS TAKEN TO MEAN ALL TYPES OF MAGNETIC DRUM (FH 432, FH 880, AND FASTRAND). MASS STORAGE IS ALLOCATED BY THE 1108 EXEC IN THREE BASIC TYPES:

1. A FIXED-LENGTH AREA FOR THE SYSTEM'S RESIDENCE.
2. A CONTIGUOUS FH432 OR FH880 AREA FOR PROGRAMS.
3. FASTRAND FORMAT FOR BOTH WORKER PROGRAMS AND THE SYSTEM USAGE.

THE FIXED LENGTH AREA OF MASS STORAGE USED BY THE SYSTEM IS ALLOCATED AT SYSTEM GENERATION TIME AND REMAINS FIXED IN SIZE AND LOCATION DURING OPERATION. THIS AREA IS LOADED WITH A COPY OF THE EXECUTIVE INCLUDING THE SYMBIONTS, AN AREA FOR STORING EXECUTIVE TABLES, AND A COPY OF THE PROCESSORS . THE NORMAL MODE CONSISTS OF ALLOCATING A CONTIGUOUS AREA OF THE FASTEST DRUM TYPE IN THE SYSTEM TO CONTAIN THESE COMPONENTS. AT SYSTEM GENERATION TIME THIS MODE CAN BE INFLUENCED BY THE 'SYSTEM' STATEMENT TO MOVE THE SYSTEM'S RESIDENCE TO A SLOWER SPEED DRUM (SEE SYSTEM GENERATION STATEMENTS - SYSTEM RESIDENCE).

THE 'DIVIDE' GENERATION STATEMENT CAN BE USED AT SYSTEM GENERATION TIME TO SPECIFY ONE OR MORE AREAS OF FH432 AND FH880 MAGNETIC DRUM WHICH IS TO BE ASSIGNED TO RUNS AS FIXED-LENGTH CONTIGUOUS AREAS (SEE SYSTEM GENERATION STATEMENTS - DIVIDE DRUM). THESE AREAS ARE INTENDED FOR THE SPECIAL CASES WHERE WORKER PROGRAMS NEED DIRECT USAGE OF HIGH SPEED MASS STORAGE AS A SCRATCH AREA WITHOUT ITEM HANDLING SUPPORT, AND AS SUCH IT IS EXPECTED TO BE A SMALL PERCENTAGE OF THE AVAILABLE AREA. THIS AREA IS STRICTLY SCRATCHING AREA AND IS NOT CATALOGUED IN THE MASTER DIRECTORY. THE 'DIVIDE' SYSTEMS GENERATION PARAMETER IS EXPRESSED AS A CHANNEL, UNIT NUMBER AND NUMBER OF LOCATIONS. THIS ASSIGNMENT IS MADE AT THE FIRST AVAILABLE LOCATIONS ON THE UNIT (LOCATION ZERO IF EXEC IS NOT ON THE SAME UNIT) AND EXTENDS AS FAR AS NECESSARY TO SATISFY THE REQUIREMENT INCLUDING ONTO THE NEXT DRUM UNIT OF A CHANNEL IF REQUIRED. ONLY ONE AREA CAN BE SPECIFIED FOR EACH UNIT IN THE CONFIGURATION.

AFTER SATISFYING THE TWO REQUIREMENTS DISCUSSED ABOVE, THE REMAINDER OF MASS STORAGE IS TREATED AS FASTRAND FORMAT AND IS ALLOCATED IN GRANULES OF 'TRACKS' AND 'POSITIONS'. A TRACK IS DEFINED AS 64 ADDRESSABLE AREAS OF 28 WORDS EACH OR 1792 WORDS OF STORAGE. A POSITION IS 64 TRACKS (4096 ADDRESSABLE AREAS OR 114,688 WORDS). AS AN EXTENSION TO THE MASTER DIRECTORY, THE EXECUTIVE MAINTAINS A TABLE LOCATING THE VARIOUS GRANULES WHICH ARE ALLOCATED TO A GIVEN FILE NAME. THIS TABLE IS STORED IN SECTOR SIZE AREAS OF FASTRAND AND ARE USED BY THE DEVICE HANDLERS TO CONVERT THE RELATIVE LOCATION FURNISHED IN THE REQUEST TO ABSOLUTE HARDWARE LOCATIONS; I.E., A

REQUEST TO READ AT ADDRESS 129 OF A FILE WITH A TRACK GRANULARITY WOULD REFER TO THE SECOND SECTOR OF THE THIRD TRACK ASSIGNED TO THE FILE. THIS REFERENCE TABLE ALLOWS VOIDS AND OVERLAPPING VARIOUS TYPES OF MASS STORAGE WITH A FILE.

WHEN A MASS STORAGE FILE IS INITIALLY ASSIGNED, ONLY THE NUMBER OF GRANULES REQUESTED IN THE ASG STATEMENT ARE ALLOCATED. AFTER THAT ONLY THOSE GRANULES NECESSARY TO SERVICE A GIVEN REQUEST ARE AUTOMATICALLY ASSIGNED; I.E., IF THE INITIAL REQUEST WAS THREE TRACKS AND A ONE TRACK WRITE WAS REQUESTED STARTING AT RELATIVE ADDRESS 256, THE RESULT WOULD BE AN ASSIGNMENT FOR TRACKS 1,2,3, AND 5, BUT NOT 4, UNTIL SUCH TIME AS A REFERENCE TO SOME RELATIVE ADDRESS FROM 192 TO 255 IS MADE.

THE HARDWARE WRITE LOCKOUT FEATURE OF FASTRAND-II IS SUPPORTED BY THE 'RESERVE' SYSTEM GENERATION PARAMETER. THIS ALLOWS ASSIGNING AN ABSOLUTE AREA OF FASTRAND TO A SPECIFIED FILE NAME AND CATALOGUING THE FILE IN THE MASTER DIRECTORY (SEE SYSTEM GENERATION STATEMENTS - RESERVE MASS STORAGE). A PROGRAM CAN THEN WRITE THE DESIRED INFORMATION INTO THIS FILE AFTER WHICH THE LOCKOUT SWITCH CAN BE SET. EXEC MAINTAINS NO KNOWLEDGE OF THE EXISTENCE OF THESE SWITCHES. THE 'RESERVE' STATEMENT SPECIFIES A CHANNEL AND UNIT NUMBER, AN ADDRESS, AND THE NUMBER OF GRANULES. IT IS NECESSARY THAT RESERVES ON A PARTICULAR UNIT START AT ADDRESS ZERO AND ARE CONTIGUOUS FOR ALL FILES INVOLVED. THE FINAL TRACK OF EACH MASS STORAGE UNIT WHICH HAS ANY PART DECLARED AS FASTRAND FORMAT IS USED AS THE START OF THE MASTER DIRECTORY FOR THE GIVEN UNIT AND HENCE CAN NOT BE INCLUDED WITHIN A 'RESERVE' STATEMENT.

12.2.2.2. SPACE AVAILABILITY CONTROL

THE FILE SUPERVISION ROUTINES AUTOMATICALLY AFFECT THE ASSIGNMENT OF ADDITIONAL INCREMENTS OF MASS STORAGE SPACE AS REQUIRED TO SATISFY THE NEEDS OF THE WORKER PROGRAMS. THE SPACE AVAILABILITY FUNCTION ALSO HANDLES RELEASE OF GRANULES TO THE AVAILABLE STATUS. RELEASE OF ANY PART OF A GRANULE WILL CAUSE THE RELEASE OF THE ENTIRE AREA.

SINCE FILES CAN BE RELEASED A GRANULE AT A TIME, IT IS POSSIBLE TO END UP WITH A VACUOUS FILE CATALOGUED IN THE SYSTEM WITH A MASTER DIRECTORY ITEM AND NO ALLOCATED SPACE.

12.2.2.3. MASS STORAGE UNIT AVAILABILITY CONTROL

IN ADDITION TO THE 'D' RESPONSE TO AN I/O ERROR MESSAGE, TWO UNSOLICITED MESSAGES ARE AVAILABLE TO CONTROL MASS STORAGE UNIT AVAILABILITY. THESE ARE:

```
DN C/U
UP C/U A
```

WHERE THE DOWN MESSAGE 'DN' CAUSES THE UNIT TO BE REMOVED FROM THE ON-LINE POOL OF DEVICES AND THE UP MESSAGE 'UP' PLACES THE DEVICE BACK INTO THE ON-LINE POOL.

ANY I/O REFERENCES TO A DOWNED UNIT CAUSES A UNIQUE STATUS CODE TO BE RETURNED IN THE REQUEST PACKET. LOOKUP FEATURES FOR PERMANENT FILES CATALOGED ON THE DOWNED UNIT REMAIN INTACT AND HENCE CAN BE RETRIEVED WHEN THE UNIT IS BROUGHT BACK UP PROVIDING THE 'A' FIELD IS NOT SPECIFIED. IF THE LETTER A IS SPECIFIED ALL OF THE MASS STORAGE UNIT IS ENTERED INTO THE AVAILABLE POOL.

WHEN THE UP UNSOLICITED MESSAGE IS USED WITHOUT THE A FIELD SPECIFIED, THE SYSTEM WILL ATTEMPT TO RESTORE THE FASTRAND FORMAT SECTION, IF ANY, TO THE CONDITION PRIOR TO BEING DECLARED DOWN. ANY DRUM FORMAT SECTION ALWAYS REVERTS TO THE AVAILABLE POOL. DURING THE COURSE OF RECOVERY,OCCURRENCE OF A FORMAT ERROR AS A RESULT OF DESTRUCTION WHILE OFF-LINE CAUSES THE MESSAGE:

```
FORMAT ERR 'FILE NAME' 'ACCT NO'
```

TO BE DISPLAYED ON THE CONSOLE, TO WHICH THE RESPONSE P FOR PURGE OR X FOR ABORT MAY BE GIVEN AS A VALID RESPONSE. THE P RESPONSE CAUSES THE SINGLE FILE TO BE PURGED FROM MASS STORAGE, AND THE X RESPONSE CAUSES THE UP OPERATION TO BE RESTARTED WITHOUT ANY RECOVERY. IT IS IMPORTANT TO NOTE THAT DURING RECOVERY, VALIDATION OF

INFORMATION READ FROM THE UNIT COMING UP IS NECESSARILY RESTRICTED TO THE DIRECTORY FOR THE UNIT AND HENCE COMPLETE ASSURANCE OF THE VALIDITY OF INFORMATION WITHIN THE FILES RECOVERED CAN NOT BE GUARANTEED. ALSO, THE ALLOCATION TABLES FOR FILES OPEN AT THE TIME THE UNIT GOES DOWN MAY NOT REFLECT THE CORRECT AREA ASSIGNMENTS AS CHANGES ARE ACCUMULATED IN CORE FOR THE OPEN FILES (THE UP FUNCTION INSURES NO CONFLICT FOR SPACE).

AT COMPLETION OF THE RE-ESTABLISHMENT OF ALL PRINT FILES, A LISTING OF THE DIRECTORY FOR THE GIVEN UNIT IS PRINTED IF THE A FIELD IS NOT SPECIFIED. OPEN FILES WITH NO CONFLICTS FOR SPACE ARE FLAGGED IN THE LISTING.

12.2.2.4. ROLLOUT OF FILES TO MAGNETIC TAPE

DEPENDING UPON THE AMOUNT OF AVAILABLE FASTRAND FORMAT MASS STORAGE, THE DEGREE OF USAGE GIVEN TO CATALOGUING FILES ON MASS STORAGE, AND THE MANNER IN WHICH FASTRAND FILES ARE ASSIGNED, THERE MAY BE OCCURRENCES DURING NORMAL OPERATION WHEN IT IS NECESSARY TO OBTAIN ADDITIONAL SPACE ON FASTRAND BY ROLLING OUT PERMANENT FILES TO MAGNETIC TAPE. THIS FEATURE IS PROVIDED AUTOMATICALLY BY THE EXECUTIVE. THE POINTS AT WHICH ROLLOUT IS TURNED ON AND OFF ARE EXPRESSED AS SYSTEM GENERATION PARAMETERS (SEE SYSTEM GENERATION STATEMENTS - ROLLOUT CONTROL POINTS). THE SYSTEM STANDARD FOR THESE PARAMETERS ARE SET SUCH THAT ROLLOUT STARTS WHEN THE AMOUNT OF AVAILABLE FASTRAND DROPS TO 6 POSITIONS (384 TRACKS) PLUS 1 POSITION FOR EACH FASTRAND UNIT OVER ONE AND ROLLS OUT FILES UNTIL THE AMOUNT OF AVAILABLE AREA HAS BEEN INCREASED BY 3 POSITIONS.

ROLLOUT TO MAGNETIC TAPE WILL OCCUR WHEN:

1. INSUFFICIENT UNUSED AREA EXISTS TO EXPAND A FILE DYNAMICALLY AFTER THE INITIAL REQUIREMENT HAS BEEN ASSIGNED BUT SUFFICIENT AREA EXISTS IN THE UNUSED PLUS PERMANENT AREA. THIS CONDITION WILL NOT OCCUR IF THE INITIAL REQUIREMENT ON THE ASG CARD IS LARGE ENOUGH.

2. AN ATTEMPT IS BEING MADE TO ASSIGN FACILITIES TO A RUN WHICH HAS A TOTAL FASTRAND REQUIREMENT GREATER THAN THE ROLLOUT START POINT AND THE LENGTH OF THE AREA REQUIRED PLUS THE TOTAL AREA ASSIGNED TO NONASSIGNED CATALOGUED FILES IS GREATER THAN THE TOTAL FASTRAND SPACE WITHIN THE CONFIGURATION. FOR THIS CASE ROLLOUT CONTINUES UNTIL THE PARTICULAR REQUIREMENT CAN BE SATISFIED.

3. AN ATTEMPT HAS BEEN MADE TO ASSIGN DIRECTORY SPACE ON A UNIT WHICH IS COMPLETELY ASSIGNED. FOR THIS CASE ONLY A SINGLE FILE IS ROLLED OUT FROM THE UNIT OF CONCERN.

THE ROLLOUT ROUTINE UTILIZES THE FILE ACTIVITY AND DATE OF CATALOGUING TO DETERMINE WHICH FILES ARE TO BE TRANSPOSED AT A GIVEN TIME. ALL FILE ACTIVITY WITH A FREQUENCY OF LESS THAN ONE REFERENCE PER 48 HOURS ARE CONSIDERED EQUAL. FILE SELECTION IS STARTED AT THE HIGHEST PRIORITY FASTRAND FORMAT UNITS WITH THE OLDEST FILES ROLLED OUT FIRST.

THE MAGNETIC TAPE IS LEFT EXTENDED BETWEEN REFERENCES TO ALLOW FUTURE TRANSFERS. EACH ROLLED OUT FILE IS MARKED IN THE DIRECTORY AS TO THE TAPE NUMBER (OPERATOR RESPONSE TO THE MOUNT MESSAGE) AND FILE POSITION ON THE TAPE.

A REQUEST TO ASSIGN A ROLLED OUT FASTRAND FILE CAUSES THE EXEC TO REQUEST MOUNTING OF THE PROPER MAGNETIC TAPE, UNLESS ALREADY MOUNTED, AND AUTOMATICALLY RETRIEVING THE FILE BACK TO FASTRAND.

ONCE THE EXEC ALLOCATES A TAPE UNIT FOR ROLLOUT PURPOSES, THE UNIT WILL REMAIN ASSIGNED TO THE EXEC UNTIL ALL FILES HAVE BEEN ROLLED BACK TO FASTRAND.

12.3. FILE REFERENCING

12.3.1. FILE CONTROL TABLE GENERAL FORMAT

FOR EACH EXTERNAL FILE NAME SPECIFIED IN THE ASG STATEMENTS THERE MUST EXIST WITHIN THE USER PROGRAMS A FILE CONTROL TABLE FOR EACH FILE NAME TO BE REFERENCED. THESE TABLES ARE USED FOR ALL COMMUNICATING BETWEEN THE PROGRAMS AND THE PERIPHERAL FILES. THE FILE CONTROL TABLE CONSISTS OF UP TO THREE PARTS DEPENDING UPON THE DEGREE OF INTERACTION ASSIGNED TO THE OPERATING SYSTEM. THESE PARTS ARE:

1. A REQUEST PACKET USED TO COMMUNICATE WITH ONE OF THE HANDLERS (TAPE OR DRUM, ARBITRARY DEVICE OR COMMUNICATIONS HANDLER).
2. AN AREA CONTAINING PARAMETERS CONTROLLING THE BLOCK BUFFERING PACKAGE.
3. AN AREA UTILIZED BY THE ITEM HANDLING PACKAGE.

THE REQUEST PACKET AREA OF THE FILE CONTROL TABLE IS ALWAYS REQUIRED. THE VARIOUS FORMATS OF THE PACKETS ARE DEFINED IN CHAPTER 10. REGARDLESS OF THE PARTS OF THE PACKET WHICH VARY WITH THE HANDLER INVOLVED, THE FIRST TWO WORDS MUST ALWAYS BE THE INTERNAL FILE NAME. ONLY THE PACKET FORMAT FOR TAPE OR MASS STORAGE CAN BE USED IF THE BLOCK BUFFERING AND ITEM HANDLING SECTIONS ARE TO BE INCLUDED IN THE FILE CONTROL TABLE. FOR THIS CASE THE PACKET LENGTH MUST BE FIXED AT 8 WORDS INCLUDING THE TWO-WORD FILE NAME. THE FORMAT OF THE BLOCK BUFFERING AND ITEM HANDLING AREAS OF THE FILE CONTROL TABLE ARE DEFINED IN DETAIL IN CHAPTER 13.

IF THE USER PROGRAMS REFERENCE THE HANDLERS DIRECTLY, THEY CAN DO SO THROUGH MORE THAN ONE FILE CONTROL TABLE EXCEPT FOR THE COMMUNICATIONS MULTIPLEXOR HANDLER WHERE ONLY ONE IS ALLOWED BECAUSE OF HANDLER INITIALIZATION BASED UPON PARAMETERS IN THE TABLE. THE SUGGESTED MODE IS A SINGLE TABLE WITH DYNAMIC CHANGES WITHIN THE TABLE BETWEEN REFERENCES. THIS PROMOTES EFFICIENT OPERATION IN THE HANDLERS. USE OF THE SAME FILE CONTROL TABLE FOR REFERENCING SEVERAL FILES AT THE DEVICE HANDLER LEVEL CAN BE ACHIEVED BY REPLACING THE INTERNAL FILE NAME IN THE FIRST TWO WORDS OF THE TABLE PROVIDING THE REFERENCES ARE SERIAL WITH A WAIT FOR COMPLETION BEFORE MAKING ANY CHANGES TO THE TABLE.

12.3.2. DIRECTORY ITEM RETRIEVAL

A LINKAGE TO THE EXECUTIVE IS AVAILABLE TO THE USER TO RETRIEVE THE MASTER DIRECTORY ITEM FOR ANY FILE CURRENTLY ASSIGNED TO THE REFERENCING RUN. THIS IS ACCOMPLISHED BY THE CALLING SEQUENCE:

```
LA A0, PKT ADDR
ER DITEM$
```

THE PACKET ADDRESSED BY THE A0 REGISTER CONSISTS OF THREE WORDS, A TWO WORD INTERNAL FILE NAME OF A FILE CURRENTLY ASSIGNED TO THE RUN FOLLOWED BY A SINGLE WORD WITH THE ADDRESS OF THE START OF A 28 WORD AREA. THE LOWER HALF OF THE THIRD WORD CONTAINS THE ADDRESS WHERE THE DIRECTIORY ITEM IS TO BE PLACED AND THE UPPER HALF CONTAINS A NUMBER WHICH DENOTES THE SECTION OF THE DIRECTORY ITEM TO DUMP. THE SECTION NUMBER STARTS WITH ZERO FOR THE FIRST SECTOR SIZE BLOCK OF THE MASTER DIRECTORY AND IS OF SIGNIFICANCE ONLY IF THE FILE HAS A GRANULE NUMBER OF 3 OR MORE OR IS 3 OR MORE REELS IN LENGTH. THE SECOND AND SUBSEQUENT DIRECTORY ITEMS CONSIST OF A LINK WORD (CHANNEL AND ADDRESS) IN WORDS 1 AND 2 (SUBSEQUENT AND PRECEDING ITEMS RESPECTIVELY) AND 26 WORDS CONTAINING REEL NUMBERS OR ABSOLUTE GRANULE LOCATIONS IF AN ASSIGNMENT EXISTS. IF AN ADDITION SECTOR IS REQUIRED, NO GRANULE ADDRESS OR REEL NUMBERS ARE MAINTAINED IN THE INITIAL DIRECTORY ITEM. THE SECOND DIRECTORY ITEM STARTS WITH GRANULE 1 OR REEL 1 IN THE THIRD WORD. THE MASTER DIRECTORY ITEM HAS THE FOLLOWING FORMAT WHERE THE BLANK FIELDS ARE USED FOR SYSTEMS INDICATORS:

```
     35          29          23          17          11          05         00
   ---------------------------------------------------------------------------
01 :U:       LINK TO NEXT SECTION OF DIRECTORY ITEM OR ZERO                  :
   ---------------------------------------------------------------------------
02 :                         EXTERNAL NAME                                   :
   ---------------------------------------------------------------------------
03 :                         EXTERNAL NAME                                   :
   ---------------------------------------------------------------------------
04 :                           QUALIFIER                                     :
   ---------------------------------------------------------------------------
05 :                           QUALIFIER                                     :
   ---------------------------------------------------------------------------
06 :                         PROJECT IDENTITY                                :
   ---------------------------------------------------------------------------
07 :                         PROJECT IDENTITY                                :
   ---------------------------------------------------------------------------
08 :                         BOOT PARAMETER                                  :
   ---------------------------------------------------------------------------
09 :                      BACK LINK TO SECTOR 0                              :
   ---------------------------------------------------------------------------
10 :                         ACCOUNT NUMBER                                  :
   ---------------------------------------------------------------------------
11 :                         ACCOUNT NUMBER                                  :
   ---------------------------------------------------------------------------
12 :                      TIME OF CATALOGUING                                :
   ---------------------------------------------------------------------------
13 :                      TIME OF LAST REFERENCE                             :
   ---------------------------------------------------------------------------
14 :ROLL OUT POSITION  : ROUT     : NUMBER OF TIMES ASSIGNED                 :
   ---------------------------------------------------------------------------
15 :INITIAL GRANULE CNT:GRANULE  :MAXIMUM GRANULE COUNT                      :
   ---------------------------------------------------------------------------
16 :NUMBER OF GRANULES/REELS     :HIGHEST GRANULE NUMBER                     :
   ---------------------------------------------------------------------------
17 :TAPE MODE SETTINGS :MED TYPE :NOISE CONSTANT VALUE                       :
   ---------------------------------------------------------------------------
18 :READ ONLY: EU       :         :CURRENT AS,MT :                           :
   ---------------------------------------------------------------------------
19 :BEING REL:PUB/PRIV ::   ABS F-CYCLE     :                                :
   ---------------------------------------------------------------------------
20 :                TRACK-MINUTES OF FH 432 USAGE                            :
   ---------------------------------------------------------------------------
21 :                TRACK-MINUTES OF FH 880 USAGE                            :
   ---------------------------------------------------------------------------
22 :                TRACK-MINUTES OF F-II OR FASTBAND USAGE                  :
   ---------------------------------------------------------------------------
23 :                                                                         :
   ---------------------------------------------------------------------------
24 :                        :  LARGEST TRACK REFERENCED                      :
   ---------------------------------------------------------------------------
25 :                LINK WORD FOR FILE CONTROL TABLE IF ANY                  :
   ---------------------------------------------------------------------------
26 :  RUN ID OF ASG IF ASG COUNT=1 OR LINK TO EXEC BUFFER AREA               :
   ---------------------------------------------------------------------------
27 :  ABSOLUTE LOCATION OF GRANULE 1 OR REEL 1 REEL NUMBER                   :
   ---------------------------------------------------------------------------
28 :  ABSOLUTE LOCATION OF GRANULE 2 OR REEL 2 REEL NUMBER                   :
   ---------------------------------------------------------------------------
```

MOST FIELDS OF THE DIRECTORY ARE SELF EXPLANATORY FROM THE ILLUSTRATION EXCEPT WHEN:

1. THE TIME FIELDS ARE RECORDED AS SECONDS SINCE MIDNIGHT IN H1, AND MONTH, DAY, YEAR IN S4, S5, AND S6

RESPECTIVELY.

2. ROUT TYPE IS THE TAPE TYPE ON WHICH THE FILE IS ROLLED OUT AND IS THE SAME FORMAT AS MED TYPE (SEE BELOW). THIS FIELD IS ALWAYS ZERO WHEN THE FILE IS ASSIGNED.
3. ROLLOUT POSITION IS THE COUNT OF END-OF-FILE MARKS ON THE ROLLOUT TAPE PRECEDING THE START OF THE FILE.
4. GRANULE IS THE GRANULARITY OF THE FILE (ZERO FOR TAPE, 1 FOR TRACK AND 2 FOR POSITION).
5. MED TYPE IS:

```
01-8C 7-TRACK
02-6C 7-TRACK
03-8C HARDWARE TRANSLATE
04-6C HARDWARE TRANSLATE
05-8C 9-TRACK
06-6C 9-TRACK
07-4C
30-F-II FASTRAND
31-FASTBAND ON F-II
32-FASTRAND FORMAT ON FH 432
33-FASTRAND FORMAT ON FH 880
```

6. THE MODE SETTING IS BY MASTER BIT AS FOLLOWS:

```
BIT 24 1= TRANSLATION MODE SET
BIT 25 AND 26
        01= LOW DENSITY
        10= MEDIUM DENSITY
        11= HIGH DENSITY
BIT 27 0= ODD PARITY
       1= EVEN PARITY
```

7. THE READ ONLY FIELD IS NOT ZERO IF THE PUBLIC MODE IS SET.
8. EU IS 0 IF NOT ASSIGNED WITH EXCLUSIVE USE, 1 IF ASSIGNED EXCLUSIVELY, OR 2 IF ANY RUN IS WAITING FOR THE EXCLUSIVELY ASSIGNED FILE.
9. 'BEING REL' IS SET IF ANY RUN HAS FREED THE FILE AND THE ASG STATEMENT SHOWED DECATALOGUING. THE DECATALOGUE FUNCTION IS DELAYED UNTIL THE ASSIGNMENT COUNT IS ZERO.
10. PUB/PRIV=1 IF THE FILE IS PUBLIC OR =0 IF PRIVATE.
11. MAX F-CYC IS THE COUNT OF BACKUPS ALLOWED BY THE SYSTEM.
12. REL F-CYC IS THE RELATIVE F-CYCLE OF THE CURRENT ITEM. A NEGATIVE RELATIVE NUMBER IS INDICATED BY BIT 17 SET.
13. IF THE FILE IS ROLLED OUT, THE ROLLOUT REEL NUMBER IS MAINTAINED IN THE LOCATION OF GRANULE 1 WORD.

12.3.3. FILE ASSIGNMENT DETERMINATION

FILE SUPERVISION PROVIDES A LINKAGE FOR THE USER TO DETERMINE THE EQUIPMENT DESCRIPTION ASSOCIATED WITH A PARTICULAR FILE NAME. THE LINKAGE IS:

```
LA A0,PKT ADDR
ER FACIL$
```

WHERE THE A0 REGISTER IS LOADED WITH AN ADDRESS OF A 9-WORD PACKET WITH THE FIRST TWO WORDS CONTAINING THE INTERNAL NAME FOR WHICH THE ASSOCIATION IS REQUESTED. THE FACIL$ ROUTINE FILLS IN THE REMAINING FIELDS OF THE PACKET. THE EXTERNAL NAME FIRST WORD IS ZERO ON RETURN IF THE FILE NAME SUPPLIED IN THE INTERNAL NAME FIELD IS NOT ATTACHED TO AN EXTERNAL NAME OR IS NOT AN EXTERNAL NAME ITSELF. THE LAST THREE WORDS ARE DEVICE DEPENDENT AS SHOWN BELOW.

```
   35       29        23        17                             00
   ---------------------------------------------------------------
 1 :                        INTERNAL NAME                        :
   ---------------------------------------------------------------
 2 :                        INTERNAL NAME                        :
   ---------------------------------------------------------------
 3 :                        EXTERNAL NAME                        :
   ---------------------------------------------------------------
 4 :                        EXTERNAL NAME                        :
   ---------------------------------------------------------------
 5 :                        QUALIFIER                            :
   ---------------------------------------------------------------
 6 :                        QUALIFIER                            :
   ---------------------------------------------------------------
 7 :EQUIP    :  DEVICE  DEPENDENT:REL F-CYC:  ABS F-CYCLE        :
   ---------------------------------------------------------------
 8 :                      DEVICE DEPENDENT                       :
   ---------------------------------------------------------------
 9 :                      DEVICE DEPENDENT                       :
   ---------------------------------------------------------------
```

```
EQUIP = 00-NO EQUIPMENT ASSIGNED TO FILE NAME OR, IF FIRST
           WORD OF EXTERNAL NAME FIELD IS 0,NO FIND ON FILE
           NAME.
      = 01-8C TAPE 7-TRACK FORMAT
      = 02-6C TAPE 7-TRACK FORMAT
      = 05-8C TAPE 9-TRACK FORMAT
      = 06-6C TAPE 9-TRACK FORMAT
      = 07-4C TAPE
      = 16 3A TAPE
      = 17 2A TAPE
      = 20-FH 432 DRUM FORMAT
      = 21-FH 880 DRUM FORMAT
      = 22-FH1782 DRUM FORMAT
      = 30-F-II FASTRAND
      = 31-FASTBAND ON F-II FASTRAND
      = 32-FASTRAND FORMAT ON FH 432
      = 33-FASTRAND FORMAT ON FH 880
      = 34-FASTRAND FORMAT ON FH1782
      = 40-CARD READER
      = 41-CARD PUNCH
      = 42-PAPER TAPE READER
      = 43-PAPER TAPE PUNCH
      = 44-HIGH SPEED PRINTER
      = 50-STANDARD 1004
      = 51-NON STANDARD 1004
      = 70-CTS
      = 71-WTS
      = 72-CTMC
      = 77-NON STANDARD DEVICE
```

THE REL F-CYC IS THE RELATIVE F-CYCLE WHERE NEGATIVE NUMBERS ARE INDICATED BY A ONE BIT IN POSITION 17.

THE FORMAT OF THE LAST 3 WORDS FOR FASTRAND IS:

```
   ---------------------------------------------------------------
 7 :  EQUIP  :EU-R/W   : GRAN    :REL F-CYC:   ABS F-CYCLE       :
   ---------------------------------------------------------------
 8 :                          ASG OPTIONS                        :
   ---------------------------------------------------------------
 9 :  INITIAL GRANULE COUNT      :    MAXIMUM GRANULE COUNT      :
   ---------------------------------------------------------------
```

```
WHERE
     EU-RW=1 IN BIT 29 IF EXCLUSIVELY ASSIGNED
          =1 IN BIT 28 IF READ KEY IS NEEDED
          =1 IN BIT 27 IF WRITE KEY IS NEEDED
          =1 IN BIT 26 IF WRITING IS INHIBITED
          =1 IN BIT 25 IF READING IS INHIBITED
     GRAN =0 IF GRANULARITY IS TRACK
          =  NON ZERO IF GRANULARITY IS POSITION
     ASG OPTIONS IN MASTER BIT NOTATION WITH BIT 25 SET FOR A,
     BIT 24 FOR B OPTION, ETC.

THE LAST 3 WORDS FOR TAPE IS:

  -------------------------------------------------------------
7 :  EQUIP  :   R/W   :UNIT CNT :REL F-CYC:   ABS F-CYCLE     :
  -------------------------------------------------------------
8 :                           ASG OPTIONS                     :
  -------------------------------------------------------------
9 :   NOISE CONSTANT             :   MODE SETTINGS            :
  -------------------------------------------------------------


WHERE,
     R/W-IS AS FOR FASTRAND
     UNIT CNT-IS THE NUMBER OF UNITS FOR ASSIGNMENT (1 OR 2)
     NOISE CONSTANT-CHARACTER COUNT FROM ASG STATEMENT
     MODE SETTING-BIT 14=0-ODD PARITY
                        =1-EVEN PARITY
                  BITS 13-12=01 LOW DENSITY
                             10-MEDIUM DENSITY
                             11-HIGH DENSITY
                  BITS 11-10=00-NO TRANSLATION
                            =10 HARDWARE TRANSLATION
                            =01 SOFTWARE TRANSLATION

THE DRUM FORMAT IS:

  -------------------------------------------------------------
7 :  EQUIP  :                    :REL F-CYC:   ABS F-CYCLE    :
  -------------------------------------------------------------
8 :                           ASG OPTIONS                     :
  -------------------------------------------------------------
9 :                   :              AREA LENGTH              :
  -------------------------------------------------------------


THE FORMAT FOR CTS,WTS, AND CTMC IS

  -------------------------------------------------------------
7 :   EQUIP :BITS/CHAR:UNIT TYPE:  LINE SPEED IN BITS/SECOND  :
  -------------------------------------------------------------
8 :                            ASG OPTIONS                    :
  -------------------------------------------------------------
9 :  SPEED  :LINE TYPE:                                       :
  -------------------------------------------------------------

WHERE,
     BITS/CHAR-IS THE NUMBER OF BITS IN 1 CHARACTER
     UNIT TYPE-IS THE UNIT OR GROUP TYPE
     SPEED-IS THE CLASSIFICATION OF THE UNIT'S SPEED

THE FORMAT FOR OTHER TYPES OF EQUIPMENT IS:
   ------------------------------------------------------------
 7 :  EQUIP  :                                                :
   ------------------------------------------------------------
 8 :                          ASG OPTIONS                     :
   ------------------------------------------------------------
 9 :                                                          :
   ------------------------------------------------------------
```

FILE SUPERVISION PROVIDES A SECOND LINKAGE FOR TAPE FILES. THE LINKAGE IS:

```
LA A0,PKT ADDRESS
ER FACIT$
```

IT IS IDENTICAL TO FACIL$ EXCEPT THAT THERE IS A TENTH WORD. THIS WORD HAS THE FOLLOWING FORMAT FOR TAPES. IT IS NOT USED FOR OTHER DEVICES.

```
------------------------------------------------------------
:  SUBSYSTEM NUMBER : UNIT NUM:ALT SUBSYSTEM NUMBER:ALT UNIT:
------------------------------------------------------------
```

WHERE,

SUBSYSTEM NUMBER AND UNIT NUMBER SPECIFY THE FIRST TAPE UNIT.
ALT SUBSYTEM NUMBER AND ALT UNIT SPECIFY THE SECOND TAPE UNIT.

12.3.4. MAGNETIC TAPE LOADING

THE FILE SUPERVISOR CONTROLS ALL TAPE MOUNTING AND TAPE SWAPPING OPERATION. THIS OPERATION IS REFERENCED BY FACILITIES SCHEDULER FOR THE MOUNTING OF THE INITIAL REELS OF A FILE OR BY THE USER ROUTINES TO MOUNT SUBSEQUENT REELS. THE TAPE SWAPPING ROUTINE, WHICH INITIATES THE ACTION DESCRIBED IN THIS SECTION IS DOCUMENTED IN THE EXECUTIVE REQUEST FUNCTIONS CHAPTER. THE ER REFERENCES TO SWAPPING ARE TSWAP$ AND TINTL$.

THE FILE SUPERVISOR COMMUNICATES WITH THE OPERATOR BY TWO MESSAGE FORMATS DEPENDING UPON WHETHER OR NOT AN ANSWER IS NECESSARY. THE MESSAGE FORMATS ARE:

```
MOUNT REEL NO. CC/UU FILENAME -REEL INDEX RUNID
LOAD REEL NO. CC/UU FILENAME -REEL INDEX RUNID
```

REEL INDEX IS THE RELATIVE REEL NUMBER AND IS SUPPLIED TO IDENTIFY THE PROPER REEL OF A SCRATCH FILE USED FOR A SECOND PASS AFTER A TINTL$ REFERENCE. THIS FIELD IS DISPLAYED ONLY IF THE WORD 'BLANK' APPEARS IN THE REEL NUMBER FIELD.

THE MOUNT MESSAGE IS TO BE ANSWERED BY THE OPERATOR BY A KEY-IN OF THE REEL NUMBER WHICH IS ENTERED INTO THE DIRECTORY ITEM OF THE FILE. THE REEL NUMBER IS LIMITED TO SIX OR LESS CHARACTERS AS DEFINED IN THE ASG STATEMENT FORMAT. THE MOUNT MESSAGE IS TYPED IF:

1. THE FILE IS TO BE CATALOGUED (ASG STATEMENT OPTIONS) AND NO REEL NUMBERS ARE FURNISHED ON THE ASG STATEMENT.
2. THE FILE IS TO BE CATALOGUED AND A SET OF REEL NUMBERS ARE ON THE ASG STATEMENT BUT A SWAP IS REQUESTED AND THE SET OF REEL NUMBERS GIVEN HAS ALREADY BEEN REQUESTED.
3. THE FILE IS ALREADY CATALOGUED AND NO REEL NUMBERS ARE SPECIFIED ON THE ASG STATEMENT AND THE SET OF NUMBERS CATALOGUED HAS ALREADY BEEN REQUESTED OR REEL NUMBERS ARE SPECIFIED BUT HAVE ALL ALREADY BEEN REQUESTED.

THE LOAD MESSAGE IS USED WHENEVER NO RESPONSE IS REQUIRED BY THE EXECUTIVE. THIS APPLIES FOR:

1. CATALOGUED FILES WHERE EITHER NO REEL NUMBERS ARE SPECIFIED, IN WHICH CASE THE NUMBERS ARE TAKEN FROM THE

DIRECTORY IN THE ORDER OF CREATION, OR REEL NUMBERS ARE SPECIFIED ON THE ASG STATEMENT IN WHICH CASE THE ORDER IS THAT SHOWN ON THE STATEMENT.

2. THE FILE IS TO BE CATALOGUED AND REEL NUMBERS ARE SPECIFIED.
3. THE FILE IS NOT TO BE CATALOGUED REGARDLESS OF WHETHER OR NOT REEL NUMBERS ARE SUPPLIED.

IF NO REEL NUMBERS ARE SUPPLIED FOR THE SCRATCH FILE, THE WORD 'BLANK' IS ENTERED INTO THE LOAD MESSAGE.

THE EXECUTIVE ASSUMES ALL UNUSED TAPE UNITS ARE REWOUND WITH INTERLOCK UNTIL SUCH TIME AS THE CORRECT TAPE REEL IS MOUNTED AND HENCE DOES NOT WAIT FOR OPERATOR INTERVENTION AFTER DISPLAYING A MOUNT OR LOAD MESSAGE. THE EXECUTIVE WILL ALLOW AN INTERLOCK TO PREVAIL ON THE FIRST REFERENCE AFTER ISSUING THE MESSAGE FOR A PERIOD OF TWO MINUTES AT WHICH TIME THE OPERATOR IS AGAIN NOTIFIED TO LOAD THE UNIT BY THE MESSAGE:

```
SERVICE  CC/UU
```

THIS CYCLE IS REPEATED UNTIL EITHER THE INTERLOCK IS REMOVED OR AN UNSOLICITED MESSAGE AFFECTS THE STATUS OF THE UNIT.

THE RESPONSE TO THE MOUNT MESSAGE CAN BE DELAYED UNTIL THE ASSIGNMENT OF THE UNIT CHANGES (FREE OR TERMINATION OF THE RUN) OR A SWAP REQUEST AFFECTS THE UNIT AT WHICH TIME THE OPERATOR IS NOTIFIED THAT A RESPPONSE IS REQUIRED .

```
MOUNT MESSAGE OUTSTANDING CC/UU
```

ON THE CONSOLE.

IF CATALOGUING IS CONDITIONAL (C OPTION) AND THE RUN IS ABORTED, THE MESSAGE

```
'FILE NAME' NOT CAT'D RELEASE REELS:
R1, R2, R3,...,RN
```

IS DISPLAYED ON THE CONSOLE TO INFORM THE OPERATOR THAT THE REELS REQUESTED TO BE MOUNTED ARE TO BE CONSIDERED BLANK REELS.

IN ORDER TO PROVIDE AN EFFICIENT MEANS OF REOPENING TWO REEL FILES CONTAINED ON TWO UNITS, THE TAPE SWAP FUNCTION DELAYS REWINDING OF UNITS AS LONG AS POSSIBLE ACCORDING TO THE FOLLOWING ALGORITHM:

1. A LOAD MESSAGE FOR THE ALTERNATE UNIT IS PUT OUT AT RUN OPENNING TIME. A MOUNT MESSAGE IS PUT OUT IF AND ONLY IF THE FILE IS TO BE CATALOGUED AND NO READ NUMBERS ARE GIVEN IN THE @ASG STATEMENT.

2. THE FIRST AND SUBSEQUENT SWAPPING FUNCTIONS INTERLOCK THE UNIT BEING SWAPPED FROM, DETERMINES WHETHER THE NEXT MESSAGE FOR THE FILE IS A LOAD OR MOUNT MESSAGE, AND ISSUES THE PROPER MESSAGE.

THE SET OF REELS FOR A TAPE FILE CAN BE REFERENCED FOR A SECOND OR SUBSEQUENT PASS ON THE SET BY REINITIALIZING THE TAPE SWAP ROUTINE BY THE TINTL$ EXECUTIVE RETURN. THE CALLING SEQUENCE IS:

```
L    A0,PKT ADDR
ER   TINTL$
```

WHERE PKT ADDR LOCATES A TWO WORD PACKET IN THE FORMAT:

```
      35                                                          00
     ----------------------------------------------------------------
  01 :                     INTERNAL FILE NAME                        :
     ----------------------------------------------------------------
  02 :                     INTERNAL FILE NAME (CONT.)                :
     ----------------------------------------------------------------
```

THE TINTL$ REFERENCE RESETS THE INDEX TO THE SET OF REELS BACK TO THE FIRST REEL ACCORDING TO THE FOLLOWING CONDITIONS:

1. ONE UNIT AND CURRENT INDEX IS ONE. REWIND WITHOUT INTERLOCK IF EXTENDED. IF REWOUND WITH INTERLOCK, ISSUE THE LOAD MESSAGE REQUESTING THE FIRST REEL OF THE SET.

2. ONE UNIT AND CURRENT REEL INDEX IS NOT ONE. REWIND WITH INTERLOCK IF EXTENDED AND ISSUE THE LOAD MESSAGE FOR THE FIRST REEL OF THE SET.

3. TWO UNITS AND CURRENT INDEX IS 1. REWIND WITHOUT INTERLOCK IF EXTENDED. IF REWOUND WITH INTERLOCK ISSUE LOAD INSTRUCTIONS FOR UNIT 1. A MOUNT OR LOAD MESSAGE IS ALREADY OUTSTANDING FOR UNIT 2.

4. TWO UNITS AND CURRENT INDEX IS 2. SWITCH UNITS, REWIND UNIT 2 WITHOUT INTERLOCK IF EXTENDED. IF UNIT 2 IS REWOUND WITH INTERLOCK, ISSUE LOADING INSTRUCTIONS. REWIND UNIT 1 WITHOUT INTERLOCK IF EXTENDED. IF UNIT 1 IS REWOUND WITH INTERLOCK ISSUE LOADING INSTRUCTIONS.

5. TWO UNITS AND CURRENT INDEX IS 3. HANDLE UNIT 2 AS IN 4 ABOVE. REWIND UNIT 1 WITH INTERLOCK UNLESS ALREADY INTERLOCKED AND ISSUE LOADING COMMAND WHETHER EXTENDED OR NOT.

6. TWO UNITS AND CURRENT INDEX IS 4 OR MORE. SWITCH UNITS IF THE INDEX IS EVEN. REWIND BOTH UNITS WITH INTERLOCK UNLESS ALREADY INTERLOCKED AND ISSUE LOADING INSTRUCTIONS FOR BOTH UNITS.

12.3.5. FILE SECURITY

WITHIN THE MASTER DIRECTORY THE EXECUTIVE MAINTAINS TWO KEYS WHICH MUST BE SPECIFIED ON THE ASG STATEMENT TO GAIN ASSIGNMENT OF A CATALOGUED FILE. THE KEY, IF ONE EXISTS, IS INITIALLY OBTAINED FROM THE ASG STATEMENT WHICH CAUSED THE FILE TO BE CATALOGUED.

THE MASTER DIRECTORY CONTAINS A COUNT OF THE NUMBER OF TIMES A FILE IS ASSIGNED AND THE TIME OF LAST ASSIGNMENT. THESE FIELDS, ALONG WITH THE SYSTEM LOG, ARE AVAILABLE TO THE USER TO MONITOR FILE USAGE AND DETECT ANY ENCROACHMENT OF INDIVIDUAL PRIVACY.

12.3.6. BUFFER SETUP

A BUFFER POOL IS A PORTION OF THE USERS CORE STORAGE AREA WHICH HAS BEEN MADE AVAILABLE FOR USE AS AN INPUT/OUTPUT AREA FOR ONE OR MORE FILES. A POOL CAN CONTAIN ANY DESIRED NUMBER OF BUFFERS.

THE BUFFER POOL IS CONTROLLED BY A TWO WORD PACKET OF THE FOLLOWING FORMAT:

```
      35                             17                           00
     ----------------------------------------------------------------
  01 : BUFFER SIZE                  :ADDRESS OF CURRENT FIRST BUFF:
     ----------------------------------------------------------------
  02 : NOT USED                     :LENGTH OF CORE AREA          :
     ----------------------------------------------------------------
```

WORD 1 H1-SIZE OF EACH BUFFER IN THE POOL. THIS SIZE INCLUDES THE AREA TO BE USED FOR DATA PLUS TWO INITIAL CONTROL WORDS IN EACH BUFFER.
H2-LOCATION OF THE CURRENT FIRST BUFFER IN THE POOL. AS A BUFFER IS REMOVED FROM THE AVAILABLE POOL THIS HALF WORD IS UPDATED TO THE NEXT BUFFER IN THE CHAIN.

WORD 2 H1-NOT USED
H2-TOTAL LENGTH OF THE CORE AREA INITIALLY DIVIDED INTO BUFFERS. AFTER THE BPOOL$ REFERENCE IS MADE WORD 2 IS NO LONGER REFERENCED.

THE SYSTEM FURNISHES TWO SUBROUTINES TO CONTROL THE BUFFER POOL. THE SUBROUTINE TO SET UP THE POOL IS ENTERED THROUGH THE LINKAGE:

```
LA A0, PKT ADDR
ER BPOOL$
```

THE A0 REGISTER IS LOADED WITH THE ADDRESS OF THE PACKET OUTLINED ABOVE. THIS LINKAGE WILL DIVIDE THE TOTAL AREA AND SET UP EACH BUFFER AS FOLLOWS:

```
 35                             17                            00
----------------------------------------------------------------
:AVAILABLE FOR STATUS ETC.    :ADDR OF NEXT BUFFER IN POOL    :
----------------------------------------------------------------
:LENGTH OF DATA AREA OF BUFFER: ADDRESS OF DATA AREA          :
----------------------------------------------------------------
:                                                              :
:                                                              :
:                     DATA AREA OF BUFFER                      :
:                                                              :
:                                                              :
----------------------------------------------------------------
```

WORD 1 H1-NOT USED, AVAILABLE FOR STATUS ETC.
H2-ADDRESS OF THE START OF THE NEXT BUFFER IN THE POOL OR ZERO FOR THE LAST BUFFER IN THE POOL.

WORD 2 AN I/O ACCESS WORD WITH THE BUFFER SIZE MINUS TWO FOR THE INITIAL CONTROL WORDS IN H1 AND THE ADDRESS OF WORD 3 IN H2.

TO ADD A SUBSEQUENT CORE AREA TO A PREVIOUSLY ESTABLISHED BUFFER POOL THE FOLLOWING SEQUENCE IS PROVIDED:

```
LA A0, PKT ADDR
ER BJOIN$
```

THE TWO-WORD PACKET ADDRESSED BY A0 CONTAINS THE ADDRESS OF THE INITIAL CONTROL WORDS (PACKET USED FOR BPOOL$) IN THE FIRST WORD AND THE LENGTH AND ADDRESS OF THE SUBSEQUENT AREA IN H1 AND H2 RESPECTIVELY OF THE SECOND WORD AS FOLLOWS:

```
    35                             17                            00
   ----------------------------------------------------------------
01 :  NOT USED                    :ADDR OF INIT CONTROL PACKET    :
   ----------------------------------------------------------------
02 :LENGTH OF ADDITIONAL AREA     :ADDRESS OF ADDITIONAL AREA     :
   ----------------------------------------------------------------
```

TO REMOVE A BUFFER FROM THE AVAILABLE POOL, THE NORMAL PROCEDURE IS TO TAKE THE BUFFER ADDRESSED BY H2 OF THE FIRST CONTROL WORD AND REPLACE THIS HALF WORD WITH H2 OF THE FIRST WORD OF THE BUFFER REMOVED FROM THE POOL. THIS PROCEDURE MUST BE PRECEDED BY A TEST FOR ZERO ON H2 OF THE FIRST CONTROL WORD WHICH WOULD INDICATE THAT THE POOL WAS EXHAUSTED. A BUFFER IS RETURNED TO THE AVAILABLE POOL BY STORING H2 OF

THE FIRST CONTROL WORD INTO H2 OF THE FIRST WORD OF THE BUFFER BEING RETURNED AND STORING THE ADDRESS OF THE BUFFER IN H2 OF THE FIRST CONTROL WORD. THIS ADDS THE BUFFER TO THE BEGINNING OF THE CHAIN. IF MORE THAN ONE ACTIVITY IS UTILIZING A SINGLE POOL A TIMING PROBLEM MAY ARISE UNLESS INTERRUPTS ARE DISABLED WHILE THE WORDS ARE UPDATED. IF THE BLOCK BUFFERING PACKAGE IS USED, THE USER NEED NOT BE CONCERNED WITH THE BUFFER CONTROL SCHEME.

THE ORIGINAL BUFFER POOL CAN ALSO BE CREATED BY A SYSTEM PROCEDURE WHICH CAN NEGATE THE NEED FOR BPOOL$ IF THE AREA CAN BE ALLOTTED AT ASSEMBLY TIME. THE PROCEDURE GENERATES A SINGLE INITIAL CONTROL WORD AT THE LINE OF THE PROC FOLLOWED IMMEDIATELY BY THE POOL. THE PROCEDURE IS:

```
   B$GPUL   NUMBER OF BUFFERS, BUFFER SIZE
```

12.3.7. COMMUNICATIONS BUFFER POOLS

A BUFFER POOL FOR USE WITH THE COMMUNICATIONS HANDLER MAY BE ESTABLISHED IN ANY PORTION OF THE USERS CORE STORAGE AREA WHICH THE USER MAY ELECT TO SET ASIDE AS AN INPUT/OUTPUT AREA. THE SYSTEM FURNISHES THREE SUBROUTINES TO CONTROL THE COMMUNICATIONS BUFFER POOL. ONE SUBROUTINE IS USED TO ESTABLISH THE POOL, ANOTHER SUBROUTINE IS AVAILABLE FOR THE USER TO REMOVE BUFFERS FROM THE POOL, AND THE OTHER SUBROUTINE PERMITS THE USER TO RETURN BUFFERS TO THE POOL. ALL PERTINENT LINKAGE FOR THE COMMUNICATIONS BUFFER POOL IS MAINTAINED BY THE SYSTEM PROVIDED SUBROUTINES EXTERNAL TO THE USER'S AREA SO THAT THE LINKAGE IS READILY AVAILABLE FOR USE BY THE INTERRUPT PROCESSING ROUTINES. THE AMOUNT OF PARAMETER INFORMATION AND ITS STORAGE IN THE USER'S BUFFER AREA IS KEPT AT A MINIMUM WHILE STILL MAKING AVAILABLE TO THE USER ALL RELEVANT INFORMATION FOR BUFFER PROCESSING. THE FOLLOWING DESCRIBES THE POOL MODE FORMAT IN THE USER'S AREA FOR BOTH INPUT AND OUTPUT BUFFERS.

```
       35                    23        17                         00
    ---------------------------------------------------------------
 01 : CHARACTER COUNT     : STATUS : ADDR OF NEXT BUFFER IN POOL  :
    ---------------------------------------------------------------
 02 :                                                             :
    :                                                             :
    :                    DATA AREA OF BUFFER                      :
    :                                                             :
    :                                                             :
    ---------------------------------------------------------------
```

WORD 1 T1-THE CHARACTER COUNT FOR THIS BUFFER. FOR INPUT BUFFERS TO BE PROCESSED BY THE INPUT COMPLETION ROUTINE, THE CHARACTER COUNT FOR COMPLETELY FILLED BUFFERS WILL BE THE VALUE WHICH IS SPECIFIED BY THE POOL START INFORMATION IN H2 OF WORD 8 OF THE LT TABLE. FOR PARTIALLY FILLED INPUT BUFFERS WHICH MAY RESULT DUE TO EITHER A TIME-OUT OR AN EXTERNAL INTERRUPT, THE CHARACTER COUNT WILL BE THE NUMBER OF CHARACTERS WHICH WERE IN THE BUFFER AT THE TIME OF OCCURRENCE OF THE EXTERNAL INTERRUPT OR TIME-OUT. FOR OUTPUT THE CHARACTER COUNT REPRESENTS THE NUMBER OF CHARACTERS TO BE TRANSMITTED FROM THIS BUFFER. THE OUTPUT CHARACTER COUNT CAN BE DYNAMICALLY SUPPLIED BY THE USER SO THAT THE USER MAY SPECIFY EITHER PARTIALLY OR COMPLETELY FILLED OUTPUT BUFFERS IN ANY ORDER. THE MAXIMUM VALUE FOR ANY OUTPUT CHARACTER COUNT IS LIMITED BY THE MAXIMUM VALUE ESTABLISHED AT SYSTEM GENERATION TIME. IF A MAXIMUM VALUE IS NOT SPECIFIED AT SYSTEM GENERATION TIME, IT WILL BE ASSUMED THAT NO MAXIMUM OUTPUT CHARACTER COUNT EXISTS.

S3-THE STATUS CODE FOR THIS BUFFER. THE CODE FOUND HERE FOR INPUT WILL BE IDENTICAL TO THOSE DESCRIBED FOR S1 OF WORD 7 FOR THE LT TABLE. THE CODE FOUND HERE FOR OUTPUT WILL BE IDENTICAL TO THOSE DESCRIBED FOR S1 OF WORD 3 FOR THE LT TABLE.

H2-THIS VALUE POINTS TO THE NEXT BUFFER IN THE CHAIN OF THOSE LINKED TOGETHER. A VALUE OF ZERO IN THIS FIELD WILL BE INTERPRETED TO MEAN THE END OF THE CHAIN.

WORD 2-THE DATA AREA FOR THE INPUT/OUTPUT BUFFER AREA STARTS AT THIS WORD.

THE APPROPRIATE SYSTEM LINKAGE TO ESTABLISH A POOL OF INPUT/OUTPUT BUFFERS FOR COMMUNICATIONS USAGE MAY BE ESTABLISHED BY THE FOLLOWING:

```
LA A0, PKT ADDR
ER CPOOL$
```

THIS LINKAGE WILL DIVIDE THE TOTAL AREA AND SET UP EACH INDIVIDUAL BUFFER IN THE FORMAT DESCRIBED ABOVE. THE A0 REGISTER IS LOADED WITH THE ADDRESS OF A PACKET CONTAINING THE FOLLOWING INFORMATION:

```
       35                    23        17                         00
    ---------------------------------------------------------------
 01 : BUFFER SIZE IN CHARACTERS   : ADDRESS OF FIRST BUFFER       :
    ---------------------------------------------------------------
 02 : NOT USED            : METHOD : LENGTH OF AREA TO BE USED    :
    ---------------------------------------------------------------
```

WORD 1 H1-THIS IS THE NUMBER OF CHARACTERS TO BE USED FOR EACH BUFFER AREA TO BE ESTABLISHED. THE VALUE SPECIFIED IN

THIS FIELD SHOULD BE THE OPTIMUM VALUE FOR THE SYSTEMS APPLICATION. THE VALUE SPECIFIED MAY BE EITHER AN ODD OR AN EVEN NUMBER, BUT IT MUST NOT EXCEED THE MAXIMUM COMMUNICATIONS BUFFER LENGTH DEFINED AT SYSTEMS GENERATION TIME. ONE CORE LOCATION WILL BE ASSIGNED TO THE INDIVIDUAL BUFFER AREA FOR EACH GROUP OF TWO CHARACTERS.

H2-THIS FIELD DESIGNATES THE STARTING CORE ADDRESS OF THE AREA TO BE SETUP AS AN INPUT/OUTPUT BUFFER POOL.

WORD 2 T1-THIS PORTION IS NOT USED BY ANY OF THE BUFFER POOL CONTROL ROUTINES AND IS AVAILABLE TO THE USER FOR ANY PURPOSE.

S3-THIS DEFINES THE METHOD OF POOL BUFFERING FOR WHICH THE ESTABLISHED BUFFERS ARE TO BE USED. TWO METHODS OF POOL BUFFERING ARE PERMITTED BY THE COMMUNICATIONS HANDLER. THE FIRST METHOD IS SIMILAR TO THAT EMPLOYED BY THE BLOCK BUFFERING PACKAGE AND WILL BE REFERRED TO AS THE OPEN CHAIN METHOD. EACH INDIVIDUAL BUFFER IS LINKED TO THE NEXT BUFFER IN THE POOL BY THE VALUE IN H2 OF WORD 1 OF EACH BUFFER EXCEPT FOR THE LAST BUFFER IN THE CHAIN WHICH HAS A VALUE OF ZERO IN ITS LINK FIELD. THE ZERO IN THE LINK FIELD CAUSES THE POOL TO HAVE AN OPEN END, HENCE THE NAME OPEN CHAIN METHOD. THE SECOND METHOD OF POOL BUFFERING IS REFERRED TO AS THE CONTINUOUS CHAIN METHOD. EACH INDIVIDUAL BUFFER IS LINKED TO THE NEXT BUFFER IN THE POOL IN THE SAME MANNER AS EMPLOYED BY THE OPEN CHAIN METHOD EXCEPT THE LAST BUFFER IN THE SEQUENCE IS LINKED BACK TO THE VERY BEGINNING OF THE POOL THUS FORMING A CONTINUOUS CHAIN. THE OPEN CHAIN METHOD IS THE PREFERRED METHOD BECAUSE EACH BUFFER IS REMOVED FROM THE POOL BY THE COMMUNICATIONS HANDLER AS INPUT DATA IS RECEIVED, AND SUCH A BUFFER WILL NOT BE REUSED AGAIN UNTIL IT HAS BEEN RETURNED TO THE POOL BY THE USER. IN THE USE OF THE CONTINUOUS CHAIN METHOD, AN INDIVIDUAL BUFFER IS NEVER REALLY REMOVED FROM THE POOL BUT RATHER THE BUFFERS IN THE POOL ARE USED IN A SEQUENTIALLY CYCLIC MANNER. THIS CAN RESULT IN THE REUSE OF A BUFFER BEFORE ALL OF ITS PREVIOUS CONTENTS HAD BEEN PROCESSED BECAUSE EITHER THE BUFFERS IS OF AN INSUFFICIENT SIZE FOR THE APPLICATION OR THE REAL-TIME PROGRAM MAY BE SPENDING EXCESSIVE TIME IN ITS BUFFER PROCESSING. PERHAPS THE MOST FREQUENT USE OF THE CONTINUOUS CHAIN METHOD IS TO EMPLOY TWO INDIVIDUAL BUFFERS CHAINED TO EACH OTHER WHEREBY THEY OPERATE IN AN ALTERNATING, TOGGLING MANNER. WHEN THE CONTINUOUS CHAIN IS EMPLOYED, THE USER ASSUMES ALL RESPONSIBILITY FOR PROCESSING INDIVIDUAL BUFFER CONTENTS IN THE REQUIRED TIME INTERVAL. FOR THE PREFERRED OPEN CHAIN METHOD, THE COMMUNICATIONS HANDLER ALWAYS INSURES THAT NO BUFFER IS REUSED UNTIL SO DIRECTED BY THE REAL-TIME PROGRAM. FOR EITHER METHOD THE USER STILL MUST EMPLOY AN OPTIMUM SIZED BUFFER FOR THE APPLICATION. THE CODE IN S3 MEANS THE OPEN CHAIN METHOD IF IT IS ZERO. ANY NON-ZERO VALUE WILL BE USED AS THE NUMBER PLUS ONE OF INDIVIDUAL BUFFERS TO BE LINKED IN A CONTINUOUS MANNER. THUS A VALUE OF ONE SPECIFIES TWO INDIVIDUAL BUFFERS LINKED TO EACH OTHER TO OPERATE IN THE ALTERNATING MANNER DESCRIBED PREVIOUSLY, A VALUE OF TWO WOULD SPECIFY THREE INDIVIDUAL BUFFERS

LINKED TOGETHER, ETC. THE REMAINING BIT IS NOT USED.

H2-THIS FIELD SPECIFIES THE LENGTH OF THE CORE AREA TO BE USED FOR THE POOL. THE SETUP ROUTINE CONTINUES TO ESTABLISH INDIVIDUAL BUFFERS OF THE SPECIFIED SIZE IN THE DESIRED METHOD UNTIL THIS LENGTH VALUE IS EXHAUSTED.

THE RETURN FROM ER CPOOL$ IS WITH A VALUE IN H2 OF A0 WHICH THE USER IS EXPECTED TO PLACE IN EVERY LT TABLE WHICH SHARES THE POOL JUST ESTABLISHED. IF THE POOL IS USED FOR OUTPUT, THAT VALUE IS PLACED IN H2 OF WORD 4 OF THE LT TABLE. IF THE POOL IS USED FOR INPUT, THAT VALUE IS PLACED IN H2 OF WORD 8 OF THE LT TABLE. A POOL MAY BE USED FOR BOTH INPUT AND OUTPUT BY PLACING THE VALUE IN H2 OF A0 AT BOTH WORD 4 AND WORD 8 OF LT TABLES. IT MUST BE UNDERSTOOD THAT THE VALUE RETURNED IN H2 OF A0 IS AN EXECUTIVE LINKING VALUE AND, AS SUCH, HAS NO USER RELATIONSHIP TO ANY OF THE BUFFERS IN THE POOL JUST ESTABLISHED AND CONSEQUENTLY, IT IS MEANINGLESS TO THE REAL-TIME PROGRAM.

WITH THE CLOSED-CHAIN METHOD NO BUFFER IS EVER REMOVED FROM THE POOL SO ALL OF THE FOLLOWING DISCUSSION ON COMMUNICATIONS BUFFER POOL CONTROL IS IRRELEVANT BECAUSE IT IS ONLY USED WHEN THE OPEN CHAIN METHOD IS EMPLOYED. IF ANY NUMBER OF BUFFERS IS REMOVED FROM THE POOL FOR FUTURE USE, THE REMOVAL MAY BE ACCOMPLISHED BY THE FOLLOWING:

```
ER CGET$
```

THE A0 REGISTER IS TO CONTAIN THE FOLLOWING INFORMATION:

```
 35                             17                             00
-----------------------------------------------------------------
: NUMBER TO BE REMOVED         : EXEC LINKING VALUE FOR POOL   :
-----------------------------------------------------------------
```

H1 OF A0 DEFINES THE NUMBER OF INDIVIDUAL BUFFERS WHICH ARE TO BE REMOVED FROM THE POOL. H2 OF A0 MUST CONTAIN THE EXECUTIVE LINKING VALUE FOR THE POOL FROM WHICH BUFFERS ARE TO BE REMOVED. THE RETURN OF CONTROL FROM ER CGET$ WILL BE WITH INFORMATION IN A0. H1 OF A0 WILL BE THE ACTUAL NUMBER OF BUFFERS WHICH WERE ACTUALLY REMOVED FROM THE SPECIFIED POOL. THIS VALUE WILL NORMALLY BE THE NUMBER OF BUFFERS REQUESTED UNLESS LESS THAN THAT SPECIFIED NUMBER EXISTED IN THE POOL AT THE TIME OF THE ER CGET$. H2 OF A0 WILL BE THE STARTING ADDRESS OF THE BUFFERS REMOVED. EACH BUFFER REMOVED WILL BE LINKED TO THE OTHERS IN THE OPEN CHAIN MANNER.

WHEN IT IS DESIRED TO RETURN ANY NUMBER OF BUFFERS TO A POOL, SUCH MAY BE ACCOMPLISHED BY::

```
LA A0, PKT ADDR
ER CADD$
```

THE A0 REGISTER IS LOADED WITH THE ADDRESS OF A PACKET CONTAINING THE FOLLOWING INFORMATION:

```
    35                             17                             00
   -----------------------------------------------------------------
01 : NUMBER TO BE ADDED           : EXEC LINKING VALUE FOR POOL   :
   -----------------------------------------------------------------
02 : NOT USED                     : ADDRESS OF FIRST BUFFER       :
   -----------------------------------------------------------------
```

WORD 1 H1-THIS FIELD SPECIFIES THE NUMBER OF INDIVIDUAL BUFFERS WHICH ARE TO BE RETURNED TO THE POOL.

H2-THIS FIELD MUST CONTAIN THE EXECUTIVE LINKING VALUE FOR THE PARENT POOL. A BUFFER CAN BE RETURNED TO ONLY THE

SAME POOL FROM WHICH IT HAD BEEN PREVIOUSLY REMOVED SO THAT BUFFER SIZE CONSISTENCY CAN BE MAINTAINED WITHIN A POOL.

WORD 2 H1-THIS PORTION IS NOT USED BY THE BUFFER POOL CONTROL ROUTINE AND IS AVAILABLE TO THE USER FOR ANY PURPOSE.

H2-THIS FIELD IS TO CONTAIN THE ADDRESS OF THE FIRST BUFFER OF THE GROUP TO BE RETURNED. EACH BUFFER TO BE ADDED TO THE POOL IS EXPECTED TO BE LINKED TO THE OTHERS IN THE OPEN CHAIN MANNER.

12.3.8. EXCLUSIVE USE OF FILES

THE FILE SUPERVISOR ROUTINES ALLOW ASSIGNMENT OF MASS STORAGE FILES TO ANY NUMBER OF RUNS AT ONE TIME PROVIDING THE EXCLUSIVE USE OPTION IS NOT EXERCISED ON THE ASG STATEMENT. THIS OPTION CAUSES A DELAY IN ASSIGNMENT OF A FILE UNTIL NO OTHER RUN HAS THE FILE ASSIGNED AND INSURES THAT OTHER RUNS ARE DELAYED UNTIL A RUN RELEASES ANY NEEDED EXCLUSIVELY ASSIGNED FILES.

ALL MAGNETIC TAPE FILES ARE EXCLUSIVELY ASSIGNED REGARDLESS OF THE PRESENCE OR ABSENCE OF THE OPTION. IF A TAPE FILE IS TO BE SHARED, IT MUST BE CATALOGUED UNDER A SECOND NAME WITH THE SAME REEL NUMBERS.

THE READ-AND-LOCK AND UNLOCK FUNCTIONS ARE AVAILABLE AT THE HANDLER LEVEL WHEREBY LOGICALLY-CONTIGUOUS AREAS (SUCCESSIVE RELATIVE ADDRESSES) CAN BE EXCLUSIVELY ASSIGNED TO ALLOW OTHER RUNS SIMULTANEOUS ACCESS OF ALL THE UNLOCKED PORTION OF THE FILE. WRITING INTO A LOCKED AREA WILL RELEASE THE EXCLUSIVE USE AS WELL AS THE UNLOCK WITHOUT WRITING METHOD. THE COMPLETE DEFINITION OF THE VARIOUS FUNCTIONS INVOLVED AND THE TIMING LIMITS TO BE CONSIDERED IS GIVEN IN THE INPUT-OUTPUT DEVICE HANDLERS CHAPTER UNDER THE FASTRAND HANDLER SECTION.

13. DATA HANDLING

13.1. GENERAL

THE DATA HANDLING ROUTINES ARE DESIGNED TO HANDLE A VARIED NUMBER OF FILE FORMATS BY USING A GENERAL TECHNIQUE WHICH PLACES A FEW BASIC RESTRICTIONS ON THE CLASSES ACCEPTABLE TO THE SYSTEM. THIS TECHNIQUE REQUIRES THE DATA HANDLING ROUTINES TO BE DIVIDED INTO THREE BASIC AREAS CALLED, (1) FORMAT DEFINITION, (2) ITEM CONTROL, AND (3) BLOCK BUFFERING CONTROL. EACH AREA IS SUBDIVIDED INTO FUNCTIONAL SUBROUTINES THAT MAY OR MAY NOT BE CALLED TO PERFORM THEIR FUNCTION.

THE FORMAT DEFINITION AREA IS THAT PORTION OF THE SYSTEM WHEREBY THE PHYSICAL ORGANIZATION OF A FILE IS DESCRIBED TO THE SYSTEM. FORMATS MAY BE DEFINED AND PROCESSED WITHIN PROGRAMS OR INDEPENDENT OF PROGRAMS. INDEPENDENTLY PROCESSED FORMAT CAN BE STORED AS AN ELEMENT ON THE MASS STORAGE DEVICE SUBJECT TO RECALL BY ANY OBJECT PROGRAM. DEPENDING UPON THE FUNCTION, A PROGRAM IS TO PERFORM, A USER MAY CALL ONE OR MORE FORMATS. THERE COULD BE ONE FORMAT BEING USED BY SEVERAL FILES OR MANY FORMATS BEING USED BY MANY FILES. IDEALLY AN INSTALLATION SHOULD DEFINE AND STORE INDEPENDENTLY ALL THE FORMATS USED AT THE SITE. INDIVIDUAL USERS THEN, WOULD NEED ONLY TO CALL THE FORMAT DESIRED. THIS WOULD NOT PRECLUDE A PROGRAMMER FROM THE CAPABILITY OF DEFINING HIS OWN PECULIAR FORMAT WHEN NECESSARY.

FORMAT DEFINITIONS ARE PROGRAM INDEPENDENT, BUT SINCE ITEM CONTROL MUST INTERACT WITH THEM OFTEN, THEY WILL OCCUPY A PORTION OF THE OBJECT PROGRAM'S AREA.

THE ITEM CONTROL AREA IS THE HIGHEST LEVEL OF OBJECT PROGRAM INTERFACE. IT INTERACTS WITH THE FORMAT DEFINITIONS WHEN NECESSARY, FOR ORGANIZATIONAL INFORMATION ABOUT THE FILE, AND BLOCK BUFFERING CONTROL FOR FUNCTIONS SUCH AS BLOCK READ AND WRITE, BUFFER ACQUISITION AND RELEASE, ETC. THE ITEM HANDLER MAINTAINS COMPLETE CONTROL OF A FILE WHILE IT PRESENTS OR RECEIVES ITEMS ON REQUEST BY THE USER. THE OBJECT PROGRAM THEN, IS CONCERNED ONLY WITH THE PROCESSING OF ITEMS; ITS WORK IS COMPLETE WHEN IT MAKES A REQUEST TO THE SYSTEM.

ITEM CONTROL IS OBJECT PROGRAM DEPENDENT AND, AS SUCH, WILL OCCUPY A PORTION OF THE OBJECT PROGRAM'S AREA.

THE BLOCK BUFFERING CONTROL AREA IS THE LOWEST LEVEL OF OBJECT PROGRAM INTERFACE AND PROVIDES A GENERAL MEANS OF FILE COMMUNICATION FOR THE USER. TO PROVIDE COMPLETE FLEXIBILITY IN FILE ORGANIZATION, FORMAT DEFINITIONS ARE NOT PROVIDED FOR BLOCK BUFFERING CONTROL. THE SYSTEM WILL MAINTAIN ONLY MINIMUM CONTROL OF THE FILE WHILE PRESENTING OR RECEIVING BLOCKS OF DATA ON REQUEST BY THE USER.

BLOCK BUFFERING CONTROL IS A RESIDENT PORTION OF THE EXECUTIVE SYSTEM. ITS SERVICES ARE DISPENSED EQUALLY TO ALL PROGRAMS OPERATING IN THE MULTI-PROGRAMMING ENVIRONMENT.

IN GENERAL, FILES HANDLED BY THE SYSTEM MAY BE CONSIDERED AS EITHER SEQUENTIAL OR RANDOM ACCORDING TO THE WAY THEY ARE REFERENCED. FURTHER, THE ITEMS OR BLOCKS IN A FILE MAY BE IDENTIFIED AS BEING EITHER FIXED OR VARIABLE IN LENGTH. FILES REFERENCED RANDOMLY MUST RESIDE ON MASS STORAGE AND MUST CONTAIN FIXED LENGTH ITEMS AND BLOCKS. FILES REFERENCED SEQUENTIALLY MAY RESIDE ON EITHER TAPE OR MASS STORAGE AND MAY CONTAIN FIXED OR VARIABLE ITEMS AND BLOCKS.

THE SYSTEM WILL PROVIDE ACCESS TO RANDOM FILES BY ITEM OR BLOCK NUMBERS RELATIVE TO THE BEGINNING OF THE FILE.

BOTH SEQUENTIAL AND RANDOM REQUESTS MAY BE USED INTERCHANGEABLY ON THE SAME MASS STORAGE FILE IF THE FILE IS ORGANIZED WITH FIXED LENGTH ITEMS AND BLOCKS. THUS IT IS POSSIBLE TO PROCESS DATA SEQUENTIALLY UP TO A CERTAIN POINT AND THEN PROCESS IT

RANDOMLY. CONVERSELY, DATA MAY BE PROCESSED AT RANDOM UNTIL SOME SPECIFIC DATA OCCURS, AND THEN PROCESSED SEQUENTIALLY.

WHENEVER ACCESS TO A SPECIFIC FILE IS REQUESTED, THE REQUEST IS SUBJECT TO BE INTERLOCKED SINCE THE SAME FILE MAY BE IN USE BY ANOTHER ACTIVITY WHICH IS EXECUTING AT THE SAME TIME. A LOCK OUT FEATURE CALLED EXCLUSIVE USE IS PROVIDED TO PERFORM THIS FUNCTION. EXCLUSIVE USE MAY SPECIFY LOCK OUT FOR A WHOLE FILE OR INDIVIDUAL ITEMS OR BLOCKS IN A FILE. A FILE BEING PROCESSED RANDOMLY MAY BE EITHER LOCKED OUT IN ITS ENTIRETY OR LOCKED OUT BY ITEMS OR BLOCKS.

LOCKING OUT AN ENTIRE FILE REQUIRES THE APPROPRIATE OPTION ON THE ASSIGN CONTROL CARD FOR THE FILE. WHENEVER THIS IS DONE, THE SYSTEM WILL LOCK OUT ALL ACTIVITIES REFERENCING THIS FILE EXCEPT THE ACTIVITY THAT INITIATED THE LOCK. LOCKING OUT INDIVIDUAL ITEMS OR BLOCKS OF A FILE IS ACCOMPLISHED BY USING THE EXCLUSIVE READ RANDOM REQUEST. IN THIS CASE, ONLY THE ITEM OR BLOCK CURRENTLY BEING READ WILL BE LOCKED OUT. REFERENCES TO THE FILE BY OTHER ACTIVITIES WILL BE HONORED ONLY IF THE REFERENCE IS NOT FOR THE LOCKED OUT ITEM OR BLOCK. AN ACTIVITY REFERENCING A LOCKED OUT ITEM OR BLOCK IS AUTOMATICALLY PLACED IN A WAIT STATE UNTIL THE DATA IS RELEASED. THE NEXT REQUEST TO THE FILE FROM THE ACTIVITY THAT INITIATED THE LOCK RELEASES THAT ITEM OR BLOCK PREVIOUSLY HELD.

A CERTAIN AMOUNT OF INDEPENDENCE FROM PERIPHERAL DEVICES CAN BE ACHIEVED WITH THIS ROUTINE. THAT IS, FILES MAY BE WRITTEN ON FASTRAND OR ANY TAPE TYPE WITHOUT PROGRAM ALTERATIONS BY SIMPLY CHANGING THE ASG CONTROL CARD(FROM TAPE TO FASTRAND OR VICE-VERSA).

THE ONLY RESTRICTIONS ARE THAT ALL FILES PROCESSED IN THE IN/OUT MODE OR REFERENCED RANDOMLY ARE LIMITED TO FASTRAND RESIDENCE.

FOR EACH FILE REFERENCED IN A USER'S PROGRAM, A FILE CONTROL TABLE WITHIN THAT PROGRAM IS REQUIRED (SEE SECTION 13.6). THESE TABLES ENABLE THE COMMUNICATION BETWEEN A USER'S PROGRAM AND THE DATA HANDLING ROUTINES FOR FILE MANIPULATION. EACH REQUEST TO THE SYSTEM MUST CONTAIN THE LOCATION OF THE FILE CONTROL TABLE FOR THAT FILE.

A PORTION OF THE OBJECT PROGRAM'S STORAGE AREA WILL BE USED AS A BUFFER POOL. THE LOCATION AND CONTROL OF THE POOL WILL BE MAINTAINED BY THE BLOCK BUFFERING PACKAGE. THE USER HOWEVER, MUST SPECIFY THE SIZE AND NUMBER OF BUFFERS THE POOL IS TO CONTAIN AND WHETHER IT IS TO BE SHARED BY OTHER FILES IN THE PROGRAM.

13.2. FORMAT DEFINITION

FORMAT DEFINITIONS ARE USED TO DEFINE THE PHYSICAL ORGANIZATION OF DATA FILES. THEY ARE DEFINED USING A PROCEDURE ELEMENT SET. LOCATION OF THE FIRST WORD OF THE PREFIX WILL BE IN WORD 28, H2 WHICH PROVIDES A CONVENIENT LANGUAGE TO ALLOW THE USER TO DESCRIBE TO THE SYSTEM THE EXACT PHYSICAL MAKEUP OF A FORMAT. DESCRIPTIONS MAY BE GIVEN FOR THE TYPE AND CONTENTS OF EACH BLOCK THE FORMAT WILL CONTAIN. THE TYPE OF BLOCK MAY BE DEFINED AS LABEL, DATA, END-OF-FILE, ETC, AND THE CONTENTS OF EACH TYPE IN TERMS OF FIELDS. EACH FIELD MAY DEFINE CONSTANTS, FLAGS, ETC, OR IT MAY BE A SYSTEM RECOGNIZABLE SUBROUTINE NAME TO PERFORM SOME ACTION ON THE DESIGNATED FIELD. ALL DEFINITIONS GIVEN ARE CONTROLLED AND MANIPULATED AUTOMATICALLY BY THE SYSTEM AND, FOR THE MOST PART, ARE NOT AVAILABLE TO THE USER. THEREFORE, THE DEFINITIONS SHOULD BE FILE INDEPENDENT. THAT IS THEY SHOULD BE CODED TO DESCRIBE THIS BLOCK (BLOCK), AN END-OF-REEL (EOR), OR FORMAT FOR A FAMILY OF FILES RATHER THAN A PARTICULAR FILE.

THERE ARE PRESENTLY FIVE TYPES OF PROCEDURES NEEDED TO DEFINE A FORMAT. THEY ARE THE FORMAT, THE RECORD-TYPE, THE SECTION-NAME, THE SUBROUTINE-NAME, AND THE END-FORMAT NAME. AT LEAST ONE OF EACH MUST APPEAR IN ANY FORMAT DEFINITION EXCEPT FOR THE FORMAT AND END-FORMAT PROCEDURES WHICH MUST APPEAR ONLY ONCE AS THE FIRST AND LAST PROCEDURE CALL, RESPECTIVELY. THE OTHER THREE PROCEDURES MAY APPEAR MORE THAN ONCE, BUT THE ORDER OF THEIR APPEARANCE MUST BE AS STATED BELOW.

- RECORD-TYPE PROCEDURES, WHICH PRIMARILY DEFINE THE LOGICAL BEGINNING OF A RECORD-TYPE, MAY APPEAR IN ANY ORDER AFTER THE FORMAT PROCEDURE
- SECTION-NAME PROCEDURES WHICH DEFINE ONE OF THREE LOGICAL AREAS WITHIN A RECORD-TYPE, MAY APPEAR IN ANY ORDER AFTER A RECORD-TYPE PROCEDURE, WITH THE EXCEPTION OF EOFMRK AS NOTED BELOW.
- SUBROUTINE-NAME PROCEDURES, WHICH NAME A PARTICULAR SUBROUTINE TO PERFORM SOME ACTION ON A DESIGNATED FIELD WITHIN A SECTION, MAY APPEAR IN ANY ORDER AFTER A SECTION-NAME PROCEDURE.

13.2.1. THE FORMAT PROCEDURE

THE FORMAT PROCEDURE MUST BE THE FIRST AND ONLY ONE IN A LIST OF DEFINITIONS OF A FORMAT. IT IS CODED AS

```
TAG    FORMAT
```

WHERE THE TAG OF THE PROCEDURE SERVES TO IDENTIFY TO THE SYSTEM A NAME BY WHICH THIS FORMAT WILL BE KNOWN. THIS NAME WILL BE USED WHENEVER THIS FORMAT IS CALLED BY A PROGRAM. THE NAME OF THE CALL IS FORMAT. IT MUST BE USED AND SPELLED AS SHOWN.

13.2.2. THE RECORD-TYPE PROCEDURE

THE RECORD-TYPE PROCEDURE DEFINES TO THE SYSTEM THE TYPE OF RECORD PRESENTLY BEING DESCRIBED. IT IS CODED AS

```
        RECORD-TYPE
```

WHERE RECORD-TYPE MAY SPECIFY A LABEL (LABEL), AN ITEM (ITEM), AN END-OF-FILE (EOF). EACH NAME INDICATES THAT THE DEFINITIONS THAT FOLLOW, UP TO ANOTHER RECORD-TYPE PROCEDURE, ARE TO BE INCLUDED AS PART OF THIS RECORD-TYPE. ANY OR ALL OF THE RECORD-TYPE NAMES MAY BE USED IN ANY ORDER; HOWEVER, THEY MAY BE USED ONLY ONCE IN ANY FORMAT DEFINITION, AND THE SPELLING MUST BE AS SHOWN. FOR EXAMPLE:

```
(1)  XYZ   FORMAT  (2) ABC FORMAT
           LABEL           LABEL
           -               -
           -               -
           -               -
           -               -
           ITEM            ITEM
           -               -
           -               -
           -               -
           -               -
           BLOCK           EOF
           -               -
           -               -
           -               -
           -               -
           EOR
           -
           -
           -
           -
           EOF
           -
           -
           -
           -
```

(1) SPECIFIES THAT THE RECORD-TYPES IN THE XYZ FORMAT WILL BE

LABELS, ITEMS, BLOCKS, END-OF-REELS AND END-OF-FILES. THE DEFINITIONS FOR EACH WILL FOLLOW THEIR RESPECTIVE NAMES.

(2) SPECIFIES THAT THE RECORD-TYPES IN THE ABC FORMAT WILL BE LABELS, ITEMS, AND END-OF-FILE. SINCE BLOCK FORMAT IS NOT DEFINED, THE SYSTEM WILL GENERATE BLOCKS CONTAINING NO OTHER DATA EXCEPT ITEMS.

13.2.3. THE SECTION-NAME PROCEDURE

THE SECTION-NAME PROCEDURE IS USED TO SUBDIVIDE THE RECORD-TYPE PROCEDURE PRESENTLY BEING DEFINED INTO ONE OF FOUR LOGICAL AREAS, AT LEAST ONE SECTION-NAME PROCEDURE MUST BE GIVEN FOR EACH RECORD-TYPE PROCEDURE. IT IS CODED AS;

```
SECTION-NAME
```

WHERE PREFIX (PREFIX), TEXT (TEXT), SUFFIX (SUFFIX), AND END OF FILE MARK (EOFMRK) ARE SECTION-NAMES AND MUST BE SPELLED AS SHOWN. THE PREFIX INDICATES TO THE SYSTEM THAT THE DEFINITIONS THAT FOLLOW DESCRIBE A LOGICAL AREA IMMEDIATELY PRECEDING A DATA ITEM, BLOCK, OR USER'S LABEL IN WHERE THE SYSTEM WOULD AUTOMATICALLY CONTROL THE PLACEMENT OR EXTRACTION OF INFORMATION. THE TEXT INDICATES TO THE SYSTEM THAT THE FOLLOWING DEFINITIONS DESCRIBE THE PORTION WHICH IS THE ACTUAL DATA ITEM OR BLOCK. THE SUFFIX INDICATES TO THE SYSTEM THAT THE FOLLOWING DEFINITIONS DESCRIBE A LOGICAL AREA IMMEDIATELY FOLLOWING A DATA ITEM, BLOCK, OR USER'S LABEL IN WHICH THE SYSTEM WOULD AUTOMATICALLY CONTROL THE PLACEMENT OR EXTRACTION OF INFORMATION. THESE THREE SECTION-NAMES MAY APPEAR IN ANY ORDER WITHIN A RECORD-TYPE.

THE FOURTH SECTION-NAME (EOFMRK) INDICATES THAT AN END-OF-FILE MARK IS TO BE WRITTEN ON TAPE AT OTHER THAN THE END OF REEL OR FILE. THE USE OF THIS SECTION-NAME CALL IS LIMITED TO THE FOLLOWING FILE LOCATIONS: ONE END-OF-FILE MARK FOLLOWING A LABEL BLOCK AND ONE END-OF-FILE MARK PRECEEDING AN END-OF-FILE OR END OF REEL FORMAT BLOCK. WITHIN A LABEL RECORD, THE EOFMRK CALL MUST BE THE LAST SECTION NAME CALLED. WITHIN AN EOF OR EOR RECORD, THE EOFMRK CALL MUST BE THE FIRST SECTION-NAME CALLED. THE MARK WILL BE WRITTEN ON AN OUTPUT TAPE FILE, CHECKED FOR ON AN INPUT TAPE FILE, AND IGNORED ON A MASS STORAGE FILE.

A LIST MAY BE USED ON PREFIX AND SUFFIX RECORDS TO SPECIFY THE LENGTH OF THE PREFIX OR SUFFIX TO BE READ OR WRITTEN. IF NOT PRESENT, THE LENGTH WILL BE COMPUTED FROM THE SUBROUTINES AS SHOWN BELOW. LGTH IS A NUMERIC VALUE.

THE FOLLOWING EXAMPLES SHOW THE USE OF SECTION-NAMES.

```
(1)   XYZ  FORMAT     (2) ABC  FORMAT
           LABEL               LABEL
           PREFIX  LGTH        PREFIX
           -                   -
           -                   -
           SUFFIX  LGTH        ITEM
           -                   TEXT
           EOFMRK              -
           ITEM                EOF
           PREFIX              EOFMRK
           -                   PREFIX
           -                   -
           -                   -
           TEXT               -
           -
           -
           SUFFIX
           -
           -
           BLOCK
           PREFIX  LGTH
           -
```

```
-
SUFFIX
-
-
EOR
PREFIX
-
-
SUFFIX
-
-
EOF
PREFIX  LGTH
-
-
SUFFIX  LGTH
-
-
```

(1) IN THE XYZ FORMAT, THE USER HAS DEFINED EACH RECORD-TYPE TO CONTAIN FORMAT INFORMATION BEFORE AND AFTER THE ACTUAL DATA. THE LENGTHS OF THE LABEL PREFIX AND SUFFIX, THE BLOCK PREFIX, AND THE EOF PREFIX AND SUFFIX HAVE BEEN DEFINED. AN END-OF-FILE MARK WILL BE WRITTEN FOLLOWING EACH LABEL BLOCK.

(2) IN THE ABC FORMAT, THE USER HAS DEFINED FORMAT INFORMATION BEFORE THE USER LABEL, IN THE ACTUAL ITEM, AND BEFORE ANY USER END-OF-FILE INFORMATION. ALSO AN END-OF-FILE MARK WILL BE WRITTEN PRECEEDING EACH EOF BLOCK.

13.2.4. THE SUBROUTINE-NAME PROCEDURE

THE SUBROUTINE-NAME PROCEDURE IS THE HEART OF THE FORMAT DEFINITION SCHEME. IT DEFINES THE DISPOSITION OF INDIVIDUAL FIELDS IN A SECTION AS WELL AS DIRECTS THE SYSTEM TO INITIATE THE COLLECTION OF THE REQUIRED SUBROUTINES TO PROCESS THIS FORMAT. AS MANY SUBROUTINE-NAME PROCEDURES AS NECESSARY TO COMPLETELY DEFINE A SECTION MAY BE USED. THERE ARE TWO PROCEDURE CALLS, SENT AND SUBR. SENT MUST BE USED IN EACH LABEL, END-OF-REEL, AND END OF FILE BLOCK TO SPECIFY THE SENTINEL. ON INPUT, THIS SUBROUTINE WILL BE GIVEN CONTROL TO CHECK THE TYPE OF BLOCK. IF A COMPARISON IS NOT MADE, RETURN TO THE ITEM HANDLER WILL BE MADE IMMEDIATELY. IF A SENTINEL IS FOUND, THE OTHER RECORD SUBROUTINES WILL BE GIVEN CONTROL TO EXAMINE FURTHER PARTS OF THE RECORD. SUBR MUST BE USED TO CALL ON ALL OTHER SUBROUTINES.

THE FIRST LIST OF THE SENT AND SUBR CALLS MUST BE THE LOCATION OF THE SUBROUTINE TO WHICH CONTROL WILL BE GIVEN. THIS SUBROUTINE WILL PERFORM A FUNCTION ON ONE WORD AT A TIME AND DISPOSE OF THE RESULTS IN THE FIELD DESIGNATED IN THE CALL LINE. SUBROUTINE SELECTION IS FROM A POOL OF AVAILABLE SYSTEM SUBROUTINES. IF HOWEVER A PARTICULAR SUBROUTINE NEEDED IS NOT IN THE SYSTEM, THE USER MAY INJECT HIS OWN.

THE FOLLOWING LIST OR LISTS SPECIFIES THE PARAMETERS TO BE WRITTEN OR CHECKED BY THE SUBROUTINE. THEY MAY APPEAR IN ONE OF THE TWO FORMS FOLLOWING.

```
(1)  WDNBR,J-DESIG,VALUE ... WDNBR,J-DESIG,VALUE
                           OR
(2)  WDNBR,CHAR(CHAR-SIZE,CHAR-POS,#-CHAR-FOR-OP),VALUE
```

WHERE:

WDNBR = THE WORD NUMBER RELATIVE TO THE START OF THE APPROPRIATE SECTION-NAME (PREFIX, TEXT, OR SUFFIX) ON WHICH THE SUBROUTINE WILL OPERATE. THE CALLS NEED NOT BE IN SEQUENCE, AND ONLY THOSE WORDS OR PORTION OF WORDS NECESSARY TO DEFINE THE FORMAT NEED BE USED, EXCEPT - IF THE PREFIX OR SUFFIX

LENGTH IS NOT SPECIFIED ON THE SECTION-NAME, THE LAST WORD IN THE SECTION MUST BE CALLED. IF AN ITEM IS DEFINED AS HAVING A 5 WORD PREFIX AND A 2 WORD SUFFIX, THEN WDNBR 0 OF THE PREFIX PINPOINTS THE WORD FURTHEST FROM THE ACTUAL ITEM, WHILE WDNBR 4 IS THE WORD IMMEDIATELY PRECEEDING THE ACTUAL ITEM OR TEXT. CONVERSLY WDNBR 0 OF THE SUFFIX IMMEDIATELY FOLLOWS THE LAST WORD OF THE ITEM.

J-DESIG = THE NUMERICAL J DESIGNATOR FROM 0-015 OF THE PART OF THE WORD TO BE ACTED UPON.

VALUE = THE VALUE TO BE PLACED IN THE APPROPRIATE PORTION OF THE WORD, AS SPECIFIED BY THE J-DESIG, ON OUTPUT, AND CHECKED ON INPUT.

CHAR-SIZE = THE NUMBER OF BITS IN A CHARACTER

CHAR-POS = THE CHARACTER POSITION OF THE LEFT-MOST CHARACTER TO BE ACTED ON, STARTING NUMBERING FROM THE LEFT SIDE OF THE WORD, FROM 1 TO N.

#-CHAR-FOR-OP = THE NUMBER OF CHARACTERS (OF THE SIZE SPECIFIED BY CHAR-SIZE) TO BE OPERATED ON BY THE SUBROUTINE.

EXAMPLES OF THE SUBROUTINE-NAME PROCEDURE ARE

```
(1)    SUBR    BLKCT    3,0
```

WHERE BLKCT IS THE NAME OF A SUBROUTINE WHICH WILL GENERATE AND CHECK A BLOCK COUNT IN THE FOURTH WORD, WHOLE WORD, OF SOME SECTION.

```
(2)    SENT    SENTNL    0,1,'/./.'    1,2,06
```

WHERE SENTNL IS A SENTINEL ROUTINE WHICH WILL PLACE THE SENTINEL '/./.' IN THE SECOND HALF OF THE FIRST WORD AND THE SENTINEL 06 IN THE FIRST HALF OF THE SECOND WORD OF A RECORD. THESE TWO PORTIONS OF WORDS WILL BE USED TO FIND THE SPECIFIC RECORD ON INPUT.

```
(3)    SUBR    CNT      2,CHAR(4,3,2),0
```

WHERE CNT IS THE NAME OF A SUBROUTINE WHICH WILL GENERATE AND CHECK SOMETHING IN THE THIRD WORD OF A RECORD. THIS WORD IS A 4-BIT CHARACTER-ORIENTATED WORD. THE COUNT WILL BE MAINTAINED IN 2 CHARACTERS STARTING WITH CHARACTER 3 (I.E. BITS 27-20).

13.2.5. THE END-FORMAT PROCEDURE

THE END-FORMAT PROCEDURE SIGNALS THE END OF A COMPLETE FORMAT DEFINITION AND MUST APPEAR AS THE LAST PROCEDURE. IT IS CODED AS

```
FOREND
```

WHERE FOREND INDICATES TO THE SYSTEM THAT NO MORE DEFINITIONS WILL BE GIVEN FOR THIS FORMAT. THERE MUST BE AS MANY END-FORMAT PROCEDURES AS THERE ARE FORMAT PROCEDURES. FOR EXAMPLE

```
LION FORMAT
     -
     -
     -
     FOREND
ABC  FORMAT
     -
     -
     -
```

```
          FOREND
```

13.2.6. EXAMPLE-COMPLETE FORMAT DEFINITION

AS AN EXAMPLE TO SHOW A COMPLETELY DEFINED FORMAT, THE FOLLOWING ARE DESCRIPTIONS TO SPECIFY LION FORMAT TO THE 1108 DATA HANDLING ROUTINE.

```
     LION FORMAT
          LABEL
          PREFIX
          SENT     LLABST    0,0,'/././.'   1,13,060
          SUBR     LBLKCT    1,0
          SUBR     LBLKLT    2,2
          SUBR     LITEMS    2,1
          SUBR     LFLWDS    3,2
          SUBR     LFREE     3,1
          SUFFIX
          SUBR     LLSRWD    0,1
          SUBR     LVERSN    0,2,4
          SUBR     LITMCT    1,0
          SUBR     LCKSUM    2,0
          SUBR     LBLKLT    3,2
          SUBR     LITEMS    3,1
          SUBR     LBLKCT    4,0
          SUBR     LLABST    4,13,060       5,0,'/././.'
          BLOCK
          PREFIX
          SUBR     LNRITM    0,2
          SUBR     LNRWDS    0,1
          SUFFIX
          SUBR     LDSRWD    0,1
          SUBR     LBCKSM    1,0
          SUBR     LNRITM    2,2
          SUBR     LNRWDS    2,1
          EOR
          PREFIX
          SENT     LEORST    0,0,'/././.'   1,13,020
          SUBR     LERBCT    1,0
          SUBR     LERBLT    2,2
          SUBR     LERITS    2,1
          SUBR     LZEROW    3,0,0          4,0,0
          SUBR     LVERSN    5,2,4
          SUBR     LZEROW    6,0,0
          SUBR     LSUFLC    7,0,0
          SUFFIX
          SUBR     LVERSN    0,2,4
          SUBR     LERICT    1,0
          SUBR     LERCKM    2,0
          SUBR     LERBLT    3,2
          SUBR     LERITS    3,1
          SUBR     LERBCT    4,0
          SUBR     LEORST    4,13,020       5,0,'/././.'
          EOF
          PREFIX
          SENT     LEOFST    0,0,'/././.'   1,13,0
          SUBR     LEFBCT    1,0
          SUBR     LEFBLT    2,2
          SUBR     LEFITS    2,1
          SUBR     LZEROW    3,0,0          4,0,0
          SUBR     LVERSN    5,2,4
          SUBR     LZEROW    6,0,0
          SUBR     LSUFLC    7,0,0
          SUFFIX
          SUBR     LVERSN    0,2,4
```

```
SUBR     LEFICT    1,0
SUBR     LEFCKM    2,0
SUBR     LEFBLT    3,2
SUBR     LEFITS    3,1
SUBR     LEFBCT    4,0
SUBR     LEOFST    4,13,0          5,0,'/./././'
FOREND
```

13.2.7. RULES FOR CODING FORMAT SUBROUTINES

WHEN A SPECIFIC FORMAT MUST BE GENERATED OR CHECKED ON A FILE, CONTROL WILL BE GIVEN TO THE APPROPRIATE SUBROUTINE BY THE ITEM HANDLER. TO DETERMINE WHETHER TO GENERATE OR CHECK A FORMAT, CONTROL WILL BE GIVEN WITH A0 POSITIVE FOR OUTPUT FILES, NEGATIVE FOR INPUT FILES.

THE FOLLOWING REGISTERS WILL BE SET AS INDICATED WHEN THE SUBROUTINE GETS CONTROL AND MUST NOT BE DESTROYED BY THE SUBROUTINE.

```
A1 - H1 - RETURN TO ITEM FROM FORMAT
     H2 - RETURN TO FORMAT FROM SUBROUTINE
A2 - H2 - ITEM REQUEST PACKET LOCATION
A3 - H1 - RETURN TO USER FROM ITEM
     H2 - FILE CONTROL TABLE LOCATION
```

THE FOLLOWING REGISTER IS SET ON ENTRY TO THE SUBROUTINE. THE UPPER SIXTH MUST NOT BE DESTROYED, THE REST MAY BE.

```
A0 - S1 - POSITIVE FOR OUTPUT FILES
          NEGATIVE FOR INPUT FILES
   - H2 - LOCATION OF PACKET CONTAINING PARAMETERS FROM
          SUBROUTINE CALL LINE. IT APPEARS AS FOLLOWS:
```

```
------------------------------------------
:                  VALUE                 :
------------------------------------------
:WDNBR :J-DESG: CODE :CH SIZ:CH POS:# CHAR:
------------------------------------------
```

```
WHERE THE FIELDS ARE AS EXPLAINED ABOVE, AND
      CODE = 0 - PREFIX SECTION
             1 - TEXT SECTION
             2 - SUFFIX SECTION
      H2 - 0 IF THE FIRST SUBROUTINE FORM IS USED.
```

THESE REGISTERS MAY BE USED FREELY: X11, A4, A5, R1, R2, R3. ALL OTHERS MUST BE SAVED AND RESTORED BEFORE EXIT FROM THE SUBROUTINE. BECAUSE THE ITEM HANDLER IS REENTRANT, ALL SUBROUTINES WILL BE TREATED AS SUCH; THEREFORE, THEY MUST BE CODED WITH THIS IN MIND.

THE EXITS FROM THE SUBROUTINES FOR NORMAL CONDITIONS MUST BE TAKEN ON 0, A1. EXITS FOR ABNORMAL CONDITIONS WILL BE TAKEN TO IHABFR WITH ONE OF THE FOLLOWING CODES IN A0, S6:

```
 01 = EXIT FROM LABEL SENTINEL ROUTINE WITH NO FIND.
 02 = EXIT FROM LABEL ROUTINE WITH ABNORMAL LABEL.
 03 = EXIT FROM BLOCK ROUTINE WITH ABNORMAL BLOCK
 04 = EXIT FROM ITEM ROUTINE WITH ABNORMAL ITEM
 05 = EXIT FROM EOR SENTINEL ROUTINE WITH NO FIND
 06 = EXIT FROM EOR ROUTINE WITH ABNORMAL EOR BLOCK.
 07 = EXIT FROM EOF SENTINEL ROUTINE WITH NO FIND
010 = EXIT FROM EOF ROUTINE WITH ABNORMAL EOF BLOCK.
```

A0, S4 AND S5, MAY BE USED TO TRANSFER ANY DATA PERTINENT TO THE USER. AFTER INTERPRETING THE CODE IN S6, THE ITEM HANDLER WILL TAKE THE USER'S ABNORMAL EXIT WITH A0 AND A1 SET AS FOLLOWS.

A0 = LENGTH AND LOCATION OF CURRENT BLOCK OR ITEM .
A1 = H1 - CODE PERTINENT TO USER
     H2 - ADDRESS OF REQUEST THAT FOUND ERROR

13.2.7.1. LABEL RECORD SUBROUTINES

ONLY THREE OF THE FOUR SECTION-NAMES MAY BE USED WITH THE LABEL RECORD. THESE ARE PREFIX, SUFFIX, AND EOFMRK. A LIST MAY BE SPECIFIED ON THE PREFIX AND SUFFIX CALLS TO DESIGNATE THE LENGTH OF THE RESPECTIVE SECTION. IF NOT SET, THE LENGTH WILL BE COMPUTED FROM THE PARAMETERS ON THE SUBROUTINE CALLS. THE TOTAL LENGTH OF THESE SECTIONS WILL BE THE TOTAL LENGTH OF THE LABEL BLOCK. IT MUST BE LESS THAN OR EQUAL TO THE MAXIMUM DATA BLOCK SIZE ALLOWED.

IF THE LABEL OR FREEWDS(SEE SECTION BELOW ON LION FORMAT) ARE SPECIFIED BY THE USER IN THE FILE CONTROL TABLE, A MOVE ROUTINE MUST BE INCLUDED AS A SUBROUTINE TO MOVE THIS AREA INTO THE LABEL BLOCK. THE LOCATION OF THE START OF THE PREFIX WITHIN THE BUFFER WILL BE FOUND IN WORD 28, H2 OF THE FCT. THE LOCATION OF THE START OF THE SUFFIX WILL BE FOUND IN WORD 26, H2 OF THE FCT.

IF EOFMRK IS SPECIFIED AS A LABEL SECTION-NAME, AN END-OFFILE MARK WILL BE WRITTEN ON OUTPUT OR EXPECTED ON INPUT IMMEDIATELY FOLLOWING A LABEL BLOCK. NO SUBROUTINE CALLS MAY FOLLOW AN EOFMRK.

LABEL OUTPUT SUBROUTINES WILL PLACE DATA IN THE LABEL PREFIX AND SUFFIX AS SPECIFIED BY THE SUBROUTINE LISTS. A SUBROUTINE CALLED BY THE PROC CALL SENT MUST BE INCLUDED AS ONE OF THE SUBROUTINE CALLS IN A LABEL RECORD. RETURN TO FORMAT FROM EACH SUBROUTINE MUST BE MADE ON 0, A1 IF THE LABEL IS GENERATED PROPERLY AND TO IHABFR WITH A0, S6 = 2 IF NOT DONE PROPERLY.

LABEL INPUT SUBROUTINES MUST VALIDATE THE LABEL BLOCK. THE SENTINEL ROUTINE WILL FIRST CHECK THE SENTINEL. IF IT IS FOUND, THE ROUTINE MUST RETURN ON 0, A1. IF IT IS NOT FOUND, RETURN MUST BE MADE TO IHABFR WITH A0, S6 = 1. SINCE THIS ROUTINE IS CALLED ON EVERY READ REQUEST, CARE MUST BE MADE TO MAKE IT VERY CONCISE. ALL OTHER LABEL INPUT SUBROUTINES MUST CHECK OTHER PARTS OF THE LABEL. IF THE LABEL IS ACCEPTED AS VALID, RETURN MUST BE MADE ON 0, A1. IF THE SUBROUTINUE WISHES TO DISCONTINE PROCESSING, IT MUST RETURN TO IHABFR WITH A0, S6 = 2.

13.2.7.2. BLOCK RECORD SUBROUTINES

WITHIN A BLOCK RECORD, PREFIX, TEXT AND SUFFIX, SECTION'NAMES MAY BE USED. THE LENGTH OF THE PREFIX AND SUFFIX MAY BE CAN BE COMPUTED FROM THE SUBROUTINE CALL. FORMAT SUBROUTINES SPECIFIED AS A LIST ON THE APPROPRIATE SECTION CALL, OR THE LENGTH FOR OUTPUT FILES WILL GET CONTROL FOR GENERATING BLOCK FORMATS AFTER ALL ITEMS HAVE BEEN MOVED INTO THE BLOCK. INPUT SUBROUTINES WILL GET CONTROL BEFORE ANY ITEMS HAVE BEEN MOVED FROM THE BLOCK. THE LOCATION OF THE FIRST WORD OF THE PREFIX WILL BE IN WORD 28, H2 OF THE FCT, THE TEXT IN WORD 28, H1 OF THE FCT AND THE SUFFIX IN WORD 26, H2 OF THE FCT.

RETURN FROM EACH SUBROUTINE SHOULD BE MADE ON 0, A1 FOR NORMAL CONDITIONS. WHEN FURTHER PROCESSING OF THE FILE IS NO LONGER DESIRED OR THE BLOCK WAS GENERATED INCORRECTLY, THE SUBROUTINE SHOULD EXIT TO IHABFR WITH A0, S6 = 3.

13.2.7.3. ITEM RECORD SUBROUTINES

WITHIN AN ITEM RECORD PREFIX, TEXT, AND SUFFIX SECTION-NAMES MAY BE USED. THE LENGTH OF THE PREFIX AND SUFFIX MAY BE SPECIFIED AS A LIST ON THE APPROPRIATE SECTION CALL, OR THE LENGTH CAN BE COMPUTED FROM THE SUBROUTINES. THE LOCATION OF THE BEGINNING OF THE PREFIX WILL BE IN WORD 28, H2 OF THE FCT, THE TEXT IN WORD 26, H2 OF THE FCT. THE SUFFIX WILL BE WORD 26, H2 OF THE FCT. CONTROL WILL BE GIVEN AT DIFFERENT TIMES.

FOR OUTPUT FILES, CONTROL WILL BE GIVEN TO THE PREFIX SUBROUTINES BEFORE THE ITEM IS MOVED UNLESS THE USER HAS ALREADY MOVED IT HIMSELF AND TO THE TEXT AND SUFFIX SUBROUTINES AFTER THE ITEM IS MOVED.

FOR INPUT ITEMS IN THE FORWARD MODE, CONTROL WILL BE GIVEN TO THE PREFIX AND TEXT SUBROUTINES BEFORE THE ITEM IS MOVED, AND TO THE SUFFIX SUBROUTINES AFTER IT IS MOVED.

FOR INPUT FILES IN THE REVERSE MODE, CONTROL WILL BE GIVEN TO THE SUFFIX AND TEXT SUBROUTINES BEFORE THE ITEM IS MOVED, AND TO THE PREFIX SUBROUTINES AFTER IT IS MOVED.

IF ITEM SIZE IS VARIABLE, A SUBROUTINE MUST BE INCLUDED UNDER BOTH THE PREFIX AND SUFFIX. THEREFORE, IF A VARIABLE SIZE ITEM HAS A PREFIX, IT MUST HAVE A SUFFIX AND VICE VERSA. THIS SUBROUTINE MUST, ON OUTPUT, INSERT THE LENGTH OF THE ITEM AND FORMATS WITHIN THE PREFIX AND SUFFIX. THIS LENGTH WILL BE FOUND IN WORD 29, H2 OF THE FCT. ON INPUT, THE SUBROUTINE MUST RETRIEVE THIS LENGTH AND PLACE IT IN WORD 29, H2 OF THE FCT. IF NEITHER PREFIX NOR SUFFIX ARE SPECIFIED, A ONE WORD CONTROL WORD WILL BE PLACED BEFORE AND AFTER EACH ITEM. IF ITEMS ARE FIXED, THE LENGTHS NEED NOT BE SAVED. IF THE FILE WILL NOT BE PROCESSED IN THE REVERSE MODE, THE SUFFIX IS NOT NEEDED.

RETURN FROM EACH SUBROUTINE SHOULD BE MADE ON 0, A1 EXCEPT WHEN FURTHER PROCESSING OF THE ITEM IS NO LONGER DESIRED. THEN THE SUBROUTINE SHOULD EXIT TO IHABFR WITH A0, S6 =4.

13.2.7.4. END OF REEL SUBROUTINES

END-OF-REEL PROCESSING IS SIMILAR TO PROCESSING OF LABEL BLOCKS. THREE SECTION NAMES, PREFIX, SUFFIX, AND EOFMRK MAY BE USED. EOFMRK INDICATES THAT AN END-OF-FILE MARK IS TO BE WRITTEN ON TAPE BEFORE THE EOR BLOCK. THE PREFIX AND SUFFIX LENGTHS WILL BE DETERMINED FROM THE LIST ON THE CALL LINE OR FROM THE PARAMETERS IN THE SUBROUTINE PACKET. THE TOTAL LENGTH MUST BE EQUAL TO OR LESS THAN THE MAXIMUM BLOCK LENGTH. OF THE FCT, THE LOCATION OF THE FIRST WORD OF THE PREFIX WILL BE IN WORD 28, H2 OF THE FCT, AND THE SUFFIX IN WORD 26,H2 OF THE FCT. THE SUFFIX IN WORD 26, H2 OF THE FCT.

ON OUTPUT AN END-OF-REEL BLOCK WILL BE WRITTEN WHEN AN ENDOF-TAPE IS ENCOUNTERED. ONE SENT SUBROUTINE MUST BE INCLUDED. EXIT SHOULD BE MADE ON 0, A1.

ON INPUT, THE SENT SUBROUTINE WILL BE GIVEN CONTROL TO CHECK FOR AN END-OF-REEL SENTINEL. IF NOT FOUND, AN EXIT SHOULD BE MADE TO IHABFR WITH A0 = 5. IF IT IS FOUND, EXIT SHOULD BE MADE ON 0, A1. CONTROL WILL THEN GO TO OTHER END-OF-REEL SUBROUTINES FOR FURTHER CHECKING. IF AN ABNORMAL END-OF-REEL BLOCK IS FOUND AND PROCESSING SHOULD BE DISCONTINUED, AN EXIT SHOULD BE MADE TO IHABFR WITH A0, S6 = 6. THE NORMAL RETURN WILL BE MADE ON 0, A1.

13.2.7.5. END-OF-FILE SUBROUTINES

END'OF'FILE PROCESSING IS SIMILAR TO PROCESSING OF LABEL BLOCKS. THREE SECTION NAMES, PREFIX, SUFFIX, AND EOFMRK MAY BE USED. EOFMRK INDICATES THAT AN END-OF-FILE MARK IS TO BE WRITTEN ON TAPE BEFORE THE EOF BLOCK. THE PREFIX AND SUFFIX LENGTHS WILL BE DETERMINED FROM THE LIST ON THE CALL LINE OR FROM THE PARAMETERS IN THE SUBROUTINE PACKET. THE TOTAL LENGTH MUST BE EQUAL TO OR LESS THAN THE MAXIMUM BLOCK LENGTH. THE LOCATION OF THE FIRST WORD OF THE PREFIX WILL BE IN WORD 28, H2 OF THE FCT, AND THE SUFFIX IN WORD 26, H2 OF THE FCT.

ON OUTPUT AN END-OF-FILE BLOCK WILL BE WRITTEN WHEN THE FILE IS CLOSED. ONE SENT SUBROUTINE MUST BE INCLUDED. EXIT SHOULD BE MADE ON 0, A1.

ON INPUT, THE SENT SUBROUTINE WILL BE GIVEN CONTROL TO CHECK FOR AN END-OF-FILE SENTINEL. IF IT IS NOT FOUND, AN EXIT SHOULD BE MADE TO IHABFR WITH A0 = 7. IF IT IS FOUND, EXIT SHOULD BE MADE ON 0, A1. CONTROL WILL THEN GO TO OTHER END-OF-FILE SUBROUTINES FOR FURTHER CHECKING. IF AN ABNORMAL END-OF-FILE BLOCK IS FOUND AND PROCESSING SHOULD BE DISCONTINUED, AN EXIT SHOULD BE MADE TO IHABFR WITH A0, S6 = 010. THE NORMAL RETURN WILL BE MADE ON 0, A1.

13.3. ITEM CONTROL

ITEM CONTROL IS RESPONSIBLE FOR THE INTERNAL MANIPULATION OF A FILE AT THE ITEM LEVEL. IT IS RESPONSIBLE FOR SUCH FUNCTIONS AS THE BLOCKING AND DEBLOCKING OF ITEMS, THE REFERENCE TO THE FORMAT DEFINITIONS TO CREATE OR VALIDATE ITEMS, THE TRANSFER OF DATA BETWEEN THE USER'S CORE AND BUFFERS, AND THE MAINTAINENCE OF CERTAIN AREAS OF THE FILE CONTROL TABLE NECESSARY WHEN PROCESSING ITEMS.

IN ADDITION CERTAIN LIAISON FUNCTIONS ARE PERFORMED WHEN APPROPRIATE. THE BLOCK BUFFERING PACKAGE IS CALLED UPON TO PERFORM BLOCK FUNCTIONS SUCH AS READ AND WRITE, BUFFER SWAPPING, ETC, AND THE TAPE HANDLING ROUTINES TO PROCESS END-OF-REEL CONDITIONS, UNIT SWAPPING, ETC.

THE ITEM HANDLER USES, AND DESTROYS, THE USER,S MINOR SET OF REGISTERS.

13.3.1. OUTPUT

THE OUTPUT SUBROUTINES PROVIDE FOR THE CREATION OF FILES ON AN EXTERNAL MEDIA. THESE SUBROUTINES ARE CONCERNED ONLY WITH THE MANIPULATION OF THE DATA PRESENTED BY THE OBJECT PROGRAM. LABELS, ITEMS, BLOCKS, END-OF-REEL AND END-OF-FILE SENTINELS ARE ALTERED BEFORE WRITING ACCORDING TO THE FORMAT DEFINITIONS GIVEN FOR EACH. ABSENCE OF A PARTICULAR DEFINITION IMPLIES THAT THIS TYPE OF RECORD IS TO BE PROCESSED UNALTERED. HOWEVER, THE IMPLICATION OF ABSENT LABEL AND SENTINEL DEFINITIONS IS THAT NONE WILL EXIST. THE WORK OF THE OUTPUT SUBROUTINES IS COMPLETE WHEN IT INSTRUCTS THE BLOCK BUFFERING PACKAGE TO PLACE THE BLOCK IN THE OUTPUT FILE.

UPON RETURN TO THE USER FROM AN OPEN, WRITE, WRITE RANDOM OR DRAIN REQUEST, REGISTER A0 WILL CONTAIN INFORMATION IN THE FORMAT OF AN ACCESS WORD.

```
35        17       0
--------------------
:NBR WORDS:ADDRESS :
--------------------
```

THIS WORD DEFINES AN AREA IN THE BUFFER INTO WHICH THE NEXT ITEM WILL BE PLACED. THE SIZE OF THIS AREA DEPENDS ON WHETHER ITEMS HAVE BEEN SPECIFIED AS FIXED OR VARIABLE. FOR FIXED SIZE ITEMS, THE AREA SIZE GIVEN IS EQUAL TO THE ITEM SIZE SPECIFIED. FOR VARIABLE SIZE ITEMS, THE AREA SIZE GIVEN IS THE REMAINING AREA IN THE BUFFER(I.E., AS THE BUFFER IS FILLED THE AREA GIVEN DECREASES).

THE FUNCTIONS PROVIDED BY THESE ROUTINES INCLUDE OPEN FILE, WRITE (SEQUENTIAL), WRITE (RANDOM), DRAIN, CLOSE REEL, CLOSE FILE AND RELEASE. THESE FUNCTIONS ARE PROCEDURAL CALLS, CODED IN THE OBJECT PROGRAM, TO REQUEST THE DESIRED FUNCTION.

WHENEVER A TAPE DEVICE IS ASSIGNED TO A FILE AND THE PHYSICAL END OF TAPE ENCOUNTERED, THE ROUTINE RETURNS CONTROL TO THE USER IF THE END OF REEL EXIT IS CODED. IF NOT, THE CLOSE REEL PROCEDURES ARE INITIATED AUTOMATICALLY.

IF THE END OF REEL EXIT IS TAKEN, REGISTER A1 WILL CONTAIN THE ADDRESS TO WHICH THE USER MUST RETURN CONTROL AFTER HIS CLOSE PROCEDURES, TO REENTER THE MAIN SEQUENCE OF EXECUTION. A1, H1 WILL CONTAIN THE FCT LOCATION AND MUST BE RESET ON RETURN.

13.3.1.1. OPEN OUTPUT

THE OPEN OUTPUT REQUEST INITIALIZES A FILE CONTROL TABLE FOR SUBSEQUENT OUTPUT OPERATIONS ON THE DEVICE ASSIGNED TO THE TABLE. THERE ARE NO RESTRICTIONS ON THE NUMBER OF OPEN REQUESTS WHICH MAY BE CODED TO REFERENCE AN OUTPUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE OPEN REQUEST IS CODED

```
   TAG  OPEN 'MODE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. OPEN IS THE NAME OF THE REQUEST AND

MUST BE USED AS SHOWN. 'MODE' SPECIFIES THE INTENDED MODE OF OPERATION FOR THIS FILE AND MUST BE CODED AS 'OUTPUT' FOR OUTPUT FILES. THE FCTABLE SUBFIELDS ARE THE SYMBOLIC LOCATIONS OF THE FILE CONTROL TABLES THAT ARE TO BE INITIATED. THE 'OPTION' SUBFIELDS, IF USED, SPECIFIES A NO REWIND OPTION ON OPENING AND MUST BE CODED AS 'N'. IF THE FILE IS TO BE REWOUND BEFORE OPENING, THE 'OPTION' SUBFIELD MUST BE OMITTED. FOR EXAMPLE THE OPEN REQUEST:

```
   ABC  OPEN 'OUTPUT' ALPHA BETA,'N' GAMMA,'N'
```

SPECIFIES TO THE SYSTEM TO OPEN FOR OUTPUT PROCESSING FILE CONTROL TABLES ALPHA, BETA AND GAMMA. IF ALL THE CONTROL TABLES ARE FOR TAPE FILES, THE TAPES ASSIGNED TO ALPHA WILL BE REWOUND WHILE THE TAPES ASSIGNED TO BETA AND GAMMA WILL NOT. IF THEY ARE CONTROL TABLES FOR MASS STORAGE FILES, THE REWIND INDICATION WILL BE IGNORED.

IF A FORMAT DEFINITION FOR LABELS HAS BEEN SPECIFIED, A LABEL BLOCK WILL BE GENERATED UTILIZING THE LABEL DEFINITIONS GIVEN AND ANY USER LABEL INFORMATION SPECIFIED IN THE FILE CONTROL TABLE. IF A FORMAT DEFINITION FOR LABELS HAS NOT BEEN SPECIFIED, A LABEL BLOCK IS NOT GENERATED.

13.3.1.2. WRITE

THE WRITE REQUEST IS A DIRECTIVE TO THE SYSTEM TO COMPLETE THE STRUCTURE OF AN ITEM ACCORDING TO THE FORMAT DEFINITION GIVEN FOR ITEMS AND RECORD IT AS THE NEXT CONSECUTIVE ITEM IN THE OUTPUT BUFFER. IF A FORMAT DEFINITION FOR ITEMS HAS NOT BEEN SPECIFIED, THE ITEM WILL BE RECORDED UNALTERED. THE WRITE REQUEST IS CODED IN ONE OF THE FOLLOWING FORMS.

```
   (1) TAG WRITE FCTABLE
   (2) TAG WRITE FCTABLE NBRWDS
   (3) TAG WRITE FCTABLE 0,ADDR
   (4) TAG WRITE FCTABLE NBRWDS,ADDR
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. WRITE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. NBRWDS IS THE NUMBER OF WORDS IN THE ITEM TO BE WRITTEN AND ADDR IS THE SYMBOLIC LOCATION OF THE ITEM.

FORM(1) IS INTERPRETED AS A REQUEST TO WRITE AN ITEM OF THE SIZE PREVIOUSLY GIVEN, AS THE NEXT CONSECUTIVE ITEM. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE ITEM IN THE GIVEN AREA.

FORM(2) IS INTERPRETED AS A REQUEST TO WRITE AN ITEM OF SIZE LESS THAN THE AREA PREVIOUSLY GIVEN, AS THE NEXT CONSECUTIVE ITEM. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE ITEM IN THE GIVEN AREA. IF ITEM SIZE WAS SPECIFIED AS VARIABLE, NBRWDS IS THE SIZE WRITTEN. HOWEVER IF ITEM SIZE WAS SPECIFIED AS FIXED, THE BALANCE OF THE ITEM AREA IS ZERO FILLED.

FORM(3) IS INTERPRETED AS A REQUEST TO MOVE AN ITEM, OF THE SIZE PREVIOUSLY GIVEN, FROM THE AREA SPECIFIED BY THE ADDR SUBFIELD INTO THE BUFFER AS THE NEXT CONSECUTIVE ITEM.

FORM(4) IS INTERPRETED AS A REQUEST TO MOVE AN ITEM, OF SIZE LESS THAN OR GREATER THAN THAT PREVIOUSLY GIVEN, FROM THE AREA SPECIFIED BY THE ADDR SUBFIELD INTO THE BUFFER AS THE NEXT CONSECUTIVE ITEM. IF NBRWDS IS LESS THAN THE AREA PREVIOUSLY GIVEN, THIS FORM OPERATES AS FORM(2) WITH A MOVE. IF NBRWDS IS GREATER THAN THE AREA PREVIOUSLY GIVEN AND BLOCK SIZE IS SPECIFIED AS FIXED, THE ITEM IS SPANNED(I.E., IT LIES PARTLY IN THIS BLOCK AND CONTINUES IN THE NEXT). IF NBRWDS IS GREATER THAN THE AREA PREVIOUSLY GIVEN AND BLOCK SIZE IS SPECIFIED AS VARIABLE, THE CURRENT BLOCK IS WRITTEN AND THE WHOLE ITEM PLACED IN THE NEXT BUFFER.

WHEN ITEM SIZE IS FIXED, THAT SIZE IS THE MAXIMUM SIZE ITEM THAT CAN BE WRITTEN. WHEN ITEM SIZE IS VARIABLE AND THE USER WISHES TO WRITE, USING FORMS (1), (2), OR (3), AN ITEM LARGER THAN THE AREA PREVIOUSLY GIVEN, THE DRAIN REQUEST MUST BE ISSUED TO WRITE THE CURRENT BUFFER AND ACQUIRE A NEW BUFFER BEFORE ISSUING THE WRITE. SEE

DRAIN REQUEST FOR USEAGE.

13.3.1.3. WRITE RANDOM

THE WRITE RANDOM REQUEST IS A DIRECTIVE TO THE SYSTEM TO COMPLETE THE STRUCTURE OF AN ITEM ACCORDING TO THE FORMAT DEFINITION GIVEN FOR ITEMS AND RECORD THE ITEM IN A SPECIFIED POSITION WITHIN THE FILE. IF A FORMAT DEFINITION FOR ITEMS HAS NOT BEEN SPECIFIED, THE ITEM IS RECORDED UNALTERED.

THE FILE IN QUESTION MUST RESIDE ON MASS STORAGE AND HAVE FIXED SIZE ITEMS AND BLOCKS. THE WRITE RANDOM REQUEST IS CODED IN ONE OF THE FOLLOWING FORMS:

```
(1) TAG  WRITRM  FCTABLE 0,ADDR  ITEM-NBR
(2) TAG  WRITRM  FCTABLE NBRWDS,ADDR  ITEM-NBR
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. WRITRM IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. ADDR IS THE SYMBOLIC LOCATION OF THE ITEM. ITEM-NBR IS THE POSITION IN THE FILE, RELATIVE TO THE BEGINNING OF THE FILE, BEGINNING WITH ITEM NUMBER 0, INTO WHICH THE ITEM WILL BE PLACED.

FORM(1) IS INTERPRETED AS A REQUEST TO MOVE AN ITEM OF THE SIZE PREVIOUSLY GIVEN, FROM THE AREA SPECIFIED BY THE 0,ADDR FIELD, INTO THE BUFFER TO BE RECORDED AS A RANDOM ITEM.

FORM(2) IS EQUIVALENT TO FORM(1) EXCEPT THAT THE NBRWDS SUBFIELD MAY SPECIFY A SIZE LESS THAN THAT PREVIOUSLY GIVEN. THE UNUSED PORTION OF THE FIXED ITEM IS EITHER ZERO FILLED OR LEFT UNDISTURBED. IT IS ZERO FILLED IF THE FILE IS CURRENTLY BEING PROCESSED AS AN OUTPUT FILE. IT IS LEFT UNDISTURBED IF THE FILE IS BEING PROCESSED AS AN IN/OUT FILE.

WHEN THE WRITE RANDOM REQUEST IS EXECUTED, THE SYSTEM USES THE ITEM-NBR FIELD TO LOCATE AND RETRIEVE, FROM THE OUTPUT MEDIUM, THE BLOCK THAT IS TO CONTAIN THE ITEM. THE ITEM IS THEN MOVED FROM THE LOCATION SPECIFIED BY THE ADDR SUBFIELD TO ITS POSITION IN THE BUFFER. THE BUFFER IS THEN REWRITTEN TO ITS ORIGINAL LOCATION.

13.3.1.4. DRAIN

THE DRAIN REQUEST CAUSES THE IMMEDIATE WRITING OF THE CURRENT BUFFER TO THE ASSIGNED FILE. THE DRAIN REQUEST IS CODED

```
TAG  DRAIN FCTABLE(1) ....FCTABLE(N)
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. DRAIN IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION.

WHEN THE DRAIN REQUEST IS EXECUTED AND THE BUFFER IS EMPTY, THE REQUEST IS IGNORED. HOWEVER IF THE BUFFER IS NOT EMPTY, ONE OF THE FOLLOWING PROCEDURES IS UTILIZED:

1. IF BLOCK SIZE WAS SPECIFIED AS VARIABLE, THE BUFFER IS TRUNCATED TO THE LAST VALID ITEM.
2. IF BLOCK SIZE WAS SPECIFIED AS FIXED, THE BUFFER IS NOT TRUNCATED. AN END OF DATA CODE(0777777777777) IS RECORDED FOLLOWING THE LAST VALID ITEM.

REGARDLESS OF WHICH OF THE ABOVE PROCEDURES WAS USED, THE FORMAT DEFINITION FOR BLOCKS, IF ANY, IS USED TO COMPLETE THE BLOCK STRUCTURE BEFORE THE BLOCK IS WRITTEN.

13.3.1.5. CLOSE REEL

THE CLOSE REEL REQUEST IS USED TO TERMINATE THE WRITING OF AN OUTPUT REEL AND AUTOMATICALLY INITIATE THE WRITING OF SUBSEQUENT REELS OF THE SAME FILE. THE CLOSE

REEL REQUEST IS CODED:

```
   TAG  CLOSE 'REEL' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. CLOSE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'REEL' IS THE TYPE OF CLOSE PROCEDURE TO BE INITIATED AND MUST BE USED AS SHOWN. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE 'OPTION' SUBFIELD IS THE REWIND OPTION. IT MAY BE CODED AS 'L' FOR REWIND WITH INTERLOCK OR 'N' FOR NO REWIND. THE REWIND WITHOUT INTERLOCK OPTION IS PERFORMED IF NONE OF THE OTHERS ARE SPECIFIED.

THE CLOSE REEL REQUEST WILL CONSTRUCT ANY UNRECORDED ITEMS IN THE BUFFER ACCORDING TO THE FORMAT DEFINITION GIVEN FOR BLOCKS AND RECORD THEM ON TAPE. IF A FORMAT DEFINITION HAS BEEN GIVEN FOR END-OF-REEL SENTINELS, THE BLOCK IS GENERATED AND WRITTEN. AN END OF FILE MARK IS THEN WRITTEN.

A FINAL END OF FILE MARK IS RECORDED BEFORE THE REWIND OPTION IS EXECUTED AND THE SWITCHING PROCEDURES INITIATED. SEE FILE PROCEDURE LAYOUT.

ON RETURN TO THE USER FROM A CLOSE REEL REQUEST REGISTER A0 WILL CONTAIN THE LOCATION OF THE NEXT ITEM AREA IN THE BUFFER.

A CLOSE REEL REQUEST IS IGNORED IF THE FILE IS ON FASTRAND.

THE TAPE IS AUTOMATICALLY REWOUND AND THE STANDARD REEL SWITCHING PROCEDURES INITIATED. SEE SECTION ON LABEL AND SENTINEL CONVENTIONS.

13.3.1.6. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE WRITING OF AN OUTPUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL OUTPUT FILES. THERE ARE NO RESTRICTIONS ON THE NUMBER OF CLOSE REQUESTS CODED TO REFERENCE AN INDIVIDUAL OUTPUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE CLOSE FILE REQUEST IS CODED.

```
   TAG  CLOSE 'FILE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. CLOSE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'FILE' IS THE TYPE OF CLOSE PROCEDURE TO BE INITIATED AND MUST BE USED AS SHOWN. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE 'OPTION' SUBFIELD IS THE REWIND OPTION. IT MAY BE CODED AS 'L' FOR REWIND WITH INTERLOCK OR 'N' FOR NO REWIND. A REWIND WITHOUT INTERLOCK IS PERFORMED IF NONE OF THE OTHERS ARE SPECIFIED.

WHEN THE CLOSE REQUEST IS ISSUED, THE BUFFER IS CONSTRUCTED ACCORDING TO THE FORMAT DEFINITION FOR BLOCKS AND WRITTEN ON THE OUTPUT DEVICE. IF A FORMAT DEFINITION HAS BEEN GIVEN FOR END-OF-FILE SENTINELS, THE BLOCK IS GENERATED AND WRITTEN. AN END-OF-FILE MARK IS THEN WRITTEN.

A FINAL END OF FILE MARK IS RECORDED BEFORE THE REWIND OPTION IS INITIATED.

IF THE FILE IS ON FASTRAND, THE NEXT ADDRESS TO BE WRITTEN IS SAVED.

A SINGLE CLOSE REQUEST MAY BE USED TO CLOSE ALL THE FILES IN A PROGRAM REGARDLESS OF THEIR PROCESSING MODES.

13.3.1.7. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN OUTPUT FILE AND ITS ASSIGNED DEVICES. THE DISPOSITION OF THE FILE DEPENDS UPON THE CATALOGING OPTIONS SPECIFIED ON THE ASSIGN CONTROL CARD AND ON THE OPTION SPECIFIED ON THE RELEASE REQUEST. THE RELEASE REQUEST IS CODED

```
   TAG  RELESE FCTABLE(1),OPTION ... FCTABLE(N),OPTION
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. RELESE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. THE FCTABLE FIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILES IN QUESTION. THE OPTION, IF SPECIFIED, MAY BE CODED AS 'S' TO HOLD THE PHYSICAL ASSIGNMENT FOR THIS FILE.

THE RELEASE REQUEST USES THE CLOSE OUTPUT FILE ROUTINES TO CLOSE THE FILE (WITH INTERLOCKED REWIND FOR TAPE DEVICES) BEFORE DISPOSITION, IF NOT ALREADY CLOSED. THEREFORE A FILE TO BE RELEASED DOES NOT HAVE TO BE TERMINATED WITH THE CLOSE REQUEST. IF THE FILE WAS NOT CATALOGUED (OPTION ON ASSIGN CARD) IT CANNOT BE OPENED AGAIN IN THE PROGRAM.

13.3.2. INPUT

THE INPUT SUBROUTINES PROVIDE FOR THE READING OF AN EXISTING FILE FROM AN EXTERNAL MEDIA. THESE SUBROUTINES ARE CONCERNED ONLY WITH THE MANIPULATION OF THE DATA REQUESTED BY THE OBJECT PROGRAM. THE FORMAT OF THE DATA IS VALIDATED BY INTERROGATING THE ASSOCIATED FORMAT DEFINITION BEFORE THE DATA IS PRESENTED TO THE OBJECT PROGRAM. ABSENCE OF A DEFINITION FOR A PARTICULAR RECORD TYPE IMPLIES THAT NO CHECKING WILL BE MADE PRIOR TO PRESENTING THE DATA TO THE OBJECT PROGRAM.

THE FUNCTIONS PROVIDED BY THESE ROUTINES INCLUDE OPEN FILE, READ (SEQUENTIAL), READ (RANDOM), DRAIN, CLOSE REEL, CLOSE FILE AND RELEASE. THESE FUNCTIONS ARE PROCEDURAL CALLS, CODED IN THE OBJECT PROGRAM, TO REQUEST THE DESIRED FUNCTION.

UPON RETURN TO THE USER FROM A READ OR READ RANDOM REQUEST, REGISTER A0 WILL CONTAIN INFORMATION IN THE FORMAT OF AN ACCESS WORD.

```
35          17        0
--------------------
:NBRWDS    : ADDRESS:
--------------------
```

THIS WORD DEFINES THE ACTUAL SIZE AND BUFFER LOCATION OF THE CURRENT ITEM REGARDLESS OF WHICH READ FORM IS USED. NOTE THAT FOR REQUESTS THAT SPECIFY A MOVE, THE USER WILL HAVE TWO IMAGES OF THE CURRENT ITEM. ONE IN THE LOCATION SPECIFIED BY THE ADDR SUBFIELD; THE OTHER IN THE BUFFER AS SPECIFIED BY THE CONTENTS OF A0. THE BUFFER IMAGE IS ALWAYS AVAILABLE UNTIL ANOTHER READ REQUEST IS ISSUED.

HOWEVER, IF THE READ REQUESTED IS A SPANNED ITEM, REGISTER A0 WILL CONTAIN THE FOLLOWING INFORMATION.

```
--------------------
35          17        0
:NBRWDS    :    0    :
--------------------
```

THE BUFFER IMAGE WILL NOT BE AVAILABLE TO THE USER.

WHEN AN END-OF-FILE BLOCK IS READ AND VALIDATED, THE END-OF-FILE EXIT WILL BE TAKEN. WHEN AN END-OF-REEL BLOCK IS READ AND VALIDATED, THE ENDOF-REEL EXIT WILL BE TAKEN. IF THE END-OF-REEL EXIT IS NOT CODED, AN AUTOMATIC CLOSE REEL IS PERFORMED, AND THE TAPE REWOUND WITH INTERLOCK. REGISTER A0 WILL CONTAIN THE LENGTH AND LOCATION OF THE FORMAT BLOCK.

IF FORMAT DEFINITIONS FOR END-OF-REEL OR END-OF-FILE SENTINELS ARE NOT SPECIFIED AND END-OF-FILE MARK IS DETECTED, THE END-OF-FILE EXIT IS TAKEN.

REGISTER A1, H2 WILL CONTAIN AN ADDRESS TO WHICH THE USER MUST RETURN CONTROL AFTER HIS CLOSE PROCEDURES, TO REENTER THE MAIN SEQUENCE OF EXECUTION. A1, H1 WILL CONTAIN THE LOCATION OF THE FILE CONTROL TABLE. THIS MUST BE RESET BEFORE RETURNING TO THE ITEM HANDLER.

13.3.2.1. OPEN INPUT

THE OPEN INPUT REQUEST INITIALIZES A FILE CONTROL TABLE FOR SUBSEQUENT INPUT OPERATIONS ON THE DEVICE ASSIGNED TO THE TABLE. THERE ARE NO RESTRICTIONS ON THE NUMBER OF OPEN REQUESTS WHICH MAY BE CODED TO REFERENCE AN INPUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE OPEN REQUEST IS CODED

```
   TAG   OPEN 'MODE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. OPEN IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'MODE' SPECIFIED THE INTENDED MODE OF OPERATION FOR THIS FILE AND MAY BE CODED AS 'INPUT' FOR FORWARD OPERATIONS AND 'REVRSE' FOR BACKWARD OPERATIONS. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE TO BE INITIATED. THE OPTION SUBFIELD, IF USED, SPECIFIES A NO REWIND OPTION AND MUST BE CODED AS 'N'. IF THE FILE IS TO BE REWOUND BEFORE OPENING, THE OPTION SUBFIELD MUST BE OMITTED. FOR EXAMPLE THE OPEN REQUESTS:

```
   ABC   OPEN 'INPUT' ALPHA,'N' BETA
   DEF   OPEN 'REVRSE' GAMMA
```

SPECIFIES TO THE SYSTEM TO OPEN FILE CONTROL TABLES ALPHA AND BETA FOR FORWARD PROCESSING AND, FILE CONTROL TABLE GAMMA FOR BACKWARD PROCESSING. IF ALPHA AND BETA ARE CONTROL TABLES FOR TAPE FILES, THE TAPES ASSIGNED TO ALPHA WILL NOT BE REWOUND BEFORE READING WHILE THE TAPES ASSIGNED TO BETA WILL. IF THEY ARE CONTROL TABLES FOR MASS STORAGE FILES THE REWIND INDICATION WILL BE IGNORED. REWINDING IS NEVER PERFORMED ON FILES OPENED FOR BACKWARD (REVRSE) PROCESSING.

IF A FORMAT DEFINITION FOR LABELS HAS BEEN SPECIFIED, THE LABEL BLOCK IS READ AND THOSE PORTIONS OF IT DEFINED BY DEFINITIONS ARE VALIDATED BY THE SYSTEM. ANY ERROR CONDITION DETECTED CAUSES THE ABNORMAL EXIT TO BE TAKEN. USER LABEL INFORMATION, IF ANY, IS PRESENTED TO THE OBJECT PROGRAM FOR CHECKING. REGISTER A0 WILL CONTAIN THE LOCATION OF THE USER LABEL AREA IN THE FORM OF AN ACCESS WORD.

```
35         17       0
--------------------
:NBRWDS    : ADDRESS:
--------------------
```

IF THERE IS NO USER LABEL INFORMATION, A0 CONTAINS ZERO.

WHEN FORMAT DEFINITIONS FOR LABELS HAVE NOT BEEN SPECIFIED FOR A FILE, THE SYSTEM ASSUMES THE FILE TO BE UNLABELED AND PROCESSES THE FIRST BLOCK AS DATA.

FILES OPENED IN THE BACKWARD DIRECTION ARE PROCESSED AS UNLABELED FILES.

13.3.2.2. READ

THE READ REQUEST IS A DIRECTIVE TO THE SYSTEM TO OBTAIN THE NEXT CONSECUTIVE ITEM IN THE FILE. THE FORMAT DEFINITION FOR ITEMS IS INTERROGATED IN ORDER TO VALIDATE THE ITEM BEFORE IT IS PRESENTED TO THE USER. IF A FORMAT DEFINITION HAS NOT BEEN SPECIFIED, THE ITEM IS PRESENTED UNALTERED. THE READ REQUEST IS CODED IN ONE OF THE FOLLOWING FORMS.

```
   (1) TAG   READ FCTABLE EOF-EXIT
   (2) TAG   READ FCTABLE 0,ADDR EOF-EXIT
   (3) TAG   READ FCTABLE NBRWDS,ADDR EOF-EXIT
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. READ IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. NBRWDS IS THE NUMBER OF WORDS IN THE ITEM TO BE MOVED AND ADDR IS THE SYMBOLIC LOCATION TO WHICH THE ITEM WILL BE MOVED. EOF-EXIT IS THE SYMBOLIC LOCATION OF A USER ROUTINE TO WHICH THE SYSTEM WILL RETURN CONTROL WHEN AN END-OF-FILE CONDITION IS MET.

THE FOLLOWING DESCRIPTIONS OF THE THREE READ FORMS APPLY TO BOTH FIXED AND VARIABLE LENGTH ITEMS.

FORM(1) IS INTERPRETED AS A REQUEST FOR THE NEXT CONECUTIVE ITEM IN THE FILE. IF THE NEXT ITEM IS NOT IN THIS BUFFER, ANOTHER BLOCK IS READ FROM THE INPUT MEDIUM AND THE FIRST ITEM OBTAINED. THE USER MAY THEN PROCESS THE ITEM WITHOUT ITS REMOVAL FROM THE BUFFER. THIS REQUEST MUST NOT BE USED FOR SPANNED ITEMS.

FORM(2) MAY BE USED WHEN READING SPANNED ITEMS (SPANNED ITEMS OVERLAP BLOCKS) AS WELL AS ITEMS OF FIXED AND VARIABLE LENGTH. IT IS INTERPRETED AS A REQUEST TO MOVE THE NEXT CONSECUTIVE ITEM, IN ITS ENTIRETY, TO THE USER AREA SPECIFIED BY THE 0,ADDR FIELD. IF THE ITEM IS SPANNED, SEVERAL REQUESTS TO THE INPUT MEDIUM FROM THE ITEM HANDLER MAY BE NECESSARY TO OBTAIN THE COMPLETE ITEM.

FORM(3) IS FUNCTIONALLY SIMILAR TO FORM(2). IT DIFFERS IN THAT IT IS INTERPRETED AS A REQUEST TO MOVE ONLY THAT NUMBER OF WORDS OF THE ITEM SPECIFIED BY THE NBRWDS SUBFIELD TO THE USER AREA SPECIFIED BY THE ADDR SUBFIELD.

13.3.2.3. READ RANDOM

THE READ RANDOM REQUEST IS A DIRECTIVE TO THE SYSTEM TO OBTAIN A SPECIFIC ITEM. THE ITEM IN QUESTION IS SPECIFIED BY AN ITEM NUMBER WHICH IS THE POSITION OF THE ITEM IN THE FILE RELATIVE TO THE BEGINNING OF THE FILE. ONCE THE ITEM IS OBTAINED, THE FORMAT DEFINITION FOR ITEMS IS INTERROGATED TO VALIDATE THE ITEM BEFORE IT IS PRESENTED TO THE USER. IF A FORMAT DEFINITION HAS NOT BEEN SPECIFIED, THE ITEM IS PRESENTED UNALTERED.

THE FILE IN QUESTION MUST RESIDE ON MASS STORAGE AND HAVE FIXED SIZE ITEMS AND BLOCKS. THE READ RANDOM REQUEST IS CODED IN ONE OF THE FOLLOWING FORMS.

```
(1) TAG   READRM  FCTABLE 0,ADDR  ITEM-NBR
(2) TAG   READRM  FCTABLE NBRWDS,ADDR  ITEM-NBR
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. READRM IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. NBRWDS IS THE NUMBER OF WORDS IN THE ITEM TO BE MOVED AND ADDR IS THE SYMBOLIC LOCATION TO WHICH THE ITEM WILL BE MOVED. ITEM-NBR SPECIES THE RELATIVE POSITION IN THE FILE OF THE ITEM TO BE READ.

WHEN THE READ RANDOM REQUEST IS EXECUTED, THE SYSTEM OBTAINS FROM THE INPUT MEDIUM THE BLOCK THAT CONTAINS THE ITEM SPECIFIED IN THE ITEM-NBR FIELD. IF THE REQUEST FORM IS (1), THE ENTIRE ITEM IS MOVED TO THE USER'S AREA SPECIFIED BY THE 0,ADDR FIELD. IF THE REQUEST FORM IS (2), ONLY THAT NUMBER OF WORDS OF THE ITEM SPECIFIED BY THE NBRWDS SUBFIELD IS MOVED TO THE AREA SPECIFIED BY THE ADDR SUBFIELD.

SINCE ITEMS IN FILES BEING READ RANDOMLY CAN BE REREAD AGAIN AND AGAIN IN ANY ORDER, THERE IS NO LOGICAL END-OF-FILE AND HENCE NO END-OF-FILE DETECTION NECESSARY.

13.3.2.4. DRAIN

THE DRAIN REQUEST CAUSES THE SYSTEM TO IMMEDIATELY RELINQUISH THE BUFFER CURRENTLY IN USE AND ACQUIRE ANOTHER BLOCK OF DATA FROM THE ASSIGNED FILE. THE DRAIN REQUEST IS CODED:

```
TAG  DRAIN FCTABLE(1) ... FCTABLE(N)
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. DRAIN IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. FCTABLE IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION.

WHEN THE DRAIN REQUEST IS EXECUTED, ANY UNUSED ITEMS IN THE CURRENT BUFFER ARE

IGNORED. IF THE BUFFER HAS A SPANNED ITEM, SEVERAL BUFFERS MAY HAVE TO BE BYPASSED TO GET TO THE NEXT VALID ITEM. THE NEXT READ REQUEST FOR THIS FILE WILL BE DIRECTED TO THE FIRST VALID ITEM IN THE NEWLY ACQUIRED BUFFER.

13.3.2.5. CLOSE REEL

THE CLOSE REEL REQUEST IS USED TO TERMINATE THE READING OF AN INPUT REEL AND AUTOMATICALLY INITIATE THE READING OF SUBSEQUENT REELS OF THE SAME FILE. THE CLOSE REEL REQUEST IS CODED

```
    TAG   CLOSE 'REEL' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. CLOSE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'REEL' IS THE TYPE OF CLOSE PROCEDURE TO BE INITIATED AND MUST BE USED AS SHOWN. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE OPTION SUBFIELD IS THE REWIND OPTION. IT MAY BE CODED AS 'L' FOR REWIND WITH INTERLOCK OR 'N' FOR NO REWIND. THE REWIND WITHOUT INTERLOCK OPTION IS PERFORMED IF NONE OF THE ABOVE OPTIONS ARE SPECIFIED.

THE CLOSE REEL REQUEST RELEASES ALL BUFFERS BACK TO THE POOL, INITIATES THE REWIND OPTION AND REEL SWAPPING PROCEDURES.

ON RETURN TO THE USER FROM A CLOSE REEL REQUEST REGISTER A0 WILL CONTAIN THE USER LABEL ACCESS WORD IF THERE WAS A USER LABEL OR ZERO IF THERE WAS NO LABEL(SEE OPEN INPUT).

13.3.2.6. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE READING OF AN INPUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL INPUT FILES. THERE ARE NO RESTRICTIONS ON THE NUMBER OF CLOSE REQUESTS CODED TO REFERENCE AN INDIVIDUAL INPUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE CLOSE FILE REQUEST IS CODED

```
    TAG   CLOSE 'FILE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. CLOSE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'FILE' IS THE TYPE OF CLOSE PROCEDURE TO BE INITIATED AND MUST BE USED AS SHOWN. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE 'OPTION' SUBFIELD IS THE REWIND OPTION. IT MAY BE CODED AS 'L' FOR REWIND WITH INTERLOCK OR 'N' FOR NO REWIND. A REWIND WITHOUT INTERLOCK IS PERFORMED IF NONE OF THE ABOVE OPTIONS ARE SPECIFIED.

A SINGLE CLOSE REQUEST MAY BE USED TO CLOSE ALL THE FILES IN A PROGRAM REGARDLESS OF THEIR PROCESSING MODES.

13.3.2.7. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN INPUT FILE AND ITS ASSIGNED DEVICES. THE DISPOSITION OF THE FILE DEPENDS UPON THE CATALOGING OPTIONS SPECIFIED ON THE ASSIGN CONTROL CARD AND ON THE OPTION SPECIFIED ON THE RELEASE REQUEST. THE RELEASE REQUEST IS CODED

```
    TAG   RELESE FCTABLE(1),OPTION ... FCTABLE(N),OPTION
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. RELESE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. THE FCTABLE FIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE OPTION, IF SPECIFIED, MAY BE CODED AS 'S' TO HOLD THE PHYSICAL ASSIGNMENT FOR THIS FILE.

THE RELEASE REQUEST USES THE CLOSE INPUT FILE ROUTINES TO CLOSE THE FILE (WITH INTERLOCKED REWIND FOR TAPE DEVICES) BEFORE DISPOSITION, IF NOT ALRE,ADY CLOSED.

THEREFORE A FILE TO BE RELEASED DOES NOT HAVE TO BE TERMINATED WITH A CLOSE REQUEST. IF THE FILE WAS NOT CATALOGUED (OPTION ON ASSIGN CARD), IT CANNOT BE OPENED AGAIN IN THE PROGRAM.

13.3.3. IN/OUT

THE IN/OUT MODE OF PROCESSING PROVIDES THE CAPABILITY TO MODIFY AN EXISTING MASS STORAGE FILE WITHOUT THE NECESSITY TO RECREATE THE FILE. THIS IS ACCOMPLISHED BY ALLOWING BOTH READ AND WRITE REQUESTS TO REFERENCE THE SAME FILE. THE USER CAN THEREFORE READ AN ITEM, ALTER THE ITEM AND REPLACE IT IN THE FILE.

FILES CAN BE MODIFIED BY UPDATING OR EXTENDING. WHEN A FILE IS BEING UPDATED, ITEMS MAY BE SIMPLY MODIFIED OR COMPLETELY REWRITTEN. IF THEY ARE REWRITTEN THE LENGTH AND FORMAT OF THE ITEM MUST BE IDENTICAL TO THE ORIGINAL ITEM. IF A FILE IS BEING EXTENDED, THE FORMAT OF THE EXTENSION MUST BE IDENTICAL TO THE PREVIOUS PORTION OF THE FILE WITH THE ITEM AND BLOCK STRUCTURE MAINTAINED.

13.3.3.1. OPEN IN/OUT

THE OPEN IN/OUT REQUEST INITIALIZES A FILE CONTROL TABLE FOR SUBSEQUENT INPUT AND OUTPUT OPERATIONS ON THE MASS STORAGE FILE ASSIGNED TO THE TABLE. THERE ARE NO RESTRICTIONS ON THE NUMBER OF OPEN REQUESTS WHICH MAY BE CODED TO REFERENCE AN IN/OUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE OPEN REQUEST IS CODED

```
TAG   OPEN 'MODE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. OPEN IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'MODE' SPECIFIES THE INTENDED MODE OF OPERATION FOR THIS FILE AND MUST BE CODED AS 'IN/OUT' FOR INPUT/OUTPUT PROCESSING. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION. THE 'OPTION' SUBFIELD, IF USED, SPECIFIES THAT THE USER WILL EXTEND THE FILE. IN THIS CASE IT MUST BE CODED AS 'E' TO INDICATE EXTENSION. ABSENCE OF THE OPTION SUBFIELD IMPLIES THAT THE FILE WILL BE UPDATED.

IF EXTENSION WAS SPECIFIED, THE FILE IS INITIALIZED AT THE END ALLOWING ITEMS TO BE ADDED TO THE FILE. IF UPDATING IS IMPLIED, THE FILE IS INITIALIZED AT THE BEGINNING TO ALLOW EACH ITEM IN THE FILE THE OPPORTUNITY TO BE UPDATED.

THE OPEN IN/OUT PROCEDURE IS A COMBINATION OF THE OPEN INPUT AND OPEN OUTPUT PROCEDURES. THAT IS, REGISTER A1 WILL CONTAIN THE ACCESS WORD FOR THE USER'S LABEL, IF ANY, AND REGISTER A0 WILL CONTAIN THE ACCESS WORD FOR THE MAXIMUM SIZE ITEM THAT COULD BE WRITTEN.

THE LABEL IS AUTOMATICALLY REWRITTEN WHEN THE FIRST READ OR WRITE REQUEST IS ISSUED. THEREFORE ANY LABEL UPDATE MUST BE MADE IN THE LABEL AREA BEFORE THE FIRST REQUEST IS ISSUED.

13.3.3.2. READ-WRITE

THE INPUT READ FORMS (2) AND (3), AND THE OUTPUT WRITE FORMS (3) AND (4) ARE THE ONLY LEGITIMATE READ/WRITE REQUESTS FOR SEQUENTIAL PROCESSING AN IN/OUT FILE.

ON RETURN TO THE USER FROM A READ OR WRITE REQUEST REGISTER A0 ALWAYS CONTAINS AN ACCESS WORD FOR THE MAXIMUM SIZE ITEM THAT CAN BE WRITTEN. IF THE FILE IS BEING UPDATED THIS SIZE REPRESENTS THE ACTUAL SIZE OF THE ITEM JUST READ OR, IT REPRESENTS THE SIZE OF THE NEXT ITEM AREA INTO WHICH AN ITEM CAN BE WRITTEN.

IF A WRITE REQUEST SPECIFIES AN ITEM OF SIZE LESS THAN THE ITEM OR ITEM AREA PREVIOUSLY GIVEN, THE UNUSED PORTION OF THE ITEM OR ITEM AREA IS MAINTAINED UNALTERED.

IF THE FILE IS BEING EXTENDED MAXIMUM ITEM SIZE IS THE MAXIMUM SIZE SPECIFIED BY THE USER.

THE FOLLOWING LOGIC IS IMPLIED WHEN SEQUENTIAL READ/WRITE REQUESTS ARE BEING EXECUTED.

- A READ REQUEST MAKES AN ITEM AVAILABLE FOR PROCESSING.
- A WRITE REQUEST FOLLOWING A READ REQUEST REWRITES THE ITEM.
- CONSECUTIVE READ REQUESTS WITHOUT INTERVENING WRITE REQUESTS MAKES AVAILABLE CONSECUTIVE ITEMS WHICH ARE NOT REWRITTEN.
- CONSECUTIVE WRITE REQUESTS WITHOUT INTERVENING READ REQUESTS WILL CAUSE THE WRITING OF SUCCESSIVE ITEMS.
- A READ REQUEST FOLLOWING SEVERAL CONSECUTIVE WRITE REQUESTS WILL OBTAIN THAT ITEM IN THE FILE WHICH SEQUENTIALLY FOLLOWS THE LAST ONE WRITTEN.

13.3.3.3. READ-WRITE RANDOM

THE READ AND WRITE RANDOM REQUESTS FOR IN/OUT FILES ARE IDENTICAL IN FORM AND FUNCTION TO THOSE SPECIFIED FOR RANDOM INPUT AND OUTPUT FILES.

13.3.3.4. EXCLUSIVE READ RANDOM

THE EXCLUSIVE READ RANDOM REQUEST ALLOWS A USER TO SPECIFY TO THE SYSTEM THAT HE WISHES EXCLUSIVE USE OF A SPECIFIC FIXED LENGTH ITEM. ALL OTHER REFERENCES TO THIS PARTICULAR ITEM ARE INTERLOCKED BY THE SYSTEM UNTIL IT HAS BEEN RELEASED. THE CODING OF THE EXCLUSIVE READ RANDOM REQUEST, AS WELL AS ITS FUNCTION, IS IDENTICAL TO THAT OF THE NORMAL READ RANDOM REQUEST EXCEPT THAT THE REQUEST NAME (READ) IS PREFIXED BY EX; E.G., EXREAD. ALL FORMS OF THE READ RANDOM REQUEST MAY BE USED EXCLUSIVELY.

WHENEVER AN ITEM IS OBTAINED FOR EXCLUSIVE USE, IT IS AUTOMATICALLY RELEASED BY THE NEXT REFERENCE TO THE FILE. IF THE ITEM IS HELD TOO LONG, CAUSING THE EXEC TO TIME-OUT, THE FILE ERROR EXIT WILL BE TAKEN WITH A1 EQUAL TO:

```
35                 17              0
------------------------------------
:        010       : EX READ ADDRESS  :
------------------------------------
```

IF THE USER CAN RECOVER, HE MUST RETURN TO THE ITEM HANDLER WITH THE ABOVE CALL.

13.3.3.5. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE READING AND WRITING OF AN IN/OUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL IN/OUT FILES. THERE ARE NO RESTRICTIONS ON THE NUMBER OF CLOSE REQUESTS CODED TO REFERENCE AN INDIVIDUAL IN/OUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE CLOSE REQUEST IS CODED

```
TAG   CLOSE 'FILE' FCTABLE(1) ... FCTABLE(N)
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. CLOSE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. 'FILE' IS THE TYPE OF CLOSE PROCEDURE TO BE INITIATED AND MUST BE USED AS SHOWN. THE FCTABLE FIELD IS THE LOCATION OF THE FILE CONTROL TABLES THAT MAINTAIN THE FILES IN QUESTION.

IF THE CLOSE FILE REQUEST IS EXECUTED WITH A FORMAT DEFINITION FOR END OF FILE SENTINELS SPECIFIED, THE SENTINEL IS WRITTEN ACCORDING TO THE FOLLOWING RULES.

1. IF THE ORIGINAL SENTINEL WAS NOT OVERWRITTEN AND RECORDING WAS NOT DONE BEYOND THE SENTINEL, IT WILL NOT BE UPDATED.
2. IF THE ORIGINAL SENTINEL WAS NOT OVERWRITTEN AND RECORDING

WAS DONE BEYOND THE SENTINEL, IT IS ERASED AND A NEW ONE RECORDED AFTER THE LAST BLOCK IN THE FILE.

3. IF THE ORIGINAL SENTINEL WAS OVER WRITTEN, A NEW SENTINEL IS RECORDED AFTER THE LAST BLOCK IN THE FILE.

SINCE IN/OUT FILES RESIDE ON MASS STORAGE DEVICES, NO REWIND PROCEDURES ARE INITIATED.

13.3.3.6. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN IN/OUT FILE AND ITS ASSIGNED MASS STORAGE DEVICE. THE DISPOSITION OF THE FILE DEPENDS ON THE CATALOGING OPTIONS SPECIFIED ON THE ASSIGN CONTROL CARD AND THE OPTION SPECIFIED ON THE RELEASE REQUEST. THE RELEASE REQUEST IS CODED

```
TAG  RELESE FCTABLE(1),OPTION ... FCTABLE(N),OPTION
```

WHERE TAG IS OPTIONAL AT THE USER'S DISCRETION. RELESE IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. THE FCTABLE SUBFIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAIN THE FILE IN QUESTION.. THE OPTION, IF SPECIFIED, MAY BE CODED AS 'S' TO HOLD THE PHYSICAL ASSIGNMENT FOR THE FILE.

THE RELEASE REQUEST USES THE CLOSE IN/OUT FILE ROUTINES TO CLOSE THE FILE NORMALLY BEFORE DISPOSITION, IF NOT ALREADY CLOSED. THEREFORE A FILE TO BE RELEASED DOES NOT HAVE TO BE TERMINATED WITH A CLOSE REQUEST. IF THE FILE WAS NOT CATALOGUED (OPTION ON ASSIGN CARD) IT CANNOT BE OPENED AGAIN IN THE PROGRAM.

13.3.4. ITEM LAYOUT

ITEMS PROCESSED BY ITEM CONTROL MAY BE SPECIFIED AS FIXED OR VARIABLE IN LENGTH. VARIABLE LENGTH ITEMS ON FASTRAND AND TAPE WILL HAVE A CONTROL WORD TO SPECIFY THE ITEM LENGTH. IF ITEM FORMAT IS SPECIFIED, THIS CONTROL WORD WILL BE IMBEDDED WITHIN THE FORMAT. FIXED LENGTH ITEMS WILL HAVE NO CONTROL INFORMATION.

13.3.4.1. SINGLE ITEM LAYOUT

FIGURE 1 ILLUSTRATES THE MINIMAL ITEM PROCESSED BY ITEM CONTROL ON TAPE OR FASTRAND FILES, WHILE FIGURE 2 ILLUSTRATES THE ITEM WITH FORMAT INFORMATION SPECIFIED BY THE USER'S FORMAT DEFINITION FOR ITEMS.

```
WITHOUT A FORMAT DEFINITION FOR ITEMS SPECIFIED

          ----------
          : A  : B  :
          ----------       ----------
          :         :      :         :
          :VARIABLE :      : FIXED   :
          : LENGTH  :      : LENGTH  :
          :  ITEM   :      :  ITEM   :
          :  AREA   :      :  AREA   :
          :         :      :         :
          :         :      :         :
          ----------       ----------
```

FIGURE 1

```
WITH A FORMAT DEFINITION FOR ITEMS SPECIFIED

          ----------       ----------
          : PREFIX  :      : PREFIX  :
          : INFOR-  :      : INFOR-  :
          : MATION  :      : MATION  :
          ----------       ----------
          :         :      :         :
          :         :      :         :
          :VARIABLE :      : FIXED   :
          : LENGTH  :      : LENGTH  :
          :  ITEM   :      :  ITEM   :
          :  AREA   :      :  AREA   :
          :         :      :         :
          ----------       ----------
          : SUFFIX  :      : SUFFIX  :
          : INFOR-  :      : INFOR-  :
          : MATION  :      : MATION  :
          ----------       ----------
```

FIGURE 2

```
A = LENGTH OF THE PREVIOUS ITEM (FOR BACKWARD PROCESSING)
B = LENGTH OF THE CURRENT ITEM
```

13.3.4.2. BLOCKED ITEM LAYOUT

ITEMS ARE BLOCKED OR DEBLOCKED AND FORMATS GENERATED OR CHECKED BY ITEM CONTROL. FIGURES 5 AND 6 ILLUSTRATES THE BLOCK AS IT IS PASSED TO OR FROM THE BLOCK BUFFERING PACKAGE. FIGURE 5 IS THE MINIMAL BLOCK AND FIGURE 6 IS A BLOCK WITH FORMAT INFORMATION SPECIFIED BY THE USER'S FORMAT DEFINITION FOR BLOCKS.

TAPE OR FASTRAND BLOCKED ITEMS (WITHOUT A FORMAT DEFINITION FOR BLOCKS).

```
-----------
:         :
:         :
:   -N-   :
:VARIABLE :
:    OR   :
:  FIXED  :
:  LENGTH :
:   ITEMS :
:         :
:         :
-----------
```

FIGURE 5

TAPE OR FASTRAND BLOCKED ITEMS (WITH A FORMAT DEFINITION FOR BLOCKS SPECIFIED).

```
-----------
: PREFIX  :
: INFOR-  :
: MATION  :
-----------
:         :
:         :
:   -N-   :
:VARIABLE :
:    OR   :
:  FIXED  :
:  LENGTH :
:   ITEMS :
:         :
-----------
: SUFFIX  :
: INFOR-  :
: MATION  :
-----------
```

FIGURE 6

13.4. LABEL AND SENTINEL CONVENTIONS

THE SYSTEM HAS A SET OF PROCEDURES THAT GOVERN THE MANNER IN WHICH IT HANDLES LABEL AND SENTINEL BLOCKS. THESE PROCEDURES ARE DIRECTLY RELATED TO THE FORMAT DEFINITIONS GIVEN FOR THE FILE AND THE TYPE OF DEVICE ASSIGNED.

13.4.1. OUTPUT

THE EXISTENCE OF FORMAT DEFINITIONS FOR LABELS AND/OR SENTINELS WILL RESULT IN THE SYSTEM GENERATING AND RECORDING THOSE LABELS AND/OR SENTINELS. IF A PARTICULAR DEFINITION WAS NOT GIVEN, THE CORRESPONDING BLOCK IS NOT PRODUCED.

MASS STORAGE

FOR MASS STORAGE FILES, LABELS AND END-OF-FILE SENTINELS, IF SPECIFIED, ARE RECORDED AS THE FIRST AND LAST BLOCKS OF THE FILE. IF EITHER IS NOT SPECIFIED THE CORRESPONDING BLOCK IS NOT PRODUCED.

MAGNETIC TAPES

FOR TAPE FILES, LABELS ARE WRITTEN AS THE FIRST BLOCK OF EACH FILE ON A REEL AND AT THE BEGINNING OF EACH REEL OF A MULTI-REEL FILE. IN THE MULTI-REEL FILE CASE, AN END-OF-REEL SENTINEL IS RECORDED AT THE END OF EACH REEL. IN EITHER CASE, AN END-OF-FILE SENTINEL IS RECORDED AS THE LAST BLOCK OF A FILE.

IF A LABEL AND AN END-OF-FILE DEFINITION WERE GIVEN IN A FORMAT THE RESULT OF AN EXECUTION COULD PRODUCE EITHER A SINGLE REEL FILE WITH A LABEL AND AN END-OF-FILE SENTINEL OR A MULTI-REEL FILE WITHOUT END-OF-REEL SENTINELS BUT WITH A LABEL AND AN END-OF-FILE SENTINEL. A SINGLE OR MULTI-REEL TAPE WILL CONTAIN DATA ONLY, IF THERE WERE NO DEFINITIONS FOR LABELS, END-OF-REELS AND END-OF-FILE SENTINELS.

ONE END-OF-FILE MARK WILL BE RECORDED BEFORE, IF SPECIFIED BY FORMAT, AND TWO AFTER, EACH END-OF-REEL OR END-OF-FILE SENTINEL. IN THE ABSENCE OF EITHER, TWO END-OF-FILE MARKS WILL FOLLOW THE LAST DATA BLOCK. FILES ON A MULTI-FILE-REEL ARE SEPARATED BY A SINGLE END-OF-FILE MARK.

REEL SWITCHING WILL BE ACCOMPLISHED IN THE EXECUTIVE ON REQUEST BY THE CLOSE REEL PROCEDURE.

13.4.2. INPUT

THE EXISTENCE OF FORMAT DEFINITIONS FOR LABELS AND/OR SENTINELS WILL IMPLY TO THE SYSTEM THAT THE CORRESPONDING BLOCKS EXIST IN THE FILE. WHEN THESE BLOCKS ARE ENCOUNTERED, THEY ARE VALIDATED ACCORDING TO THEIR RESPECTIVE DEFINITIONS BEFORE PROCESSING CONTINUES. THE ABSENCE OF A DEFINITION WILL IMPLY THAT THE CORRESPONDING BLOCK DOES NOT EXIST IN THE FILE.

IF THIS IMPLICATION, BETWEEN THE FORMAT DEFINITIONS FOR LABELS AND/OR SENTINELS AND THE ACTUAL FILE, IS NOT TRUE AN ERROR CONDITION COULD RESULT.

MAGNETIC TAPES

WHEN AN END OF FILE MARK IS DETECTED AND FORMAT DEFINITIONS HAVE BEEN SPECIFIED, THE SENTINEL BLOCK IS READ BEFORE THE APPROPRIATE EXIT IS TAKEN. WHEN AN END OF FILE MARK IS DETECTED AND FORMAT DEFINITIONS HAVE NOT BEEN SPECIFIED, THE END OF FILE EXIT IS ALWAYS TAKEN. IF TWO END OF FILE MARKS ARE DETECTED IN SUCCESSION, AN END-OF-FILE EXIT IS TAKEN.

13.5. BLOCK BUFFERING PACKAGE

THE BLOCK BUFFERING PACKAGE IS RESPONSIBLE FOR THE INTERNAL MANIPULATION OF A FILE AT THE BLOCK LEVEL. IT IS DESIGNED TO PROVIDE THE USER WITH A GENERAL MEANS OF FILE PROCESSING WITHOUT THE USE OF FORMAT DEFINITIONS. BLOCKS ARE READ AND WRITTEN UNALTERED, BY THE SYSTEM.

WHENEVER A SENTINEL, END-OF-FILE MARK, OR END-OF-TAPE IS DETECTED, THE ABNORMAL EXIT IS TAKEN WITH REGISTER A1 CONTAINING AN ERROR CODE TO DEFINE THE TYPE OF ABNORMAL CONDITION, AND AN ADDRESS TO WHICH THE USER MUST RETURN CONTROL, AFTER HIS CLOSE PROCEDURES, TO RE-ENTER THE MAIN SEQUENCE OF EXECUTION. REGISTER A0 WILL CONTAIN AN ACCESS WORD TO DEFINE THE LENGTH AND LOCATION OF THE SENTINEL.

CONTROL OF A BUFFER POOL IS MAINTAINED BY THIS PACKAGE. EACH FILE IS ASSOCIATED WITH A BUFFER POOL WHICH MAY BE SHARED BY OTHER FILES IN THE OBJECT PROGRAM.

THE ITEM CONTROL ROUTINES USE THIS PACKAGE FOR MANIPULATION OF THE BLOCKS IT CREATES.

13.5.1. OUTPUT

THE OUTPUT SUBROUTINES PROVIDE FOR THE CREATION OF FILES ON AN EXTERNAL MEDIA. THESE SUBROUTINES ARE CONCERNED ONLY WITH TRANSFERING BLOCKS OF DATA TO THE I/O HANDLER FOR WRITING AND SECURING EMPTY BUFFERS FROM THE BUFFER POOL FOR THE USER.

THE FUNCTIONS PROVIDED BY THESE ROUTINES INCLUDE OPEN FILE, WRITE (SEQUENTIAL), WRITE (RANDOM), CLOSE REEL, CLOSE FILE, AND RELEASE. THESE FUNCTIONS ARE PROCEDURE CALLS CODED IN THE OBJECT PROGRAM TO REQUEST THE DESIRED FUNCTION.

13.5.1.1. OPEN OUTPUT

THE OPEN OUTPUT REQUEST INITIALIZES A FILE CONTROL TABLE FOR SUBSEQUENT OUTPUT OPERATIONS ON THE DEVICE ASSIGNED TO THE TABLE. THERE ARE NO RESTRICTIONS ON THE NUMBER OF OPEN REQUESTS WHICH MAY BE CODED TO REFERENCE AN OUTPUT FILE, BUT ONLY ONE OF THEM MAY BE SELECTED BY THE PROGRAM TO PERFORM THE ACTUAL FUNCTION. THE OPEN REQUEST IS CODED

```
   TAG  BOPEN 'MODE' FCTABLE(1),'OPTION ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE FUNCTIONS OF THE OPEN REQUEST ARE TO INITIALIZE THE FILE CONTROL TABLE FOR PROCESSING A FILE BY BLOCKS WITHOUT RESPECT TO ANY FORMAT DEFINITION OR LABEL PROCESSING, AND TO REQUEST A BUFFER FROM THE BUFFER POOL FOR THE FIRST OUTPUT BLOCK OF DATA. THE LOCATION AND LENGTH OF THE AVAILABLE BUFFER IS LOADED AS AN ACCESS WORD IN REGISTER A0 BEFORE RETURN IS MADE.

13.5.1.2. WRITE

THE WRITE REQUEST IS A DIRECTIVE TO THE SYSTEM TO RECORD A BLOCK OF DATA, UNALTERED, ON THE OUTPUT DEVICE ASSIGNED TO THE FILE CONTROL TABLE. THE WRITE REQUEST MAY BE CODED IN ONE OF THE FOLLOWING FORMS:

```
(1)TAG  BWRIT FCTABLE
(2)TAG  BWRIT FCTABLE NBRWDS
(3)TAG  BWRIT FCTABLE 0,ADDR
(4)TAG  BWRIT FCTABLE NBRWDS,ADDR
```

WHERE DEFINITIONS OF THE WRITE FORMS ARE IDENTICAL TO THOSE SPECIFIED FOR ITEM PROCESSING, EXCEPT THAT THE REFERENCE HERE IS TO BLOCKS.

FORM(1) IS INTERPRETED AS A REQUEST TO WRITE A BLOCK OF THE SIZE PREVIOUSLY GIVEN, AS THE NEXT CONSECUTIVE BLOCK. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE BLOCK IN THE GIVEN BUFFER.

FORM(2) IS INTERPRETED AS A REQUEST TO WRITE A BLOCK OF SIZE LESS THAN THE AREA PREVIOUSLY GIVEN, AS THE NEXT CONSECUTIVE BLOCK. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE BLOCK IN THE GIVEN BUFFER. IF BLOCK SIZE WAS SPECIFIED AS VARIABLE, NBRWDS IS THE BLOCK SIZE WRITTEN. HOWEVER IF BLOCK SIZE WAS SPECIFIED AS FIXED, THE BALANCE OF THE BUFFER AREA IS ZERO FILLED BEFORE WRITING.

FORM(3) IS INTERPRETED AS A REQUEST TO MOVE A BLOCK OF THE SIZE PREVIOUSLY GIVEN, FROM THE AREA SPECIFIED BY THE 0,ADDR FIELD, INTO THE BUFFER TO BE RECORDED AS THE NEXT CONSECUTIVE BLOCK.

FORM(4) IS EQUIVELANT TO FORM(2) EXCEPT THAT THE ADDR SUBFIELD SPECIFIES AN AREA FROM WHICH THE BLOCK IS TO BE MOVED.

ON RETURN TO THE USER REGISTER A0 WILL CONTAIN AN ACCESS WORD FOR A NEWLY ACQUIRED BUFFER. FOR FORMS(2-4), THE USERS MINOR SET OF CONTROL REGISTERS ARE DESTROYED.

13.5.1.3. WRITE RANDOM

THE WRITE RANDOM REQUEST IS A DIRECTIVE TO THE SYSTEM TO RECORD A BLOCK OF DATA, UNALTERED, IN A SPECIFIED POSITION WITHIN A FILE. THE FILE IN QUESTION MUST RESIDE ON MASS STORAGE AND HAVE FIXED BLOCK SIZE. WHEN THE RANDOM REQUEST IS EXECUTED THE SYSTEM WRITES THE BLOCK AT THE POSITION SPECIFIED BY THE BLKNBR. THE RANDOM REQUEST IS CODED

```
(1) TAG  BRWRIT  FCTABLE BLKNBR
(2) TAG  BRWRIT  FCTABLE BLKNBR NBRWDS
(3) TAG  BRWRIT  FCTABLE BLKNBR 0,ADDR
(4) TAG  BRWRIT  FCTABLE BLKNBR NBRWDS,ADDR
```

FORM(1) IS INTERPRETED AS A REQUEST TO RECORD A RANDOM BLOCK OF THE SIZE PREVIOUSLY GIVEN. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE BLOCK IN THE GIVEN BUFFER.

FORM(2) IS INTERPRETED AS A REQUEST TO RECORD A RANDOM BLOCK OF SIZE LESS THAN PREVIOUSLY GIVEN. THIS FORM IS USED WHEN THE USER HAS ALREADY PLACED THE BLOCK IN THE GIVEN BUFFER. IF THE NBRWDS IS LESS THAN THE SPECIFIED BLOCK SIZE FOR THE FILE, THE BALANCE OF THE BUFFER AREA IS ZERO FILLED. IT IS LEFT UNDISTURBED IF THE FILE IS BEING PROCESSED AS AN IN/OUT FILE.

FORM(3) IS INTERPRETED AS A REQUEST TO MOVE A BLOCK OF THE SIZE PREVIOUSLY GIVEN, FROM THE AREA SPECIFIED BY THE 0,ADDR FIELD, INTO THE BUFFER TO BE RECORDED AS A RANDOM BLOCK.

FORM(4) IS EQUIVALENT TO FORM(3) EXCEPT THAT NBRWDS SUBFIELD MAY SPECIFY A SIZE LESS THAN THAT PREVIOUSLY GIVEN. THE UNUSED PORTION OF THE FIXED BLOCK IS EITHER ZERO FILLED OR LEFT UNDISTURBED. IT IS ZERO FILLED IF THE FILE IS BEING PROCESSED AS AN OUTPUT FILE.
IT IS LEFT UNDISTURBED IF THE FILE IS BEING PROCESSED AS AN IN/OUT FILE.

ON RETURN TO THE USER REGISTER A0 WILL CONTAIN AN ACCESS WORD FOR A NEWLY ACQUIRED BUFFER. FOR FORMS(2-4), THE USERS MINOR SET OF CONTROL REGISTERS ARE DESTROYED.

13.5.1.4. CLOSE REEL

THE CLOSE REEL REQUEST IS USED TO TERMINATE THE WRITING OF AN OUTPUT REEL AND AUTOMATICALLY INITIATE THE WRITING OF SUBSEQUENT REELS OF THE SAME FILE. THE CLOSE REQUEST IS CODED:

```
TAG  BCLOSE 'REEL' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE CLOSE REEL REQUEST EXECUTES THE REWIND OPTION AND REFERENCES THE EXECUTIVE TO SWAP REELS. ON RETURN TO THE USER REGISTER A_0 WILL CONTAIN THE ACCESS WORD FOR A NEWLY ACQUIRED BUFFER AND PROCESSING MAY CONTINUE ON THE ALTERNATE REEL. THIS REQUEST IS IGNORED IF THE FILE IS ON FASTRAND.

13.5.1.5. MARK

THE MARK REQUEST WILL RECORD AN END OF FILE MARK ON THE DEVICE ASSIGNED TO THE FILE. THE MARK REQUEST IS CODED:

```
    TAG BMARK FCTABLE(1) ... FCTABLE(N)
```

WHERE LABEL IS OPTIONAL AT THE USER'S DISCRETION. MARK IS THE NAME OF THE REQUEST AND MUST BE USED AS SHOWN. THE FCTABLE FIELD IS THE SYMBOLIC LOCATION OF THE FILE CONTROL TABLE THAT MAINTAINS THE FILE IN QUESTION.

THIS REQUEST MAY BE ISSUED ANY TIME THE USER DESIRES, BUT IT SHOULD ALWAYS BE ISSUED BEFORE A CLOSE REEL OR CLOSE FILE REQUEST. IF FILE IS ON FASTRAND, THE REQUEST IS IGNORED AS ONLY ONE END-OF-FILE FOR A FILE IS RECORDED. WHEN THE FILE IS CLOSED USING THE CLOSE PROCEDURE, THE HIGHEST FASTRAND ADDRESS WRITTEN FOR THE FILE IS RECORDED. ON RETURN TO THE USER REGISTER A_0 WILL CONTAIN THE ACCESS WORD FOR A NEWLY ACQUIRED BUFFER.

13.5.1.6. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE WRITING OF AN OUTPUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL OUTPUT FILES. THE CLOSE FILE REQUEST IS CODED

```
   TAG  BCLOSE 'FILE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE CLOSE FILE REQUEST CLOSES OUT THE FILE CONTROL TABLE, RELEASES ALL BUFFERS BACK TO THE POOL, AND EXECUTES THE REWIND OPTION IF THE FILE IS ON TAPE. IF THE FILE IS ON FASTRAND THE REWIND OPTION IS IGNORED AND HIGHEST FASTRAND ADDRESS RECORDED IS SAVED IN THE FILE CONTROL TABLE. FOR A CATOLOGUED FILE THE BBP WILL RECORD PERTINENT PARAMETERS IN THE FILES DIRECTORY ITEM.

13.5.1.7. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN OUTPUT FILE AND ITS ASSIGNED DEVICES. THE DISPOSITION OF THE FILE DEPENDS UPON THE CATALOGING OPTION SPECIFIED ON THE ASSIGN CONTROL CARD AND THE RELEASE OPTION SPECIFIED ON REQUEST. THE RELEASE REQUEST IS CODED

```
    TAG BREL FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION ON THE REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

13.5.2. INPUT

THE INPUT SUBROUTINES PROVIDE FOR THE READING OF AN EXISTING FILE FROM AN EXTERNAL MEDIA. THE SUBROUTINES ARE CONCERNED ONLY WITH THE TRANSFERING OF BLOCKS OF DATA TO THE USER ON REQUEST.

THE FUNCTIONS PROVIDED BY THESE ROUTINES INCLUDE OPEN FILE, READ (SEQUENTIAL), READ (RANDOM), CLOSE REEL, CLOSE FILE, AND RELEASE. THESE FUNCTIONS ARE PROCEDURE CALLS CODED IN THE OBJECT PROGRAM TO REQUEST THE DESIRED FUNCTION.

UPON RETURN TO THE USER FROM A READ OR READ RANDOM REQUEST, REGISTER A0 WILL CONTAIN AN ACCESS WORD THAT DEFINES THE ACTUAL LENGTH AND LOCATION OF THE CURRENT BLOCK. NOTE THAT FOR REQUESTS THAT SPECIFIES A MOVE, THE USER WILL HAVE TWO IMAGES OF THE CURRENT BLOCK. ONE IN THE LOCATION SPECIFIED BY THE ADDR SUBFIELD; THE OTHER IN THE BUFFER AS SPECIFIED BY THE CONTENTS OF A0. THE BUFFER IMAGE IS ALWAYS AVAILABLE UNTIL ANOTHER READ REQUEST IS ISSUED.

13.5.2.1. OPEN INPUT

THE OPEN REQUEST INITIALIZES THE FILE CONTROL TABLE FOR SUBSEQUENT INPUT OPERATIONS ON THE DEVICE ASSIGNED TO THE TABLE. THE OPEN REQUEST IS COD:ED

```
    TAG  BOPEN ,MODE, FCTABLE(1),,OPTION, ... FCTABLE(N),,OPTION,
```

WHERE THE DEFINITION OF THE REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE FUNCTIONS OF THE OPEN REQUEST ARE TO INITIALIZE THE FILE CONTROL TABLE FOR PROCESSING A FILE BY BLOCKS, IN THE FORWARD OR BACKWARD DIRECTION, WITHOUT RESPECT TO ANY FORMAT DEFINITION.

FOR A CATOLOGUED FILE THAT HAS BEEN PROCESSED BY THE BLOCK BUFFERING PACKAGE, PERTINENT PARAMETERS PERTAINING TO THE FILE MAY BE RETRIVED FROM THE FILE'S DIRECTORY ITEM BY REFERENCING DITEM$. IF THE FILE HAD BEEN CATOLOGUED, THE FOLLOWING PARAMETERS ARE AVAILABLE:

1. BLOCK SIZE - IF NEGATIVE VARIABLE BLOCK SIZE.
2. ITEM SIZE - IF ZERO VARIABLE ITEM SIZE.
3. BLOCK SENTINEL
4. FASTRAND END-OF-FILE ADDRESS IF FASTRAND FILE.

ALL PARAMETERS AS DEFINED BY FILE PROCEDURE (13.6.1) MUST BE PROVIDE IN FILE CONTROL TABLE BEFORE OPEN REQUEST IS INITIATE.

13.5.2.2. READ

THE READ REQUEST IS A DIRECTIVE TO THE SYSTEM TO PRESENT TO THE USER, UNALTERED, THE NEXT CONSECUTIVE BLOCK OF DATA. THE READ REQUEST IS CODED IN ONE OF THE FOLLOWING FORMS.

```
    (1)TAG   BREAD FCTABLE
    (2)TAG   BREAD FCTABLE 0,ADDR
    (3)TAG   BREAD FCTABLE NBRWDS,ADDR
```

WHERE THE DEFINITIONS OF THE READ FORMS ARE IDENTICAL TO THOSE SPECIFIED FOR ITEM PROCESSING, EXCEPT THAT THE REFERENCE HERE IS TO BLOCKS. THE END-OF-FILE EXIT, CODED AT THE ITEM LEVEL, IS NOT REQUIRED SINCE THE BLOCK BUFFERING PACKAGE HAS NO KNOWLEDGE OF BLOCK CONTENTS.

FORM(1) IS INTERPRETED AS A REQUEST FOR THE NEXT CONSECUTIVE BLOCK IN THE FILE.

FORM (2) IS EQUIVALENT TO FORM(1) EXCEPT THAT THE 0,ADDR FIELD SPECIFIES AN AREA TO MOVE THE ENTIRE BLOCK.

FORM (3) IS SIMILAR TO FORM (2) EXCEPT THAT THE NBRWDS FIELD SPECIFIES THE NUMBER OF WORDS FROM THE BLOCK TO MOVE.

EACH READ REQUEST AUTOMATICALLY FREES THE BUFFER WHICH WAS PRESENTED BY THE PREVIOUS READ REQUEST. FOR FORMS(2-3), THE USERS MINOR SET OF CONTROL REGISTERS ARE DESTROYED.

13.5.2.3. READ RANDOM

THE READ RANDOM REQUEST IS A DIRECTIVE TO THE SYSTEM TO PRESENT TO THE USER, UNALTERED, A SPECIFIC BLOCK OF DATA FROM A FILE. THE FILE IN QUESTION MUST RESIDE ON

MASS STORAGE AND HAVE FIXED SIZE BLOCKS. THE READ RANDOM REQUEST IS CODED:

```
(1) TAG BRREAD FCTABLE BLKNBR
(2) TAG BRREAD FCTABLE BLKNBR  0,ADDR
(3) TAG BRREAD FCTABLE BLKNBR  NBRWDS,ADDR
```

WHEN THE READ RANDOM REQUEST IS EXECUTED, THE SYSTEM OBTAINS FROM THE INPUT MEDIUM THE BLOCK SPECIFIED IN THE BLK-NBR FIELD. FORM(1) IS INTERPRETED AS A REQUEST FOR THE BLOCK SPECIFIED BY THE BLKNBR FOR THE FILE. IF THE REQUEST IS OF FORM(2), THE ENTIRE BLOCK IS MOVED TO THE USER'S AREA SPECIFIED BY THE 0,ADDR FIELD. IF THE REQUEST IS OF FORM(3), ONLY THAT NUMBER OF WORDS OF THE BLOCK SPECIFIED BY THE NBRWDS SUBFIELD IS MOVED TO THE AREA SPECIFIED BY THE ADDR SUBFIELD.

EACH READ RANDOM REQUEST AUTOMATICALLY FREES THE BUFFER WHICH WAS PRESENTED BY THE PREVIOUS READ REQUEST. FOR FORM(2-3), THE USERS MINOR SET OF CONTROL REGISTERS ARE DESTROYED.

13.5.2.4. CLOSE REEL

THE CLOSE REEL REQUEST IS USED TO TERMINATE THE READING OF AN INPUT REEL. THE CLOSE REEL REQUEST IS CODED:

```
TAG BCLOSE 'REEL' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE CLOSE REEL REQUEST EXECUTES THE REWIND OPTION AND REFERENCES THE EXECUTIVE TO SWAP REELS. THIS REQUEST IS IGNORED IF THE FILE IS ON FASTRAND.

13.5.2.5. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE READING OF AN INPUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL INPUT FILES. THE CLOSE FILE REQUEST IS CODED

```
TAG  BCLOSE 'FILE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE CLOSE FILE REQUEST CLOSES OUT THE FILE CONTROL TABLE, RELEASES ALL BUFFERS BACK TO THE POOL, AND EXECUTES THE REWIND OPTION FOR ALL EXTENDED REELS IF THE FILE IS ON TAPE. IF IT IS ON FASTRAND THE REWIND OPTION IS IGNORED.

13.5.2.6. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN INPUT FILE AND ITS ASSIGNED DEVICE. THE DISPOSITION OF THE FILE DEPENDS UPON THE CATOLOGING OPTION SPECIFIED ON THE ASSIGNED CONTROL CARD AND RELEASE OPTION SPECIFIED ON REQUEST. THE INPUT RELEASE REQUEST IS CODED

```
TAG BREL FCTABLE(1),'OPTION'... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION ON THE REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

13.5.3. IN/OUT

THE IN/OUT MODE OF PROCESSING PROVIDES THE CAPABILITY TO MODIFY AN EXISTING MASS SOFRAGE FILE WITHOUT THE NECESSITY TO RECREATING THE FILE. THIS IS ACCOMPLISHED BY ALLOWING BOTH READ AND WRITE REQUESTS TO REFERENCE THE SAME FILE. THE USER MAY THEREFORE READ A BLOCK, ALTER THE BLOCK AND REPLACE IT IN THE FILE.

IN/OUT FILES MAY BE MODIFIED BY UPDATING OR EXTENDING. WHEN UPDATING, BLOCKS MAY BE MODIFIED OR COMPLETELY REWRITTEN. IF THEY ARE REWRITTEN, THE SIZE OF THE BLOCK REWRITTEN MUST BE EQUAL TO THE SIZE OF THE ORIGINAL BLOCK. IF A FILE IS BEING EXTENDED, THE BLOCK SIZE IS OF NO CONSEQUENCE.

THE FUNCTIONS PROVIDED BY THESE ROUTINES INCLUDE OPEN FILE, READ (SEQUENTIAL), READ (READ RANDOM), EXCLUSIVE READ RANDOM, WRITE (SEQUENTIAL), WRITE (RANDOM), CLOSE FILE, AND RELEASE. THESE FUNCTIONS ARE PROCEDURE CALLS CODED IN THE OBJECT PROGRAM TO REQUEST THE DESIRED FUNCTION.

13.5.3.1. OPEN IN/OUT

THE OPEN IN/OUT REQUEST INITIALIZES A FILE CONTROL TABLE FOR SUBSEQUENT INPUT AND OUTPUT OPERATIONS ON A MASS STORAGE FILE ASSIGNED TO THE TABLE. THE OPEN IN/OUT REQUEST IS CODED

```
   TAG  BOPEN 'MODE' FCTABLE(1),'OPTION' ... FCTABLE(N),'OPTION'
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM IN/OUT PROCESSING.

THE FUNCTION OF THE OPEN IN/OUT REQUEST IS TO INITIALIZE THE FILE CONTROL TABLE FOR PROCESSING A FILE BY BLOCKS WITHOUT RESPECT TO ANY FORMAT DEFINITION AND TO ISSUE A READ FOR THE FIRST BLOCK IN THE FILE. A NORMAL RETURN TO THE USER IS MADE WITH REGISTER A0 CONTAINING THE LENGTH AND LOCATION OF THE BLOCK READ IF AN ERROR CONDITION WAS NOT DETECTED; OTHERWISE, THE USERS ABNORMAL EXIT IS TAKEN. THE READ REQUEST INITIATED IN THE OPEN PROCEDURE IS PROVIDED TO GIVE THE USER AN OPPORTUNITY TO UPDATE THE LABEL BLOCK OF A FILE. THEREFORE, ANY LABEL UPDATE MUST BE MADE IN THE LABEL AREA BEFORE THE FIRST REQUEST. IF THE FILE IS EXTENDING, THE FIRST WRITE REQUEST WILL AUTOMATICALLY WRITE OUT THE BLOCK READ FROM THE OPEN REQUEST. RETURN IS MADE TO THE USER WITH REGISTER A0 CONTAINING THE LENGTH AND LOCATION OF THE NEXT AVAILABLE BUFFER TO WRITE. BEFORE THE FIRST WRITE REQUEST THE USER MUST SUPPLY THE FASTRAND END-OF-FILE ADDRESS IN THE FCT, WORD 21, FROM WHICH EXTENSION OF THE FILE WILL BEGIN.

13.5.3.2. READ-WRITE

ALL INPUT READ FORMS AND OUTPUT WRITE FORMS ARE LEGITIMATE REQUESTS FOR AN IN/OUT UPDATING FILE. FOR AN IN/OUT ENTENDING FILE, ONLY THE OUTPUT WRITE FORMS ARE ALLOWED.

THE FUNCTIONS AND LOGIC OF THESE READ FORMS ARE IDENTICAL TO THAT SPECIFIED FOR SEQUENTIAL PROCESSING IN/OUT FILES AT THE ITEM LEVEL, EXCEPT THE REFERENCE HERE IS TO BLOCKS.

13.5.3.3. READ-WRITE RANDOM

THE READ AND WRITE RANDOM REQUESTS FOR IN/OUT FILES ARE IDENTICAL IN FORM AND FUNCTION TO THOSE SPECIFIED FOR RANDOM INPUT AND OUTPUT FILES. THE LOGIC IMPLIED WHEN RANDOMLY REFERENCING A FILE AT THE ITEM LEVEL, ALSO APPLIES HERE EXCEPT THAT THE REFERENCE IS TO BLOCKS. ALL RANDOM READ AND WRITE FORMS ARE LEGITIMATE WHEN PROCESSING ON THE BLOCK LEVEL.

13.5.3.4. EXCLUSIVE READ RANDOM

THE EXCLUSIVE READ RANDOM REQUEST SPECIFIES TO THE SYSTEM THAT EXCLUSIVE USE OF A FIXED LENGTH BLOCK IS REQUESTED. ALL OTHER REFERENCES TO THIS PARTICULAR BLOCK ARE INTERLOCKED UNTIL IT HAS BEEN RELEASED. WHENEVER A BLOCK IS OBTAINED FOR EXCLUSIVE USE, IT IS AUTOMATICALLY RELEASED BY THE NEXT REFERENCE TO THE FILE. THE READ RANDOM EXCLUSIVE REQUEST IS CODED

```
     TAG  BXREAD FCTABLE BLKNR
     TAG  BXREAD FCTABLE BLKNR   0,ADDR
     TAG  BXREAD FCTABLE BLKNR   NBRWDS,ADDR
```

WHERE THE DEFINITION OF THE REQUEST IS IDENTICAL TO THAT SPECIFIED FOR RANDOM READ REQUESTS.

13.5.3.5. CLOSE FILE

THE CLOSE FILE REQUEST IS USED TO TERMINATE THE READING AND/OR WRITING OF AN IN/OUT FILE AND MUST BE INCLUDED IN THE CODING FOR ALL IN/OUT FILES. THE CLOSE FILE REQUEST IS CODED:

```
   TAG  BCLOSE 'FILE' FCTABLE(1) ... FCTABLE(N)
```

WHERE THE DEFINITION OF THIS REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

THE CLOSE FILE REQUEST CLOSES OUT THE FILE CONTROL TABLE AND RELEASES ALL BUFFERS BACK TO THE POOL.

SINCE IN/OUT FILES RESIDE ON MASS STORAGE DEVICES, REWIND PROCEDURES, IF CODED, ARE IGNORED.

13.5.3.6. RELEASE

THE RELEASE REQUEST IS USED TO IMMEDIATELY RELINQUISH A PROGRAM'S ASSOCIATION WITH AN IN/OUT FILE AND ITS ASSIGNED DEVICES. THE DISPOSITION OF THE FILE DEPENDS UPON THE CATALOGING OPTION SPECIFIED ON THE ASSIGN CONTROL CARD AND THE RELEASE OPTION SPECIFIED ON THE REQUEST. THE RELEASE REQUEST IS CODED

```
    TAG BREL FCTABLE,'OPTION',., FCTABLE,'OPTION'
```

WHERE THE DEFINITION ON THE REQUEST IS IDENTICAL TO THAT SPECIFIED FOR ITEM PROCESSING.

13.5.4. BLOCK LAYOUT

BLOCKS PROCESSED BY THE BLOCK BUFFERING PACKAGE MUST BE SPECIFIED AS FIXED OR VARIABLE IN LENGTH. BLOCKS WRITTEN ON FASTRAND WILL HAVE A CONTROL WORD TO SPECIFY THE BLOCK SIZE FOR VARIABLE LENGTH BLOCKS.

TAPE FILES (FOR FIXED AND VARIABLE LENGTH BLOCKS)

```
-----------
:         :
:         :
:         :
: DATA    :
:  AREA   :
:         :
:         :
:         :
:         :
-----------
```

FASTRAND FILES

```
-----------
:    : A  :
-----------         -----------
:         :         :         :
:         :         :         :
:         :         :         :
:         :         :         :
: DATA    :         : DATA    :
:  AREA   :         :  AREA   :
:         :         :         :
:         :         :         :
:         :         :         :
-----------         -----------
 VARIABLE             FIXED
  LENGTH             LENGTH
   BLOCK              BLOCK
```

A = LENGTH OF THE BLOCK (BLOCK SIZE)

13.6. FILE CONTROL TABLE

THE DATA HANDLING ROUTINES UTILIZE THE FILE CONTROL TABLE TO COMMUNICATE BETWEEN THE USER PROGRAM AND THE FILE. THE COMPLEXITY AND LENGTH OF SUCH A TABLE DEPENDS UPON THE LEVEL OF INPUT/OUTPUT CONTROL BEING USED TO ACCESS THE FILE(SEE CHAPTER 12, SECTION 3.1).

THE CONTENTS OF THE FILE CONTROL TABLE ARE SUPPLIED BY USING A PROCEDURE NAMED FILE. THIS CALL ALLOWS THE USER TO DESCRIBE THE OPERATING CHARACTERISTICS OF A FILE IN TERMS OF THE DATA THEREIN AND ITS MANIPULATION.

WHEN A FILE IS CREATED AND CATALOGUED THE SYSTEM WILL RETAIN A LIMITED NUMBER OF PARAMETERS FROM THE FCT IN THE FILES DIRECTORY ITEM. THE FOLLOWING DEFINES THE FORMAT AND LOCATIONS IN THE DIRECTORY ITEM AND THE PARAMETERS SAVED. THE LOCATION WITHIN THE DIRECTORY OF THE SAVED PARAMETERS DIFFER DEPENDING ON TYPE OF FILE. THE FOLLOWING DEFINES THE FORMAT FOR A TAPE AND FASTRAND FILE:

```
TAPE:
        WORD-
              14        BLOCK SENTINEL.
              24-H1     BLOCK SIZE, NEGATIVE IF VARIABLE
                        BLOCK SIZE.
                -H2     ITEM SIZE, ZERO IF VARIABLE ITEM
                        SIZE.
FASTRAND:
        WORD-
              24-H1     BLOCK SIZE, NEGATIVE IF VARIABLE
                        BLOCK SIZE.
                -H2     ITEM SIZE, ZERO IF VARIBLE ITEM
                        SIZE.
              26        FASTRAND EOF ADDRESS.
              27        BLOCK SENTINEL.
```

FOR IN/OUT UPDATING FILES WHEN THE END-OF-FILE HAS BEEN DETECTED, THE MODE FOR THE FILE IS SET TO OUTPUT AND ONLY OUTPUT REQUESTS FOR THE FILE WILL BE ALLOWED. ON DETECTION OF THE END-OF-FILE CONDITION, THE USERS ABNORMAL EXIT WILL BE TAKEN WITH REGISTER A0 CONTAINING THE LENGTH AND LOCATION OF THE NEXT BUFFER TO WRITE.

13.6.1. THE FILE PROCEDURE

THE FILE PROCEDURE CALL CONTAINS SEVERAL LISTS SOME OF WHICH ARE OPTIONAL. EACH LIST, EXCEPT THE 'FILENAME', FORMAT LIST, HAS A TITLE ENCLOSED IN QUOTATION MARKS AND SUBFIELDS SEPARATED BY COMMAS. THE TITLE IS THE FIRST PARAMETER OF A LIST AND SERVES TO IDENTIFY THE LIST. THE ORDER OF THE LISTS IS OF NO CONSEQUENCE, BUT THE ORDER OF THE SUBFIELDS MUST BE MAINTAINED AS SHOWN.

THE COMPLETE PROCEDURE CALL IS ILLUSTRATED BELOW. THE VARIOUS LISTS ARE SHOWN ON SEPARATE LINES ONLY TO FACILITATE CLARITY.

```
TAG  FILE 'FILENAME',FORMAT ;
          'SIZE',BLOCK,ITEM ;
          'POOL',LINK,LAF ;
          'ERROR',FILE,DEVICE,ABNORMAL ;
          'SENT',VALUE,MASK,POSITION ;
          'LABEL',NBRWDS,ADDR ;
          'FREEWD',NBRWDS,ADDR ;
          'EOR',ADDR
```

13.6.1.1. 'FILENAME' LIST

THE 'FILENAME',FORMAT LIST IS MANDATORY AND MUST BE THE FIRST LIST IN EVERY FILE PROCEDURE CALL. THE 'FILENAME' SUBFIELD IS THE EXTERNAL NAME OF THE FILE IN QUESTION AND MAY BE UP TO 12 CHARACTERS IN LENGTH. THIS NAME IS GENERALLY IDENTICAL TO THAT

SPECIFIED IN AN @ASG CONTROL STATEMENT. IF IT IS NOT IT MUST BE LINKED TO THAT NAME BY A @USE CONTROL STATEMENT. THE FORMAT SUBFIELD IS THE NAME OF THE FORMAT DEFINITION ENTRY POINT TO USE WITH THIS FILE. IT IS MANDATORY AND MAY BE UP TO 6 CHARACTERS IN LENGTH. IF NO SPECIFIC FORMAT IS DESIRED; THAT IS, THE FILE IS TO CONSIST PURELY OF ITEMS WITHIN BLOCKS, THE SYSTEM DEFINED FORMAT NULFOR MUST BE USED. THE SYSTEM WILL RECOGNIZE THE NAME 'BBP' TO INDICATE PROCESSING AT THE BLOCK BUFFERING LEVEL. THE 'FILENAME',FORMAT LIST MAY THEREFORE BE CODED:

```
'ALPHA    ',LION ; - TO PROCESS A FILE NAMED ALPHA AT THE ITEM
                     LEVEL USING A FORMAT ELEMENT CALLED LION OR,
'ALPHA    ',NULFOR ; - TO PROCESS A FILE NAMED ALPHA AT THE ITEM
                     LEVEL WITHOUT A FORMAT ELEMENT OR,
'ALPHA    ','BBP' ; - TO PROCESS A FILE NAMED ALPHA AT THE BLOCK
                      BUFFERING LEVEL.
```

IF THE SYSTEM'S PROVIDED LION FORMAT IS TO BE USED, THE FORMAT SUBFIELD 'LION' MUST BE PRECEDED BY AN ASTERISK-*LION. THIS INDICATES TO THE SYSTEM THAT ADDITIONAL AREA MUST BE RESERVED FOR VARIOUS COUNTERS USED BY LION FORMAT SUBROUTINES.

13.6.1.2. 'SIZE' LIST

A LIST TITLED 'SIZE' DESCRIBES THE ITEM/BLOCK RELATIONSHIP FOR A FILE. IF THIS LIST IS NOT CODED, THE SIZE LIST SAVED IS USED. IF THE FILE IS NOT CATALOGUED AND THIS LIST NOT CODED, THE SYSTEM WILL SUBSTITUTE BUFFER SIZE MINUS 3 FOR BLOCK SIZE, AND EXPECT VARIABLE LENGTH ITEMS IF PROCESSING AT THE ITEM LEVEL. THE LIST FORM IS:

```
'SIZE',BLOCK,ITEM ;
```

WHERE,

```
 ITEM - IS THE NUMBER OF ACTUAL DATA WORDS IN EACH
        FIXED LENGTH ITEM.
BLOCK - IS THE MAXIMUM NUMBER OF WORDS IN EACH BLOCK.
        THIS VALUE MUST INCLUDE THE FOLLOWING:
           1. THE NUMBER OF WORDS IN THE FORMAT
              DEFINITION FOR EACH ITEM,
           2. THE NUMBER OF WORDS IN THE FORMAT
              DEFINITION FOR THE BLOCK.
```

THE LIST 'SIZE' MAY THEREFORE BE CODED IN ONE OF THE FOLLOWING FORMS:

```
(1) 'SIZE'' BLOCK,ITEM  ;
(2) 'SIZE',*BLOCK,ITEM ;
(3) 'SIZE', BLOCK ;
(4) 'SIZE',*BLOCK ;
```

FORM(1) SPECIFIES FIXED LENGTH ITEM PROCESSING IN VARIABLE LENGTH BLOCKS. THE ROUTINE RECORDS FIXED LENGTH ITEMS UNTIL THE AREA REMAINING IS LESS THAN THE ITEM SIZE. THE BLOCK IS THEN WRITTEN AND A NEW BUFFER ACQUIRED. WITH THIS FORM THE USER MAY VARY THE NUMBER OF FIXED ITEMS IN A BLOCK (THEREBY VARYING THE BLOCK LENGTH), BY USING THE DRAIN REQUEST WHICH TRUNCATES THE BLOCK BEFORE IT IS WRITTEN.

FORM(2) SPECIFIES FIXED LENGTH ITEM PROCESSING IN FIXED LENGTH BLOCKS. THE ASTERISK IN THE BLOCK SUBFIELD DENOTES THAT THE BLOCK VALUE IS TO BE USED TO COMPUTE A BLOCK LENGTH THAT CONTAINS AN INTEGRAL NUMBER OF ITEMS. THIS BLOCK LENGTH IS THE SIZE OF EACH BLOCK WRITTEN IN THE FILE. THE USER MAY VARY THE NUMBER OF ITEMS IN A BLOCK BY USING THE DRAIN REQUEST BUT TRUNCATION WILL NOT OCCUR. THIS FORM MUST BE USED WHEN RANDOMLY PROCESSING A FILE.

FORM(3) SPECIFIES VARIABLE LENGTH ITEM PROCESSING IN VARIABLE LENGTH BLOCKS. THE ROUTINE WILL RECORD VARIABLE LENGTH ITEMS UNTIL THE AREA REMAINING IN THE BLOCK IS SMALLER THAN THE NEXT ITEM. THE BLOCK IS THEN WRITTEN AND A NEW BUFFER ACQUIRED.

IF BLOCK BUFFERING LEVEL PROCESSING IS INDICATED (BBP IN THE FORMAT SUBFIELD), THIS FORM SPECIFIES THAT VARIABLE LENGTH BLOCKS WILL BE PROCESSED.

FORM(4) SPECIFIES VARIABLE LENGTH ITEM PROCESSING IN FIXED LENGTH BLOCKS. THE ASTERISK IN THE BLOCK SUBFIELD DENOTES THAT THE VALUE GIVEN IS THE ACTUAL BLOCK SIZE TO USE. THIS FORM ALLOWS SPANNING(OVERLAPPING BLOCKS WITH AN ITEM). THE SYSTEM SPANS BLOCKS WITH AN ITEM WHENEVER THE ITEM PRESENTED IS LARGER THAN THE BLOCK SIZE SPECIFIED OR THE AREA REMAINING IN A BLOCK.

IF THE BLOCK BUFFERING LEVEL IS INDICATED, THIS FORM SPECIFIES THAT FIXED LENGTH BLOCKS WILL BE PROCESSED.

13.6.1.3. 'POOL' LIST

A LIST TITLED 'POOL' DESCRIBES THE BUFFERING FOR A FILE. THE LIST FORM IS:

```
        'POOL',LINK,LAF ;
```

WHERE,

```
              LAF - IS THE LOOK AHEAD FACTOR. THIS VALUE IS THE
                    NUMBER OF BLOCKS TO READ AHEAD FOR INPUT
                    FILES.
             LINK - IS THE ADDRESS OF THE POOL CONTROL WORD.
```

13.6.1.4. 'ERROR' LIST

A LIST TITLED 'ERROR' ALLOWS THE USER TO SPECIFY THOSE ERROR CONDITIONS HE WISHES TO PROCESS. IF THIS LIST IS NOT CODED, THE SYSTEM INSTITUTES A STANDARD PROCEDURE FOR EACH TYPE OF ERROR. THE LIST FORM IS:

```
        'ERROR',FILE,DEVICE,ABNORMAL ;
```

WHERE,

```
             FILE - SPECIFIES A SYMBOLIC LOCATION IN THE USER'S
                    PROGRAM TO WHICH CONTROL IS RETURNED
                    WHENEVER A CONTIGENCY ARISES PERTAINING TO
                    THE USER'S OPERATION OF HIS FILE, E.G.,
                    OPENING AN ALREADY OPEN FILE.
           DEVICE - SPECIFIES A SYMBOLIC LOCATION IN THE USER'S
                    PROGRAM TO WHICH CONTROL IS RETURNED
                    WHENEVER A CONTIGENCY ARISES PERTAINING TO
                    THE PHYSICAL OPERATION OF A PERIPHERAL
                    DEVICE.
         ABNORMAL - SPECIFIES A SYMBOLIC LOCATION IN THE USER'S
                    PROGRAM TO WHICH CONTROL IS RETURNED
                    WHENEVER ONE OF THE FOLLOWING CONDITIONS
                    ARISES.
                      1. SENTINEL FOUND
                      2. END OF RECORDED INFORMATION
                      3. END OF TAPE
                      4. END OF MASS STORAGE AREA

                    AT THE ITEM LEVEL CONTROL WILL BE RETURNED
                    TO THIS LOCATION WHENEVER A FORMAT ERROR IS
                    DISCOVERED BY A FORMAT SUBROUTINE.
```

THE ERROR PROCEDURE PROVIDED BY THE SYSTEM CAUSES PROGRAM TERMINATION AFTER AN APPROPRIATE ERROR CODE AND THE LOCATION OF THE ACTIVE REFERENCE IS RECORDED IN THE PROGRAM LOG.

13.6.1.5. 'SENT' LIST

A LIST TITLED 'SENT' DESCRIBES THE SENTINEL TO LOOK FOR WHEN READING A FILE AT THE BLOCK BUFFERING LEVEL(ON OUTPUT THE USER WRITES HIS OWN SENTINEL BLOCKS). IF THE

FILE IS NOT CATALOGUED AND THIS LIST NOT CODED, NO SENTINEL CHECK IS MADE. THE LIST FORM IS:

```
'SENT','VALUE,MASK,POSITION ;
```

WHERE,

VALUE - IS THE NUMERIC OR ALPHANUMERIC CHARACTERS WHICH THE SYSTEM USES TO DETECT A SENTINEL BLOCK. IF ALPHANUMERIC CHARACTERS ARE USED, THEY MUST BE SURROUNDED BY QUOTATION MARKS. THE SIZE LIMIT OF THIS FIELD IS 6 CHARACTERS.

MASK-MASK APPLIED AGAINST DATA WORD IN BLOCK TO DETERMINE SENTINEL BLOCK. IF OMITTED OR ZERO, A MASK OF ALL BITS (FOR FULL WORD) WILL BE GENERATED.

POSITION- POSITION OF DATA WORD IN BLOCK TO APPLY MASK TO DETERMINE SENTINEL BLOCK. IF OMITTED, THE FIRST WORD (POSITION 0) IS ASSUMED.

WHEN THE BLOCK MASK IS PROVIDED IN THE FILE CONTROL TABLE THE BBP WILL BYPASS CHECKING FOR FASTRAND EOF ADDRESS, IF A FASTRAND FILE, AND INPUT WILL BE STOPPED ONLY WHEN A SENTINEL BLOCK IS DETECTED. HOWEVER, WHEN OPERATING ON BBP LEVEL ONLY, THE SENTINEL WORD PROVIDED IN FCT IS COMPARED WITH THE FIRST WORD OF EACH BLOCK, UNLESS ITEM FLAG (WORD 16,S1) IS NON-ZERO, TO DETERMINE THE SENTINEL BLOCK.

WHEN THIS LIST IS OMITTED AND THE FORMAT IS LION (FORMAT PARAMETER PREFIXED WITH *), THEN THE STANDARD LION SENTINEL VALUES ARE GENERATED (JUST AS IF THE USER HAD CODED 'SENT','/./././').

13.6.1.6. 'LABEL' LIST

A LIST TITLED 'LABEL' DEFINES THE LOCATION AND LENGTH OF THE USER'S LABEL IMAGE. IF THIS LIST IS NOT CODED, NO USER LABEL PROCESSING IS DONE. THE LIST FORM IS:

```
'LABEL',NBR-WDS,ADDR ;
```

WHERE,

NBR-WDS - IS THE NUMBER OF WORDS COMPRISING THE USER'S LABEL
ADDR - IS THE SYMBOLIC LOCATION OF THE USER'S LABEL

THIS LIST IS IGNORED BY THE SYSTEM WHEN PROCESSING AT THE BLOCK BUFFERING LEVEL AND, AT THE ITEM LEVEL IF A FORMAT DEFINITION FOR A LABEL BLOCK WAS NOT GIVEN.

13.6.1.7. 'FREEWD' LIST

A LIST TITLED 'FREEWD' DEFINES THE 1107 LION LABEL BLOCK AREA KNOWN AS FREE WORDS. IF THIS LIST IS NOT CODED, NO FREE WORD PROCESSING IS DONE. THE INCLUSION OF THIS LIST SIMPLY PROVIDES COMPATIBILITY WITH LION. THE LIST FORM IS:

```
'FREEWD',NBR-WDS,ADDR ;
```

WHERE,

NBR-WDS - IS THE NUMBER OF WORDS COMPRISING THE FREE WORD INFORMATION
ADDR - IS THE SYMBOLIC LOCATION OF THE FREE WORD INFORMATION

13.6.1.8. 'EOR' LIST

A LIST TITLED 'EOR' DEFINES THE END-OF-REEL EXIT. IF THIS LIST IS NOT CODED, THE ROUTINE WILL AUTOMATICALLY PERFORM END-OF-REEL PROCEDURES. THE LIST FORM IS:

```
'EOR', ADDR
```

WHERE,

ADDR - IS A SYMBOLIC LOCATION IN THE USER'S PROGRAM TO WHICH CONTROL WILL BE RETURNED WHENEVER AN END-OF-REEL CONDITION IS DETECTED.

THIS LIST IS IGNORED WHEN PROCESSING AT THE BLOCK BUFFERING LEVEL.

13.6.2. FILE PROCEDURE LAYOUT

EXECUTION OF THE FILE PROCEDURE CAUSES THE GENERATION OF THE FILE CONTROL TABLE (FCT). THE COMPLETE TABLE IS DIVIDED INTO 3 SECTIONS IN THE FOLLOWING ORDER.

1. I/O CONTROL,
2. BLOCK BUFFERING, AND
3. ITEM CONTROL

THE I/O CONTROL SECTION IS ALWAYS GENERATED AND USED BY THE ROUTINE REGARDLESS OF THE LEVEL OF PROCESSING.

THE BLOCK BUFFERING SECTION IS GENERATED ALONG WITH THE I/O CONTROL SECTION WHEN THE FILE PROCEDURE SPECIFIES BLOCK BUFFERING LEVEL OF PROCESSING.

THE ITEM CONTROL SECTION IS GENERATED ALONG WITH THE BLOCK BUFFERING AND I/O CONTROL SECTIONS WHEN THE FILE PROCEDURE IMPLIES ITEM LEVEL OF PROCESSING. A TOTAL OF 46 LOCATIONS ARE USED WHEN PROCESSING AT THE BLOCK BUFFERING AND ITEM LEVELS-25 LOCATIONS FOR BLOCK BUFFERING CONTROL AND 21 LOCATIONS FOR ITEM CONTROL. IF THE SYSTEMS LION FORMAT IS EMPLOYED, 10 ADDITIONAL LOCATIONS ARE NECESSARY FOR VARIOUS COUNTERS USED BY THE FORMAT SUBROUTINES. IF THE FILE PROCEDURE IS NOT EMPLOYED, THE USER MUST PROVIDE THE REQUIRED AREA AND PROVIDE THE LIST PARAMETERS DESCRIBED ABOVE IN THEIR PROPER LOCATIONS AS INDICATED IN THE FILE CONTROL LAYOUT DESCRIBED BELOW.

```
    35        29        23        17        11        05        00
    --------------------------------------------------------------
 00 :                   (FILE NAME)  WORD 1                       :
    --------------------------------------------------------------
 01 :                   (FILE NAME)  WORD 2                       :
    --------------------------------------------------------------
 02 :  USED BY EXEC     :INT ACT ID: INTERRUPT ACTIVITY START     :
    --------------------------------------------------------------
 03 :  STATUS :FUNCTION :   AFC     :        SUBSTATUS            :
    --------------------------------------------------------------
 04 :                      ACCESS  WORD                           :
    --------------------------------------------------------------
 05 :                       FASTRAND  ADDRESS                     :
    --------------------------------------------------------------
 06 :OPEN FLAG: (LAF)    :  MODE    :  CURRENT BUFFER HELD BY I/O :
    --------------------------------------------------------------
 07 : (USER MAX BLOCK SIZE)         :FILE FLAG:  LOCK   :FAST FLAG:
    --------------------------------------------------------------
 08 : (DEVICE ERROR EXIT)           :     (FILE ERROR EXIT)       :
    --------------------------------------------------------------
 09 :  BEG ADDR OF USER BUFFER      :      (ABNORMAL ERROR EXIT)  :
    --------------------------------------------------------------
 10 :                 (SENTINEL     VALUE)                        :
    --------------------------------------------------------------
 11 :             (LINK)            : ACTIVITY RE-ENTRY LOCATION  :
    --------------------------------------------------------------
 12 :EXEC      :   RQ     :  Q      :      SECTOR COUNT           :
    --------------------------------------------------------------
 13 :   END OF QUEUE                :       BEG OF QUEUE          :
    --------------------------------------------------------------
 14 : CURRENT DATA LOC IN BUFFER    :    ROUTINE RETURN LOCATION  :
    --------------------------------------------------------------
 15 :   SIZE OF DATA BLOCK          :    BEG LOCATION OF DATA     :
    --------------------------------------------------------------
 16 :ITEM/FLAG:FRAME CT :CHECK/PT   :    BLOCK COUNT              :
    --------------------------------------------------------------
 17 : EXCLUSIVE READ RETURN         :   RELATIVE FCTABLE LOCATION :
    :------------------------------------------------------------:
 18 :                  EXEC USEAGE                                :
    :------------------------------------------------------------:
 19 :  EXEC    : EQUIP   :READ/FLAG:        EXEC                  :
    :------------------------------------------------------------:
 20 :            ACCUMULATIVE FASTRAND ADDRESS                    :
    :------------------------------------------------------------:
 21 :               FASTRAND EOF ADDRESS                          :
    :------------------------------------------------------------:
 22 :          :DATA POS : EXEC    USAGE                          :
    --------------------------------------------------------------
 23 :                     BLOCK MASK                              :
    --------------------------------------------------------------
 24 :                       EXEC USAGE                            :
    --------------------------------------------------------------
 25 : IO FLAG :DEBUG FLG:DO IO FLG:      (FORMAT ENTRY NAME)      :
    --------------------------------------------------------------
 26 : LGTH DATA AREA LEFT IN BLK    : ADD CURRENT ITEM DATA AREA  :
    --------------------------------------------------------------
 27 : NBR WORDS LEFT TO PROCESS     : NBR WDS REQUESTED FOR MOVE  :
    --------------------------------------------------------------
 28 : LGTH LEFT IN BLK PLS FORMAT   : ADD CURRENT ITEM PREFIX     :
    --------------------------------------------------------------
 29 :       (END OF REEL EXIT)      : LGTH CURRENT ITEM PLS FORMAT:
    --------------------------------------------------------------
```

```
30 :       (FIXED ITEM SIZE)       : NUMBER OF ITEMS IN BLOCK      :
   -----------------------------------------------------------------
31 :BLOCK FLG: REEL FLG:WRITE FLG: READ FLG: LOCK FLG:USER FUNC:
   -----------------------------------------------------------------
32 : T-S FLG : SPAN FLG: EOF FLG : EXCLUSIVE READ ADDRESS        :
   -----------------------------------------------------------------
33 : MRK FLG :LBL FRMAT:ITM FRMAT:BLK FRMAT:EOR FRMAT:EOF FRMAT:
   -----------------------------------------------------------------
34 :                     (USER LABEL WORDS)                      :
   -----------------------------------------------------------------
35 :                     (USER FREE WORDS)                       :
   -----------------------------------------------------------------
36 :                     THREE WORD PACKET                       :
   -----------------------------------------------------------------
37 :                FOR BLOCK BUFFERING REQUESTS                 :
   -----------------------------------------------------------------
38 :                     FROM ITEM HANDLER                       :
   -----------------------------------------------------------------
39 :  USER DEVICE ERROR EXIT      :  USER FILE ERROR EXIT        :
   -----------------------------------------------------------------
40 :  USER ABNORMAL EXIT          : SDF FLAG:          :         :
   -----------------------------------------------------------------
41 : ROUTINE RETURN LOCATION      : ROUTINE RETURN LOCATION      :
   -----------------------------------------------------------------
42 : ITEM PREFIX LENGTH           : ITEM SUFFIX LENGTH           :
   -----------------------------------------------------------------
43 : BLOCK PREFIX LENGTH          : BLOCK SUFFIX LENGTH          :
   -----------------------------------------------------------------
44 : ROUTINE RETURN LOCATION      :                              :
   -----------------------------------------------------------------
45 :                              :  DIAG DUMT ROUTINE LOCATION  :
   -----------------------------------------------------------------
```

THE PARENTHESIZED PARAMETERS ARE THOSE THAT ARE SPECIFIED IN THE FILE PROCEDURE CALL. SHOULD IT BE DESIRABLE TO GENERATE A FILE CONTROL TABLE NOT USING THE FILE PROCEDURE THOSE PARENTHESIZED PARAMETERS MUST BE PROVIDED IN THEIR SPECIFIED FIELD BEFORE ANY REFERENCE IS MADE TO THE FILE CONTROL TABLE. ALSO, THE OPEN FLAG FIELD MUST BE SET NON-ZERO, AND FILE FLAG SET EQUAL TO 1 IF FIXED BLOCK SIZE.

13.7. BUFFER POOL

FOR EACH FILE ASSIGNED TO AN ACTIVITY ,THERE MUST BE AN ASSOCIATED BUFFER POOL. A BUFFER POOL MAY BE ASSIGNED TO ONE PARTICULAR FILE OR IT MAY BE ASSIGNED TO MANY FILES. THE SIZE OF EACH BUFFER MUST BE EQUAL TO THE MAXIMUM BLOCK SIZE SPECIFIED FOR THE FILES PLUS 3 EXTRA WORDS FOR CONTROL PURPOSES.

THE PROPER PROCEDURES FOR GENERATION OF A BUFFER POOL ARE PROVIDED IN 13.3.6. THE NUMBER OF BUFFERS PROVIDED FOR A FILE IS LEFT TO THE DISCRETION OF THE USER, BUT FOR MAXIMUM EFFICIENCY, THE NUMBER OF BUFFERS FOR EACH FILE SHOULD BE ONE MORE THAN THE FILES LOOK-A-HEAD FACTOR(LAF). THE SYSTEM WILL ASSUME THAT LAF PLUS 1 BUFFERS HAVE BEEN PROVIDED. IT IS THE RESPONSIBILITY OF THE USER TO PROVIDE THE NECESSARY BUFFERS, AND PRIOR TO REFERENCING THE BLOCK BUFFERING PACKAGE, THE EXECUTIVE REQUEST OR PROCEDURE IN 12.3.6 IS EXECUTED. IF THE LAF CANNOT BE SATISIFIED, DUE TO INSUFFICIENT BUFFERS IN THE USERS BUFFER POOL THE USER'S FILE ERROR EXIT WILL BE TAKEN WITH AN ERROR CODE OF 011. IF THE LINK CONTROL WORD IS NOT PROVIDED OR THE BUFFER SIZE IS LESS THAN MAXIMUM BLOCK SIZE,A FILE ERROR EXIT WILL BE TAKEN.

13.8. ERROR PROCESSING

WHEN A DEVICE, FILE, OR ABNORMAL ERROR CONDITION OCCURS, CONTROL WILL BE RETURNED TO THE USER VIA THE EXIT PROVIDED IN THE FILE CONTROL TABLE.
A DESCRIPTION FOR EACH ERROR CONDITION AND ITS RESPECTIVE CODE RETURNED TO THE USER IS PROVIDED.

13.8.1. DEVICE ERRORS

FOR EACH OCCURRENCE OF A DEVICE ERROR, CONTROL WILL BE RETURNED TO THE USER VIA THE FILE CONTROL TABLE'S DEVICE EXIT. REGISTER A1 WILL CONTAIN THE STATUS CODE FOR THE PARTICULAR DEVICE ERROR IN ITS UPPER HALF AND THE USER'S RE-ENTRY LOCATION IN ITS LOWER HALF. FOR AN INPUT FILE REGISTER, A0 WILL CONTAIN THE SIZE AND DATA LOCATION OF THE BLOCK BEING READ WHEN THE DEVICE ERROR OCCURRED. SEE 10.1.4.1 FOR DECRIPTION OF STATUS CODES.

13.8.2. FILE ERRORS

FILE ERRORS ARE CREATED WHEN A REQUEST HAS BEEN GIVEN BY THE USER WHICH VIOLATES PRESCRIBED BBP PROCEDURES. FOR EACH OCCURRENCE OF A FILE ERROR CONTROL WILL BE RETURNED TO THE USER VIA THE FILE CONTROL TABLES FILE ERROR EXIT. REGISTER A1 WILL CONTAIN AN ERROR CODE IN ITS UPPER HALF AND USER RE-ENTRY LOCATION IN ITS LOWER HALF. POSSIBLE ERROR CODES AND THEIR DESCRIPTION ARE:

CODE (OCTAL)	DESCRIPTION
01	BUFFER POOL LINK NOT PROVIDED
02	REQUEST TO CLOSE A FILE ALREADY CLOSED.
03	REQUEST TO OPEN A FILE ALREADY OPENED.
04	REQUEST TO READ OR WRITE A CLOSED FILE.
05	REQUEST TO WRITE A BLOCK GREATER THAN MAXIMUM BLOCK SIZE OR A REQUEST TO REWRITE A BLOCK IN IN/OUT MODE AND SIZE REQUESTED TO WRITE IS GREATER THAN SIZE READ.
06	FASTRAND VARIABLE BLOCK SIZE SPECIFIED ON BLOCK READ LARGER THEN MAX BLOCK SIZE OR READING VARIABLE BLOCKS FROM FIXED BLOCK FILE.
07	RANDOM REQUEST AND FILE NOT ASSIGNED TO FASTRAND.
010	RANDOM REQUEST AND BLOCK SIZE NOT FIXED.
011	INSUFFICIENT BUFFERS IN POOL TO SATISFY LAF FOR INPUT OR OUTPUT REQUEST.
012	INVALID BLOCK NUMBER FOR RANDOM READ REQUEST.
013	READ REQUEST FOR A BLOCK GREATER THAN BLOCK SIZE READ.
014	FILE NOT ASSIGNED TO FASTRAND FOR IN/OUT MODE.
015	RANDOM WRITE REQUEST FOR INPUT FILE.
016	RANDOM READ REQUEST FOR OUTPUT FILE.
017	READ REQUEST WITH NUMBER WORDS

	PARAMETER SPECIFIED BUT NO ADDRESS SPECIFIED.
020	READ REQUEST IN OUTPUT FILE.
021	BUFFER SIZE LESS THAN SPECIFIED BLOCK SIZE.
023	LOCATION OF LINK OR BUFFER AREA OUTSIDE OF USERS ASSIGNED AREA.
024	BLOCK SIZE NOT FIXED FOR REVERSE MODE FOR FASTRAND FILE.
025	NO I/O FACILITIES ASSIGNED OR IMPROPER EQUIPMENT TYPE.
026	WRITE REQUEST IN INPUT FILE.
027	MARK REQUEST FOR INPUT FILE.
064	INVALID MODE PARAMETER FOR OPEN REQUEST.

THE FOLLOWING ARE ERROR CODES WHICH WILL OCCUR ONLY WHEN PROCESSING AT THE ITEM LEVEL:

030	MOVE ADDRESS NOT GIVEN ON IN/OUT READ OR WRITE REQUEST.
031	MOVE ADDRESS NOT GIVEN ON READ OR WRITE REQUEST OF SPANNED ITEM.
032	ITEM SIZE FIXED ON REQUEST TO WRITE A SPANNED ITEM.
033	ADDRESS OF ITEM = MOVE ADDRESS GIVEN ON IN/OUT READ OR WRITE REQUEST.
034	ADDRESS OF ITEM = MOVE ADDRESS GIVEN ON RANDOM READ OR WRITE REQUEST.
035	RANDOM REQUEST MADE IN FILE WITHOUT FIXED ITEMS AND BLOCKS.
036	OPERATION ATTEMPTED ON A CLOSED FILE.
037	LABEL SPECIFIED BUT NOT FOUND ON OPEN INPUT REQUEST.
040	UPDATE FLAG NOT SET ON IN/OUT FILE READ REQUEST.
041	HIGHEST ADDRESS WRITTEN NOT SET ON IN/OUT FILE WRITE EXTEND REQUEST.
042	DRAIN REQUESTED ON IN/OUT FILE.
043	A REQUEST WAS MADE TO WRITE A SPANNED ITEM AFTER READING IT.
044	A REQUEST WAS MADE TO WRITE MORE THAN THE SIZE READ ON IN/OUT FILE.
045	READ BACKWARDS REQUESTED ON A SPANNED ITEM.
046	FORMAT NOT SET IN FILE CONTROL TABLE.
047	TAPE POSITIONED IMPROPERLY ON OPEN REQUEST
050	FRAME COUNT ERROR DETECTED AND USER DID NOT SPECIFY TO ACCEPT AS NORMAL.
051	AN EOF BLOCK WAS DETECETD ON A RANDOM READ REQUEST.
052	FIXED ITEM LARGER THAN MAX BLOCK SIZE.

13.8.3. ABNORMAL CONDITIONS

WHEN AN ABNORMAL CONDITION OCCURS, CONTROL IS RETURNED TO THE USER VIA THE FILE CONTROL TABLE'S ABNORMAL EXIT. FOR EACH OCCURRENCE OF AN ABNORMAL CONDITION, REGISTER A1 WILL CONTAIN THE ABNORMAL CODE IN ITS UPPER HALF AND THE USER'S RE-ENTRY LOCATION IN ITS LOWER HALF. REGISTER A0 WILL CONTAIN THE SIZE AND DATA LOCATION OF THE BLOCK READ OR TO BE WRITTEN.

ABNORMAL CODES RETURNED TO USER ARE:

CODE(OCTAL)	DESCRIPTION
01	EOF-OF-FILE MARK OR LOAD POINT HAS BEEN DETECTED FOR INPUT TAPE FILE OR FASTRAND HIGHEST ADDRESS (EOF) HAS BEEN DETECTED.
02	END-OF-TAPE MARK HAS BEEN DETECTED FOR OUTPUT FILE
06	SENTINEL BLOCK HAS BEEN DETECTED FOR FILE.
010	BLOCK PREVIOUSLY READ EXCLUSIVELY HAS BEEN TIMED OUT BY THE SYSTEM FOR AN IN/OUT FILE.

THE FOLLOWING ERROR CODES ARE RETURNED TO THE USER VIA THE ABNORMAL ERROR EXIT WHILE AN ERROR IS DETECTED IN A SENTINEL BLOCK WHEN USING THE SYSTEMS LION FORMAT:

CODE (OCTAL)	DESCRIPTION
004	LABEL WORD ERROR HAS BEEN DETECTED IN LABEL BLOCK FOR AN INPUT FILE.
012	LABEL BLOCK CANNOT BE LOCATED, BAD TAPE POSITION.
013	BLOCK SIZE SPECIFIED IN FILE CONTROL TABLE IS LESS THAN 14 WORDS FOR AN OUTPUT FILE. FOR AN INPUT FILE BLOCK SIZE OF SENTINEL IS GREATER THAN BLOCK SIZE SPECIFIED IN FILE CONTROL TABLE.
014	ITEM SIZE SPECIFIED IN FILE CONTROL TABLE NOT EQUAL TO ITEM SIZE IN SENTINEL BLOCK FOR AN INPUT FILE.

RECOVERY CAN BE MADE FOR ERROR CODES 004, 013, AND 014 BY CLOSING THE FILE, MODIFYING THE RESPECTIVE FIELDS IN THE FILE CONTROL TABLE, AND INITIATING THE OPEN PROCEDURE AGAIN. FOR ERROR CODE 012, THE TAPE MUST BE REPOSITIONED.

13.9. BLOCK BUFFERING ER REQUEST CALLING SEQUENCES

THE FOLLOWING CALLING SEQUENCES MAY BE USED BY THE USER TO REFERENCE BLOCK BUFFERING FUNCTIONS. THESE CALLING SEQUENCES DESCRIBE THE ACTUAL CODING GENERATED BY USING THE PROCEDURE CALLS FOR THE BLOCK BUFFERING FUNCTIONS AND MAY BE USED DIRECTLY BY THE USER. FOR EACH CALL TO BLOCK BUFFERING, A0 MUST CONTAIN THE ADDRESS OF A ONE OR TWO WORD PACKET THAT WILL CONTAIN THE REQUIRED PARAMETERS FOR THE REQUEST.

13.9.1. OPEN FUNCTION

```
LA,U    A0,PKT ADDR
ER      BOPEN$
NORMAL  RETURN
```

```
PACKET:
       35     30      23       17                    0
       ---------------------------------------------------
   1   :       : MODE  : REWIND  :     FCTABLE ADDRESS   :
       ---------------------------------------------------
```

```
WORD 1 S2-INDICATES THE REQUESTED MODE FOR FILE.
          MODES:  02        INPUT(FORWARD)
                  03        INPUT(REVERSE)
                 040        OUTPUT
                 041        IN/OUT
       S3-INDICATES REWIND OPTION FOR TAPE FILE OR IN/OUT
          EXTENDING/UPDATING OPTION.
          OPTIONS:  0       REQUEST FOR REWIND ONLY.
                            UPDATING FOR IN/OUT MODE.
                  013       EXTENDING FOR IN/OUT MODE.
                  021       REQUEST FOR REWIND WITH LOCK.
                  023       REQUEST FOR NO REWIND.
       H2-ADDRESS OF THE FILE CONTROL TABLE THAT MAINTAIN
          THE FILE IN QUESTION.
```

13.9.2. WRITE FUNCTIONS

13.9.2.1. FORM 1

```
    LA,U   A0,PKT ADDR
    ER     BWRIT$
   NORMAL RETURN

    PACKET:
           35                       17                       0
           ---------------------------------------------------
     1     :                        :     FCTABLE ADDRESS    :
           ---------------------------------------------------
  WORD 1   H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS
             THE FILE IN QUESTION.
```

13.9.2.2. FORMS 2-4

```
   LA,U    A0,PKT ADDR*
   LMJ     X11,B$WMOV
   LA,U    A0,PKT ADDR*
   ER      BWRIT$
   NORMAL  RETURN

  *PACKET:
           35                       17                       0
           ---------------------------------------------------
     1     :                        :     FCTABLE ADDRESS    :
           ---------------------------------------------------
     2     :       NBR-WDS          :   DATA ADDRESS         :
           ---------------------------------------------------
  WORD 1   H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS
             THE FILE IN QUESTION.
  WORD 2   H1-SPECIFIES THE NUMBER OF WORDS TO MOVE OR 0.
           H2-SPECIFIES THE LOCATION IN USERS AREA DATA WILL
             BE MOVED FROM OR 0.
```

13.9.3. READ FUNCTIONS

13.9.3.1. FORM 1

```
   LA,U    A0,PKT ADDR
   ER      BREAD$
   NORMAL RETURN
```

13.9.3.2.   FORMS 2-3

```
   LA,U    A0,PKT ADDR**
   ER      BREAD$
   LA,U    A0,PKT ADDR**
   LMJ     X11,B$MOVE
   NORMAL  RETURN

 **PACKET:
           35                       17                      0
           ---------------------------------------------------
     1     :                        :    FCTABLE ADDRESS     :
           ---------------------------------------------------
     2     :       NBR-WDS          :  DATA ADDRESS          :
           ---------------------------------------------------
  WORD 1   H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS
             THE FILE IN QUESTION.
  WORD 2   H1-SPECIFIES THE NUMBER OF WORDS TO MOVE OR 0.
           H2-SPECIFIES THE LOCATION IN USERS AREA DATA WILL
              BE MOVED TO.
```

13.9.4. RANDOM WRITE FUNCTIONS

13.9.4.1.   FORM 1

```
   LA,U    A0,PKT ADDR***
   ER      BRWRT$
   NORMAL  RETURN
```

13.9.4.2.   FORMS 2-4

```
   LA,U    A0,PKT ADDR***
   LMJ     X11,B$WMOV
   LA,U    A0,PKT ADDR***
   ER      BRWRT$
   NORMAL  RETURN

 ***PACKET:
           35                       17                      0
           ---------------------------------------------------
     1     :                        :    FCTABLE ADDRESS     :
           ---------------------------------------------------
     2     :       NBR-WDS          :  DATA ADDRESS          :
           ---------------------------------------------------
     3     :                  BLOCK NUMBER                   :
           ---------------------------------------------------
  WORD 1   H1-NON-ZERO FOR AN EXCLUSIVE READ REQUEST.
           H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS
              THE FILE IN QUESTION.
  WORD 2   H1-SPECIFIES THE NUMBER OF WORD TO MOVE FROM/OR TO
              USERS AREA.
           H2-SPECIFIES THE LOCATION IN USERS AREA WHERE
              DATA WILL BE MOVED FROM OR TO.
              NOTE:  FOR FORM 1 READ OR WRITE FUNCTIONS WORD 2
                     IS ZERO.
  WORD 3      POSITION IN FILE WHERE BLOCK WILL BE READ OR
              WRITTEN.
```

13.9.5. RANDOM READ FUNCTIONS

13.9.5.1. FORM 1

```
LA,U    A0,PKT ADDR***
ER      BRRED$
NORMAL  RETURN
```

13.9.5.2. FORMS 2-3

```
LA,U    A0,PKT ADDR***
ER      BRRED$
LA,U    A0,PKT ADDR***
LMJ     X11,B$MOVE
NORMAL  RETURN
```

13.9.6. CLOSE REEL/FILE FUNCTIONS

```
LA,U    A0,PKT ADDR
ER      BCLOR$
NORMAL  RETURN
    OR
LA,U    A0,PKT ADDR
ER      BCLOF$
NORMAL  RETURN

PACKET:
        35     30      23       17                        0
        ---------------------------------------------------
   1    :      :       : REWIND :     FCTABLE ADDRESS    :
        ---------------------------------------------------
WORD 1  S3-REWIND OPTION FOR TAPE FILES
           OPTIONS:  0  REQUEST REWIND ONLY
                    021 REQUEST REWIND WITH LOCK
                    023 REQUEST NO REWIND
        H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS THE
           FILE IN QUESTION.
```

13.9.7. WRITE EOF FUNCTION

```
LA,U    A0,PKT ADDR
ER      BMARK$
NORMAL  RETURN

PACKET:
        35      30      23        17
        ---------------------------------------------------
   1    :                       :        FCTABLE ADDRESS   :
        ---------------------------------------------------
WORD 1  H2-ADDRESS OF FILE CONTROL TABLE THAT MAINTAINS THE
           FILE IN QUESTION.
```

13.10. ITEM HANDLER CALLING SEQUENCES

IF THE USER WISHES TO USE THE ITEM HANDLER WITHOUT CALLING THE ITEM HANDLER

PROCEDURES, THE FOLLOWING METHOD MUST BE USED. THE LOCATION OF A PACKET IS LOADED IN A0; AN LMJ IS MADE ON X11 TO THE APPROPRIATE ENTRY IN THE ITEM HANDLER. EG:

```
                    L,U       A0,PACKET LOCATION
                    LMJ       X11,ITEM ENTRY
```

THE ITEM HANDLER ENTRIES FOR THE VARIOUS FUNCTIONS ARE:

```
         OPEN FILE . . . . . IHOPN
         WRITE SEQUENTIAL  . IHWRT
         WRITE RANDOM  . . . IHWTRN
         READ SEQUENTIAL . . IHRD
         READ RANDOM . . . . IHRDRN
         DRAIN . . . . . . . IHDRN
         CLOSE REEL  . . . . IHCLR
         CLOSE FILE  . . . . IHCLF
```

THE REQUEST PACKETS ARE 1, 2, OR 3 WORDS LONG, AND CONTAIN THE FOLLOWING INFORMATION:

```
   OPEN FILE
                                    ------------------------------------
                                    : 0  :  *  : **  : FCTABLE REQUEST :
                                    ------------------------------------

   WRITE SEQUENTIAL
                                    ------------------------------------
                                    :        0       : FCTABLE ADDRESS :
                                    ------------------------------------
                                    :   MOVE LENGTH  :  MOVE ADDRESS   :
                                    ------------------------------------

   READ SEQUENTIAL
                                    ------------------------------------
                                    :        0       : FCTABLE ADDRESS :
                                    ------------------------------------
                                    :   MOVE LENGTH  :  MOVE ADDRESS   :
                                    ------------------------------------
                                    :        0       :    EOF EXIT     :
                                    ------------------------------------

   READ AND WRITE RANDOM
                                    ------------------------------------
                                    :       ***      : FCTABLE ADDRESS :
                                    ------------------------------------
                                    :                :  MOVE ADDRESS   :
                                    ------------------------------------
                                    :        0       :    EOF EXIT     :
                                    ------------------------------------

   DRAIN
                                    ------------------------------------
                                    :        0       : FCTABLE ADDRESS :
                                    ------------------------------------

   CLOSE FILE AND REEL
                                    ------------------------------------
                                    : 0  :  0  : **  : FCTABLE ADDRESS :
                                    ------------------------------------
```

THE FOLLOWING FIELDS NEED FURTHER DEFINING;

```
        *  MODE  -  040  -  OUTPUT FILE
                     02  -  INPUT FILE, FORWARD MODE
```

```
                  03  -  INPUT FILE, REVERSE MODE
                 041  -  IN/OUT FILE

 **  REWIND OPTION  -           0  -  REWIND WITHOUT
                                      INTERLOCK
                        FIELDATA L  -  REWIND WITH
                                      INTERLOCK
                        FIELDATA N  -  NO REWIND
     EXTENSION      -  FIELDATA E  -  OPEN IN/OUT FILE
                                      FOR EXTENSION

***  0  -  NORMAL RANDOM REQUEST
     1  -  EXCLUSIVE READ REQUESTED
```

13.11. FILE ORGANIZATION

13.11.1. ITEM LEVEL TAPE FILES

TAPE FILES PROCESSED BY ITEM CONTROL WILL HAVE END-OF-FILE MARKS RECORDED AUTOMATICALLY BY THE ROUTINE. THE PLACEMENT OF THESE MARKS IS AS FOLLOWS:

(A) TWO MARKS ARE RECORDED AFTER AN END OF REEL OR END OF FILE SENTINEL BLOCK.

(B) IF AN END OF REEL OR END OF FILE SENTINEL IS NOT WRITTEN, TWO END-OF-FILE MARKS FOLLOW THE LAST VALID DATA BLOCK.

(C) WHEN A REEL CONTAINS MORE THAN ONE FILE THE 2ND OF THE TWO MARKS IS OVERWRITTEN THEREBY SEPARATING FILES WITH ONE MARK.

(D) A SPECIAL SEPARATOR BLOCK IS RECORDED BY ITEM CONTROL WHEN A FILE IS OPENED AND CLOSED ON TAPE WITHOUT RECORDING ANY DATA AND END-OF-FILE SENTINELS HAVE NOT BEEN SPECIFIED. THE BLOCK DENOTES AN EMPTY (VOID) FILE AND SERVES TO SEPARATE THE END-OF-FILE MARKS.

```
SINGLE FILE REELS

      ----------     ----------
      : LABEL  :     : LABEL  :
      : BLOCK  :     : BLOCK  :
      ----------     ----------

      ----------     ----------     ----------     ----------
      :        :     :        :     :        :     :        :
      :        :     :        :     :        :     :        :
      :  FILE  :     :  FILE  :     :  FILE  :     :  FILE  :
      :        :     :        :     :        :     :        :
      :  AREA  :     :  AREA  :     :  AREA  :     :  AREA  :
      :        :     :        :     :        :     :        :
      :        :     :        :     :        :     :        :
      ----------     ----------     ----------     ----------
                        MARK           MARK
                      MARK           MARK
      ----------                                   ----------
      : EOF    :                                   : EOF    :
      :  BLOCK :                                   :  BLOCK :
      ----------                                   ----------
        MARK                                         MARK
        MARK                                         MARK
```

MULTI-FILE REELS

```
----------     ----------
: LABEL  :     : LABEL  :
: BLOCK  :     : BLOCK  :
----------     ----------

----------     ----------      ----------     ----------
:        :     :        :      :        :     :        :
:        :     :        :      :        :     :        :
:  FILE  :     :  FILE  :      :  FILE  :     :  FILE  :
:        :     :        :      :        :     :        :
:  AREA  :     :  AREA  :      :  AREA  :     :  AREA  :
:        :     :        :      :        :     :        :
:        :     :        :      :        :     :        :
:        :     :        :      :        :     :        :
----------     ----------      ----------     ----------
                  MARK            MARK
----------     ----------      ----------     ----------
: EOF    :     : LABEL  :      :        :     : EOF    :
:  BLOCK :     : BLOCK  :      :        :     :  BLOCK :
----------     ----------      :  FILE  :     ----------
 MARK                          :        :      MARK
----------     ----------      :  AREA  :     ----------
: LABEL  :     :        :      :        :     :        :
: BLOCK  :     :        :      :        :     :        :
----------     :  FILE  :      :        :     :  FILE  :
               :        :      ----------     :        :
----------     :  AREA  :        MARK         :  AREA  :
:        :     :        :        MARK         :        :
:        :     :        :                     :        :
:  FILE  :     :        :                     :        :
:        :     ----------                     ----------
:  AREA  :       MARK
:        :       MARK                         ----------
:        :                                    : EOF    :
:        :                                    :  BLOCK :
----------                                    ----------
                                                 MARK
----------                                       MARK
: EOF    :
:  BLOCK :
----------
 MARK
 MARK
```

MULTI-FILE REELS WITH AN EMPTY (VOID) FILE

```
----------     ----------
: LABEL  :     : LABEL  :
: BLOCK  :     : BLOCK  :
----------     ----------

----------     ----------     ----------     ----------
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:  FILE  :     :  FILE  :     :  FILE  :     :  FILE  :
:        :     :        :     :        :     :        :
:  AREA  :     :  AREA  :     :  AREA  :     :  AREA  :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
----------     ----------     ----------     ----------
                  MARK           MARK
----------     ----------     ----------     ----------
: EOF    :     : LABEL  :     :SEPARA- :     : EOF    :
:  BLOCK :     : BLOCK  :     : TOR BLK:     :  BLOCK :
----------     ----------     ----------     ----------
  MARK           MARK           MARK           MARK
----------       MARK           MARK        ----------
: LABEL  :                                  : EOF    :
: BLOCK  :                                  :  BLOCK :
----------                                  ----------
                                              MARK
----------                                    MARK
: EOF    :
:  BLOCK :
----------
  MARK
  MARK
```

```
MULTI-REEL FILES (REEL 1 SHOWN - SUBSQUENT REEL LAYOUT IS SAME)

      ----------     ----------
      : LABEL  :     : LABEL  :
      : BLOCK  :     : BLOCK  :
      ----------     ----------

      ----------     ----------     ----------     ----------
      :        :     :        :     :        :     :        :
      :        :     :        :     :        :     :        :
      :  FILE  :     :  FILE  :     :  FILE  :     :  FILE  :
      :        :     :        :     :        :     :        :
      :  AREA  :     :  AREA  :     :  AREA  :     :  AREA  :
      :        :     :        :     :        :     :        :
      :        :     :        :     :        :     :        :
      ----------     ----------     ----------     ----------
                        MARK           MARK
                       MARK            MARK
      ----------                                   ----------
      : EOR    :                                   : EOR    :
      :  BLOCK :                                   :  BLOCK :
      ----------                                   ----------
        MARK                                         MARK
        MARK                                         MARK
```

13.11.2. ITEM LEVEL FASTRAND FILES

```
----------     ----------
: LABEL  :     : LABEL  :
: BLOCK  :     : BLOCK  :
----------     ----------     ----------     ----------
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:  FILE  :     :  FILE  :     :  FILE  :     :  FILE  :
:        :     :        :     :        :     :        :
: AREA   :     :  AREA  :     :  AREA  :     :  AREA  :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
:        :     :        :     :        :     :        :
----------     ----------     ----------     ----------
: EOF    :                                   : EOF    :
:  BLOCK :                                   :  BLOCK :
----------                                   ----------
```

VOID FASTRAND FILES

```
----------     ----------                    ----------
:        :     :        :                    :        :
: LABEL  :     : LABEL  :                    : EOF    :
: BLOCK  :     : BLOCK  :                    :  BLOCK :
----------     ----------                    ----------
: EOF    :
:  BLOCK :
----------
```

13.11.3. BLOCK BUFFERING LEVEL TAPE AND FASTRAND FILES

TAPE AND FASTRAND FILES AT THE BLOCK BUFFERING LEVEL ARE ASSUMED TO CONTAIN PURE DATA. END-OF-FILE MARKS WILL NOT BE AUTOMATICALLY RECORDED. THE PLACEMENT OF THESE MARKS IS AT THE DISCRETION OF THE USER. IT IS SUGGESTED HOWEVER, THAT THEY BE USED TO ELIMINATE THE INPUT OF INVALID DATA WHEN READING.

13.12. SYSTEM DATA FILE FORMAT (SDF)

SDF PROVIDES THE SYSTEM WITH A BASIC FORMAT FOR DATA HANDLING BETWEEN THE VARIOUS SYSTEM COMPONENTS, AND BETWEEN THE SYSTEM AND THE USER. SDF FORMAT IS PRODUCED BY THE @DATA AND @FILE PROCESSORS, AND IS THE FINAL FORMAT OF THE SYSTEM ACCOUNTING FILE. ALL SYMBIONT PROCESSED FILES ARE OF THIS FORMAT, INCLUDING THOSE FILES TO BE INSERTED INTO THE CONTROL STREAM WITH THE @ADD CONTROL STATEMENT. SOURCE ELEMENTS ARE NECESSARILY OF THE SAME FORMAT.

SDF HAS INTEGRATED FASTRAND AND MAGNETIC TAPE COMPATIBILITY BY 1) ADHERING TO BLOCK LENGTHS OF FASTRAND SECTOR MULTIPLES,2) IMBEDDING FILE LABELS IN THE DATA AS THE INITIAL IMAGE OF THE FILE, 3) WRITING THE FILE AS A CONTINUOUS SET OF DATA WITHOUT BLOCK CONTROL WORDS AND 4) INCORPORATING THE BLOCK BUFFERING PACKAGE'S FACILITY OF SEQUENTIAL FILE PROCESSING.

13.12.1. FORMAT DEFINITION

DATA OF THE FILE IS RECORDED IN VARIABLE LENGTH IMAGES, WITH EACH IMAGE BEING PRECEDED WITH A LENGTH. FILE IMAGES FALL INTO TWO CATEGORIES:

1. CONTROL IMAGES

THESE IMAGES PROVIDE FILE CONTROL INFORMATION SUCH AS THE FILE LABEL, END OF FILE SENTINEL, ETC. THE FORMAT IS

```
35  30   24                  00
--------------------------------
:1:S : CL :                    :
--------------------------------
: CONTROL IMAGE.  NUMBER       :
--------------------------------
: OF WORDS DEFINED BY          :
--------------------------------
: CONTROL IMAGE LENGTH (CL).   :
--------------------------------
```

BIT 35 IS SET EQUAL TO 1 FOR ALL CONTROL IMAGES AND THE 'S' FIELD SPECIFIES THE TYPE. THE DEFINITION OF S VARIES WITH THE FILE TYPE. E.G. FILES GENERATED FOR PRINT SYMBIONTS USE UNIQUE S DEFINITIONS. HOWEVER, CERTAIN S DEFINITIONS APPLY THROUGHOUT THE SYSTEM. THEY ARE

TYPE	S VALUE
LABEL IMAGE.............	10
END OF FILE SENTINEL....	37
BYPASS IMAGE SENTINEL...	00

2. DATA IMAGES

```
35         24                  00
--------------------------------
:0:  IL    :                   :
--------------------------------
:     DATA IMAGE. NUMBER       :
--------------------------------
:     OF WORDS DEFINED         :
--------------------------------
:     BY IMAGE LENGTH (IL).    :
--------------------------------
```

DATA IMAGES HAVE BIT 35 SET EQUAL TO 0. THE REMAINDER OF

THIS WORD IS DEFINED ACCORDING TO THE TYPE OF DATA FILE PRODUCED, E.G, FILES GENERATED FOR PRINTING USE THIS FIELD FOR SPACING INCREMENTS, AND SOURCE ELEMENTS USE THIS FIELD FOR CYCLE NUMBERS.

13.12.2. FILE CONVENTIONS

FILES RESIDING ON MAGNETIC TAPE ARE RECORDED IN 224 WORD BLOCKS. IMAGES ARE ALLOWED TO SPAN TWO CONSECUTIVE BLOCKS. FASTRAND FILES ARE RECORDED AS A CONTINUOUS DATA SET. THE END-OFFILE SENTINEL CONTROL IMAGE TERMINATES FILE PROCESSING.

A SINGLE TAPE MARK IS USED TO SEPARATE FILES RESIDING ON MAGNETIC TAPE, AND TWO CONSECUTIVE TAPE MARKS SPECIFY END OF RECORDING. MASS STORAGE FILES USE A SOFTWARE DEFINED END-OF-FILE MARK FOR FILE OPERATIONS AT THE BLOCK LEVEL. THIS IS DEFINED AS A BLOCK WHICH HAS AS ITS FIRST WORD THE FIELDATA SENTINEL '$@EOF$'. THE BLOCK BUFFERING PACKAGE INTERPRETS THIS SENTINEL AS IT WOULD A TAPE MARK.

13.12.3. LABEL IMAGES

THE FILE LABEL IMAGE IS THE INITIAL IMAGE OF AN SDF FORMATTED FILE PRECEDED BY ITS FILE IMAGE CONTROL WORD. THE CONTROL WORD'S S FIELD MUST BE SET = OCTAL 10, AND CL SET TO THE LABEL IMAGE LENGTH THE MINIMUM IMAGE CONSISTS OF THE FOLLOWING

```
WORD 1    T    B
WORD 2-3  FILELABEL
```

WHERE T = THE TYPE OF FILE, AND IS LOCATED IN S1 (BITS 35-30) OF THE FIRST WORD. THE T FIELD IS SET EQUAL TO THE FIELDATA CHARACTER 'P' FOR THOSE FILES PRODUCED FOR PRINTING, OR 'C' FOR CARD FILES OR 'T' FOR PAPER TAPE FILES. THE B FIELD IS USED TO DENOTE BREAKPOINT NUMBERS OF A PRINT OR PUNCH FILE AND IS LOCATED IN S6 (BITS 5-0) OF WORD 1, AND SERVES AS AN EXTENSION TO THE FILE LABEL IMAGE. THE FILELABEL FIELD IS USED TO IDENTIFY THE FILE AND IS A 12 CHARACTER FIELD.

14. FILE UTILITY ROUTINES (FURPUR)

14.1. GENERAL

IN ADDITION TO THE EXECUTIVE CONTROL STATEMENTS DISCUSSED PREVIOUSLY, THERE IS A SET OF STATEMENTS RECOGNIZED BY THE EXECUTIVE AS CALLS FOR THE FILE UTILITY ROUTINES (FURPUR). WHEN THE EXECUTIVE ENCOUNTERS A FURPUR STATEMENT IT LOADS THE FURPUR PROCESSOR. FURPUR CONTINUES BY PROCESSING STATEMENTS UNTIL THE EXECUTIVE SIGNALS THE NEXT STATEMENT IS NOT A FURPUR STATEMENT.

STATEMENTS PROCESSED BY FURPUR ARE LISTED BELOW BY COMMAND NAME WITH A BRIEF DESCRIPTION OF THE FUNCTIONS THEY PERFORM.

COMMAND	FUNCTION
@CHG	CHANGES THE KEYS,MODE, RELATIVE F-CYCLE, AND FILE NAME OF A FILE, OR CHANGES THE NAME AND/OR VERSION OF PROGRAM FILE ELEMENT.
@CLOSE	WRITES TWO HARDWARE END OF FILE MARKS ON A TAPE AND REWINDS IT.
@COPIN	READS ELEMENTS IN ELEMENT FILE FORMAT ON TAPE AND INSERTS THEM IN A PROGRAM FILE.
@COPOUT	WRITES ELEMENTS OF PROGRAM FILES TO TAPE IN ELEMENT FILE FORMAT.
@COPY	TRANSFERS FILES OR PROGRAM FILE ELEMENTS FROM ONE FILE TO ANOTHER.
@CYCLE	SETS A NEW MAXIMUM NUMBER OF F-CYCLES TO BE RETAINED FOR FILES IN THE DIRECTORY OR SETS A NEW MAXIMUM NUMBER OF CYCLES TO BE RETAINED FOR A SYMBOLIC PROGRAM FILE ELEMENT.
@DELETE	DROPS A FILE IN THE DIRECTORY OR MARKS A PROGRAM FILE ELEMENT DELETED.
@ENABLE	REMOVES THE DISABLE FLAG FROM TEMPORARILY DISABLED CATALOGUED FILES.
@ERS	RELEASES SPACE ALLOCATED TO PROGRAM FILES.
@FIND	LOCATES AN ELEMENT ON A TAPE IN ELEMENT FILE FORMAT AND POSITIONS THE TAPE BEFORE THE ELEMENT'S LABEL BLOCK.
@MARK	WRITES A HARDWARE END OF FILE MARK ON TAPE.
@MOVE	MOVES A TAPE FORWARD OR BACKWARD OVER A SPECIFIED NUMBER OF HARDWARE END OF FILE MARKS.
@PACK	REWRITES A PROGRAM FILE TO EXCLUDE ELEMENTS MARKED DELETED.
@PCH	PUNCHES PROGRAM FILE ELEMENTS.
@PREP	CREATES AN ENTRY POINT TABLE FOR A PROGRAM FILE.
@PRT	LISTS DIRECTORY ITEMS FOR CATALOGUED FILES, THE TABLE OF CONTENTS OF A PROGRAM FILE, OR THE TEXT OF A SYMBOLIC ELEMENT.
@REWIND	REWINDS TAPES.

14.2. STATEMENT FORMAT

THE GENERAL FORM OF A FURPUR STATEMENT IS:

```
@LABEL:COMMAND,OPTIONS SPEC1,SPEC2,...,SPECN
```

14.2.1. CONTENTS OF SPECIFICATION FIELDS.

A SPECIFICATION FIELD MAY CONTAIN A FILE NAME, A FILE NAME AND ELEMENT NAME, OR A PARAMETER VALUE DEPENDING ON THE STATEMENT AND ITS INTENDED USE. EXTERNAL OR

INTERNAL FILE NAMES MAY BE USED. THE INTERNAL NAMES $FILEA,$FILEB AND THE EXTERNAL FILE NAME DLOC$ SHOULD NOT BE USED IN RUNS CONTAINING FURPUR STATEMENTS. EXTERNAL FILE NAMES TAKE THE FORM:

```
QUAL*FILE(C)/KEY1/KEY2.
```

ELEMENT NAMES TAKE THE FORM:

```
ELEMENT/VERSION(S)
```

THE FILE NAME SHOULD BE FOLLOWED BY A PERIOD. IF THE SPECIFICATION REQUIRES AN ELEMENT NAME, IT SHOULD FOLLOW THE PERIOD. THE ELEMENT CYCLE (S), OR THE F-CYCLE (C), MAY BE EXCLUDED WHEN RELATIVE ZERO (0) IS INTENDED. SYSTEM DROPOUT RULES FOR THE QUALIFIER APPLY.

14.2.2. FILE ASSIGNMENTS

FURPUR WILL AUTOMATICALLY ATTEMPT TO ASSIGN CATALOGUED FILES NOT ASSIGNED AT THE TIME THE FURPUR STATEMENT IS ENCOUNTERED. FURPUR WILL REQUIRE EXCLUSIVE USE OF THE FILES NAMED IN MANY CASES, AND THEREFORE WILL ATTACH EXCLUSIVE USE AS NECESSARY TO FILES ASSIGNED BY THE USER. FURPUR RETURNS THE FILE TO THE ASSIGNED STATUS IT HAD PRIOR TO THE STATEMENT, EXCEPT WHEN THE FUNCTION OF THE STATEMENT ITSELF IS TO CHANGE THE STATUS (E.G. @DELETE). TEMPORARY FILES MUST BE ASSIGNED BY THE USER.

14.2.3. OPTIONS FIELD

IN GENERAL, THE OPTIONS USED VARY WITH THE STATEMENT. THE OPTIONS BELOW HAVE THE SAME MEANING FOR ALL FURPUR STATMENTS.

C	- REQUESTS THAT FURPUR EXIT NORMALLY AFTER AN ERROR CONDITION. A DIAGNOSTIC MESSAGE WILL STILL BE PRINTED. WITHOUT THE C OPTION, AN ERROR CONDITION WILL CAUSE FURPUR TO EXIT TO ERR$. THE C OPTION MAY BE USED ON ANY FURPUR STATEMENT.
A	- PROCESS ABSOLUTE ELEMENTS
R	- PROCESS RELOCATABLE ELEMENTS
S	- PROCESS SYMBOLIC ELEMENTS AND PROCEDURE ELEMENTS.

ADDITIONAL OPTIONS ARE DISCUSSED WITH THE COMMANDS TO WHICH THEY APPLY.

14.3. SHORTHAND NOTATION.

THE FILE NAME MAY BE OMITTED FROM THE SPECIFICATION FIELD ON ALL FURPUR STATMENTS. IF THE FILE NAME IS OMITTED IN SPECIFICATION 1, TPF$ IS SUBSTITUTED. IF THE FILE NAME USED IN SPECIFICATION N IS THE SAME AS THAT IN SPECIFICATION N-1 IT MAY BE OMITTED.

THE PERIOD MAY BE OMITTED ON ANY FURPUR STATEMENT NOT CONTAINING AN ASTERISK (*) OTHER THAN @CYCLE,@PRT,@COPIN(V OPTION), AND @COPOUT(V OPTION) THAT DOES NOT SPECIFY A,R, OR S OPTIONS. IF THE FILE NAME TPF$ WAS OMITTED, THE PERIOD MAY ALSO BE OMITTED ON THE @PCH STATEMENT AND @PRT STATMENT. THE PERIOD IS NOT NEEDED ON THE @PRT STATMENT IF OPTIONS (OTHER THAN C) ARE SPECIFIED.

14.4. FURPUR STATEMENTS

THE FOLLOWING PARAGRAPHS DISCUSS IN DETAIL THE STATEMENTS PROCESSED BY FURPUR.

14.4.1. @COPY

THE COPY COMMAND IS USED TO COPY A FILE OR ELEMENT FROM ONE FILE TO ANOTHER.

14.4.1.1. FORMATTING THE @COPY STATEMENT

THE @COPY STATEMENT HAS THE FOLLOWING FORMAT:

```
@COPY,OPTIONS SPEC1,SPEC2,SPEC3
```

SPEC1 IS THE INPUT FILE OR ELEMENT TO BE COPIED.

SPEC2 IS THE OUTPUT FILE TO BE COPIED INTO.

SPEC3 IS USED ONLY FOR TAPE TO TAPE COPYING OF ENTIRE FILES WITH NO OPTIONS OR WITH THE 'M' OPTION. IT SPECIFIES THE NUMBER OF INPUT FILES TO COPY TO THE OUTPUT TAPE. IF SPEC3 IS OMITTED, ONE FILE WILL BE COPIED. THE COPY IS TERMINATED REGARDLESS OF THE VALUE OF SPEC3 IF A VOID FILE IS COPIED. THE INPUT TAPE WILL BE LEFT POSITIONED FOLLOWING THE END OF FILE AFTER THE LAST FILE COPIED. THE NUMBER OF BLOCKS IN EACH FILE COPIED, AND THE NUMBER OF FILES COPIED WILL BE INDICATED BY MESSAGES.

THE OPTIONS ALLOWED ARE:

(NO OPTION) OR 'M' ONLY

THE 'M' OPTION, WHEN USED, IS VALID ONLY IF THE OUTPUT FILE IS ON TAPE. IT INDICATES THE OUTPUT TAPE IS TO BE MARKED WITH A HARDWARE END OF FILE MARK AFTER EACH NON-VOID FILE COPIED FROM THE INPUT TAPE. IN ADDITION, A SECOND END OF FILE IS WRITTEN FOLLOWING THE LAST NON-VOID FILE COPIED, AND THE TAPE IS THEN BACKSPACED ONE END OF FILE MARK.

IF THE INPUT FILE IS ON TAPE, AND THE OUTPUT FILE IS ON FASTRAND, THE BLOCKS WILL BE COPIED TO FASTRAND IN A CONTIGUOUS MANNER BEGINNING IN SECTOR 0. NOTE THAT A BLOCK SIZE NOT DIVISIBLE BY 28 WILL LEAVE A GARBAGE AREA IN THE LAST SECTOR OF THE BLOCK AS IT APPEARS ON FASTRAND. A BLOCK BUFFERING END OF FILE SENTINEL WILL BE WRITTEN IN THE SECTOR FOLLOWING THE LAST DATA BLOCK.

IF THE INPUT FILE IS ON FASTRAND AND THE OUTPUT FILE IS ON TAPE, AN ATTEMPT IS MADE TO DETERMINE THE FORMAT OF THE INPUT FILE. IF THE FILE IS CATALOGUED AND WAS CREATED BY BLOCK BUFFERING, THE DIRECTORY ITEM WILL INDICATE THE BLOCK SIZE TO BE WRITTEN ON THE OUTPUT TAPE. IF VARIABLE BLOCK SIZE IS INDICATED, THE CONTROL WORDS THAT INDICATE BLOCK LENGTH ARE STRIPPED BEFORE THE BLOCKS ARE WRITTEN TO TAPE, TO MAINTAIN COMPATIBILITY WITH BLOCK BUFFERING. IF THE FILE IS NOT CATALOGUED, OR WAS NOT CREATED BY BLOCK BUFFERING, TRACK SIZE BLOCKS WILL BE WRITTEN.

V (AND 'M') - A FILE IS TO BE COPIED. SPEC1 NAMES THE INPUT FILE AND SPEC2 NAMES THE OUTPUT FILE. THE INPUT FILE AND OUTPUT FILE MAY NOT BOTH BE ON TAPE OR ON FASTRAND. IF THE INPUT FILE IS ON FASTRAND, VARIABLE BLOCK SIZE WILL BE ASSUMED. THIS MEANS THE FIRST WORD OF EACH BLOCK CONTAINS THE BLOCK SIZE. THIS WORD IS STRIPPED BEFORE THE BLOCK IS WRITTEN TO TAPE TO MAINTAIN COMPATIBILITY WITH BLOCK BUFFERING. THE 'M' OPTION IS VALID ONLY WHEN THE OUTPUT FILE IS ON TAPE. IT INDICATES THAT AFTER THE FILE IS COPIED, A HARDWARE END OF FILE MARK IS TO BE WRITTEN ON THE OUTPUT TAPE. THIS IS DONE BY WRITING TWO END OF FILE MARKS AND

BACKSPACING ONE.

IF THE INPUT FILE IS ON TAPE, A WORD CONTAINING THE BLOCK SIZE WILL BE PREFIXED TO THE BLOCK TO MAINTAIN BLOCK INTEGRITY BEFORE IT IS WRITTEN ON FASTRAND.

NOTE THAT A @COPY TO OR FROM FASTRAND BEGINS IN SECTOR 0. EACH BLOCK ON FASTRAND STARTS IN A NEW SECTOR. A GARBAGE AREA EXISTS IN THE LAST SECTOR OF BLOCKS WHOSE SIZE IS NOT DIVISIBLE BY 28. THE INPUT FILE MUST BE FOLLOWED BY A HARDWARE END OF FILE IF ON TAPE.

G (AND 'M') - A FILE IS TO BE COPIED. SPEC1 NAMES THE INPUT FILE AND SPEC2 NAMES THE OUTPUT FILE. THE INPUT AND OUTPUT FILES SHOULD NOT BOTH BE ON FASTRAND OR BOTH ON TAPE. THE 'M' OPTION IS VALID ONLY IF THE OUTPUT FILE IS ON TAPE. IT INDICATES THAT AFTER THE FILE IS COPIED TWO HARDWARE END OF FILE MARKS ARE TO BE WRITTEN, AND THE TAPE BACKSPACED ONE.

IF THE INPUT FILE IS ON FASTRAND, EACH ALLOCATED TRACK BEGINNING WITH RELATIVE TRACK ZERO WILL BE PREFIXED WITH ITS RELATIVE TRACK NUMBER AND THEN WRITTEN TO TAPE (ROLLOUT FORMAT). IN THIS CASE THE @COPY IS TERMINATED BY AN ATTEMPT TO READ OUTSIDE THE FILE LIMITS. THE FIRST BLOCK WRITTEN ON TAPE WILL BE A LABEL BLOCK THAT INDICATES THE FORMAT.

IF THE INPUT FILE IS ON TAPE, THE FIRST WORD OF EACH BLOCK INDICATES THE RELATIVE TRACK THE BLOCK (MINUS THE FIRST WORD) IS TO BE COPIED ONTO ON FASTRAND. THE @COPY IS TERMINATED WHEN A HARDWARE END OF FILE MARK IS ENCOUNTERED ON THE INPUT TAPE.

THIS OPTION SUPPLIES THE USER WITH AN EFFICIENT MEANS OF SAVING AND RECREATING A FASTRAND FILE THAT CONTAINED VOIDS.

F - A FILE IS TO BE COPIED. SPEC1 NAMES THE INPUT FILE, AND SPEC2 NAMES THE OUTPUT FILE. THE INPUT FILE MUST BE IN SYSTEM DATA FORMAT (SDF). READING OF THE INPUT FILE IS TERMINATED BY AN SDF END OF FILE SENTINEL. IF THE OUTPUT FILE IS PUT ON TAPE, TWO HARDWARE END OF FILE MARKS WILL BE WRITTEN AND THE TAPE BACKSPACED ONE. BLOCK SIZES FOR FILES ON TAPE MUST BE 224 WORDS. PROGRAM FILES OR ELEMENT FILES MAY NOT BE COPIED WITH THIS OPTION.

I - THE I OPTION ADDS A FILE IN SDF FORMAT TO A PROGRAM FILE AS AN ELEMENT. SPEC1 NAMES AN INPUT FILE WHICH MUST BE SDF FORMAT. SPEC2 NAMES A FILE IN PROGRAM FILE FORMAT ON FASTRAND AND THE ELEMENT NAME TO BE ENTERED IN THE PROGRAM FILE ELEMENT TABLE. THE NEW ELEMENT WILL BE ENTERED AS A SYMBOLIC ELEMENT, CYCLE 0. THE CYCLE LIMIT IS SET TO THE SYSTEM STANDARD.

A,R,S - SELECTED ELEMENTS FROM A FILE IN PROGRAM FILE FORMAT ON FASTRAND ARE TO BE REPRODUCED IN ANOTHER FILE IN PROGRAM FILE FORMAT ON FASTRAND. SPEC1 NAMES THE INPUT FILE, AND SPEC2 NAMES THE OUTPUT FILE; OR, SPEC1 NAMES THE INPUT FILE AND INPUT ELEMENT, AND SPEC2 NAMES THE OUTPUT FILE AND OUTPUT FILE ELEMENT. ONLY NON-DELETED ELEMENTS OF THE INPUT FILE MAY BE INSERTED IN THE OUTPUT FILE. THE OPTIONS INDICATE THE ELEMENT TYPES TO BE REPRODUCED. IF NO ELEMENT NAME IS GIVEN, ALL ELEMENTS OF THE TYPES INDICATED WILL BE REPRODUCED IN THE OUTPUT FILE. ANY COMBINATION OF A,R,S MAY BE GIVEN.

P - ALL NON-DELETED ELEMENTS OF ONE FILE IN PROGRAM FILE FORMAT ARE TO BE INSERTED IN ANOTHER FILE IN PROGRAM FILE FORMAT. SPEC1 NAMES THE INPUT FILE. SPEC2 NAMES THE OUTPUT FILE.

WHEN REPRODUCING ELEMENTS VIA THE A,R,S, OR P OPTIONS, THE RELATED PROCEDURE NAME

ENTRIES ARE ADDED TO THE OUTPUT FILE'S PROCEDURE NAME TABLE ENTRIES. IF A RELOCATABLE ELEMENT IS REPRODUCED THE OUTPUT FILE'S ENTRY POINT TABLE WILL BE DESTROYED. A @PREP STATEMENT CAN BE USED TO RECREATE THE ENTRY POINT TABLE WHEN NECESSARY.

14.4.1.2. EXAMPLES OF THE @COPY STATEMENT.

T,T1,T2,... WILL BE USED TO INDICATE TAPE FILES, F,F1,F2,... WILL BE USED TO INDICATE FASTRAND FILES.

SOME TYPICAL @COPY STATEMENTS ARE GIVEN BELOW.

(1) @COPY F1.,F2.

THE FASTRAND FILE F1 IS COPIED INTO THE FASTRAND FILE NAMED F2.

(2) @COPY,M T1.,T2.,9

9 FILES ON TAPE WITH FILE NAME T1 ARE COPIED ONTO TAPE WITH FILE NAME T2. EACH FILE ON T2 IS SEPARATED BY END OF FILE MARKS AS DIRECTED BY THE 'M' OPTION. THE LAST FILE ON T2 IS FOLLOWED BY 2 END OF FILE MARKS. T2 IS LEFT POSITIONED BETWEEN THE LAST TWO FILE MARKS.

(3) @COPY,GM F.,T.

FILE F IS COPIED TO TAPE IN ROLLOUT FORMAT WITH FILE NAME T. TWO HARDWARE END OF FILE MARKS ARE WRITTEN FOLLOWING THE FILE. THE TAPE IS THEN BACKSPACED ONE END OF FILE MARK. IF F WAS A PROGRAM FILE, SEQUENTIAL ACCESS OF ELEMENTS IN THE OUTPUT FILE VIA @FIND AND @COPIN ARE NOT PERMITTED SINCE THEY REQUIRE ELEMENT FILE FORMAT; HOWEVER, ROLLOUT FORMAT IS MORE ECONOMICAL OF TIME AND SPACE. NOTE THAT THE ENTIRE FILE AS IT WAS BEFORE THE COPY, INCLUDING ALL TABLES OF CONTENTS AND DELETED ELEMENTS, WILL BE REPRODUCED WHEN THE FILE IS RETURNED TO FASTRAND VIA THE 'G' OPTION. TWO PROGRAM FILES SAVED ON TAPE VIA THIS OPTION MAY NOT BE MERGED AS EACH FILE WOULD OVERLAY THE OTHER.

(4) @COPY,P F1.,F2.

THE NON-DELETED ELEMENTS OF PROGRAM FILE F1 ARE REPRODUCED IN PROGRAM FILE F2. F1 IS UNCHANGED.

(5)@COPY,I F.,F1.ELT1/VERS

THE FILE F IN SDF FORMAT IS REPRODUCED IN PROGRAM FILE F1. AN ENTRY IS CREATED IN THE ELEMENT TABLE OF F1 WITH AN ELEMENT NAME ELT1, VERSION NAME VERS, CYCLE O, WHOSE TEXT AREA CONTAINS THE CONTENTS OF F.

(6) @COPY,I T.,F1.ELT1/VERS

SAME AS (5) EXCEPT THE INPUT FILE IS ON TAPE.

(7) @COPY,RS F1.,F2.

THE NON-DELETED RELOCATABLE ELEMENTS AND SYMBOLIC ELEMENTS OF PROGRAM FILE F1 ARE REPRODUCED IN PROGRAM FILE F2. F1 REMAINS UNCHANGED.

(8) @COPY,RS F1.A,F2.B

THE SYMBOLIC AND RELOCATABLE ELEMENTS WITH NAME A IN PROGRAM FILE F1 ARE ADDED TO PROGRAM FILE F2 WITH THE NAME B. BOTH TYPES R AND S MUST EXIST IN F1 AS NON-DELETED ELEMENTS.

14.4.2. @COPOUT

COPOUT IS USED TO WRITE A PROGRAM FILE, OR SELECTED ELEMENTS OF A PROGRAM FILE TO TAPE IN ELEMENT FILE FORMAT.

14.4.2.1. FORMATTING THE @COPOUT STATEMENT.

THE FORMAT OF THE @COPOUT STATEMENT IS :

```
@COPOUT,OPTIONS SPEC1,SPEC2
```

THE @COPOUT STATEMENT IS USED TO WRITE ELEMENTS OF PROGRAM FILES ON FASTRAND ONTO TAPE IN ELEMENT FILE FORMAT. PROCEDURE NAME ENTRIES WILL BE PRESERVED. ENTRY POINTS ARE NOT PRESERVED. ELEMENT FILE FORMAT WAS DESIGNED TO REDUCE TAPE MOVEMENT WHEN IT IS NECESSARY TO READ SELECTED ELEMENTS FROM TAPE, AS OPPOSED TO TREATING A GROUP OF ELEMENTS AS A SINGLE FILE. TAPES MUST BE IN ELEMENT FILE FORMAT IN ORDER TO USE @FIND OR @COPIN.

THE AVAILABLE OPTIONS ARE:

NO OPTION - ALL NON-DELETED ELEMENTS OF A PROGRAM FILE ARE WRITTEN ONTO A TAPE IN ELEMENT FILE FORMAT. TWO HARDWARE END OF FILE MARKS ARE THEN WRITTEN ON THE TAPE AND THE TAPE IS THEN BACKSPACED OVER THE SECOND ONE. SPEC1 NAMES A PROGRAM FILE ON FASTRAND. SPEC2 CONTAINS A FILE NAME THAT REFERS TO TAPE.

A,R,S - NON-DELETED ELEMENTS OF THE TYPES SPECIFIED BY THE OPTIONS ARE WRITTEN ONTO TAPE IN ELEMENT FILE FORMAT. SPEC1 NAMES A PROGRAM FILE ON FASTRAND AND SPEC2 CONTAINS A FILE NAME THAT REFERS TO TAPE; OR, SPEC1 NAMES A PROGRAM FILE ON FASTRAND AND AN ELEMENT WITHIN THE FILE, AND SPEC2 CONTAINS A FILE NAME THAT REFERS TO TAPE AND THE NAME TO BE GIVEN TO THE ELEMENT WRITTEN TO TAPE. ANY COMBINATION OF OPTIONS MAY BE USED WHETHER APPLIED TO THE ENTIRE FILE OR TO A SINGLE ELEMENT NAME. IF NO ELEMENT NAME IS GIVEN IN SPEC1, ALL NON-DELETED ELEMENTS OF THE TYPES SPECIFIED BY THE OPTIONS ARE WRITTEN TO THE TAPE WITH THE FILE NAME GIVEN IN SPEC2. IF AN ELEMENT NAME IS INCLUDED IN SPEC1, ALL TYPES SPECIFIED BY THE OPTIONS FOR THE ELEMENT NAME GIVEN WILL BE WRITTEN TO THE TAPE WITH THE ELEMENT NAME GIVEN IN SPEC2. EACH TYPE SPECIFIED MUST REFER TO A NONDELETED ELEMENT.

V - NON-DELETED ELEMENTS OF A PROGRAM FILE SELECTED BY VERSION NAME AND TYPE ARE WRITTEN IN ELEMENT FILE FORMAT ON TAPE. THE V OPTION MAY BE USED IN COMBINATION WITH THE A,R, AND S OPTIONS. THE V OPTION ALONE IMPLIES ALL ELEMENT TYPES ARE TO BE CONSIDERED. SPEC1 NAMES A PROGRAM FILE AND AN ELEMENT VERSION NAME. SPEC2 CONTAINS THE FILE NAME OF A TAPE AND AN ELEMENT VERSION NAME. THE ELEMENTS WITH THE VERSION NAME GIVEN IN SPEC1 OF THE TYPES SPECIFIED BY THE OPTIONS WILL BE WRITTEN TO THE TAPE INDICATED BY SPEC2 IN ELEMENT FILE FORMAT. THE VERSION NAME GIVEN IN SPEC2 REPLACES THEIR ORIGINAL VERSION NAME. IF THE VERSION NAME IS OMITTED IN SPEC1, ONLY THOSE ELEMENTS WITH A BLANK VERSION NAME ARE CONSIDERED. IF THE VERSION NAME IN SPEC2 IS OMITTED, THE ELEMENTS WRITTEN TO TAPE WILL HAVE THE SAME VERSION NAME AS IN SPEC1.

14.4.2.2. EXAMPLE OF THE USE OF @COPOUT

THE @COPOUT STATMENT IS TYPICALLY USED IN THE FOLLOWING MANNER:

```
@COPOUT   PROGRAM.,HOLDPROG.
```

THE PROGRAM FILE NAMED PROGRAM WILL BE COPIED ONTO THE OUTPUT FILE HOLDPROG. IT WILL BE REFORMATTED AS AN ELEMENT FILE. THE R,S,AND A OPTIONS APPLY AS WITH THE COPY STATEMENT.

```
@COPOUT   A,TAPE
```

IN THE ABOVE EXAMPLE THE A IN SPEC1 IS PRESUMED TO BE A FILE NAME AND THE ENTIRE FILE WILL BE COPIED TO TAPE. THE FILE IS MARKED WITH AN END-OF-FILE MARK BECAUSE NO OPTIONS ARE PRESENT.

```
@COPOUT,ARS   C.,D.
```

THE CONTENTS OF FILE C ARE COPIED TO FILE D AND NO EOF MARK IS WRITTEN.

```
@COPOUT,R   A.,B.
```

IN THE ABOVE CASE ALL RELOCATABLE ELEMENTS ARE COPIED FROM FILE A TO B.

```
@COPOUT,S  A.B,C.D
```

IN THE ABOVE CASE THE 'S' OPTION INDICATES TO THE PROCESSOR THAT IT IS TO HANDLE ONE ELEMENT, OR TYPE OF ELEMENT. THAT ELEMENT WILL BE COPIED TO THE FILE NAMED IN SPEC2 ('C') AND GIVEN THE ELEMENT NAME NAMED IN SPEC FIELD 2.

```
@COPOUT   A.B,C.
```

IN THIS CASE THE ELEMENT NAME 'B' IS IGNORED AND THE ENTIRE FILE A IS COPIED TO FILE C, BECAUSE NO OPTION LETTERS ARE PRESENT. ENTRY POINTS WILL NOT BE COPIED.

```
@COPOUT,SV   A./B,C.
```

ALL SYMBOLIC ELEMENTS IN FILE A WITH A VERSION 'B' WILL BE COPIED TO FILE C.

```
@COPOUT,AV   A.,C.
```

ALL ABSOLUTE ELEMENTS IN FILE A WITH NO VERSION NAME WILL BE COPIED TO FILE C.

14.4.3. @COPIN

THE @COPIN STATEMENT IS USED TO READ ELEMENTS FROM A TAPE IN ELEMENT FILE FORMAT AND INSERT THEM IN A PROGRAM FILE ON FASTRAND. RELATED PROCEDURE NAMES ARE ENTERED IN THE PROGRAM FILE'S PROCEDURE NAME TABLE. ENTRY POINTS ARE NOT ADDED TO THE ENTRY POINT TABLE. IF A RELOCATABLE ELEMENT IS ADDED TO THE PROGRAM FILE AS A RESULT OF THE @COPIN THE ENTRY POINT TABLE WILL BE DESTROYED IF ONE EXISTED PREVIOUS TO THE @COPIN. THE @PREP STATEMENT MAY BE USED TO RECREATE THE ENTRY POINT TABLE WHEN NECESSARY. IF A TAPE ERROR OCCURS, ONLY THOSE ELEMENTS ENTERED PROPERLY PRIOR TO THE ERROR WILL APPEAR IN THE PROGRAM FILE'S TABLE OF CONTENTS.

14.4.3.1. FORMATTING THE @COPIN STATEMENT.

THE FORMAT OF THE @COPIN STATEMENT IS:

```
@COPIN,OPTIONS SPEC1,SPEC2
```

THE ALLOWABLE OPTIONS ARE:

NO OPTION - SPEC1 CONTAINS THE FILE NAME OF A TAPE POSITIONED AT THE LABEL BLOCK OF AN ELEMENT IN ELEMENT FILE FORMAT. SPEC2 NAMES A PROGRAM FILE ON FASTRAND. ELEMENTS ARE READ FROM THE TAPE NAMED BY SPEC1 AND INSERTED IN THE PROGRAM FILE NAMED BY SPEC2 UNTIL A HARDWARE END OF FILE MARK IS ENCOUNTERED.

A,R,S - SPEC1 CONTAINS THE FILE NAME OF A TAPE POSITIONED AT THE LABEL BLOCK OF AN ELEMENT IN ELEMENT FILE FORMAT. SPEC2 NAMES A PROGRAM FILE ON FASTRAND; OR,SPEC1 CONTAINS THE FILE NAME OF A TAPE AND THE NAME OF THE ELEMENT IN ELEMENT FILE FORMAT AT WHOSE LABEL BLOCK THE TAPE MUST BE POSITIONED, AND SPEC2 CONTAINS THE NAME

OF A PROGRAM FILE ON FASTRAND AND THE ELEMENT NAME TO BE GIVEN TO THE ELEMENT WHEN IT IS ADDED TO THE FILE.

V SPEC1 MUST CONTAIN THE FILE NAME OF A TAPE POSITIONED AT THE LABEL BLOCK OF AN ELEMENT IN ELEMENT FILE FORMAT, AND THE VERSION NAME OF THE ELEMENTS TO BE ADDED TO THE PROGRAM FILE ON FASTRAND NAMED BY SPEC2. SPEC2 ALSO CONTAINS THE VERSION NAME TO BE GIVEN THE ELEMENTS ADDED. ELEMENTS NOT HAVING THE SAME VERSION NAME AS GIVEN IN SPEC1 ARE SKIPPED. THE @COPIN IS TERMINATED WHEN A HARDWARE END OF FILE IS ENCOUNTERED. IF ONLY THOSE ELEMENTS WITH NO VERSION NAME ARE TO BE ADDED, THE VERSION SUBFIELD MAY BE OMITTED IN SPEC1. IF THE VERSION SUBFIELD IS OMITTED IN SPEC2, THE ELEMENTS WILL RETAIN THEIR SAME VERSION NAME. THE V OPTION MAY BE USED IN ANY COMBINATION WITH THE A,R,AND S OPTIONS, IF ONLY SELECTED TYPES ARE TO BE ADDED.

14.4.3.2. EXAMPLES OF THE @COPIN STATMENT

THE @COPIN STATMENT IS TYPICALLY USED IN THE FOLLOWING MANNER:

```
@COPIN   HOLDPROG.,PROGRAM.
```

IN THIS EXAMPLE, THE ELEMENT FILE HOLDPROG IS COPIED AND REFORMATTED ON THE FASTRAND AREA ASSIGNED TO FILE PROGRAM. WHEN THE @COPIN OPERATION IS COMPLETE, FILE PROGRAM WILL BE IN THE STANDARD PROGRAM FILE FORMAT AND MAY BE TREATED AS A PROGRAM FILE IN ANY SUBSEQUENT OPERATION.

```
@COPIN,R   TEMP.ELTA,PF1.
```

THE ABOVE COMMAND CAUSES THE RELOCATABLE ELEMENT ELTA TO BE READ FROM THE ELEMENT FILE TEMP AND ADDED TO THE PROGRAM FILE PF1. THE ELEMENT FILE MUST BE POSITIONED AT ELTA (E.G. WITH FIND). THE ENTRY POINT TABLE IS NOT UPDATED (THIS MAY BE DONE WITH PREP).

```
@COPIN,RV   A./B,C.
```

ALL RELOCATABLE ELEMENTS IN FILE A WITH A VERSION 'B' WILL BE COPIED TO FILE C.

```
@COPIN,SV   A.,C.
```

ALL SYMBOLIC ELEMENTS IN FILE A WITH NO VERSION NAME WILL BE COPIED TO FILE C.

14.4.4. @DELETE

THE @DELETE COMMAND MAY BE USED TO DROP A CATALOGUED FILE OR TO MARK AN ELEMENT OF A PROGRAM FILE ON FASTRAND DELETED. THE EFFECT OF THE @DELETE COMMAND FOR CATALOGUED FILES IS THE SAME AS THE SEQUENCE:

```
@ASG,AY FILEA
@FREE,D FILEA
```

WHEN USED TO MARK A PROGRAM FILE ELEMENT DELETED, THE ELEMENT ENTRY AND ITS RELATED PROCEDURE NAMES ARE MARKED DELETED. THE ELEMENT MAY BE REMOVED SUBSEQUENTLY BY USING A @PACK STATEMENT.

14.4.4.1. FORMATTING THE @DELETE STATEMENT

THE @DELETE STATEMENT HAS THE FOLLOWING FORMAT:

```
@DELETE,OPTIONS SPEC1,SPEC2,... SPECN
```

THE AVAILABLE OPTIONS ARE:

(NO OPTIONS) - A CATALOGUED FILE IS TO BE DROPPED. EACH SPECIFICATION FIELD NAMES A CATALOGUED FILE. THE FILE NAMES GIVEN MAY BE EXTERNAL OR INTERNAL.

IF AN EXTERNAL NAME IS GIVEN THE F-CYCLE TO BE DROPPED SHOULD BE SPECIFIED. THE LATEST CYCLE IS UNDERSTOOD IF NONE IS GIVEN. SECURITY KEYS MUST BE GIVEN IF THE FILE WAS CATALOGUED WITH KEYS AND THE FILE IS TO BE ASSIGNED BY FURPUR. THE KEYS WILL BE IGNORED IF THE FILE IS ALREADY ASSIGNED.

IF AN INTERNAL NAME IS GIVEN THE EXTERNAL NAME ON THE ASSOCIATED @USE STATEMENT MUST SATISFY THE SAME RULES AS IF IT HAD APPEARED ON THE STATEMENT ITSELF. NOTE THAT WHEN THE FILE IS ACTUALLY DROPPED FROM THE DIRECTORY OLDER F-CYCLES WILL HAVE THEIR RELATIVE F-CYCLE VALUE INCREASED ONE. THE FILE IS NOT ACTUALLY DROPPED FROM THE DIRECTORY UNTIL ALL ASSIGNMENTS MADE BEFORE THE DROP FLAG WAS SET HAVE BEEN @FREE'D.

A,R,S - AN ELEMENT OF A PROGRAM FILE IS TO BE DELETED. EACH SPECIFICATION FIELD GIVEN NAMES AN ELEMENT AND THE PROGRAM FILE THAT CONTAINS IT. THE TYPES GIVEN BY THE OPTIONS WILL BE MARKED DELETED. ANY COMBINATION OF A,R, AND S OPTIONS MAY BE USED, BUT AT LEAST ONE MUST BE GIVEN. @DELETE REQUIRES EACH ELEMENT AS SPECIFIED BY NAME AND OPTIONS ON THE STATEMENT EXIST IN A NON-DELETED STATE PRIOR TO THE @DELETE STATEMENT. INCLUDING A CYCLE NUMBER FOR SYMBOLIC ELEMENTS IS ILLEGAL ON THE @DELETE STATEMENT. ASSOCIATED ENTRY POINTS AND PROCEDURE NAMES WILL ALSO BE MARKED DELETED.

14.4.4.2. EXAMPLES OF THE @DELETE STATEMENT.

(1) @DELETE,S F.ELT1/VERS,F1.ELTY

THE SYMBOLIC ELEMENT ELT1/VERS IN PROGRAM FILE F, AND THE SYMBOLIC ELEMENT ELTY IN PROGRAM FILE F1 WILL BE MARKED DELETED. ASSOCIATED PROCEDURE NAMES IF ANY WILL ALSO BE MARKED DELETED.

(2) @DELETE F.,T1.

THE CATALOGUED FILES F AND T1 ARE DROPPED FROM THE DIRECTORY (ONLY THE F-CYCLE SPECIFIED).

14.4.5. @PRT

PRT IS USED TO PRINT THE TABLE OF CONTENTS OF A PROGRAM FILE, OR TO PRINT THE MASTER FILE DIRECTORY ITEM OR ITEMS OF CATALOGUED FILES.

14.4.5.1. FORMATTING THE @PRT STATEMENT

THE @PRT STATEMENT HAS THE FOLLOWING FORMAT

```
@PRT,OPTIONS SPEC1,SPEC2,...,SPECN
```

THE AVAILABLE OPTIONS ARE :

T - LIST THE TABLE OF CONTENTS OF A PROGRAM FILE ON FASTRAND. EACH SPECIFICATION FIELD GIVEN NAMES A PROGRAM FILE WHOSE TABLE OF CONTENTS IS TO BE LISTED.

(NO OPTION) - IF NO SPECIFICATION FIELDS ARE GIVEN, THE ENTIRE DIRECTORY WILL BE LISTED. THE DIRECTORY ITEMS LISTED WILL NOT INCLUDE SECURITY KEYS. ITEMS ARE LISTED IN ALPHABETIC ORDER, FIRST BY PROJECT, THEN BY ACCOUNT, AND THEN BY QUALIFIER AND FILE NAME.

IF AT LEAST ONE SPECIFICATION FIELD IS GIVEN, A SYMBOLIC ELEMENT OF A PROGRAM FILE WILL BE LISTED WITH LINE NUMBERS. EACH SPECIFICATION FIELD GIVEN SHOULD CONTAIN THE NAME OF A PROGRAM FILE ON FASTRAND AND THE NAME OF A SYMBOLIC ELEMENT WITHIN THE FILE. AN ELEMENT CYCLE MAY BE SPECIFIED. IF NONE IS SPECIFIED, THE LATEST CYCLE IS LISTED.

F - LIST THE DIRECTORY ITEM FOR A CATALOGUED FILE. EACH SPECIFICATION FIELD NAMES A CATALOGUED FILE. SECURITY KEYS ARE NOT LISTED.

P - LIST THE DIRECTORY ITEMS FOR A PROJECT. SECURITY KEYS ARE NOT LISTED. EACH SPECIFICATION FIELD GIVEN NAMES A PROJECT. IF NO SPECIFICATION FIELDS ARE GIVEN THE ENTIRE DIRECTORY IS LISTED IN ALPHABETIC ORDER FIRST BY PROJECT, THEN BY ACCOUNT, AND THEN BY QUALIFIER AND FILE NAME.

N - LIST THE DIRECTORY ITEMS FOR ALL FILES CATALOGUED WITH THE ACCOUNT NUMBER SPECIFIED. SECURITY KEYS ARE NOT LISTED. EACH SPECIFICATION FIELD NAMES AN ACCOUNT. IF NO SPECIFICATION FIELDS ARE GIVEN, THE ENTIRE DIRECTORY IS LISTED IN ALPHABETIC ORDER BY ACCOUNT, THEN BY PROJECT, THEN BY QUALIFIER AND FILE NAME.

14.4.5.2. EXAMPLES OF THE @PRT STATEMENT

(1) PRT,T PROGFILE

THE TABLE OF CONTENTS OF THE PROGRAM FILE ON FASTRAND PROGFILE ARE LISTED. THE TABLES LISTED INCLUDE THE ELEMENT TABLE, PROCEDURE NAME TABLES, AND THE ENTRY POINT TABLE. ELEMENT TABLE ENTRIES ARE LISTED IN THE ORDER THE ELEMENTS WERE INTRODUCED INTO THE FILE. OTHER TABLES ARE LISTED IN ALPHABETIC ORDER.

(2) @PRT PROGFILE.SAM/XYZ

THE LATEST CYCLE OF THE ELEMENT SAM,VERSION XYZ, IN PROGRAM FILE PROGFILE IS LISTED.

(3) @PRT,P MERCURY

THE DIRECTORY ITEMS FOR THOSE CATALOGUED FILES WHOSE PROJECT NAME IS MERCURY ARE LISTED.

14.4.6. @PCH

THE @PCH STATEMENT IS USED TO PUNCH PROGRAM FILE ELEMENTS ON 80-COLUMN CARDS.

14.4.6.1. FORMATTING THE @PCH STATEMENT

THE FORMAT OF THE @PCH STATEMENT IS:

```
@PCH,OPTIONS SPEC1,SPEC2
```

THE ALLOWABLE OPTIONS FOR THE @PCH STATEMENT ARE :

A,R,S - AN ELEMENT OF A PROGRAM FILE IS TO BE PUNCHED ON 80 COLUMN CARDS. SPEC1 NAMES A PROGRAM FILE ON FASTRAND AND THE NAME OF THE ELEMENT TO BE PUNCHED. THE OPTIONS DESIGNATE WHICH TYPES OF THE ELEMENT ARE TO BE PUNCHED. ANY COMBINATION OF A,R,S MAY BE GIVEN, BUT AT LEAST ONE OPTION MUST BE GIVEN. THE ELEMENT, BY NAME AND TYPES GIVEN, MUST EXIST IN A NON-DELETED STATE.

THE ELEMENTS PUNCHED WILL CONTAIN THE NECESSARY CONTROL CARDS TO REINSERT THEM IN A PROGRAM FILE. THE FIRST CARD OF PROCEDURE ELEMENTS WILL BE A @PDP CONTROL CARD.

THE FILE NAME REFERRED TO ON THE CONTROL CARD WILL BE THE SAME AS THE FILE FROM WHICH THE ELEMENT IS PUNCHED.

SEVERAL OTHER OPTIONS ARE AVAILABLE THAT APPLY ONLY TO SYMBOLIC ELEMENTS. THE S OPTION MUST BE INCLUDED OR THEY WILL BE IGNORED. THE OPTIONS ARE:

G - THE INPUT CARD IMAGES ARE TO BE COMPRESSED.

H - THE INPUT IMAGES ARE TO HAVE COLUMNS 72 THROUGH 80 OVERLAYED WITH A SEQUENCE NUMBER. SPEC2 SHOULD CONTAIN UP TO THREE ALPHANUMERIC CHARACTERS. THE CHARACTERS WILL BE LEFT ADJUSTED AND OVERLAY COLUMNS 72 THROUGH 75 OF THE INPUT IMAGES.

J - THE INPUT IMAGES ARE TO BE COMPRESSED. THE OUTPUT IMAGES ARE TO BE SEQUENCED IN COLUMNS 72 THROUGH 80.

THE G AND J OPTIONS MAY NOT BOTH APPEAR ON THE SAME STATEMENT. SEQUENCE NUMBERS ARE 100 (DECIMAL) APART. RELOCATABLE AND ABSOLUTE ELEMENTS ARE SEQUENCED AUTOMATICALLY IN COLUMNS 79,80. THE SEQUENCING STARTS WITH AA AND ENDS WITH ZZ. IT IS REPEATED IF NECESSARY. THE COMPRESSED FORM OF PUNCHED OUTPUT USES A SPACE COUNT TO STRIP SPACES FROM THE INPUT IMAGES.

14.4.6.2. EXAMPLES OF THE @PCH STATEMENT

(1) @PCH,S TPF$.RUNPROG

THE SYMBOLIC ELEMENT RUNPROG IN THE PROGRAM FILE TPF$ IS PUNCHED ON 80-COLUMN CARDS, ONE IMAGE PER CARD.

(2) @PCH,SRJH A.B,XYZ

THE SYMBOLIC ELEMENT B IN PROGRAM FILE A IS PUNCHED ON 80-COLUMN CARDS. THE INPUT IMAGES ARE SEQUENCED IN COLUMNS 76-80. THE IDENTIFICATION FIELD XYZ IS PLACED IN COLUMNS 73-75. THE SEQUENCED IMAGES ARE COMPRESSED IN COLUMNS 1-72 ON THE PUNCHED CARDS, AND COLUMNS 73-80 ARE THEN SEQUENCED. THE RELOCATABLE ELEMENT B IN PROGRAM FILE B IS PUNCHED. THE TEXT HAS BEEN PREVIOUSLY SEQUENCED BY THE ASSEMBLER. FURPUR SEQUENCES THE PREAMBLE. IF THE SYMBOLIC ELEMENT WAS A PROCEDURE ELEMENT, THE DECK SHOULD BE USED AS INPUT TO @PDP, OTHERWISE TO @ELT. THE RELOCATABLE DECK MAY BE USED AS INPUT TO @ELT.

14.4.7. @CHG

THE CHANGE COMMAND MAY BE USED TO ALTER THE NAME OF CATALOGUED FILES (BOTH PROGRAM AND DATA FILES), OR TO CHANGE THE NAME OF ELEMENTS IN A PROGRAM FILE. THE FILE MUST BE CATALOGUED OR ASSIGNED TO THE RUN.

THE FORMAT OF THE @CHG STATEMENT IS:

@CHG,(OPTIONS) SPEC1,SPEC2

14.4.7.1. CATALOGUED FILES

ONE F-CYCLE SERIES EXISTS FOR EACH SET OF FILES WITH THE SAME QUALIFIER AND EXTERNAL NAME. EACH CATALOGUED FILE BELONGS TO ONE AND ONLY ONE F-CYCLE SERIES. SECURITY KEYS, IF ANY, ARE THE SAME FOR ALL MEMBERS OF THE SERIES. THE DIRECTORY CONTAINS A MAIN ITEM FOR EACH F-CYCLE SERIES THAT LISTS THE SECURITY KEYS FOR THE SERIES AND POINTS TO A FILE ITEM FOR EACH MEMBER OF THE SERIES. THE READ ONLY MODE AND WRITE ONLY MODE INDICATORS ARE KEPT IN THE FILE ITEM FOR THAT MEMBER.

THE @CHG STATEMENT MAY BE USED TO PERFORM THE FOLLOWING FUNCTIONS RELATED TO CATALOGUED FILES:

1) CHANGE THE SECURITY KEYS FOR ALL FILES OF A GIVEN F-CYCLE SERIES.
2) REMOVE OR SET READ-ONLY OR WRITE-ONLY MODES ON A FILE.
3) REMOVE A FILE FROM AN F-CYCLE SERIES AND ADD IT TO ANOTHER

SERIES AS THE LATEST F-CYCLE.

IF AN F-CYCLE SERIES CONTAINS ONLY ONE MEMBER, 1) IS EQUIVALENT TO CHANGING THE KEYS FOR A FILE, AND 3) IS EQUIVALENT TO CHANGING THE NAME OF A FILE.

SPEC1 NAMES A CATALOGUED FILE. READ ONLY AND WRITE ONLY MODES WILL BE REMOVED FROM THE FILE. TWO OPTIONS ARE AVAILABLE TO RESET THE MODES:

V	-	SET READ-ONLY MODE
W	-	SET WRITE-ONLY MODE

THE CONTENTS OF SPEC2 DETERMINE WHAT OTHER FUNCTONS ARE TO BE PERFORMED. SPEC2 CONTAINS A FILE NAME. THE FUNCTIONS PERFORMED ARE GIVEN ACCORDING TO CASES.

CASE 1 THE QUALIFIER AND EXTERNAL NAME SPECIFIED BY SPEC2 ARE THE SAME AS FOR THE FILE NAMED IN SPEC1.

THE SECURITY KEYS FOR THE F-CYCLE SERIES CONTAINING THE FILE NAMED IN SPEC1 WILL BE CHANGED TO THOSE GIVEN IN SPEC2.

IF THE FILE NAMED IN SPEC1 IS NOT THE LATEST F-CYCLE OF THE SERIES IT WILL BE REMOVED FROM THE SERIES AND RECATALOGUED AS THE LATEST.

CASE 2 SPEC2 NAMES A CATALOGUED FILE IN THE DIRECTORY WITH A DIFFERENT QUALIFIER AND/OR EXTERNAL NAME THAN THE FILE NAMED IN SPEC1.

THE FILE NAMED IN SPEC1 WILL BE REMOVED FROM THE SERIES IT CURRENTLY BELONGS TO AND RECATALOGUED AS THE LATEST F-CYCLE OF THE SERIES TO WHICH THE FILE NAMED IN SPEC2 BELONGS.

CASE 3 THE FILE NAME IN SPEC2 DOES NOT CONTAIN THE QUALIFIER AND EXTERNAL NAME OF AN F-CYCLE SERIES CURRENTLY IN THE DIRECTORY. THE FILE NAMED IN SPEC1 WILL BE REMOVED FROM THE SERIES IT CURRENTLY BELONGS TO AND RECATALOGUED UNDER THE NEW NAME.

ALTHOUGH THE FUNCTIONS PERFORMED BY @CHG DO NOT INCLUDE READING OR WRITING IN TEXT AREAS OF THE FILES NAMED, SECURITY KEYS, IF THE FILES HAVE ANY, ARE REQUIRED IN ORDER FOR @CHG TO MODIFY THEIR DIRECTORY ITEMS. THIS MEANS THAT THE FILE NAME ON THE FIRST @ASG STATEMENT GIVEN TO THE EXECUTIVE MUST INCLUDE THE KEYS IF AN EXTERNAL NAME IS USED. IF AN INTERNAL NAME IS USED, IT MUST BE ASSOCIATED BY A @USE STATEMENT STILL IN EFFECT THAT INCLUDES THE KEYS. FURPUR WILL PERFORM THE INITIAL ASSIGNMENT, IF THE USER HAS NOT ASSIGNED THE FILE. IN THIS CASE THE SAME RULES APPLY TO THE NAME FURNISHED ON THE @CHG STATEMENT AS FOR THE @ASG STATEMENT FURNISHED BY THE USER.

14.4.7.2. PROGRAM FILE ELEMENTS

THE @CHG STATEMENT MAY BE USED TO CHANGE THE ELEMENT NAME OF A PROGRAM FILE ELEMENT. ONE OR MORE OF THE A,R, AND S OPTIONS MUST BE SPECIFIED. SPEC1 NAMES A PROGRAM FILE AND AN ELEMENT NAME. SPEC2 NAMES THE SAME PROGRAM FILE AND THE NEW ELEMENT NAME. THE ELEMENT NAME SUBFIELDS INCLUDE THE ELEMENT NAME AND VERSION. ELEMENT CYCLE MAY NOT BE SPECIFIED. ONLY THOSE TYPES SPECIFIED BY THE OPTIONS WILL HAVE THEIR NAMES CHANGED.

14.4.7.3. EXAMPLES OF THE @CHG STATEMENT

```
@CHG,S   FILE1.ELT2,FILE1.ELT5/VERS3
```

CHANGE SYMBOLIC ELEMENT NAME 'ELT2' OF PROGRAM FILE 'FILE1', TO ELEMENT NAME 'ELT5' AND ADD A VERSION NAME 'VERS3'.

```
@CHG,R   FILE1/KEY1/KEY2
```

CHANGE THE MODE OF CATALOGUED FILE 'FILE1' FROM ITS PRESENT STATE TO A READ ONLY MODE.

```
@CHG   FILE1/KEY1/KEY2,FILE1/KEYA
```

CHANGE THE READ KEY FROM 'KEY1' TO 'KEYA', AND DROP THE WRITE KEY.

```
@CHG   FILE1(-2)/KEY1/KEY2,FILE2/KEY3/KEY4
```

FILE1 MUST BE A CATALOGUED FILE. IF FILE2 IS A CATALOGUED FILE, RELATIVE F-CYCLE (-2) OF FILE1 WILL BECOME THE LATEST F-CYCLE OF FILE2, AND F-CYCLE (2) WILL BE DELETED FROM FILE1. ALL F-CYCLES OF FILE1 OLDER THAN (-2) WILL BE MOVED UP.

IF FILE2 IS NOT CATALOGUED, A DYNAMIC @ASG WILL BE MADE BY FURPUR TO CATALOGUE THE FILE WITH THE NAME AND KEYS GIVEN IN SPEC2. THE F-CYCLE SPECIFIED IN SPEC1 WILL BE RELATIVE F-CYCLE (0) OF FILE 2.

14.4.8. @PAC^K

THE @PACK STATEMENT IS USED TO REWRITE AN ENTIRE PROGRAM FILE SO AS TO EXCLUDE DELETED ELEMENTS AND THEIR ASSOCIATED ENTRIES IN THE TABLE OF CONTENTS. THE ENTRY POINT TABLE IS DESTROYED.

THE FORMAT OF THE @PACK STATEMENT IS :

```
@PACK SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD GIVEN MUST NAME A PROGRAM FILE ON FASTRAND TO BE @PACK'ED. ALL GRANULES NO LONGER NEEDED TO CONTAIN THE FILE ARE RELEASED TO THE SYSTEM.

14.4.9. @PREP

THE @PREP STATEMENT IS USED TO CREATE AN ENTRY POINT TABLE FROM THE PREAMBLES OF THE NON-DELETED ELEMENTS OF A PROGRAM FILE. IF A PREVIOUS ENTRY POINT TABLE EXISTED, IT IS DESTROYED PRIOR TO CREATING THE NEW ONE. THE ENTRY POINT TABLE IS REQUIRED IN SOME CASES FOR USE BY THE COLLECTOR. THE ENTRY POINT TABLE MAY BE LISTED BY @PRT,T.

THE FORMAT OF THE @PREP STATEMENT IS:

```
@PREP SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD GIVEN SHOULD NAME A PROGRAM FILE ON FASTRAND TO BE @PREP'ED.

14.4.10. @ERS

THE @ERS STATEMENT IS USED TO RETURN ALL GRANULES ALLOCATED TO A PROGRAM FILE BACK TO THE SYSTEM.

THE FORMAT OF THE @ERS STATEMENT IS:

```
@ERS SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD GIVEN SHOULD NAME A PROGRAM FILE ON FASTRAND TO BE @ERS'ED.

14.4.11. @REWIND.

THE @REWIND STATEMENT IS USED TO REWIND TAPES.
THE FORMAT OF THE @REWIND STATEMENT IS:

```
@REWIND,OPTION SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION GIVEN SHOULD GIVE THE NAME OF A TAPE FILE. EACH TAPE REFERRED

TO BY FILE NAME WILL BE REWOUND. THE ONLY OPTION ALLOWED ON THIS STATMENT IS THE I OPTION. IF THE I OPTION IS PRESENT, THE TAPE WILL BE REWOUND WITH INTERLOCK. THE EXECUTIVE ROUTINE TINTL$ IS UTILIZED BY THIS FUNCTION TO RETURN TO THE FIRST REEL OF TAPE ASSIGNED THE GIVEN FILE NAME IF NECESSARY.

14.4.12. @MARK

THE @MARK STATEMENT IS USED TO WRITE A HARDWARE END OF FILE MARK ON MAGNETIC TAPE. THE FUNCTION IS ACCOMPLISHED BY WRITING TWO END OF FILE MARKS AND BACKSPACING OVER THE SECOND ONE.

THE FORMAT OF THE @MARK STATEMENT IS :

```
@MARK SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD GIVEN MUST CONTAIN A FILE NAME THAT REFERS TO TAPE.

14.4.13. @CLOSE.

THE @CLOSE STATEMENT IS USED TO WRITE TWO HARDWARE END OF FILE MARKS ON A TAPE AND THEN REWIND THE TAPE.

THE FORMAT OF THE @CLOSE STATEMENT IS:

```
@CLOSE,OPTION SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD GIVEN MUST CONTAIN A FILE NAME THAT REFERS TO TAPE. THE I OPTION IS THE ONLY OPTION ALLOWED. THE I OPTION INDICATES THE TAPE IS TO BE REWOUND WITH INTERLOCK.

14.4.14. @MOVE.

THE @MOVE STATEMENT IS USED TO MOVE A MAGNETIC TAPE OVER A SPECIFIED NUMBER OF FILE MARKS.

THE FORMAT OF THE MOVE STATEMENT IS;

```
@MOVE,OPTION SPEC1,N
```

SPEC1 MUST CONTAIN A FILE NAME THAT REFERS TO TAPE. N IS THE NUMBER OF EOF MARKS TO MOVE PAST. B IS THE ONLY OPTION AVAILABLE WITH THE @MOVE STATMENT. THE TAPE IS MOVED FORWARD WITHOUT THE B OPTION, BACKWARD WITH THE B OPTION.

14.4.15. @FIND

THE @FIND STATEMENT IS USED TO LOCATE AN ELEMENT ON TAPE IN ELEMENT FILE FORMAT AND POSITION THE TAPE IMMEDIATELY BEFORE THE LABEL BLOCK OF THE ELEMENT. @FIND SEARCHES FORWARD UNTIL THE ELEMENT IS FOUND, OR UNTIL AN END OF FILE IS ENCOUNTERED. IN THE LATTER CASE, THE TAPE IS BACKSPACED TO THE PREVIOUS END OF FILE MARK AND THE SEARCH IS REPEATED. ENCOUNTERING AN END OF FILE THIS SECOND PASS IS AN ERROR EXIT FOR THE FUNCTION. NORMALLY THE @FIND STATEMENT IS USED JUST PRIOR TO A @COPIN STATEMENT REQUESTING THE ELEMENT JUST LOCATED TO BE INSERTED IN A PROGRAM FILE ON FASTRAND, OR TO INSERT ALL THOSE READ UP TO THE NEXT HARDWARE END OF FILE MARK.

THE FORMAT OF THE @FIND STATEMENT IS :

```
@FIND,OPTION SPEC1
```

ONE AND ONLY ONE OF THE OPTIONS A,R,S MUST BE GIVEN TO SPECIFY THE TYPE OF ELEMENT TO BE LOCATED. SPEC1 NAME A FILE ON TAPE AND THE ELEMENT TO BE LOCATED.

14.4.16. @CYCLE.

THE @CYCLE STATEMENT IS USED TO SET THE MAXIMUM NUMBER OF F-CYCLES TO BE RETAINED FOR A GIVEN FILE NAME EXISTING IN THE DIRECTORY, OR TO SET THE MAXIMUM NUMBER OF CYCLES TO BE MAINTAINED FOR A PROGRAM FILE SYMBOLIC ELEMENT ON FASTRAND. PROCEDURE ELEMENTS MAY NOT HAVE THEIR MAXIMUM CHANGED (ONE CYCLE). THE ORIGINAL MAXIMUMS FOR F-CYCLES AND ELEMENT CYCLES ARE SYSTEM STANDARDS.

14.4.16.1. FORMATTING THE @CYCLE STATEMENT

THE FORMAT OF THE @CYCLE STATEMENT IS:

```
@CYCLE SPEC1,N
```

THE FUNCTION PERFORMED IS DETERMINED BY THE CONTENTS OF SPEC1. THERE ARE TWO CASES:

(1) SPEC1 NAMES A CATALOGUED FILE. IN THIS CASE THE STATEMENT APPLIES TO THE NUMBER OF F-CYCLES. THE FILE NAME AND THE F-CYCLE SPECIFIED MUST BOTH BE IN THE DIRECTORY. N SPECIFIES THE NEW MAXIMUM NUMBER OF F-CYCLES TO BE RETAINED. IF N IS 0 THE ENTIRE F-CYCLE SERIES AND THE FILE NAME ENTRY WILL BE DROPPED. IF THE CURRENT NUMBER OF F-CYCLES FOR THIS FILE NAME IS GREATER THAN THE NEW MAXIMUM, THE DROP FLAG WILL BE SET IN THOSE F-CYCLES,STARTING WITH THE OLDEST, NECESSARY TO SATISFY THE NEW MAXIMUM. BOTH SECURITY KEYS, IF THE FILE WAS CATALOGUED WITH BOTH, MUST BE GIVEN WHEN THE FILE IS ORIGINALLY ASSIGNED TO THE RUN. NONE OF THE F-CYCLES MAY CURRENTLY BE ASSIGNED EXCLUSIVELY TO ANOTHER RUN, OR AN ERROR EXIT WILL BE TAKEN.

THE NUMBER OF F-CYCLES IS TAKEN TO BE THE RANGE OF THE NEWEST AND OLDEST ABSOLUTE F-CYCLES.

(2) SPEC1 CONTAINS THE NAME OF A PROGRAM FILE ON FASTRAND AND THE NAME OF A SYMBOLIC ELEMENT IN THE FILE OTHER THAN A PROCEDURE ELEMENT, THAT IS TO HAVE ITS MAXIMUM CHANGED FOR NUMBER OF CYCLES TO BE RETAINED. N IS THE NEW MAXIMUM NUMBER TO RETAIN. IF THE NUMBER OF CYCLES CURRENTLY RETAINED IS ACTUALLY GREATER THAN THE NEW MAXIMUM, A NEW ELEMENT WITH THE SAME NAME WILL BE CREATED WITH THOSE OLDEST CYCLES OF THE ELEMEMT ELIMINATED NECESSARY TO SATISFY THE NEW MAXIMUM. THE OLD ELEMENT IS MARKED DELETED.

14.4.16.2. EXAMPLES OF THE @CYCLE STATEMENT

(1)@CYCLE Q*A.B,2

SUPPOSE THE SYMBOLIC ELEMENT B IN PROGRAM FILE Q*A ORIGINALLY CONTAINED 4 CYCLES, CYCLES 5,6,7,8. THE NEW MAXIMUM REQUIRES THAT A NEW ELEMENT B CONTAINING ONLY CYCLES 7 AND 8 BE CREATED. THE OLD ELEMENT B IS DELETED.

IF THE NEW LIMIT WERE 5, ONLY THE PARAMETER FIELD OF THE ELEMENT ENTRY WOULD BE CHANGED. NO NEW ELEMENT WOULD BE CREATED.

(2)@CYCLE Q*A.,2

SUPPOSE THE DIRECTORY ENTRY FOR THE FILE Q*A INDICATES THAT 4 F-CYCLES EXIST WITH THE NAME Q*A, THEN THE DROP FLAG IN F-CYCLES (-3) AND (-2) WILL BE SET.

14.4.17. @ENABLE

THE @ENABLE STATEMENT IS USED TO REMOVE THE DISABLE FLAG, WHICH MAY HAVE BEEN SET BY THE EXECUTIVE AS THE RESULT OF SOME TYPE OF MALFUNCTION, FROM A CATALOGUED FILE IN THE DIRECTORY. UNDER NORMAL OPERATION THE USER WILL NOT NEED THE @ENABLE STATEMENT IN HIS RUN.

THE FORMAT OF THE @ENABLE STATEMENT IS:

```
@ENABLE SPEC1,SPEC2,...,SPECN
```

EACH SPECIFICATION FIELD SHOULD CONTAIN THE NAME OF A CATALOGUED FILE THAT IS TO BE @ENABLE'D. A MESSAGE WILL BE PRINTED IF THE FILE WAS NOT PREVIOUSLY DISABLED. A NORMAL EXIT IS TAKEN WHETHER THE FILE WAS DISABLED OR NOT.

14.5. MULTIREEL FILES

SOME PROVISIONS EXIST IN FURPUR FOR THE CREATION OF TAPE FILES THAT EXTEND OVER MORE THAN ONE REEL. THE @COPOUT FUNCTION MAKES ITS OWN CALL TO TSWAP$ IN THE EVENT AN END OF TAPE CONDITION IS ENCOUNTERED. @COPOUT WRITES A 14 WORD END OF REEL SENTINEL WHICH IS UNDERSTOOD BY @COPIN AS AN INDICATION THAT THE ELEMENT BEING READ EXTENDS ONTO A SECOND REEL. THE REWIND STATEMENT RETURNS A USER TO THE FIRST REEL OF TAPE ASSIGNED THE GIVEN FILE NAME. @COPY,F AND @COPY,I ALSO PERMIT THE READING AND WRITING OF MULTIREEL FILES. END OF REEL SENTINELS CREATED BY FUNCTIONS OTHER THAN FURPUR MAY NOT BE INTERPRETED CORRECTLY BY FURPUR. THE @MOVE STATMENT CHECKS FOR END OF REEL SENTINELS CREATED BY FURPUR.

15. AUXILIARY PROCESSORS

15.1. THE COLLECTOR

15.1.1. GENERAL

THE COLLECTOR FOR THE UNIVAC 1108 IS DESIGNED TO PROVIDE THE USER A STRAIGHT-FORWARD MEANS OF COLLECTING AND INTERCONNECTING RELOCATABLE ELEMENTS TO PRODUCE A PROGRAM WHICH IS IN A FORM READY FOR EXECUTION UNDER CONTROL OF THE 1108 EXECUTIVE SYSTEM. THIS PROGRAM FORM IS CALLED AN ABSOLUTE ELEMENT. INTERNAL REFERENCES ARE LINKED TOGETHER AND NO MODIFICATION IS NECESSARY TO LOAD THE PROGRAM ANYWHERE IN CORE. OPTIONALLY, THE COLLECTOR CAN BE USED TO PRODUCE ONE RELOCATABLE ELEMENT FROM A COLLECTION OF SEVERAL RELOCATABLE ELEMENTS.

THE COLLECTOR IS CONCERNED WITH THREE BASIC INPUTS. THESE ARE:

1. PARAMETERS FROM THE EXECUTIVE CONTROL STATEMENT CAUSING THE COLLECTION.
2. SOURCE LANGUAGE CONTROL STATEMENTS.
3. RELOCATABLE ELEMENTS FROM A VARIABLE NUMBER OF SOURCES.

ALL OF THESE INPUTS ARE DISCUSSED IN DETAIL LATER; HOWEVER, A BRIEF DESCRIPTION OF EACH IS GIVEN HERE FOR INTRODUCTORY PURPOSES.

BASICALLY, THE COLLECTOR IS CALLED WHENEVER A @MAP EXECUTIVE CONTROL STATEMENT IS ENCOUNTERED WITHIN A CONTROL INPUT FILE. THE @MAP STATEMENT SPECIFIES THE INPUT AND OUTPUT ELEMENTS TO THE COLLECTOR AS IT DOES FOR OTHER SYSTEM PROCESSORS. THE INFORMATION CONTAINED WITHIN THE @MAP CONTROL STATEMENT IS COMPREHENSIVE ENOUGH TO DIRECT THE ALLOCATION OF MOST PROGRAMS.

FOR PERFORMING THE COLLECTION OF COMPLEX PROGRAMS WHICH REQUIRE RELOCATABLE INPUT FROM MANY SOURCES, CONSTRUCTION OF OVERLAY SEGMENTS, OR THE USE OF MULTIPLE LIBRARIES, THE USER MUST PREPARE A SET OF SOURCE LANGUAGE CONTROL STATEMENTS. THESE STATEMENTS MAY FOLLOW THE @MAP EXECUTIVE CONTROL STATEMENT OR BE CONTAINED IN AN ELEMENT IN A PROGRAM FILE.

COMPLETE CAPABILITIES ARE AVAILABLE THROUGH THE COLLECTOR FOR UPDATING THE SYMBOLIC ELEMENT IN THE PROGRAM FILE. THE PROCEDURE IS THE SAME AS FOR UPDATING ANY OTHER SOURCE LANGUAGE (FORTRAN, COBOL, ETC.) ELEMENT PROCESSED BY THE SYSTEM.

RELOCATABLE ELEMENTS TO ENTER INTO THE COLLECTION ARE INDICATED TO THE COLLECTOR BY WAY OF THE TWO INPUT SOURCES JUST DESCRIBED. RELOCATABLE ELEMENTS FROM LIBRARIES MAY BE SPECIFICALLY NAMED TO BE INCLUDED IN THE PROGRAM OR INCLUDED ONLY IF AN EXTERNAL REFERENCE IS MADE TO THE ELEMENT. GENERALLY, ALL THE RELOCATABLE ELEMENTS IN THE TEMPORARY PROGRAM FILE (TPF$.) ARE ARBITRARILY INCLUDED IN THE PROGRAM BEING COLLECTED. USE OF THE SYSTEM RELOCATABLE LIBRARY TO SATISFY EXTERNAL REFERENCES IS AUTOMATICALLY IMPLIED; THE USE OF USER LIBRARIES IS UNDER CONTROL OF A SOURCE LANGUAGE CONTROL STATEMENT TO THE COLLECTOR. ANY SPECIFIED USER LIBRARIES ARE ALWAYS SEARCHED BEFORE THE SYSTEM RELOCATABLE LIBRARY.

THE OUTPUTS OF THE COLLECTOR ARE AS FOLLOWS:

1. AN ABSOLUTE OR RELOCATABLE ELEMENT
2. A SOURCE LANGUAGE CONTROL ELEMENT AS DISCUSSED ABOVE.
3. LISTING INFORMATION

THE PRIMARY OUTPUT OF THE COLLECTOR IS THE RELOCATABLE OR ABSOLUTE ELEMENT WHICH RESULTS FROM THE COLLECTING AND LINKING OF THE VARIOUS RELOCATABLE ELEMENTS. THIS ELEMENT IS GIVEN A NAME AND PLACED WITHIN A PROGRAM FILE FOR SUBSEQUENT USE. BOTH THE ELEMENT NAME AND THE FILE IN WHICH THE ELEMENT IS PLACED MAY BE DICTATED BY THE

USER.

NORMALLY, THE COLLECTOR INCLUDES WITHIN AN ABSOLUTE ELEMENT, A SET OF TABLES FOR USE BY THE DIAGNOSTIC SYSTEM. AS DISCUSSED LATER, THIS OUTPUT CAN BE SUPPRESSED BY THE USER.

FOR ANY ERROR CONDITION ENCOUNTERED, THE COLLECTOR PRODUCES AN ERROR MESSAGE WHICH IS PLACED IN THE PRINT FILE ASSIGNED TO THE RUN (PRINT$).

THE ENSUING SECTIONS DESCRIBE IN DETAIL THE EXECUTIVE CONTROL STATEMENTS INVOLVING THE COLLECTOR, THE SOURCE LANGUAGE CONTROL STATEMENTS PROCESSED BY THE COLLECTOR, THE OPERATIONAL CHARACTERISTICS OF IT, AND PROCEDURES FOR SEGMENTING A PROGRAM.

15.1.2. EXECUTIVE CONTROL STATEMENTS

15.1.2.1. THE @MAP CONTROL STATEMENT

THE @MAP EXECUTIVE CONTROL STATEMENT IS USED FOR SPECIFYING THAT THE COLLECTOR IS TO BE USED TO COMBINE A SET OF RELOCATABLE ELEMENTS INTO A SINGLE ABSOLUTE ELEMENT OR INTO A SINGLE RELOCATABLE ELEMENT. THE @MAP STATEMENT HAS THE SAME FORMAT AS OTHER PROCESSOR STATEMENTS AND IS :

```
@MAP,<OPTIONS>     <SPEC1>,<SPEC2>,<SPEC3>
```

THE AVAILABLE OPTIONS ARE:

I INITIAL INSERTION: INTRODUCE SOURCE LANGUAGE ELEMENT INTO PROGRAM FILE FROM THE CONTROL STREAM. <SPEC1> IDENTIFIES THIS ELEMENT. <SPEC3> IS NOT USED. IF NO SOURCE LANGUAGE FOLLOWS, THE COLLECTOR ASSUMES: IN TPF$. (SEE: COLLECTOR CONTROL STATEMENTS).

U UPDATE: PRODUCE A NEW CYCLE OF THE SOURCE LANGUAGE ELEMENT IDENTIFIED IN <SPEC1>. IF CORRECTIONS FOLLOW, THEY WILL BE INCLUDED IN THIS NEW CYCLE. <SPEC3> IS NOT USED.

L PRODUCE A COMPLETE LISTING CONTAINING A SUMMARY OF THE CORE SPACE USED BY THE PROGRAM, THE SPACE ALLOCATED TO EACH ELEMENT, THE PROGRAM ADDRESS OF ALL DEFINITIONS, AND THE EXTERNAL REFERENCES OF EACH ELEMENT.

N PRODUCE NO LISTING. DIAGNOSTIC MESSAGES ARE ALWAYS PRINTED. IF NEITHER L NOR N ARE CODED, ONLY SUMMARY INFORMATION WILL BE PRINTED.

X IF ANY ERRORS ARE DETECTED, INHIBIT EXECUTION OF THE REMAINDER OF THE PROGRAM (EXCEPT A @PMD STATEMENT, IF ONE FOLLOWS). NORMAL ACTION, WITHOUT THE X OPTION, IS TO ACCEPT THE RESULTS OF THE COLLECTION SO LONG AS AN ABSOLUTE OUTPUT ELEMENT IS PRODUCED.

Z INHIBIT GENERATION OF THE DIAGNOSTIC INFORMATION NORMALLY PROVIDED TO THE DIAGNOSTIC SYSTEM.

R PRODUCE A RELOCATABLE ELEMENT RATHER THAN AN ABSOLUTE ELEMENT. (ENTRY POINTS IN THE RELOCATABLE ELEMENTS BEING COMBINED IN AN R-OPTION COLLECTION, WHICH ARE STILL TO BE DEFINED AS ENTRY POINTS IN THE SINGLE RELOCATABLE OUTPUT ELEMENT, MUST BE NAMED ON THE COLLECTOR DEF STATEMENT. THE DEF STATEMENT IS EXPLAINED BELOW.)

E ALLOW PROGRAM TO EXCEED ADDRESS 177777 (DECIMAL 65K).

D   GIVE A DIAGNOSTIC MESSAGE FOR ALL POSSIBLE ADDRESS FIELDS OVER 65K.

<SPEC1> IS 'FILE1.ELT1/VERS1(CYCLE)'. IT NORMALLY IDENTIFIES THE SYMBOLIC INPUT ELEMENT. WHEN THE I OPTION IS USED, IT IDENTIFIES THE SYMBOLIC OUTPUT ELEMENT.

<SPEC2> IS 'FILE2.ELT2/VERS2'. IT IDENTIFIES THE ABSOLUTE OUTPUT ELEMENT OF THE COLLECTION. (IF THE R OPTION IS ON, THIS IS INSTEAD A RELOCATABLE ELEMENT.)

<SPEC3> IS 'FILE3.ELT3/VERS3'. IT IDENTIFIES THE (OPTIONAL) SYMBOLIC OUTPUT ELEMENT WHEN THE I OR U OPTION IS NOT USED. IF CORRECTIONS FOLLOW, THEY WILL BE INCLUDED IN THE NEW ELEMENT. THE CYCLE NUMBER OF THE NEW ELEMENT IS SET TO ZERO.

STANDARD SYSTEM DROPOUT RULES APPLY TO THESE 3 SPECIFICATION SUBFIELDS. A SINGLE NAME (WITH NO PERIOD) IN ANY OF THESE SUBFIELDS IS ASSUMED TO BE AN ELEMENT IN THE RUN'S TEMPORARY PROGRAM FILE (TPF$). A SINGLE NAME WITH A LEADING PERIOD IN <SPEC2> OR <SPEC3> IS ASSUMED TO BE AN ELEMENT IN THE FILE GIVEN IN THE <SPEC> PRECEDING. A SINGLE NAME WITH A LEADING PERIOD IN <SPEC1> IS ASSUMED TO BE IN TPF$.

HERE ARE SOME EXAMPLES OF THE @MAP STATEMENT:

@MAP   SYMIN/C,BACKUP.ABSOUT

(ELEMENT SYMIN, VERSION C, LATEST CYCLE, FROM TPF$, IS USED TO DIRECT THE COLLECTION OF ELEMENT ABSOUT WRITTEN INTO FILE BACKUP. IF ANY CORRECTIONS FOLLOW, THEY WILL BE USED BUT NOT SAVED, BECAUSE NO OUTPUT SYMBOLIC ELEMENT IS PRODUCED.)

@MAP,I BACKUP.SYMOUT,.ABSOUT

(THE STATEMENTS FOLLOWING THE @MAP STATEMENT ARE USED TO DIRECT THE COLLECTION AND ARE OUTPUT AS ELEMENT SYMOUT IN FILE BACKUP. THE ABSOLUTE OUTPUT ELEMENT IS ALSO PUT INTO FILE BACKUP.)

@MAP,U SYMIN(3),ABSOUT/REVISED

(CYCLE 4 OF ELEMENT SYMIN IN TPF$ IS PRODUCED, SAVING ANY CORRECTIONS THAT FOLLOW THE @MAP STATEMENT. ABSOLUTE OUTPUT ELEMENT ABSOUT, VERSION REVISED, IS ALSO PUT INTO TPF$.)

@MAP,I

(THIS IS A SPECIAL CASE WHERE THE SYSTEM PICKS OUT ITS OWN NAME FOR THE SYMBOLIC AND ABSOLUTE OUTPUT ELEMENTS. BOTH INTERNAL TABLE ENTRIES, AND THE PRINTED OUTPUT PRODUCED BY A @PRT,T STATEMENT FOLLOWING THE COLLECTION, WILL LOOK AS IF THE @MAP STATEMENT HAD CONTAINED: @MAP,I TPF$.NAME$,.NAME$.)

@MAP

(THIS IS TREATED AS IF IT HAS AN I OPTION, WHICH IS THE SAME AS THE SPECIAL CASE ABOVE.)

@MAP,IRXLED A,A

(THE STATEMENTS FOLLOWING THE @MAP STATEMENT

```
                    ARE USED TO DIRECT THE COLLECTION AND ARE
                    OUTPUT TO TPF$ AS SYMBOLIC ELEMENT A.  THE
                    RESULT OF THE COLLECTION IS OUTPUT TO TPF$
                    AS RELOCATABLE ELEMENT A.  IF ERRORS OF ANY
                    KIND ARE ENCOUNTERED, INHIBIT CONTINUATION OF
                    PROGRAM.  PRODUCE A FULL LISTING.  ALLOW DBANK
                    TO EXCEED 65K.  PRINT DIAGNOSTICS FOR ADDRESS
                    FIELDS OVER 65K.)

   @MAP     OLDFILENAME.OLDELEMENT,A,NEWFILENAME.NEWELEMENT

                    (THE SYMBOLIC LIST OF COLLECTOR COMMANDS
                    CONTAINED IN OLDELEMENT, AS AMENDED BY ANY
                    CORRECTIONS FOLLOWING THIS @MAP STATEMENT, IS
                    USED TO DIRECT THE COLLECTION, AND IS OUTPUT
                    INTO A NEW FILE WITH A NEW ELEMENT NAME.
                    THE ABSOLUTE OUTPUT ELEMENT, A, GOES TO TPF$.)
```

THESE 3 EXAMPLES FOLLOW ONE ANOTHER IN SEQUENCE, AND SHOW SOME ACTUAL COLLECTOR CONTROL STATEMENTS (WHICH ARE EXPLAINED LATER ON):

```
   @MAP,I   F8.MAPSYM,.MYPROGRAM
   LIB        F8.
   SEG        CONTROL
   IN         ELEMENT1
   SEG        OVERLAY5,(CONTROL)
   IN         F9.

   @MAP,U   F8.MAPSYM,.MYPROGRAM
   LIB        F10.
   -3,3
   IN         ELEMENT0
   SEG        OVERLAY4,(CONTROL)
   IN         F7.
   -5
   SEG        HIGHCORE,(OVERLAY4,OVERLAY5)
   IN         DATAELEMENT
   CLASS      REVISED
   NOT        ELEMENT1
                    (AFTER THIS SECOND @MAP, ELEMENT MAPSYM WILL
                    SAVE THE CORRECTIONS, BECAUSE AN UPDATE OF
                    CYCLE WAS SPECIFIED.  THE ABSOLUTE ELEMENT
                    MYPROGRAM PUT INTO FILE F8 BY THE FIRST @MAP
                    WILL BE DELETED AND REPLACED BY THE DIFFERENT
                    MYPROGRAM PRODUCED BY THIS SECOND @MAP.)

   @MAP,I   F8.MAPSYM,.MYPROGRAM
   LIB        F10.
   LIB        F8.
   SEG        CONTROL
   IN         ELEMENT0
   SEG        OVERLAY4,(CONTROL)
   IN         F7.
   SEG        OVERLAY5,(CONTROL)
   IN         F9.
   SEG        HIGHCORE,(OVERLAY4,OVERLAY5)
   IN         DATAELEMENT
   CLASS      REVISED
   NOT        ELEMENT1
                    (ELEMENT MYPROGRAM COMES OUT EXACTLY THE SAME
                    HERE AS THE PRECEDING MYPROGRAM, WHICH IT
                    REPLACES IN FILE F8.  ELEMENT MAPSYM, SET HERE
                    TO INTITAL CYCLE 0, ALSO COMES OUT THE SAME
                    AS THE UPDATE CYCLE 1 OF MAPSYM, WHICH IT
                    REPLACES IN FILE F8.)
```

IF A PROGRAM FILE NAMED ON A @MAP STATEMENT (OR ON A 'LIB' OR 'IN' COLLECTOR CONTROL STATEMENT) HAS NOT BEEN ASSIGNED, BUT HAS BEEN PREVIOUSLY CATALOGED, IT WILL BE ASSIGNED AUTOMATICALLY DURING THE COLLECTION. AT THE END OF THE COLLECTION, IT WILL BE RETURNED TO ITS ORIGINAL STATE WITH A @FREE (PLUS X, R, AND/OR A OPTION). THE METHOD USED TO @ASG AND @FREE THESE FILES FROM WITHIN THE COLLECTOR IS THE SAME AS THAT AVAILABLE TO EVERY USER PROGRAM VIA THE ER CSF$ EXECUTIVE REQUEST (DOCUMENTED EARLIER IN THE PRM).

15.1.2.2. THE @XQT CONTROL STATEMENT

FOR EXECUTING AN ABSOLUTE PROGRAM CREATED BY THE COLLECTOR, THE FOLLOWING CONTROL STATEMENT IS USED:

```
@XQT,OPTIONS  FILE1.ELT1/VERS1
```

ANY OPTIONS SPECIFIED ARE AVAILABLE TO THE USER'S PROGRAM BY THE OPT$ EXECUTIVE REQUEST WHENEVER IT IS INITIATED. THE FIELD 'FILE1.ELT1/VERS1' SPECIFIES THE ABSOLUTE PROGRAM TO EXECUTE. THE SUB-FIELD 'FILE1' IS THE NAME OF THE FILE; 'ELT1/VERS1' IDENTIFIES THE ELEMENT (AND VERSION, IF ANY). THIS FIELD IS THE COUNTERPART OF FILE2.ELT2/VERS2' IN THE @MAP CONTROL STATEMENT. IF NO ELEMENT IS SPECIFIED, THE LAST ABSOLUTE ELEMENT PLACED IN THE FILE GIVEN (TEMPORARY IF NO FILE SPECIFIED) WILL BE LOADED AND EXECUTED. IN THE ABSENCE OF AN ABSOLUTE ELEMENT, ALL OF THE RELOCATABLE ELEMENTS IN THE FILE ARE BOTH COLLECTED AND EXECUTED (A RELOCATABLE ELEMENT MAY NOT BE EXPLICITELY NAMED ON AN @XQT STATEMENT, HOWEVER).

EXAMPLES OF THE @XQT STATEMENT IN TYPICAL RUN STREAMS:

```
1.   ----       COMPILATIONS OR ASSEMBLIES TO PRODUCE
                RELOCATABLE ELEMENTS IN THE RUN'S TEMPORARY
                PROGRAM FILE, TPF$.
     @MAP       GENERATE AN ABSOLUTE ELEMENT CONSISTING OF ALL
                THE RELOCATABLE ELEMENTS IN THE TEMPORARY FILE.
     @XQT       EXECUTE THE ABSOLUTE ELEMENT GENERATED ABOVE.

2.   ----       COMPILATIONS OR ASSEMBLIES PRODUCING
                RELOCATABLE ELEMENTS IN THE USER SPECIFIED
                FILE, FILEA.
     @MAP  FILEA.SYMBOLIC,FILEA.XYZ
                GENERATE THE ABSOLUTE ELEMENT XYZ IN FILEA
                AS DIRECTED BY THE SOURCE ELEMENT
                FILEA.SYMBOLIC.
     @XQT FILEA.XYZ
                EXECUTE THE ABSOLUTE ELEMENT GENERATED ABOVE.
```

15.1.3. COLLECTOR CONTROL STATEMENTS

15.1.3.1. GENERAL

IN ADDITION TO THE INFORMATION SPECIFIED IN THE @MAP CONTROL STATEMENT, A SET OF SOURCE LANGUAGE CONTROL STATEMENTS CAN BE PROCESSED BY THE COLLECTOR TO PROVIDE THE USER THE CAPABILITY OF CONTROLLING THE CONSTRUCTION OF EVEN THE MOST COMPLEX PROGRAMS. THE USER CAN ENTER THESE CONTROL STATEMENTS VIA HIS CONTROL INPUT STREAM FOR EACH COLLECTION, OR HE CAN CREATE WITHIN A PROGRAM FILE A SOURCE LANGUAGE CONTROL ELEMENT CONTAINING THE STATEMENTS. THIS ELEMENT CAN BE UPDATED BY ENTERING THE CORRECTIONS VIA THE CONTROL INPUT STREAM.

THE CONTROL STATEMENTS RECOGNIZED BY THE COLLECTOR INCLUDE THE FOLLOWING:

```
IN      INCLUDE SPECIFIC ELEMENTS IN THE COLLECTION.
```

NOT EXCLUDE SPECIFIC ELEMENTS FROM THE COLLECTION.

LIB SPECIFY LIBRARIES TO BE SEARCHED.

SEG DIRECT THE SEGMENTATION OF A PROGRAM.

DSEG SPECIFY A DYNAMIC SEGMENT.

RSEG SPECIFY A RELOCATABLE SEGMENT.

DEF SPECIFY EXTERNAL DEFINITIONS TO BE RETAINED IN THE ABSOLUTE OR RELOCATABLE ELEMENT.

REF SPECIFY THE EXTERNAL REFERENCES TO BE RETAINED IN THE ABSOLUTE OR RELOCATABLE ELEMENT.

ENT SPECIFY THE STARTING ADDRESS OF A PROGRAM.

EQU GIVE VALUES TO UNDEFINED SYMBOLS AT THE TIME OF THE COLLECTION.

CLASS SPECIFY A MASK TO USE IN SELECTING ELEMENTS FOR COLLECTION.

COR SPECIFY THAT CORRECTIONS ARE TO BE MADE TO AN ELEMENT.

SNAP DIRECT POSITIONING OF SNAPSHOT DUMPS.

END END OF SOURCE LANGUAGE STATEMENTS TO BE PROCESSED.

15.1.3.2. THE IN STATEMENT

THE IN CONTROL STATEMENT ALLOWS THE USER TO INCLUDE ANY OR ALL ELEMENTS FROM ANY NUMBER OF FILES IN HIS COLLECTION AND SPECIFICALLY IN THE SEGMENT NAMED BY THE PRECEDING SEG STATEMENT. THE FORMAT OF THIS STATEMENT IS THE FOLLOWING:

```
      IN   FILE1.ELT1/VER1,FILE2.ELT2,...
```

THE FIELDS 'FILE1.ELT1/VER1,' ETC., IDENTIFY SPECIFIC ELEMENTS TO BE INCLUDED IN THE COLLECTION. BY SPECIFYING ONLY 'FILE1.', THE USER CAN SPECIFY THE INCLUSION OF ALL ELEMENTS IN A PROGRAM FILE.

NORMALLY ALL THE RELOCATABLE ELEMENTS IN THE RUN'S TEMPORARY PROGRAM FILE, TPF$, ARE INCLUDED IN EVERY COLLECTION. IF THE EXTERNAL DEFINITIONS OF THE TEMPORARY FILE HAVE BEEN COLLECTED WITH A @PREP CONTROL STATEMENT, THE ELEMENTS ARE INCLUDED SELECTIVELY. IN THIS CASE TPF$. IS THE FIRST PROGRAM FILE EXAMINED FOR ELEMENT INCLUSION. ELEMENTS THAT ARE NOT ASSOCIATED WITH FILES MAY BE INCLUDED FROM TPF$. OR ANY PROGRAM FILE NAMED IN LIB STATEMENTS.

AN ELEMENT NAME MAY APPEAR ON ONLY ONE IN STATEMENT AND ONLY ONCE IN A COLLECTION.

IT IS IMPORTANT, FOR FORTRAN PROGRAMMERS ESPECIALLY, TO NOTE THAT NO COMMON BLOCK NAME (LABELLED, OR NAMED COMMON) IN THE COLLECTION MAY BE IDENTICAL TO ANY ELEMENT NAME. THROUGHOUT MOST OF THE COLLECTION COMMON BLOCKS AND ELEMENTS ARE HANDLED IN A VERY SIMILAR MANNER, AND THEIR NAMES MUST DISTINGUISH THEM FROM ONE ANOTHER.

THE FOLLOWING ARE EXAMPLES ON THE USE OF THE IN STATEMENT:

```
IN   FILEA.,FILEB.
               (ALL RELOCATABLE ELEMENTS IN FILEA AND FILEB
               ARE INCLUDED.)

IN   FILEB.BB,.CC,DD
               (ELEMENTS BB AND CC FROM FILEB, AND ELEMENT DD,
                    WHOSE FILE ISN'T INDICATED, ARE INCLUDED IN
                    THE COLLECTION.)
```

15.1.3.3. THE NOT STATEMENT

THE NOT CONTROL STATEMENT IS ESSENTIALLY THE INVERSE OF THE IN STATEMENT. IT ALLOWS THE USER TO STATE EXPLICITLY WHICH ELEMENTS WITHIN FILES ARE NOT TO BE INCLUDED IN A COLLECTION. THE FORMAT OF THIS STATEMENT IS AS FOLLOWS:

```
NOT  FILE1.ELT1/VER1,FILE2.ELT2/VER2,...
```

WHERE THE SUCCESSIVE FIELDS INDICATE ELEMENTS NOT TO BE INCLUDED. IF THE VERSION NAME OR FILE NAME IS OMITTED, ALL ELEMENTS WITH THE SPECIFIED NAME ARE BYPASSED.

THE FOLLOWING ARE EXAMPLES ON THE USE OF THE NOT STATEMENT:

```
1.   @MAP,I  A,A
     NOT  AA,BB
                (ALL RELOCATABLE ELEMENTS IN THE TEMPORARY FILE
                EXCEPT AA AND BB ARE INCLUDED IN THE
                COLLECTION.)

2.   @MAP,I  A,A
     IN    FILEA.
     NOT   FILEA.AA,.BB
                (ALL RELOCATABLE ELEMENTS IN FILEA EXCEPT
                AA AND BB ARE INCLUDED.)

3.   @MAP,I  A,A
     IN    FILEA.,FILEB.
     NOT   FILEA.AA,.BB,FILEB.CC,.DD
                (ALL RELOCATABLE ELEMENTS FROM FILEA EXCEPT
                AA AND BB, AND ALL RELOCATABLE ELEMENTS FROM
                FILEB EXCEPT CC AND DD ARE INCLUDED.)
```

15.1.3.4. THE LIB STATEMENT

THE LIB CONTROL STATEMENT ALLOWS THE USER TO SPECIFY LIBRARIES TO BE SEARCHED BY THE COLLECTOR FOR THE PURPOSE OF SATISFYING EXTERNAL REFERENCES AND/OR OF FINDING ELEMENTS SPECIFIED WITHOUT FILE NAMES. THE FORMAT OF THE LIB STATEMENT IS:

```
LIB  FILE1,FILE2,...
```

THE NAMES OF FILES TO BE TREATED AS LIBRARIES ARE SPECIFIED IN SUCCESIVE FIELDS. THESE LIBRARIES ARE SEARCHED IN THE ORDER IN WHICH THEY ARE GIVEN AND BEFORE THE SYSTEM LIBRARY. FILES MAY BE SPECIFIED TO BE SEARCHED MORE THAN ONCE BY NAMING IT MORE THAN ONCE IN A LIB STATEMENT. SEVERAL LIB STATEMENTS MAY BE SPECIFIED AND THEIR EFFECT IS CUMULATIVE. FILES ARE NOT SEARCHED FOR EXTERNAL DEFINITIONS IF THE FILE HAS NOT BEEN PREPARED BY THE PREP OPERATION.

TYPICAL LIB CONTROL STATEMENTS ARE AS FOLLOWS:

```
LIB USER1       (FILE USER1 IS SEARCHED BEFORE THE SYSTEM
                LIBRARY.)

LIB  USER1,USER2
                (FILE USER1 AND THEN FILE USER2 ARE SEARCHED
                BEFORE THE SYSTEM LIBRARY.)
```

15.1.3.5. THE SEG STATEMENT

THE SEG CONTROL STATEMENT IS USED TO INFORM THE COLLECTOR OF THE BEGINNING OF A NEW SEGMENT IN THOSE PROGRAMS REQUIRING SEGMENTATION. THE FORMAT IS:

```
SEG  NAME1, NAME2
```

```
          OR
      SEG  NAME1, (NAME2,NAME3,...)
```

THE FIELD 'NAME1' IS THE NAME OF THE SEGMENT AND MUST BE SPECIFIED. THE FIELD 'NAME2',ETC. GIVES THE NAMES OF OTHER SEGMENTS TO WHICH THE SEGMENT 'NAME1' IS BEING RELATED. A SEGMENT CAN BE SPECIFIED FOR AUTOMATIC (INDIRECT) LOADING WHEN REFERENCED BY SUFFIXING AN ASTERISK TO 'NAME1': 'NAME1*'.

IF THE FIELD 'NAME2' IS VOID, THE SEGMENT BEING SPECIFIED IS ORIGINATED IMMEDIATELY FOLLOWING THE SEGMENT DEFINED BY THE PRECEDING SEG STATEMENT. THE FIELD 'NAME2' (NOT INCLUDED IN PARENTHESES) SPECIFIES THAT THE SEGMENT BEING DEFINED IS TO ORIGINATE AT THE SAME LOCATION AS DOES SEGMENT 'NAME2'. IF THE RIGHT HAND FIELD CONTAINS ONE OR MORE SEGMENT NAMES ENCLOSED IN PARENTHESES, THE SEGMENT 'NAME1' IS STARTED FOLLOWING THE HIGHEST ADDRESS OCCUPIED BY ANY OF THESE SEGMENTS.

EACH SEGMENT MAY HAVE TWO PROGRAM AREAS (BANKS) NAMELY, THE INSTRUCTION AREA AND THE DATA AREA (ALSO REFERRED TO AS IBANK AND DBANK). THEREFORE, SEGMENTS SPECIFIED TO FOLLOW THE HIGHEST ADDRESS OF SEVERAL SEGMENTS MAY HAVE ITS INSTRUCTION AREA FOLLOW THE INSTRUCTION AREA OF ONE SEGMENT AND ITS DATA AREA FOLLOW THE DATA AREA OF A DIFFERENT SEGMENT.

THE FIRST SEGMENT NAMED IN THE SOURCE INPUT IS CALLED THE MAIN SEGMENT AND IS NOT OVERLAYED BY OTHER SEGMENTS.

SEGMENTS MAY BE LOADED AND EXECUTED INDEPENDENTLY OF ONE ANOTHER. HOWEVER, THE PLACEMENT OF ELEMENTS COMMON TO SEVERAL SEGMENTS MAY DICTATE THAT SOME SEGMENTS MUST BE IN MEMORY WHEN OTHERS ARE BEING EXECUTED. ELEMENTS ARE NOT NECESSARILY ATTACHED TO THE MAIN SEGMENT WHEN THEY ARE REFERENCED IN MORE THAN ONE SEGMENT BUT NOT EXPLICITLY INCLUDED IN ANY SEGMENT. EACH SEGMENT HAS A PATH LEADING TO THE MAIN SEGMENT. ELEMENTS REFERENCED BY TWO (OR MORE) SEGMENTS ARE ATTACHED TO THE SEGMENT THAT IS IN THE PATH OF ALL THE REFERENCING SEGMENTS. NAMED COMMON BLOCKS ARE LIKEWISE IN THE PATH OF ALL SEGMENTS REFERENCING THE BLOCK.

THE PATH TO THE MAIN SEGMENT FOLLOWS THE PATH OF THE FIRST SEGMENT IN ITS PATH. THE FIRST SEGMENT IN ITS PATH IS DETERMINED BY ITS RELATION SPECIFICATIONS.

```
   SEG   A,(B)          SEGMENT A'S PATH STARTS WITH SEGMENT B AND
OR:                     FOLLOWS B'S PATH TO THE MAIN SEGMENT.
   SEG   B
   SEG   A

   SEG   A,B            SEGMENT A'S PATH IS IDENTICAL TO SEGMENT B'S.

   SEG   A,(B,C,D)      THE FIRST SEGMENT IN A'S PATH IS THE SEGMENT
                        COMMON TO THE PATHS OF SEGMENTS B,C AND D.
```

AT LEAST ONE IN STATEMENT MUST FOLLOW THE SEG STATEMENT.

ALL THE ELEMENTS SPECIFIED ON IN STATEMENTS AFTER THIS SEG STATEMENT AND BEFORE THE FOLLOWING SEG STATEMENT ARE A PART OF THIS SEGMENT. OTHER ELEMENTS REFERENCED ARE INCLUDED IN THIS SEGMENT OR IN A SEGMENT IN ITS PATH TO THE MAIN SEGMENT.

15.1.3.6. THE DSEG STATEMENT

THE DSEG CONTROL STATEMENT IS USED TO INFORM THE COLLECTOR OF THE BEGINNING OF A SEGMENT WITH SPECIAL CHARACTERISTICS. THIS TYPE OF SEGMENT IS CALLED A DYNAMIC SEGMENT. THE CORE AREA OF THE SEGMENT IN EXCESS OF NORMAL SEGMENTS MAY BE TEMPORARILY RELEASED TO THE EXECUTIVE SYSTEM WITH A REFERENCE TO DREL$. THE AREA MAY BE RELEASED AUTOMATICALLY WHEN A DYNAMIC SEGMENT IS OVERLAYED. THE AREA IS RELEASED ONLY WHEN IT IS AT THE END OF THE PROGRAM'S AREA. SINCE THE EXECUTIVE SYSTEM MAY NEED TO MOVE OTHER PROGRAMS TO LOAD A DYNAMIC SEGMENT, DISCRETION SHOULD BE USED IN DESIGNATING WHAT SEGMENTS ARE DYNAMIC SEGMENTS.

THE DSEG STATEMENT HAS THE SAME FORMAT AS THE SEG STATEMENT.

15.1.3.7. THE RSEG STATEMENT

THE RSEG STATEMENT IS USED TO INFORM THE COLLECTOR OF THE BEGINNING OF A SEGMENT THAT IS RELOCATABLE. THE SEGMENT CONTAINS ONLY AN INSTRUCTION AREA. RELOCATION OF ADDRESS FIELDS IS ACCOMPLISHED BY ADDING THE BEGINNING ADDRESS OF THE SEGMENT TO THE RIGHT OR LEFT HALF OF THE WORDS TO BE RELOCATED. RELOCATABLE SEGMENTS MAY NOT BE DESIGNATED FOR INDIRECT LOADING. THE ELEMENTS TO MAKE UP A RSEG SEGMENT MUST BE EXPLICITLY NAMED ON IN STATEMENTS FOLLOWING THE RSEG STATEMENT.

15.1.3.8. THE DEF STATEMENT

THE DEF CONTROL STATEMENT IS USED TO LIST THOSE EXTERNAL DEFINITIONS TO BE RETAINED BY THE RESULTING ABSOLUTE OR RELOCATABLE ELEMENT. FOR ABSOLUTE ELEMENTS, THE PROGRAM MAY BE ENTERED BY INTERPRETIVE CODE (OUTPUT OF THE CONVERSATIONAL PROCESSORS) AT ANY OF THE EXTERNAL DEFINITIONS LISTED. THE ADDRESS OF THIS TABLE IS DEFINED BY THE COLLECTOR TO BE ENTRY$. IT IS ADDRESSABLE BY THE PROGRAM USING THE TAG ENTRY$. ALSO, DEF AND/OR REF STATEMENTS CAUSE THE COLLECTOR TO BUILD A TABLE DEFINING THE LABELLED COMMON BLOCKS IN THE PROGRAM. THIS TABLE IS ADDRESSABLE BY THE COLLECTOR DEFINED TAG COMMN$. THE FORMAT OF THE STATEMENT IS:

```
        DEF  DEF1,DEF2,DEF3,...
```

WHERE THE SUCCESSIVE 'DEF1' FIELDS ARE THE NAMES OF EXTERNAL DEFINITIONS TO BE RETAINED. AN EXAMPLE OF THIS STATEMENT IS AS FOLLOWS:

```
   DEF SIN,COS,SORT
                   (THE LISTED EXTERNAL DEFINITIONS ARE RETAINED
                   BY THE RESULTING ELEMENT.)
```

15.1.3.9. THE REF STATEMENT

THE REF CONTROL STATEMENT IS USED TO LIST THOSE EXTERNAL REFERENCES TO BE RETAINED BY THE RESULTING ABSOLUTE OR RELOCATABLE ELEMENT. FOR ABSOLUTE ELEMENTS, THE EXTERNAL REFERENCES LISTED MAY BE LINKED TO INTERPRETIVE CODE BY THE INTERPRETER. THE TABLE IS ADDRESSABLE BY THE COLLECTOR DEFINED TAG XREF$. THE FORMAT OF THIS STATEMENT IS AS FOLLOWS:

```
        REF  REF1,REF2,REF3,...
```

WHERE THE SUCCESSIVE 'REF1' FIELDS ARE THE NAMES OF THE EXTERNAL REFERENCES TO BE RETAINED. NO ATTEMPT IS MADE TO SATISFY THESE REFERENCES FROM EITHER USER LIBRARIES OR THE SYSTEM LIBRARY. AN EXAMPLE OF THE REF STATEMENT IS:

```
  REF SIN,COS,SQRT
                   (THE LISTED EXTERNAL REFERENCES ARE RETAINED BY
                   THE NEW ELEMENT)
```

15.1.3.10. THE ENT STATEMENT

THE ENT CONTROL STATEMENT PROVIDES THE USER THE CAPABILITY OF OVERRIDING THE STARTING ADDRESS SPECIFIED VIA THE END ASSEMBLER DIRECTIVE OR THE ENTRANCE TO A MAIN PROGRAM GENERATED BY FORTRAN OR COBOL. THE FORMAT OF THIS STATEMENT IS :

```
        ENT  NAME
```

WHERE NAME IS AN EXTERNALLY DEFINED SYMBOL. CONTROL IS TRANSFERRED TO THE ABSOLUTE LOCATION GENERATED FOR THIS SYMBOL WHENEVER THE PROGRAM IS SUBSEQUENTLY EXECUTED. IN THE ABSENCE OF AN ENT STATEMENT, THE STARTING ADDRESS WILL BE TAKEN TO BE A TRANSFER ADDRESS ENCOUNTERED IN THE PROCESSING OF RELOCATABLE ELEMENTS. THE STARTING ADDRESS MUST BE IN THE MAIN SEGMENT.

15.1.3.11. THE EQU STATEMENT

THE EQU CONTROL STATEMENT MAY BE USED TO GIVE A VALUE TO AN UNDEFINED SYMBOL AT

THE TIME OF COLLECTION. THE FORMAT OF THIS STATEMENT IS:

```
      EQU  NAME1/VALUE1,NAME2/VALUE2,...
```

WHERE 'NAME1' IS A SYMBOL TO BE DEFINED AND 'VALUE' IS THE VALUE TO BE GIVEN TO THE SYMBOL. THE SAME IS TRUE FOR 'NAME2/VALUE2', ETC. THE SUB-FIELD 'VALUE' MAY BE AN OCTAL OR DECIMAL INTEGER, A SYMBOL, OR A SYMBOL WITH AN OFFSET. IF A SYMBOL IS USED, IT MUST BE EXTERNALLY DEFINED BY ONE OF THE ELEMENTS TO BE INCLUDED. EXAMPLES OF THE USE OF THE EQU STATEMENT ARE AS FOLLOWS:

```
   1.  EQU JOE/0200
                  (EXTERNAL REFERENCE JOE IS DEFINED TO BE 0200.)
   2.  EQU AL/SAM+10
                  (EXTERNAL REFERENCE AL IS DEFINED TO BE SAM+10;
                  SAM MUST BE EXTERNALLY DEFINED).
   3.  EQU JOE/0200, AL/SAM+10
                  (SAME AS 1 AND 2)
```

15.1.3.12. THE CLASS STATEMENT

THE CLASS STATEMENT MAY BE USED TO SPECIFY THE RELOCATABLE ELEMENT TO BE INCLUDED IN THE COLLECTION WHEN OTHERWISE MORE THAN ONE ELEMENT COULD QUALIFY. THERE ARE TWO CONDITIONS WHERE MORE THAN ONE ELEMENT MAY QUALIFY:

1. THE VERSION IS NOT SPECIFIED ON AN IN STATEMENT AND MORE THAN ONE RELOCATABLE ELEMENT HAS THAT SAME NAME.
2. MORE THAN ONE RELOCATABLE ELEMENT DEFINES AN EXTERNAL REFERENCE AND NONE OF THE ELEMENTS HAS BEEN EXPLICITLY INCLUDED IN THE COLLECTION, OR ALL BUT ONE EXPLICITLY EXCLUDED FROM THE COLLECTION.

THE FORMAT OF THE CLASS STATEMENT IS :

```
      CLASS  STRING
```

WHERE THE FIELD 'STRING' IS TWELVE CHARACTERS INCLUDING THE SPECIAL CHARACTER ASTERISK (*). THE ASTERISK DESIGNATES A CHARACTER POSITION THAT IS TO BE IGNORED WHEN MAKING COMPARISONS. WHEN SEVERAL ELEMENTS QUALIFY TO BE INCLUDED IN THE COLLECTION, THE COLLECTOR COMPARES THIS STRING OF CHARACTERS WITH THE VERSIONS OF THE QUALIFYING ELEMENTS. IF THE ELEMENT DOES NOT HAVE THE SAME CHARACTERS IN THE VERSION AS THE CHARACTERS OF THE 'STRING' (FOR EACH CHARACTER POSITION), IT NO LONGER QUALIFIES FOR INCLUSION.

WHEN ONLY ONE ELEMENT REMAINS QUALIFIED AFTER THE COMPARISONS, THAT ELEMENT IS INCLUDED IN THE COLLECTION. WHEN MORE THAN ONE ELEMENT STILL QUALIFIES, THE VERSIONS OF THESE ELEMENTS ARE COMPARED TO THE CHARACTER STRING OF THE NEXT CLASS STATEMENT. IF MORE THAN ONE ELEMENT QUALIFIES AFTER THE CLASS STATEMENT PARAMETERS HAVE BEEN EXHAUSTED A DIAGNOSTIC MESSAGE IS GIVEN. NONE OF THE 'QUALIFING' ELEMENTS IS INCLUDED IN THE PROGRAM SINCE A UNIQUE ELEMENT MAY BE FOUND IN THE NEXT LIBRARY EXAMINED. IT SHOULD BE NOTED THAT DIFFERENT ORDERS OF CLASS STATEMENTS MAY GIVE DIFFERENT RESULTS.

ASSUME THAT THE ELEMENT NAMED SIZE IS NAMED ON AN IN STATEMENT AND THE FOLLOWING RELOCATABLE ELEMENTS ARE IN THE TEMPORARY LIBRARY:

```
 SIZE/A2SMALL
 SIZE/B3LARGE
 SIZE/D3SMALL
 SIZE/D2LARGE
```

THE SOURCE LANGUAGE TO THE COLLECTOR IS:

```
 @MAP,I   S,A
 SEG   AA
 IN    SIZE
 CLASS    **LA********
 CLASS    D***********
```

THE ELEMENT SIZE/D2LARGE IS INCLUDED IN THE COLLECTION. THE ONE CLASS STATEMENT:

```
 CLASS    D*LA********
```

WILL GIVE THE SAME RESULTS.

15.1.3.13. THE COR STATEMENT

THE COR STATEMENT IS USED TO SPECIFY THAT RELOCATABLE CORRECTIONS ARE TO BE MADE TO AN ELEMENT INCLUDED IN THE COLLECTION. THE FORMAT OF THE COR STATEMENT IS:

```
        COR  ELT
```

THE RELOCATABLE CORRECTIONS FOR THE ELEMENT 'ELT' FOLLOW THE COR STATEMENT. RELOCATABLE CORRECTIONS MAY BE ONE OF THREE FORMATS:

```
      ADDRESS,LC1 F J A B H I U,LC2,ELT1
      ADDRESS,LC1 DATAWORD
      ADDRESS,LC1 DATA,LC2 DATA,LC2
```

THE FIELD 'ADDRESS' SPECIFIES THE RELATIVE ADDRESS UNDER LOCATION COUNTER 'LC1' TO MAKE THE CORRECTION. THE F,J,A,B,H AND I FIELDS CORRESPOND TO PORTIONS OF THE UNIVAC 1108 INSTRUCTION WORD. THE FIELDS 'U' AND 'DATA' MAY BE A SYMBOL, SYMBOL AND OFFSET OR AN OCTAL OR DECIMAL NUMBER. OCTAL NUMBERS REQUIRE A LEADING ZERO. THE FIELD 'DATAWORD' MUST BE NUMERIC. THE OPTIONAL FIELD 'LC2' INDICATES THAT THE 'U' OR 'DATA' FIELDS ARE RELATIVE TO THE VALUE OF THE LOCATION COUNTER 'LC2'. THE OPTIONAL FIELD 'ELT1' SPECIFIES THE ELEMENT IN WHICH 'LC2' BELONGS, IF IT IS OTHER THAN THE ELEMENT BEING CORRECTED. THE DATA FIELDS REPRESENT THE UPPER AND LOWER HALVES OF THE WORD.

COR STATEMENTS ARE BYPASSED IN AN R-OPTION COLLECTION.

15.1.3.14. THE SNAP STATEMENT

THE SNAP CONTROL STATEMENT SPECIFIES ELEMENTS IN WHICH SNAPSHOT DUMPS ARE TO BE TAKEN. THE FORMAT OF THE STATEMENT IS:

```
       SNAP   ELT
```

THE FIELD 'ELT' IS AN ELEMENT INCLUDED IN THE COLLECTION. STATEMENTS FOLLOWING THE SNAP STATEMENT GIVE THE PARAMETERS FOR THE SNAPSHOT DUMP. THE FORMAT IS:

```
      ADDRESS,LC1  ADDRESS  LENGTH,R  TIMES,FREQUENCY
```

THE FIELD 'ADDRESS,LC1' SPECIFIES THE ADDRESS OF THE INSTRUCTION TO BE REPLACED WITH A DUMP REQUEST. THE FIELD 'ADDRESS' GIVES THE ADDRESS TO START THE DUMP. THE FIELD MAY BE 'U,LC2,ELT1' OR SYMBOL AND OFFSET AS IN THE COR STATEMENT PARAMETERS. THE FIELD 'LENGTH' SPECIFIES THE LENGTH OF THE MEMORY AREA TO DUMP. THE FIELD 'R' IS USED TO INDICATE WHICH OF THE REGISTERS IS TO BE PRINTED ACCORDING TO THE VALUES:

```
       R=           0-NO REGISTERS
                    1-R  REGISTERS
                    2-A  REGISTERS
                    3-A AND R  REGISTERS
                    4-X REGISTERS
                    5-X AND R REGISTERS
                    6-X AND A REGISTERS
                    7-X, A AND R REGISTERS
```

THE 'TIMES' FIELD SPECIFIES THE NUMBER OF TIMES THE SNAPSHOT IS TO BE TAKEN. IF OMITTED, THE VALUE IS 100. THE FIELD 'FREQUENCY' SPECIFIES AT WHAT INTERVALS THE DUMP IS TO BE TAKEN. THE VALUE THREE SPECIFIES THE DUMP TO BE TAKEN EVERY THIRD REFERENCE;FIVE EVERY FIFTH REFERENCE,ETC. THE VALUE OF ONE IS ASSUMED IF THE FIELD IS OMITTED.

AT MOST SIXTEEN SNAPSHOT PARAMETER STATEMENTS MAY BE INCLUDED IN ONE COLLECTION.

WHAT ACTUALLY HAPPENS WITH A SNAP SHOT DUMP REQUEST IS THAT AN 'SLJ SNAP$$' INSTRUCTION IS INSERTED AT THE LOCATION AT WHICH THE SNAP IS CALLED. SNAP$$ IS AN ENTRY POINT IN ELEMENT SNAP$. THE REPLACED INSTRUCTION WORD IS SAVED IN A TABLE IN ELEMENT SNAP$. AFTER THE DUMP IS TAKEN, THE SAVED INSTRUCTION IS EXECUTED FROM WITHIN SNAP$ AS IF IT HAD NOT BEEN MOVED. IF THE SAVED INSTRUCTION IS A JUMP, CONTROL GOES IMMEDIATELY TO THE JUMP DESIGNATION. OTHERWISE, CONTROL IS TRANSFERED TO THE LOCATION FOLLOWING THE LOCATION AT WHICH THE SNAP IS CALLED.

BECAUSE OF THIS EXECUTION OF THE REPLACED INSTRUCTION FROM WITHIN SNAP$, THE REPLACED INSTRUCTION MUST NOT BE -

- ALTERED DURING THE COURSE OF PROGRAM EXECUTION.
- REFERENCED AS DATA.
- REFERENCED BY INDIRECT ADDRESSING.
- AN SLJ WHICH SPECIFIES INDIRECT ADDRESSING OR INDEXING.
- AN LMJ WHICH SPECIFIES INDEXING.
- AN EX WHICH INDIRECTLY REFERENCES AN LMJ OR SLJ.
- A TEST SKIP INSTRUCTION.

15.1.3.15. THE END STATEMENT

IF A DATA STATEMENT (AS OPPOSED TO A CONTROL STATEMENT -- @ IN COLUMN 1) WHICH IS UNRELATED TO THE COLLECTION FOLLOWS A @MAP STATEMENT, AN END STATEMENT PLACED AFTER THE LAST COLLECTOR SOURCE STATEMENT MAY BE NECESSARY TO TELL THE COLLECTOR TO DISREGARD THE DATA STATEMENTS FOLLOWING. THE FORMAT OF THE STATEMENT IS:

```
END
```

15.1.4. FUNCTIONAL ASPECTS

FUNCTIONALLY, THE COLLECTOR MUST INTERPRET THE SOURCE INPUT LANGAUGE, FIND THE ELEMENTS TO INCLUDE, AND GENERATE THE OUTPUT LISTING. THE FOLLOWING DESCRIPTION PERTAINS TO THE PROCEDURE FOR THE MORE GENERAL CASE OF A SEGMENTED PROGRAM; HOWEVER, A NON SEGMENTED PROGRAM CAN BE CONSIDERED AS BEING A SEGMENTED PROGRAM WITH ONLY A MAIN SEGMENT.

INITIALLY, PARAMETERS FROM THE @MAP (OR @XQT) CONTROL STATEMENT ARE OBTAINED AND INTERPRETED. ALL OF THE COLLECTOR CONTROL STATEMENTS ARE INTERPRETED AND SAVED IN TABLES INTERNAL TO THE COLLECTOR. DIAGNOSTIC MESSAGES ARE GIVEN WHERE APPROPRIATE. ELEMENTS NAMED ON IN STATEMENTS BUT NOT PRECEDED BY A FILE NAME ARE MAINTAINED IN A LIST APART FROM ELEMENTS NAMED WITH A DESIGNATED FILE NAME.

ELEMENTS ARE ADDED TO THE COLLECTION IN TWO STEPS. THE FIRST STEP INVOLVES FINDING THE ELEMENTS EXPLICITELY NAMED ON IN STATEMENTS, AND PROCESSING THE INFORMATION CONTAINED IN THE PREAMBLE SECTION OF EACH ELEMENT. IN ADDITION TO THE THE EXPLICITELY NAMED ELEMENTS, ALL THE ELEMENTS IN THE RUN'S TEMPORARY PROGRAM FILE (TPF$) MAY BE AUTOMATICALLY PROCESSED IN STEP ONE. WHEN TPF$ HAS BEEN PREPARED BY THE @PREP STATEMENT (A BLANK NAME FIELD ON A @PREP STATEMENT ALWAYS IMPLIES TPF$), THE AUTOMATIC PROCESSING OF ITS ELEMENTS IS INHIBITED. TPF$ ELEMENTS AUTOMATICALLY PROCESSED ARE ADDED TO THE PROGRAM ONLY IF REFERENCES ARE MADE TO THEM, OR IF THERE ARE NO IN STATEMENTS AT ALL (WHICH THE COLLECTOR TREATS AS AN IMPLIED: IN TPF$).

TPF$ IS ALWAYS THE FIRST LIBRARY SEARCHED FOR ELEMENTS EXPLICITELY NAMED BUT WITHOUT FILE DESIGNATION. (THE COLLECTOR INCLUDES TPF$ AS THE FIRST ENTRY IN ITS INTERNAL LIB TABLE, AS IF THERE HAD BEEN A LIB TPF$ STATEMENT.) FILES ACTUALLY NAMED ON LIB STATEMENTS MAKE UP THE SECOND LEVEL OF SEARCHING. THE LAST LEVEL OF SEARCHING

COVERS SYS$*RLIB$ (SEE EXECUTIVE CONTROL LANGUAGE CHAPTER: SYSTEM LIBRARIES). (SYS$*RLIB$ IS @PREP'ED AT SYSTEM BOOT TIME, AND THE COLLECTOR INCLUDES IT AUTOMATICALLY AS THE LAST ENTRY OF THE INTERNAL LIB TABLE, EXCEPT IN AN R-OPTION @MAP).

THE PREAMBLE OF AN ELEMENT INCLUDES THE DEFINITION OF EACH ENTRY POINT IN THE ELEMENT, THE LENGTH OF EACH LOCATION COUNTER USED, EVERY SYMBOL YET UNDEFINED IN THE ELEMENT, AND COMMON BLOCKS DEFINED BY THE ELEMENT. IN PROCESSING THE PREAMBLES, THE ENTRY POINTS OF THE ELEMENT ARE PUT INTO THE INTERNAL COLLECTOR EP TABLE. THE UNDEFINED SYMBOLS ARE LINKED TO AN ENTRY POINT BY THE SAME NAME IN THE EP TABLE, OR ADDED TO THE LIST OF SYMBOLS YET UNDEFINED (THE UNDE LIST) FROM ELEMENTS PREVIOUSLY PROCESSED. SYMBOLS ARE REMOVED FROM THE UNDE LIST AS ENTRY POINTS OF THE SAME NAMES ARE ENCOUNTERED IN PROCESSING PREAMBLES.

THE SECOND STEP IN ADDING ELEMENTS TO THE COLLECTION INVOLVES SEARCHING LIBRARIES FOR ELEMENTS WITH ENTRY POINTS OF THE SAME NAMES AS THOSE IN THE UNDE LIST, ACCUMULATED IN PROCESSING PREAMBLES OF INCLUDED ELEMENTS. ONLY LIBRARIES THAT HAVE BEEN @PREP'ED (ENTRY POINT TABLE PREPARED) WILL BE SEARCHED IN THE SECOND STEP. EVERY UNDEFINED SYMBOL CURRENTLY IN THE UNDE LIST IS LOOKED FOR IN EACH LIBRARY. WHEN AN ELEMENT IS FOUND WITH AN ENTRY POINT BY THE SAME NAME, THE PREAMBLE OF THAT ELEMENT IS THEN ALSO PROCESSED AS DESCRIBED ABOVE. THE UNDE LIST IS PROCESSED FROM TOP TO BOTTOM WITH NEW SYMBOLS ADDED TO THE BOTTOM. TPF$ IS SEARCHED FIRST, THEN FILES NAMED ON LIB STATEMENTS, AND THEN SYS$*RLIB$.

THE ORDER OF THE APPEARANCE OF USER-SPECIFIED ELEMENTS IN ANY SEGMENT OF A PROGRAM WILL BE THE SAME AS THAT IN WHICH THEY WERE SPECIFIED SO LONG AS EACH WAS SPECIFICALLY NAMED ON AN IN STATEMENT. WHEN ALL ELEMENTS WITHIN A FILE ARE INCLUDED IN A SEGMENT, THE ORDERING OF THE ELEMENTS WITHIN A GROUP SO SPECIFIED WILL BE RANDOM. AN ELEMENT INCLUDED BY A LIBRARY SEARCH APPEARS IMMEDIATELY PRECEDING THE USER SPECIFIED ELEMENTS OF THE SEGMENT IN THE PATH OF ALL SEGMENTS REFERENCING THE ELEMENT.

THE MOST EFFICIENT COLLECTION RESULTS WHEN EVERY ELEMENT DESIRED IN THE COLLECTION IS EXPLICITELY NAMED, INCLUDING FILE NAME. THE REASON FOR THIS IS THE @PREP REQUIREMENTS AND LIBRARY SEARCHES ARE ELIMINATED.

THE FIRST ADDRESS OF THE INSTRUCTION AREA IS ASSIGNED 01000 (OCTAL). THE ADDRESS OF THE DATA AREA IS ALWAYS GREATER THAN THE HIGHEST ADDRESS OF ANY REENTRANT PROCESSOR IN THE SYSTEM. ODD LOCATION COUNTERS OF AN ELEMENT (1,3,5.ETC.) ARE ASSIGNED TO THE INSTRUCTION AREA. EVEN NUMBERED COUNTERS (0,2,4 ETC.) ARE ASSIGNED TO THE DATA AREA. BLANK COMMON IS ASSIGNED TO THE DATA AREA OF THE MAIN SEGMENT. A NAMED COMMON BLOCK IS ATTACHED TO THE SEGMENT (IF NOT NAMED ON AN IN STATEMENT) IN THE PATH TO THE MAIN SEGMENT OF ALL SEGMENTS REFERENCING IT.

SYMBOLIC NAMES OF EXTERNAL DEFINITIONS AND EXTERNAL REFERENCES (SEE COMMENTS ON THE COLLECTOR DEF AND REF STATEMENTS), SEGMENT NAMES, QUALIFIER NAMES, FILE NAMES, ELEMENT NAMES, VERSION NAMES, AND COMMON BLOCK NAMES, INSOFAR AS COLLECTOR RESTRICTIONS ARE CONCERNED, MAY BE UP TO TWELVE CHARACTERS IN LENGTH AND MAY CONTAIN ANY COMBINATION OF ALPHABETIC, NUMERIC, $ OR HYPHEN CHARACTERS.

15.1.5. COLLECTOR DEFINED SYMBOLS

IN ADDITION TO THE COLLECTOR DEFINED SYMBOLS ENTRY$, COMMN$ AND XREF$, THREE MORE SYMBOLS ARE AVAILABLE TO THE USER. THE TAG LASTD$ IS GIVEN THE VALUE OF THE LAST ADDRESS OF THE DATA AREA AT COLLECTION TIME. LIKEWISE, LASTI$ IS THE LAST ADDRESS OF THE INSTRUCTION AREA. THE ADDRESS OF THE SEGMENT LOAD TABLE IS MADE AVAILABLE TO THE DIAGNOSTIC SYSTEM BY DEFINING THE TAG SLT$. THIS TAG SHOULD BE OF LITTLE INTEREST TO THE USER.

15.1.6. PROGRAM SEGMENTATION AND LOADING

THE CAPABILITIES OFFERED BY THE COLLECTOR FOR SEGMENTING A PROGRAM PROVIDE THE USER WITH A STRAIGHTFORWARD MEANS OF CONSTRUCTING OVERLAY SEGMENTS EXACTLY AS HE

WISHES. FOR EACH SEGMENT THE USER PREPARES IN AND NOT CONTROL STATEMENTS TO SPECIFY THE RELOCATABLE ELEMENTS TO BE INCLUDED WITHIN THAT PARTICULAR SEGMENT. THESE STATEMENTS ARE PRECEDED BY A SEG OR DSEG STATEMENT SPECIFYING THE NAME OF THE SEGMENT AND ITS LOGICAL ORIGIN. WHEN A SEGMENTED PROGRAM IS CALLED FOR BY THE @XQT EXECUTIVE CONTROL STATEMENT, ONLY THE MAIN SEGMENT IS INITIALLY LOADED. THERE ARE TWO WAYS BY WHICH OTHER SEGMENTS MAY BE LOADED. THE DIRECT METHOD IS WHENEVER THE USER MAKES A DIRECT CALL TO THE SEGMENT LOADER SPECIFYING THE SEGMENT TO BE LOADED AND THE LOCATION TO WHICH CONTROL IS TO BE TRANSFERRED AFTER IT IS LOADED.

THE DIRECT METHOD OF LOADING IS ACCOMPLISHED WITH THE PROCEDURE CALL:

```
        L$OAD      NAME,JUMP,CLEAR,ADDRESS
```

WHERE 'NAME' IS THE NAME OF THE SEGMENT TO LOAD. THE PARAMETER 'JUMP' INDICATES THE LOCATION TO WHICH CONTROL WILL BE TRANSFERRED WHEN THE LOAD IS COMPLETE. THE USE OF AN EXIT FOR THE LOADER OF THIS FORM IS REQUIRED SINCE THE SEGMENT IN WHICH THE CALLING SEQUENCE RESIDES MAY BE OVERLAYED BY THE SEGMENT BEING LOADED. IF THIS FIELD IS OMITTED, CONTROL IS RETURNED FOLLOWING THE CALL. WHEN THE FIELD 'CLEAR' IS NONZERO THE AREA OCCUPIED BY THE SEGMENT BEING LOADED IS NOT CLEARED TO ZERO. THE OPTIONAL FIELD 'ADDRESS' IS THE ADDRESS TO LOAD A RELOCATABLE SEGMENT. IF THIS FIELD IS OMITTED WHEN LOADING A RELOCATABLE SEGMENT, THE ADDRESS MUST BE IN A2 BEFORE THE CALL IS MADE.

THE CALL

```
        L$OAD      NAME,JUMP
```

WILL GENERATE:

```
        L,U        A1,JUMP
        L,U        A0,NAME
        ER         LOAD$
```

THE INDIRECT METHOD PROVIDES FOR AUTOMATIC LOADING OF A SEGMENT REFERENCED BY ANY JUMP TYPE COMMAND TO AN INSTRUCTION (IBANK) AREA, WHENEVER THE SEGMENT IS NOT IN CORE. SEGMENTS TO BE LOADED BY THE INDIRECT METHOD MUST BE SO MARKED ON THE SEG OR DSEG STATEMENT. THE MECHANICS FOR SUCH LOADING ARE SET UP BY THE COLLECTOR AND CARRIED OUT BY THE SEGMENT LOADER. THE COLLECTOR REPLACES THE ADDRESS PORTION OF THE JUMP COMMAND WITH THE ADDRESS OF A INDIRECT LOAD TABLE ENTRY. THE INDIRECT LOAD TABLE PERFORMS AN SLJ TO THE INDIRECT LOAD ROUTINE WHICH IN TURN PERFORMS AN ER TO THE SEGMENT LOADER (IF THE SEGMENT IS NOT ALREADY LOADED) AND JUMPS TO THE LOCATION OF THE EXTERNALLY DEFINED SYMBOL. ALL REGISTERS ARE PRESERVED BY THE PROCESS. THE INDIRECT LOAD TABLE AND THE SEGMENT LOAD TABLE ARE ASSIGNED TO THE DATA AREA OF THE MAIN SEGMENT.

IF INDIRECT LOADING IS USED, THE REFERENCE MAY NOT BE MADE TO THE EXTERNAL SYMBOL WITH AN OFFSET.

THE AREA ASSIGNED A DYNAMIC SEGMENT MAY BE RELEASED WITH THE PROCEDURE CALL:

```
        D$REL      NAME,JUMP
```

WHERE 'NAME' SPECIFIES THE DYNAMIC SEGMENT TO RELEASE AND 'JUMP' SPECIFIES THE LOCATION TO TRANSFER CONTROL WHEN THE SEGMENT IS RELEASED. ANY REFERENCE TO THE DYNAMIC SEGMENT WITHOUT AGAIN CAUSING IT TO BE LOADED MAY RESULT IN A MEMORY PROTECTION VIOLATION.

THE ABOVE PROCEDURE CALL GENERATES THE FOLLOWING INSTRUCTIONS:

```
        L,U        A1,JUMP
        L,U        A0,NAME
        ER         DREL$
```

THE FOLLOWING EXAMPLE IS GIVEN TO ILLUSTRATE THE USE OF THE SEGMENTATION FACILITIES OF THE COLLECTOR.

ASSUME FILEA HAS THE FOLLOWING RELOCATABLE ELEMENTS WITH THE INDICATED REFERENCES OUTSIDE OF THE FILE:

```
FILEA ELEMENTS   REFERENCES OUTSIDE OF FILEA REQUIRED
NAME/VERSION     FILE.NAME/VERSION

MAIN             FILEA.A1,B1,F1
A1/A
A2/A             LIB1.SIN/X
A3/A             LIB2.COS/X
B1/B             LIB1.SQRT/X
B2/B
B3/B
C1/C             LIB1.SQRT/X
C2/C
D1/D             LIB2.CAT/Y
D2/D
E1/E             LIB2.CAT/Y
E2/E
F1
F2
G1/G             LIB1.SIN/X
G2/G             LIB2.COS/X
G3/G
```

A PARTICULAR COLLECTION SETUP FOR SEGMENTING A PROGRAM FROM THIS FILE MIGHT BE AS FOLLOWS:

```
@PREP       LIB1.
@PREP       LIB2.
@MAP,LI     MAPSYM,MAPABS
SEG     MAIN
IN      FILEA.MAIN
SEG     A*, (MAIN)
IN      FILEA.A1/A,.A2/A,.A3/A
SEG     B*, (A)
IN      FILEA.B1/B,.B2/B,.B3/B
SEG     C*, B
IN      FILEA.C1/C,.C2/C
SEG     D*, (B,C)
IN      FILEA.D1/D,.D2/D
SEG     E*, D
IN      FILEA.E1/E,.E2/E
DSEG    F*, (D,G)
IN      FILEA.F1,.F2
SEG     G*, (MAIN)
IN      FILEA.G1/G,.G2/G
LIB     LIB1,LIB2
@XQT
```

THIS PARTICULAR SET OF CONTROL STATEMENTS WOULD RESULT IN THE MEMORY STRUCTURE ILLUSTRATED BELOW. THE HORIZONTAL COORDINATE IS USED TO DENOTE INCREASING MEMORY ADDRESSES FROM LEFT TO RIGHT. SEGMENTS WITH COMMON HORIZONTAL COORDINATES MAY NOT BE IN MEMORY SIMULTANEOUSLY.

```
         INSTRUCTION AREA MEMORY MAP
 01000                                      K            M
          CAT       --B1-B2-B3----
          SQRT                    -D1-D2--
  COS    -A1-A2-A3--
  SIN                             -E1-E2-----
 -MAIN---           --C1-C2--               --F1--F2-----
         ---G1---G2-------------------------


         DATA AREA MEMORY MAP
N                                                O            P
               CAT      --B1-B2-B3---
```

```
    ILD$   SQRT                    -D1-D2--
    COS   -A1-A2-A3--              -E1-E2-----
    SIN             ---C1-C2-------
-LT-BC--MAIN--                              --F1-F2---------
          --G1--G2------
```

ILD$ IS THE NAME OF THE INDIRECT LOAD ROUTINE AND IS ALWAYS IN THE MAIN SEGMENT.
LT REPRESENTS THE SEGMENT LOAD TABLE AND INDIRECT LOAD TABLE GENERATED BY THE COLLECTOR.
BC STANDS FOR BLANK COMMON.
NOTE THAT THE ELEMENT CAT IS ATTACHED TO SEGMENT A AND NOT SEGMENT B OR SEGMENT C.
THE CORE AREA FROM K TO M AND FROM O TO P IS RELEASED TO THE EXECUTIVE SYSTEM WHENEVER THE SEGMENT F IS RELEASED BY THE PROGRAM.

THE FIRST ADDRESS OF THE DATA AREA ,N, IS GREATER THAN OR EQUAL TO THE MINIMUM DATA AREA ADDRESS SPECIFIED AT SYSTEM GENERATION TIME. THIS ENABLES THE DATA AREA TO BE LINKED TO REENTRANT PROCESSORS. N IS ALWAYS A MULTIPLE OF 01000.

15.2. THE PROCEDURE DEFINITION PROCESSOR

15.2.1. GENERAL

THE PROCEDURE DEFINITION PROCESSOR (PDP) ACCEPTS SOURCE LANGUAGE DEFINING 1108 ASSEMBLER, FORTRAN, OR COBOL PROCEDURES AND BUILDS AN ELEMENT IN THE USER DEFINED PROGRAM-FILE. THESE PROCEDURES MAY SUBSEQUENTLY BE REFERENCED IN AN ASSEMBLY OR COMPILATION WITHOUT DEFINITION.

ONE TABLE WILL BE GENERATED FOR EACH TYPE OF PROCEDURE (ASSEMBLER, FORTRAN, COBOL) IN A PROGRAM-FILE. THIS TABLE WILL CONTAIN ANY LABELS THAT ARE DEFINED EXTERNALLY TO THE PROCEDURE. IN THE CASE OF FORTRAN AND COBOL PRECEDURES, THESE WILL BE THE LABELS ON THE PROC LINE. IN THE CASE OF ASSEMBLER PROCEDURES THESE WILL BE THOSE LABELS DEFINED EXTERNALLY (TRAILING ASTERISK AFTER LABEL) ON THE PROC AND NAME LINES OF A FIRST LEVEL PROCEDURE. FOR EVERY LABEL ENTERED IN A TABLE THE LOCATION OF THE PROCEDURE OR NAME LINE IS NOTED. WHEN A CALL IS MADE FOR A PROCEDURE IN A SOURCE PROGRAM THE SYSTEM AUTOMATICALLY RETRIEVES THE PROCEDURE. IF MORE THAN ONE PROCEDURE OF THE SAME TYPE (ASSEMBLER, FORTRAN, COBOL) HAS THE SAME LABEL AN ENTRY WILL BE MADE IN THE TABLE FOR EACH PROCEDURE, BUT A CALL ON THAT PROCEDURE WILL PRODUCE THE LAST ONE ENTERED.

THE PROCEDURE DEFINITION PROCESSOR IS CALLED WHENEVER A PDP EXECUTIVE CONTROL STATEMENT IS ENCOUNTERED. THE FORMAT OF THIS STATEMENT IS AS FOLLOWS:

```
@PDP,OPTIONS ELEMENT1,ELEMENT2
```

THE FIELD 'OPTIONS' MAY CONTAIN ANY OF THE FOLLOWING LETTERS TO INDICATE DIRECTIONS TO THE PROCEDURE DEFINITION PROCESSOR:

F INDICATES A FORTRAN PROCEDURE ELEMENT.
C INDICATES A COBOL PROCEDURE ELEMENT.
L PRODUCE A COMPLETE LISTING OF THE OUTPUT ELEMENT WITH
A ACCEPT THE RESULTS AS CORRECT EVEN IF ERRORS ARE DETECTED.
X ABORT THE REMAINDER OF THE RUN IF ANY ERRORS ARE DETECTED.
I INTRODUCE SOURCE LANGUAGE ELEMENT INTO PROGRAM FILE FROM THE CONTROL STREAM.
U PRODUCE A NEW CYCLE OF THE SOURCE LANGUAGE ELEMENT.

WHEN THE 'F' OPTION IS PRESENT, PDP ASSUMES IT IS INSERTING OR UPDATING A FORTRAN PROCEDURE ELEMENT. WHEN THE 'C' OPTION IS PRESENT, THE PDP ASSUMES IT IS INSERTING OR UPDATING A COBOL PROCEDURE ELEMENT. WHEN NEITHER OPTION IS PRESENT, THE PDP

ASSUMES IT IS INSERTING OR UPDATING AN 1108 ASSEMBLER PROCEDURE ELEMENT.

THE FIELDS 'ELEMENT1' AND 'ELEMENT2' ARE OF THE STANDARD FORMAT FOR SYMBOLIC ELEMENT DESCRIPTION. THE FIELD 'ELEMENT1' NORMALLY IDENTIFIES AN INPUT ELEMENT BY FILE, ELEMENT NAME, VERSION, AND CYCLE. HOWEVER, WHEN THE I OPTION IS USED, 'ELEMENT1' IS THE IDENTIFICATION TO BE GIVEN TO THE NEW PROGRAM FILE ELEMENT.

THE FIELD 'ELEMENT2' IS USED AS THE IDENTIFICATION OF A NEW OUTPUT ELEMENT WHENEVER IT APPEARS. STANDARD SYSTEM DROPOUT RULES APPLY TO BOTH 'ELEMENT1' AND 'ELEMENT2'.

THE I OPTION IS USED SOLELY TO INTRODUCE SOURCE CARD IMAGES INTO A PROGRAM FILE . WHEN APPLYING CORRECTIONS TO AN ELEMENT, THE I OPTION IS NOT PERMITTED .

PDP WILL PERMIT PROCESSING PROCEDURAL ELEMENTS FROM A TAPE FILE THAT IS IN ELEMENT FILE FORMAT . FURTHERMORE, CORRECTIONS TO THIS ELEMENT ARE PERMITTED IF A SOURCE ELEMENT IS PRODUCED IN A PROGRAM FILE. PDP WILL NOT ATTEMPT TO INTERPRET THE NAMES ON A CONTROL CARD; I.E., IT WILL MAKE NO EFFORT TO INSURE UNIQUENESS OR AVOID POSSIBLE DUPLICATION OF NAMES IN SPEC1 AND SPEC2 FIELDS.

CYCLING OF PROCEDURES IS PERMITTED. THE CYCLE NUMBER MAY BE INCREASED IF THE U OPTION IS ON, BUT ONLY ONE CYCLE IS RETAINED, AND WHEN THE PROCEDURE IS CALLED FOR INCLUSION , THE LATEST CYCLE IS GIVEN .
SOME EXAMPLES ON THE USE OF THE @PDP STATEMENT ARE AS FOLLOWS:

@PDP,L	A.B,C	PRODUCE A PROCEDURE ELEMENT FROM FILE A (IF IT IS A PF OR TAPE FILE) ELEMENT B AND PUT THE NEW SOURCE CODE IN THE TEMPORARY PROGRAM FILE. PRODUCE A COMPLETE LISTING.
@PDP,L	A.B,.C	PRODUCE A PROCEDURE ELEMENT FROM PROGRAM FILE A ELEMENT B , CALL IT ELEMENT C AND PUT IT IN PROGRAM FILE A . PRODUCE A COMPLETE LISTING . BOTH ELEMENTS B AND C ARE RETAINED . A TAPE FILE IS NOT PERMITTED FOR SPEC2 . PDP WILL ERROR OUT BUT WILL NOT ABORT UNLESS THE 'X' OPTION IS ON .
@PDP,L	A.B	PRODUCE A COMPLETE LISTING OF ELEMENT B FROM FILE A .
@PDP,U	A.B	IF FILE A IS A PROGRAM FILE ELEMENT B HAS ITS CYCLE INCREASED BY 1 , AND THE ELEMENT IS ENTERED INTO THE PROGRAM FILE NAMED 'A' . SPEC 2 MAY NOT DESIGNATE A TAPE FILE .
@PDP,I	AFILE.PROCS/AB	(PROCEDURE DEFINITIONS FOLLOWING THE @PDP ARE INTRODUCED INTO AFILE AS ELEMENT PROCS, VERSION AB, CYCLE 0.)
@PDP,U	BFILE.PAT/DE	(CORRECTIONS ARE APPLIED TO ELEMENT PAT, VERSION DE, LATEST CYCLE, IN BFILE TO PRODUCE AN UPDATED CYCLE OF THE SAME ELEMENT IN THE SAME

```
                                          FILE.)
   @PDP        AF.PR1,BF.PR2              (ANY CORRECTIONS FOLLOWING
                                          THE @PDP STATEMENT ARE MERGED
                                          WITH THE MOST RECENT CYCLE OF
                                          ELEMENT PR1 FROM FILE AF TO
                                          PRODUCE CYCLE 0 OF ELEMENT
                                          PR2 IN FILE BF.)
```

IF PDP IS PROCESSING ELEMENTS FROM A TAPE FILE, THE FILE MUST BE POSITIONED SO THE LABEL BLOCK WILL BE READ IN . IF THE NAME IN THE LABEL BLOCK DOES NOT AGREE WITH THE SPEC1 ELEMENT NAME, PDP TAKES THE ERROR EXIT .

15.2.2. ASSEMBLER PROCEDURES

THE PDP WILL ACCEPT SOURCE LANGUAGE DEFINING 1108 ASSEMBLER PROCEDURES THAT ARE TO BE CALLED BY NAME THROUGH SOURCE INPUT TO THE 1108 ASSEMBLER. WHEN A GIVEN PROCEDURE IS CALLED, THE ASSEMBLER WILL USE ANY DEFINITION OF THAT PROCEDURE INCLUDED IN ITS INPUT SOURCE LANGUAGE. IF NONE IS PRESENT, THE PROGRAM FILE FROM WHICH THE SOURCE INPUT WAS OBTAINED WILL BE INTERROGATED FOR A DEFINITION. IF NO DEFINITION IS FOUND THERE, THE SYSTEMS LIBRARY IS INTERROGATED. IF NO DEFINITION IS FOUND, THE ASSEMBLER GIVES AN ERROR INDICATOR.

THE 1108 ASSEMBLER PROCEDURE HAS THE FORM:

```
        AA*    PROC
               -
        BB*    NAME
               -
               -
               END
```

WHERE 'PROC' AND 'NAME' ARE THE STANDARD ASSEMBLER DIRECTIVES. AN ENTRY WILL BE MADE IN THE PROGRAM-FILE ASSEMBLER PROCEDURE TABLE FOR LABELS 'AA' AND 'BB'.

THE DEF DIRECTIVE.

THE FUNCTION OF THE DEF DIRECTIVE IS TO PRECEDE THE PROC DIRECTIVE WITH ANY NUMBER OF EQU OR FORM DIRECTIVES FOR A PROCEDURE. THE ENTRY IN THE ASSEMBLER PROCEDURE TABLE WILL THEN POINT TO THE DEF LINE RATHER THAN THE PROC LINE. THE PURPOSE OF THE DEF DIRECTIVE IS TO SAVE PROCEDURE STORAGE SPACE IN CORE AT ASSEMBLY TIME. THUS:

```
               DEF
        A0     EQU    12
        A1     EQU    13
               -
               -
               -
        A15    EQU    27
        AREG*  PROC    0,0
               END
```

WILL ALLOW A0,A1,...,A15 TO BE DEFINED IN A PROGRAM BY CALLING THE PROC AREG, YET ONLY TWO WORDS OF PROCEDURE STORAGE IN CORE ARE USED.

15.2.3. FORTRAN PROCEDURE

A FORTRAN PROCEDURE CONTAINS FORTRAN SOURCE LANGUAGE THAT IS TO BE INCLUDED IN A COMPILATION BY USE OF THE FORTRAN INCLUDE STATEMENT. WHEN SUCH A CALL IS MADE THE PROGRAM-FILE FROM WHICH THE SOURCE INPUT WAS OBTAINED IS INTERROGATED FOR THE PROCEDURE. IF NONE, THE SYSTEM LIBRARY IS INTERROGATED. IF NO DEFINITION IS FOUND, THE COMPILER GIVES AN ERROR INDICATION.

THE FORTRAN PROCEDURE HAS THE FORM:

```
        AA     PROC
               -
               (FORTRAN STATEMENTS)
               END
```

AN ENTRY WILL BE MADE IN THE PROGRAM-FILE FORTRAN PROCEDURE TABLE FOR THE LABEL 'AA'. THE END STATEMENT FOR FORTRAN PROCEDURES MUST BEGIN IN COLUMN TWO (2).

15.2.4. COBOL PROCEDURE

A COBOL PROCEDURE CONTAINS COBOL SOURCE LANGUAGE THAT IS TO BE INCLUDED IN A COMPILATION BY USE OF THE COBOL INCLUDE AND COPY VERBS. WHEN SUCH A CALL IS MADE, THE PROGRAM-FILE FROM WHICH THE SOURCE INPUT WAS OBTAINED IS INTERROGATED FOR THE PROCEDURE. IF NONE, THE SYSTEMS LIBRARY IS INTERROGATED. IF NO DEFINITION IS FOUND, THE COMPILER GIVES AN ERROR INDICATOR.

THE COBOL PROCEDURE HAS THE FORM:

```
      AA    PROC
            -
            (COBOL STATEMENTS)
            -
            END
```

AN ENTRY WILL BE MADE IN THE PROGRAM-FILE COBOL PROCEDURE TABLE FOR THE LABEL 'AA'. 'END' MUST BEGIN IN COLUMN 2 .

15.3. THE @ED STATEMENT

THE 'ED' STATEMENT MAY BE USED TO INSERT IMAGES INTO A PROGRAM FILE ELEMENT OR DATA FILE. IT MAY ALSO BE USED TO CHANGE THE IMAGE OR A PORTION OF THE IMAGE, OR IT MAY BE USED TO DELETE THE ENTIRE IMAGE.

THE COOPERATION OF DR. J. SALTZER IN PROVIDING PROGRAM LISTINGS IS HEREWITH EXPRESSLY APPRECIATED.

THE 'ED' PROCESSOR,BY NATURE OF ITS TECHNIQUE, WILL NOT ACCEPT CORRECTION LINES OF THE FORM -N,M AS DO OTHER SYSTEM PROCESSORS. THERE ARE TWO BASIC MODES IN WHICH THE PROCESSOR OPERATES: (1) THE INPUT MODE, AND (2) THE EDIT MODE.

THE INPUT FILE NAMED AS SPECIFICATION FIELD ONE MUST HAVE BEEN PREVIOUSLY ASSIGNED OR BE CATALOGUED. SPECIFICATION FIELD TWO FILENAME NEED NOT BE ASSIGNED OR CATALOGUED. 'ED' WILL ASSIGN THE FILE ON FASTRAND, BUT IT IS UP TO THE USER TO PERFORM THE ACTUAL CATALOGING OF THE FILE IF DESIRED.

THE FORMAT OF THE PROCESSOR CALL IMAGE IS A FOLLOWS:

```
@ED,OPTIONS  NAME1,NAME2 .
```

NAME1(INPUT) AND NAME2(OUTPUT) ARE THE NAMES DEFINING THE PROGRAM FILE ELEMENT, AND VERSION NAMES BY WHICH THE UNIT IS KNOWN. THE THREE OPTIONS WHICH ARE AVAILABLE ARE:

I- ENTER 'ED' WITH THE INPUT MODE. NAME2 IS IGNORED, AND NAME1 IS THE NAME OF THE OUTPUT FILE CONTAINING THE SOURCE IMAGES.

D- IF THE 'D' OPTION IS PRESENT NAME1 AND/OR NAME2 ARE THE NAMES OF A DATA FILE- AS OPPOSED TO A PROGRAM FILE.

U- CREATE A NEW CYCLE OF THE DATA FILE OR PROGRAM FILE ELEMENT. NAME2 IS IGNORED.

E- ECHO (I.E., PRINT OUT) ALL COMMANDS FROM THE INPUT STREAM. THEY WILL BE INDENTED AND FLAGGED WITH THREE ASTERISKS. THIS MAY BE PARTICULARLY USEFUL FOR CARD INPUT (BATCH) RUNS.

- ABSENCE OF ALL OF THE OPTIONS I, D, AND U REQUIRES NAME1 AS THE INPUT FILE NAME AND NAME2 AS THE OUTPUT FILE NAME.

NOTE: THE FOLLOWING DEFINES THE ONLY DIFFERENCE IN THE UTILIZATION OF 'ED' FROM A

TERMINAL (SUCH AS A TELETYPE) AND FROM A CARD READER. A CARRIAGE RETURN ON A TTY SIGNIFIES THE END OF THE IMAGE, WHILE ON A CARD READER THE IMAGE IS DETERMINED BY THE IMAGE LENGTH PASSED ON BY READ$.

A SINGLE CR ON A TELETYPE OR A BLANK CARD CHANGES 'ED' FROM THE INPUT TO THE EDIT MODE, OR VICE VERSA.

THE FOLLOWING COMMANDS ARE AVAILABLE TO 'ED' TO PERMIT THE MANIPULATION OF IMAGES IN A FILE OR ELEMENT. NOTE THAT COMMANDS A THROUGH J MAY BE WRITTEN COMPLETELY OR ABBREVIATED BY A SINGLE LETTER .

```
     COMMAND                        FUNCTION
     -------    ------------------------------------------
A.   TOP         POSITION A POINTER TO THE TOP OF A FILE. IF
     T           THE CURRENT POSITION IS NOT AT THE TOP OR
                 BOTTOM OF A FILE REMAINING IMAGES FROM THE
                 INPUT ARE COPIED TO THE OUTPUT FILE, AND THE
                 POINTER IS POSITIONED AT THE 'TOP' OF THE
                 OUTPUT FILE.

B,   BOTTOM      ANY IMAGES FROM THE OLD FILE (PLUS THOSE
     B           BEING PROCESSED) ARE COPIED TO THE NEW FILE.
                 'ED' THEN ENTERS THE INPUT MODE AND ANY
                 FOLLOWING IMAGES ARE WRITTEN TO THE NEW
                 FILE. THE INPUT MODE MAY BE TERMINATED BY
                 TYPING A CARRIAGE RETURN (CR).

C.   NEXT M     THE M IMAGES ARE COPIED FROM NAME1 TO NAME2.
     N M        THE MTH IMAGE IS NOT WRITTEN TO NAME2.

D.   DELETE M   THE CURRENT IMAGE IS SKIPPED AS ARE THE NEXT
     D M        M-1 IMAGES. THE POINTER INTO NAME1 IS ADVAN-
                CED M-1 IMAGES.

E.   INSERT(CR) THE INPUT MODE IS ENTERED.
     I(CR)
     INSERT LINE  THE LINE AS TYPED IS ACCEPTED FOLLOWING
     I LINE       THE CURRENT IMAGE.

F.   RETYPE LINE  THE LINE AS TYPED IS ACCEPTED,
     R LINE       AND THE CURRENT IMAGE IS REPLACED.

G.   PRINT N    THE CURRENT IMAGE IS PRINTED. IF N>1 SUCCEED-
     P N        ING IMAGES ARE PRINTED. THE PROCESS IS
                SIMILAR TO 'NEXT N-1'.

H.   FIND LINEMASK. EACH IMAGE IS SCANNED TO FIND A GROUP
     F LINEMASK     OF CHARACTERS WHICH CORRESPONDS TO THE
                    LINEMASK EXCLUDING BLANKS. OBSERVE THAT
                    CARD COLUMNS IN THIS COMMAND ARE IMPOR-
                    TANT. SEE EXAMPLE BELOW:
                    1   5   9
                    F    B H M -
                    THIS SAYS TO FIND A MASK STARTING IN
                    COLUMN 3 WHICH CORRESPONDS TO  B H M OR
                    BXHXM, ETC. THE POSITIONS OF THE COMMAND
                    WORD 'F ' OR 'FIND ' MUST NOT BE COUNTED
                    AS PART OF THE IMAGE. BLANKS IN THE LINE
                    MASK CORRESPOND TO CHARACTERS WHICH MAY
                    BE IGNORED.
                 IF NO FIND APPLIES THE SCAN CONTINUES TO END
                 OF FILE. A MESSAGE THEN IS GIVEN STATING
                 THAT END OF FILE WAS ENCOUNTERED.

I. LOCATE STRING
   L STRING     THIS COMMAND SCANS IMAGES FOR A MATCH ANY-
                WHERE WITH 'STRING'. STRING STARTS WITH THE
                SECOND CHARACTER FOLLOWING THE COMMAND.
                EXAMPLE:
                    L LA A9  WILL LOCATE:
                      LA   A9  OR,
                      L  A A9, BUT NOT  L  AA9.

                 IF NO FIND APPLIES THE SCAN CONTINUES TO
                 END OF FILE.
```

J. CHANGE *STRING1*STRING2* N G
C *STRING1*STRING2* N G

THIS COMMAND WILL EXAMINE 'N' IMAGES LOOKING FOR A CHARACTER BY CHARACTER MATCH TO STRING1 (INCLUDING BLANKS). WHEN A FIND IS MADE STRING1 IS REPLACED BY STRING2. IF G IS ABSENT REPLACEMENT OCCURS ON ONE STRING ONLY. IF G IS PRESENT EVERY APPEARANCE OF STRING1 IS REPLACED. ACCOUNT FOR DIFFERENT STRING LENGTHS IS MADE. IF N IS MISSING ONE IMAGE ONLY IS SCANNED. VIZ. C *A*B* G . THIS WILL CHANGE ALL A'S TO B'S IN THE IMAGE. IF A SAVE COMMAND IS IN EFFECT NO A'S WILL BE CHANGED AFTER THE SAVE COLUMN. ANY CHARACTER MAY BE USED TO DELIMIT STRING1 AND STRING2. G IN THE EXAMPLE IS THE CHARACTER G.

EXAMPLE 1:

```
@RUN
@ED,I    IN1
     AXR$.
START*.
      LA,U   A0,MESS1      . THE ELEMENT NAMED IN1
      ER     COMX          . WILL BE INSERTED IN THE
      ER     EXIT$         . TEMPORARY RUN FILE

MESS1 +0                   . TOGETHER WITH THESE
      +8,$+2               . COMMENTS WHEN 'ED' IS
      +0                   . TERMINATED
      'HI THERE'
      END START
      . BLANK CARD HERE (COMMENTS NOT PERMITTED)
TOP
N 4
CHANGE *X*$*
@EOF          ( ANY CONTROL IMAGE )
```

THE FOLLOWING 'ED' COMMANDS HAVE NO ABBREVIATIONS.

K. SAVE M — THIS COMMANDS RESTRICTS THE CHANGE COMMAND TO COLUMNS UP TO AND INCLUDING COLUMN M. IT MIGHT BE DESIRABLE FOR EXAMPLE, TO PROHIBIT CHANGING ID'S CONTAINED IN COLUMNS 73-80. IF M IS NOT SPECIFIED ON THE COMMAND 72 IS ASSUMED.

L. VERIFY — THIS COMMAND NULLIFIES THE 'BRIEF' COMMAND. ALL LINES IN THE WORK AREA 'ED' USES ARE PRINTED.

M. BRIEF — THIS COMMAND TURNS OFF THE VERIFY MODE. IF NOT SPECIFIED, THE VERIFY MODE IS ALWAYS ON.

N. SPLIT NAME3 — A PARTIAL FILE (OR ELEMENT) MAY BE FILED UNDER NAME3 BY USING THIS COMMAND. ALL IMAGES IN THE INPUT FILE PRECEEDING THE ONE CURRENTLY BEING PROCESSED ARE FILED IN NAME3. SUCCEEDING IMAGES ARE NOT FILED IN NAME3.

O. ADD NAME4 — THIS COMMAND IS USED TO ENTER SUCCESSIVE IMAGES FROM THE FILE NAMED NAME4 (EITHER A PREGRAM FILE ELEMENT OR DATA FILE) TO THE END OF THE FILE 'ED' IS PROCESSING.

```
P. END           TERMINATE 'ED', DO NCT FILE THE ELEMENT.

Q. SEQ,ID I SV   THIS COMMAND MAY BE USED IF IT IS DESIRED TO
                 PROVIDE SEQUENCE NUMBERS TO THE IMAGES. ID IS
                 THE CONSTANT IDENTIFICATION TO BE SUPPLIED.
                 'I' IS THE INCREMENT TO BE USED IN COUNTING.
                 'SV' IS THE STARTING VALUE. IF THEY ARE NOT
                 SUPPLIED 'SV'= 100 AND 'I'= 100.

R. TAB Q         THIS SETS ANY CHARACTER 'Q' AS THE TAB
                 CHARACTER TO BE USED IN THE IMAGE STRINGS OF
                 THE TYPE AND INSERT COMMANDS OR IN INPUT
                 IMAGES.

S. SET T1 T2 TN
                 THIS COMMAND SETS TAB STOPS AT TN (1-80) ALONG
                 THE IMAGE.

T. GOTO N        THE FIRST N-1 IMAGES ARE PUT INTO A WORK FILE.
   GO N          THE POINTER IS POSITIONED AT IMAGE N.
```

16. PROCESSOR INTERFACE ROUTINES

16.1. GENERAL

TO ENHANCE THE MODULARITY OF THE UNIVAC 1108 EXECUTIVE SYSTEM, A SET OF ROUTINES IS PROVIDED WHICH ALLOWS ALL PROCESSORS TO MAINTAIN A STANDARD INTERFACE WITH THE EXECUTIVE SYSTEM. THIS MAKES INCORPORATION OF ADDITIONAL PROCESSORS BY ANY INSTALLATION A RELATIVELY MINOR TASK INSOFAR AS INTEGRATING THEM INTO THE SOFTWARE SYSTEM IS CONCERNED. THESE ROUTINES ARE AVAILABLE FROM THE SYSTEMS LIBRARY AND, FOR ANY PROCESSOR, ONLY THESE ROUTINES WHICH ARE ACTUALLY REQUIRED WILL BE ALLOCATED WITH THE PROCESSORS.

IN GENERAL, PROCESSORS ARE CONCERNED WITH A SOURCE INPUT, CORRECTION INPUT, SOURCE OUTPUT, AND RELOCATABLE OUTPUT. WHEN USING THE PROCESSOR INTERFACE ROUTINES, THE PROCESSOR NEED ONLY BE CONCERNED WITH REQUESTING THE NEXT SOURCE IMAGE, OR OUTPUTTING A RELOCATABLE WORD. THE SOURCE INPUT ROUTINE (SIR) PROVIDES THE PROCESSOR WITH THE NEXT SOURCE IMAGE WHILE AUTOMATICALLY MERGING CORRECTIONS, IF NECESSARY, AND PROVIDING AN UPDATED SOURCE ELEMENT WHEN REQUESTED. THE RELOCATABLE OUTPUT ROUTINE (ROR) FORMATS AND OUTPUTS THE RELOCATABLE BINARY IN STANDARD BLOCK AND ITEM SIZES ACCEPTABLE TO THE COLLECTOR.

PERTINENT INFORMATION FROM THE PROCESSOR CALL STATEMENT, SUCH AS FILE NAMES, ELEMENT NAMES, AND OPTIONS, IS PASSED ALONG TO THE PROCESSOR WHEN IT IS INITIATED BY THE SUPERVISOR. A PROCESSOR CALL STATEMENT HAS THE FOLLOWING FORMAT:

```
@PROCESSOR,OPTIONS    ELEMENT1,ELEMENT2,ELEMENT3,...
```

16.2. SOURCE INPUT ROUTINE (SIR)

THE SOURCE INPUT ROUTINE INPUTS TO THE PROCESSOR THE SYMBOLIC ELEMENT IDENTIFIED BY THE 'ELEMENT1' FIELD OF THE CALL STATEMENT. THIS FIELD HAS THE FORM:

```
FILE.ELEMENT(CYCLE)
```

WHERE 'FILE' IS A EITHER A PROGRAM-FILE OR A MAGNETIC TAPE FILE, AND 'ELEMENT' IDENTIFIES THE SYMBOLIC ELEMENT WITHIN THAT FILE. IF THE FILE IS A TAPE FILE THE ELEMENT MUST BE IN THE STANDARD TAPE ELEMENT FORMAT.

IF THE 'I' OPTION IS USED IN THE CALL STATEMENT, THE SOURCE INPUT WILL BE TAKEN FROM THE CONTROL STREAM. IN THIS CASE, THE 'ELEMENT1' FIELD IDENTIFIES THE SOURCE OUTPUT ELEMENT AND PROGRAM-FILE.

IF THE SOURCE INPUT IS FROM A PROGRAM-FILE OR TAPE, CORRECTIONS MAY FOLLOW THE CALL STATEMENT IN THE CONTROL STREAM. SIR WILL MERGE THESE CORRECTIONS WITH THE SYMBOLIC INPUT BY LINE NUMBER. (SEE CHAPTER 5.6.3)

IF THE 'I' OR 'U' OPTION IS NOT USED IN THE CALL STATEMENT, THE CORRECTIONS FOLLOWING THE CALL STATEMENT ARE APPLIED AGAINST THE SOURCE INPUT ELEMENT SPECIFIED IN THE 'ELEMENT1' FIELD. A NEW SOURCE ELEMENT WHOSE NAME IS SPECIFIED IN THE 'ELEMENT3' FIELD IS GENERATED AND INSERTED IN THE PROGRAM-FILE SPECIFIED IN THE 'ELEMENT3' FIELD. IF THE 'ELEMENT3' FIELD IS VOID, THE NEW SOURCE ELEMENT WILL NOT BE KEPT.

IF THE 'U' OPTION IS USED IN THE CALL STATEMENT, THE CORRECTIONS FOLLOWING THE CALL STATEMENT ARE APPLIED AS A NEW CYCLE OF THE SOURCE INPUT ELEMENT SPECIFIED IN THE 'ELEMENT1' FIELD. IF THE CYCLE LIMIT HAS ALREADY BEEN REACHED FOR THIS ELEMENT, SIR WILL APPLY THE FIRST CYCLE OF CORRECTIONS TO THE SOURCE TO PRODUCE A NEW SOURCE

WHILE ADDING THE LATEST CORRECTION CYCLE.

SIR IMPLEMENTS THE CYCLE FEATURE BY USING A RECURSIVE TECHNIQUE FOR THE MULTI-LEVELS OF CORRECTIONS TO BE APPLIED TO THE SINGLE ELEMENT BEING UPDATED. THE CONTROL INPUT PROVIDES THE CURRENT LEVEL OF CORRECTIONS. THE OLDEST COMPLETE SOURCE AND THE CORRECTIONS OF THE CYCLES NOT YET IMBEDDED IN THE SOURCE ARE THE SOURCE INPUT AND ARE MAINTAINED IN THE PROGRAM LIBRARY AS ONE ELEMENT. THE CORRECTION LINES UP THROUGH THE CYCLE REQUESTED ARE APPLIED TO THE SOURCE INPUT AS IT IS PROVIDED TO THE PROCESSOR.

SIR FORMATS SYMBOLIC ELEMENTS IN STANDARD DATA FORMAT (SDF). EACH IMAGE HAS AN IMAGE CONTROL WORD PRECEDING IT. THE UPPER THIRD CONTAINS THE IMAGE LENGTH, THE MIDDLE THIRD CONTAINS THE CYCLE NUMBER OF THE CYCLE THAT DELETED THE IMAGE. IF THE IMAGE HAS NOT BEEN DELETED, THIS THIRD IS ZERO, AND THE LOWER THIRD CONTAINS THE ACTUAL CYCLE NUMBER OF THE IMAGE. IF AN IMAGE HAS NOT BEEN DELETED, IT BELONGS TO ITS CYCLE AND ANY CYCLE WITH A LATER CYCLE NUMBER. IF AN IMAGE IS MARKED DELETED, IT BELONGS TO ITS CYCLE AND ANY LATER CYCLES UP TO THE CYCLE THAT DELETED IT.

WHEN THE 'I' OPTION IS ON, SIR GENERATES A ONE WORD LABEL IMAGE (*SDFF*) AS THE FIRST IMAGE OF THE SYMBOLIC ELEMENT. THE IMAGE CONTROL WORD FOR A LABEL IMAGE HAS BIT 35 SET TO ONE. SIR PASSES ALL IMAGES TO THE USER, BUT ONLY COUNTS THOSE IMAGES WITH BIT 35 SET TO ZERO AS ITEMS WHEN APPLYING CORRECTIONS.

NOTE: THE ABOVE DISCUSSION MENTIONS THIRD'S OF A WORD AS DOES THE EXAMPLE BELOW. IN REALITY THE IMAGE LENGTH, DELETE CYCLE, AND CYCLE NUMBER FIELDS UTILIZE ONLY A SIXTH OF A WORD .

THE FOLLOWING EXAMPLE SHOWS HOW SIR FORMATS SYMBOLIC ELEMENTS.

```
THE FOLLOWING
ASSEMBLY:

@ASM,I  PF.E
START NOP
 ER EXIT$
 END START
@...

WOULD PRODUCE:

   S2  S4  S6
--------------
:5001   0   0:
--------------
: * S D F F *:
--------------
:  2    0   0:
--------------
: S T A R T  :
--------------
: N O P      :
--------------
:  2    0   0:
--------------
:   E R   E X:
--------------
: I T $      :
--------------
:  2    0   0:
--------------
:   E N D    :
--------------
:S T A R T   :
--------------
:7777        :
--------------
```

```
THE FOLLOWING
UPDATE:

@ASM,U  PF.E
-1
 NOP
@...

WOULD PRODUCE:

   S2  S4  S6
--------------
:5001   0   0:
--------------
: * S D F F *:
--------------
:  2    0   0:
--------------
: S T A R T  :
--------------
: N O P      :
--------------
:  1    0   1:
--------------
: N O P      :
--------------
:  2    0   0:
--------------
:   E R   E X:
--------------
: I T $      :
--------------
: 2    0    0:
--------------
:   E N D    :
--------------
:S T A R T   :
--------------
:7777        :
--------------
```

```
THE FOLLOWING
UPDATE:

@ASM,U  PF.E
-2,2
TAG NOP
@...

WOULD PRODUCE:

   S2  S4  S6
--------------
:5001   0   0:
--------------
: * S D F F *:
--------------
:  2    0   0:
--------------
: S T A R T  :
--------------
: N O P      :
--------------
:  1    2   1:
--------------
: N O P      :
--------------
:  2    0   2:
--------------
: T A G   N O:
--------------
: P          :
--------------
:  2    0   0:
--------------
:   E R   E X:
--------------
:I T $       :
--------------
:  2    0   0:
--------------
:   E N D    :
--------------
:S T A R T   :
--------------
:7777        :
--------------
```

IF THE 'W' OPTION IS USED IN THE CALL STATEMENT, SIR WILL ALSO OUTPUT THE CORRECTIONS TO THE PRINT OUTPUT FILE.

TO OPEN THE SOURCE INPUT AND OUTPUT FILES, THE PROCESSOR MUST MAKE THE FOLLOWING CALL:

```
LMJ   11,OPNSRC
      ERROR RETURN
      NORMAL RETURN
```

THEREAFTER, THE PROCESSOR WILL EXECUTE THE FOLLOWING CALL TO ACQUIRE A SOURCE IMAGE AT THE ADDRESS SPECIFIED IN A0.

```
H1 OF A0 -- LENGTH OF USER IMAGE AREA
H2 OF A0 -- LOCATION OF USER IMAGE AREA
LMJ     11,GETSRC
```

```
                  ERROR RETURN
                  END-OF-FILE RETURN
                  NORMAL RETURN
```

THE END-OF-FILE RETURN IS TAKEN WHEN AN SDF END-OF-FILE IMAGE IS ENCOUNTERED IN THE SOURCE INPUT ELEMENT OR WHEN READ$ GIVES AN END-OF-FILE RETURN WITH THE 'I' OPTION ON.

WHEN THE NORMAL RETURN IS TAKEN, A1 CONTAINS THE SDF IMAGE CONTROL WORD FOR THE IMAGE, AND H2 OF A2 CONTAINS THE 3 DIGIT FIELDATA CYCLE NUMBER FOR THE IMAGE. IF THE IMAGE CAME FROM THE RUN STREAM, S3 OF A2 CONTAINS A FIELDATA ASTERISK (050); IF NOT, IT CONTAINS A FIELDATA SPACE (05). S1 AND S2 OF A2 ALWAYS CONTAIN FIELDATA SPACES.

AFTER THE LAST IMAGE HAS BEEN PROCESSED, THE FOLLOWING CALL IS MADE TO CLOSE THE SOURCE INPUT AND OUTPUT FILES.

```
          LMJ     11,CLOSRC
                  ERROR RETURN
                  NORMAL RETURN
```

16.3. PREPROCESSOR ROUTINE (PREPRO)

A PREPROCESSOR ROUTINE MAY BE UTILIZED BY ANY WORKER PROGRAM TO ENABLE IT TO OBTAIN FILE NAMES, OPTIONS, FILE TYPES , AND THE NEXT WRITE LOCATIONS TO HOLD THE TEXT. THIS ROUTINE IS CALLED PREPRO, AND IT MUST BE ENTERED PRIOR TO OPENING SIR.

THE PURPOSE OF PREPRO IS TO DETERMINE IF THE FILE NAMES ON THE PROCESSOR CALL CARD ARE PROGRAM OR TAPE FILES; I.E., IF THE PROGRAM FILE DOES EXIST. IF IT DOES EXIST WHERE CAN THE PROCESSOR WRITE THE TEXT AND PREAMBLE. PREPRO PRODUCES A TABLE WHICH HAS THE FILE NAMES AND WRITE LOCATION FOR SIR AND THE PROCESSORS. THIS TABLE IS DOCUMENTED MORE FULLY IN THE TECHNICAL DOCUMENT. THE FIRST CELL OF THE TABLE MUST BE NAMED 'PARTBL' AND IS EXTERNALLY DEFINED.

TO CALL THE PREPROCESSOR:

```
       LMJ    X11,PREPRO
       - ERROR RETURN
       - NORMAL RETURN
```

IF THE ERROR RETURN IS TAKEN A MESSAGE HAS BEEN DELIVERED TO PRINT$ A0 HAS A CODE VALUE = 0 IF A HARDWARE ERROR CAUSED THE MALFUNCTION , NON ZERO VALUE IF THE ERROR IS A RESULT OF A CONTROL CARD, UNDEFINED PROGRAM, OR AN IMPROPERLY POSITIONED TAPE.

16.4. SOURCE OUTPUT ROUTINE (SOR)

CERTAIN PROCESSORS (E. G., LIFT) GENERATE SOURCE LANGUAGE AS THEIR OUTPUT RATHER THAN RELOCATABLE TEXT. FOR THESE PROCESSORS, THE 'ELEMENT2' FIELD OF THE PROCESSOR CALL CARD SPECIFIES THE OUTPUT SOURCE LANGUAGE WHICH MUST NOT BE CONFUSED WITH 'ELEMENT3', THE UPDATED SOURCE LANGUAGE. THE 'ELEMENT2' FIELD SHOULD CONFORM TO ALL NORMAL SPECIFICATIONS FOR A SYMBOLIC ELEMENT IN A PROGRAM FILE, WITH THE EXCEPTION THAT NO CYCLE NUMBER MAY BE GIVEN. THE CYCLE PRODUCED IS ALWAYS CYCLE 0.

THE SYMBOLIC TEXT PRODUCED BY SUCH A PROCESSOR MAY BE WRITTEN OUT USING SOR. SOR PRODUCES AN ELEMENT IN THE SYSTEM DATA FORMAT (SDF), WHICH IS FULLY DESCRIBED IN THE SECTION ON SIR (16.2).

SOR EXPECTS THAT PARTBL WILL HAVE BEEN INITIALIZED WITH THE INFORMATION FROM THE PROCESSOR CALL STATEMENT, GENERALLY BY A CALL TO PREPRO (SEE 16.3). MOREOVER, NO OTHER ROUTINE SHOULD HAVE THE SAME PROGRAM FILE OPENED FOR OUTPUT. IN PARTICULAR, THE FIRST PASS BY SIR SHOULD BE COMPLETED BEFORE CALLING SOR.

TO OPEN THE SOURCE OUTPUT FILE, THE PROCESSOR MUST MAKE THE FOLLOWING CALL:

```
LMJ     11,SSOR
        ERROR RETURN
        NORMAL RETURN
```

THEREAFTER, THE PROCESSOR WILL MAKE THE FOLLOWING CALL TO OUTPUT A SOURCE IMAGE AT THE ADDRESS SPECIFIED IN A0.

```
H1 OF A0 -- LENGTH OF USER IMAGE AREA (<64)
H2 OF A0 -- LOCATION OF USER IMAGE AREA
LMJ     11,SOR
        ERROR RETURN
        NORMAL RETURN
```

AFTER THE LAST IMAGE HAS BEEN OUTPUT, THE FOLLOWING CALL IS MADE TO CLOSE THE SOURCE OUTPUT FILE.

```
LMJ     11,ESOR
        ERROR RETURN
        NORMAL RETURN
```

IF THE 'ELEMENT2' FIELD HAS THE SAME FILE, ELEMENT, AND VERSION NAMES AS THAT FOR THE UPDATED SOURCE LANGUAGE ('ELEMENT1' WITH 'I' OR 'U' OPTION, ELSE 'ELEMENT3'), THE UPDATED SOURCE ELEMENT WILL BE DELETED FROM ITS PROGRAM FILE. IF THIS IS NOT TAKEN INTO CONSIDERATION, UNEXPECTED RESULTS MAY OCCUR.

FINALLY, IT IS IMPORTANT THAT THE DIFFERENCE BETWEEN THE OUTPUT FUNCTION OF SOR AND THE UPDATING FUNCTION OF SIR BE THOROUGHLY UNDERSTOOD. SIR APPLIES CORRECTIONS TO AN EXISTING ELEMENT TO UPDATE IT; SOR AIDS IN THE CREATION OF AN ENTIRELY NEW ELEMENT WHICH NEED BEAR NO RESEMBLANCE WHATEVER TO THE INPUT ELEMENT.

16.5. RELOCATABLE OUTPUT ROUTINE (ROR)

THE 'ELEMENT2' FIELD OF THE PROCESSOR CALL STATEMENT SPECIFIES THE RELOCATABLE ELEMENT NAME AND THE PROGRAM-FILE IN WHICH THE RELOCATABLE ELEMENT IS TO BE PLACED.
PRIOR TO OUTPUTTING ANY RELOCATABLE TEXT WORDS, THE PROCESSOR MUST INITIALIZE THE RELOCATABLE OUTPUT ROUTINE WITH THE CALL.

```
A0 --   K-BIT LIMIT
LMJ     11,SROR
        ERROR RETURN
        NORMAL RETURN
```

WHERE 'K-BIT LIMIT' IS THE NUMBER OF BITS REQUIRED TO CONTAIN THE LARGEST CONTROL COUNTER USED OR THE NUMBER OF UNDEFINED SYMBOLS FOR THE RELOCATION, WHICHEVER IS LARGER.
THEREAFTER, FOR EVERY WORD OF RELOCATABLE TEXT TO BE PLACED INTO THE RELOCATABLE ELEMENT, THE PROCESSOR WILL EXECUTE THE FOLLOWING CALL:

```
A0 --   LOCATION OF ITEM
LMJ     11,ROR
        ERROR RETURN
        NORMAL RETURN
```

WHERE 'LOCATION' IS THE ADDRESS OF AN ITEM DESCRIBING THE TEXT WORD TO BE INSERTED IN THE RELOCATABLE ELEMENT.
AFTER THE LAST TEXT WORD HAS BEEN PROCESSED THROUGH ROR, THE OPERATION MUST BE TERMINATED WITH THE CALL:

```
A0 -- TRANSFER ADDRESS,LOCATION COUNTER
LMJ    11,EROR
       ERROR RETURN
       NORMAL RETURN
```

THIS WILL OUTPUT THE LAST RELOCATABLE IMAGE, AND GENERATE A TRANSFER IMAGE IF A0 IS POSITIVE. THE UPPER HALF OF A0 CONTAINS THE TRANSFER ADDRESS, AND THE LOWER HALF SPECIFIES THE LOCATION COUNTER OF THE TRANSFER ADDRESS. IF A0 IS NEGATIVE WHEN EROR IS ENTERED NO TRANSFER IMAGE IS GENERATED.

THE PREAMBLE MUST BE CONSTRUCTED BY THE INDIVIDUAL PROCESSOR USING THE ROUTINES.

16.6. PREAMBLE OUTPUT ROUTINE (TBLWR)

WHEN A PROCESSOR HAS CLOSED ROR , IT THEN MUST ENSURE THAT THE PREAMBLE WHICH IT PRODUCED GETS WRITTEN BECAUSE THESE DATA ARE REQUIRED FOR COLLECTION , THE FORMAT OF THE PREAMBLE IS DEFINED IN THE INTERNAL SYSTEMS MANUAL (C.F. CHAPTER CONCERNING THE COLLECTOR).

TO WRITE OUT THE PREAMBLE, LOAD CR A0 WITH THE LOCATION OF THE TABLE, LOAD CR A1 WITH THE LENGTH OF THE TABLE AND:

```
LMJ       X11,TBLWR
- ERROR RETURN
- NORMAL RETURN
```

AN ERROR RETURN IS CAUSED BY A HARDWARE MALFUNCTION. THE TABLE HAS NOT BEEN PROPERLY WRITTEN.

THE NORMAL RETURN IMPLIES ALL IS SATISFACTORY AND THE PROGRAM FILE IS READY TO RECEIVE MORE DATA.

17. THE DIAGNOSTIC SYSTEM

17.1. GENERAL

THE 1108 EXECUTIVE SYSTEMS PROVIDES A COMPREHENSIVE DIAGNOSTIC SYSTEM TO AID IN THE EFFICIENT CHECKOUT OF USER PROGRAMS. DUMPS ARE PROVIDED IN TWO FORMS: SNAPSHOT (DYNAMIC) DUMPS, AND POSTMORTEM (STATIC) DUMPS . DYNAMIC DUMPS ARE ASSEMBLED WITH THE USERS SOURCE CODE (ASSEMBLER CODING ONLY) AND THE RESULTANT CALLS ON THE DUMPS ARE SATISFIED AT COLLECTION TIME . POST MORTEM DUMPS ARE CALLED VIA THE @PMD CONTROL CARD

17.2. SNAPSHOT DUMPS VIA A PROCESSOR

17.2.1. BASIC OPERATION

THE 1108 EXECUTIVE SYSTEM INCLUDES THE CAPABILITY OF USING SNAPSHOT DUMPS EMPLOYING INFORMATION GENERATED BY THE PROCESSORS AND THE COLLECTOR. DUMPED INFORMATION IS WRITTEN TO A DIAGNOSTIC FILE, AND READ BACK LATER, FOR EDITING AFTER THE PROGRAM UNDER TEST HAS TERMINATED. ASSEMBLY PROCEDURES ARE USED IN GENERATING CALLS TO LIBRARY SUBROUTINES WHICH ACTUALLY WRITE OUT THE DUMPS TO THE DIAGNOSTIC FILE. THESE SUBROUTINES WILL SAVE AND RESTORE ALL THE PROGRAM'S ENVIRONMENT. THE AMOUNT OF INFORMATION WRITTEN INTO THE DIAGNOSTIC FILE IS LIMITED BY A PARAMETER SET AT SYSTEM GENERATION TIME.

THIS SECTION SPECIFICALLY DISCUSSES SNAPSHOT DUMPS FROM THE STANDPOINT OF A USER OF THE 1108 ASSEMBLER. THERE IS NO INHERENT RESTRICTION ON THE EMPLOYMENT OF THE SNAPSHOT FACILITY WITH ANY OTHER PROCESSOR. ALL THAT IS NEEDED IS THAT THE PROPER INFORMATION BE WRITTEN TO THE DIAGNOSTIC FILE. LIBRARY ROUTINES ARE PROVIDED TO ASSIST IN THIS PROCESS. THE USE OF THE SNAPSHOT FACILITY BY A HIGH-LEVEL LANGUAGE PROCESSOR FALLS OUTSIDE THE SCOPE OF THIS DOCUMENT.

17.2.2. AVAILABLE SNAPSHOT DUMP TYPES

A GROUP OF 17 PROCEDURES IS AVAILABLE FOR THE PURPOSE OF TAKING SPECIALIZED SNAPSHOT DUMPS.

(1) CONDITIONAL (FOUR). XIF, XAND, X$OR, AND X$TALY ARE USED TO DETERMINE WHEN A GIVEN DUMP OR SERIES OF DUMPS SHOULD OCCUR.

(2) DUMP (SEVEN). X$MESG, X$CORE, X$DUMP, X$TAPE, X$DRUM, X$FILE, AND X$CREG ARE USED TO RECORD DATA IN THE DIAGNOSTIC FILE.

(3) SPECIFICATION (SIX). X$FRMT, X$BUF, X$MARK, X$BACK, X$ON, AND X$OFF ARE USED TO SPECIFY ARBITRARY PRINT LINE FORMATS, CORE SPACE FOR DRUM AND TAPE FILE DUMPS, DELETION OF RECORDED DUMPS NOT OF INTEREST, AND A CONTROL FOR NULLIFYING AND ACTIVATING DIAGNOSTIC PROCEDURES.

17.2.3. THE CONDITIONAL PROCEDURES

A STRING OF CONDITIONAL PROCEDURES MAY EITHER PRECEDE, OR BE INTERSPERSED AMONG, A SERIES OF DUMP PROCEDURES. THE PURPOSE OF CONDITIONAL PROCEDURES IS TO DETERMINE WHEN OR IF THE DUMP PROCEDURES WHICH IMMEDIATELY FOLLOW ARE EFFECTIVE. A STRING OF CONDITIONAL PROCEDURES MAY BE STARTED BY EITHER THE X$IF OR X$TALY STATEMENT . THE DIAGNOSTIC SYSTEM CONTAINS AN INTERNAL CONDITION SWITCH WHICH MAY BE ALTERED BY THE EXECUTION OF THESE CONDITIONAL PROCEDURES. DUMP PROCEDURE CALLS IMMEDIATELY FOLLOWING A CONDITIONAL PROCEDURE CALL WILL BE EXECUTED ONLY IF THE PRECEDING

CONDITIONAL PROCEDURE HAS LEFT THE CONDITION SWITCH SET ON .

17.2.3.1. THE X$IF PROCEDURE

THE X$IF PROCEDURE BY DEFINITION BEGINS A STRING OF DIAGNOSTIC CALLS. THE CONDITION SWITCH IS SET TO TRUE OR FALSE DEPENDING ON THE CURRENT VALUE OF THE FIRST LOGICAL EXPRESSION (OPERAND 'REL' OPERAND) IN THE X$IF PROCEDURE CALL. IT IS CALLED BY:

```
X$IF M1, X1, J1 'REL' M2, X2, J2
```

IN WHICH M IS A LOCATION IN CORE OR CONTROL MEMORY, X IS AN INDEX REGISTER DESIGNATION, AND J IS A 'J FIELD' SPECIFYING ANY DESIRED PARTIAL WORD. OMITTED FIELDS ARE TREATED AS ZERO. LITERALS MAY BE USED FOR M, IF DESIRED, AND INDIRECT ADDRESSING IS ALLOWED; HOWEVER, THERE IS NO PROVISION FOR INDEX INCREMENTATION.

THE 'REL' FIELD SPECIFIES THE RELATION BETWEEN THE TWO OPERANDS TO BE TESTED. THE ALLOWABLE CODES ARE:

CODE	MEANING
'GT'	GREATER THAN
'LT'	LESS THAN
'EQ'	EQUAL TO
'GE'	GREATER THAN OR EQUAL TO
'LE'	LESS THAN OR EQUAL TO
'NE'	NOT EQUAL

17.2.3.2. THE X$AND PROCEDURE

THE X$AND PROCEDURE HAS PRECISELY THE SAME FORMAT AND FUNCTION AS DOES THE X$IF PROCEDURE, EXCEPT THAT THE CONDITION SWITCH IS SET 'ON' IF, AND ONLY IF, IT IS ALREADY 'ON' AND IF THE CURRENT CONDITION IS 'TRUE'.

17.2.3.3. THE X$OR PROCEDURE

THE X$OR PROCEDURE HAS PRECISELY THE SAME FORMAT AS THE X$IF AND X$AND PROCEDURES, EXCEPT THAT THE CONDITION SWITCH IS SET 'ON' IF IT IS ALREADY 'ON' AND/OR IF THE CURRENT CONDITION IS 'TRUE'.

17.2.3.4. THE X$TALY PROCEDURE

THE X$TALY PROCEDURE SERVES TO SET THE CONDITION SWITCH 'OFF' IF CERTAIN CRITERIA ARE NOT SATISFIED. IT IS CALLED BY:

```
X$TALY      START, UNTIL, EVERY
```

EVERY TIME A X$TALY PROCEDURE CALL IS EXECUTED WHILE THE CONDITION SWITCH IS 'ON', THE FOLLOWING TESTS ARE MADE TO A FIELD 'Z' WITHIN THE GENERATED CODE: IF [START<Z<UNTIL] AND [Z-START/EVERY] PRODUCES A ZERO REMAINDER, THE CONDITION SWITCH IS LEFT 'ON' AND THE FIELD 'Z' IS INCREMENTED BY ONE. IF BOTH OF THE ABOVE CONDITIONS DO NOT HOLD, THE CONDITION SWITCH IS SET 'OFF.' DUMP PROCEDURES IMMEDIATELY FOLLOWING THE X$TALY PROCEDURE WILL NOT BE EXECUTED IF THE CONDITION SWITCH HAS BEEN SET 'OFF'.

17.2.3.5. EXAMPLE.

AN EXAMPLE OF THE USE OF CONDITIONAL PROCEDURES WILL INDICATE HOW THE CONDITION SWITCH IS SET. NOTE THAT IF DUMP PROCEDURES (DESCRIBED IN NEXT SECTION) ARE INTERSPERSED WITH CONDITIONALS, THEY WILL BE EFFECTIVE IF AND ONLY IF THE CONDITION SWITCH IS SET TO 'TRUE' AT THE TIME THEY ARE ENTERED AND WILL HAVE NO EFFECT ON THE SETTING OF THE CONDITION SWITCH.

ASSUME THAT A PROGRAM CONTAINS THE VARIABLES X, Y, AND Z, AND THE CONSTANTS A, B, AND C. ALSO ASSUME THAT THE FOLLOWING PROCEDURES ARE EXECUTED SEQUENTIALLY (WITH OR WITHOUT OTHER PROCEDURES OR INSTRUCTIONS INTERSPERSED) AND THAT THEY ARE PART OF A

LOOP WHICH WILL BE EXECUTED 4000 TIMES. WE COUNT THE NUMBER OF EXECUTIONS STARTING WITH 0 AND ENDING WITH 3999.

		EXECUTION LOOP COUNT:	0	1	99	100
PROCEDURE:	(1) X$IF	X 'EQ' A	FALSE	FALSE	FALSE	FALSE
	(2) X$OR	X 'LT' Z	TRUE	TRUE	TRUE	TRUE
	(3) X$AND	Y 'GT' B	FALSE	FALSE	FALSE	FALSE
	(4) X$OR	Y 'NE' Z	TRUE	TRUE	TRUE	TRUE
	(5) X$OR	A 'EQ' 90	TRUE	TRUE	TRUE	TRUE
	(6) X$OR	A 'EQ' 'A'	TRUE	TRUE	TRUE	TRUE
	(7) X$TALY	0,4000,100	TRUE	FALSE	FALSE	TRUE

IF X, Y, AND Z HAVE VALUES OF 78, 80, AND 88 RESPECTIVELY, AND A, B, AND C HAVE VALUES OF 'A' (FIELDATA CHARACTER), 180 AND OCTAL 040 RESPECTIVELY, THEN THE CONDITION SWITCH WILL HAVE THE ABOVE VALUES (TRUE OR FALSE) AFTER EXECUTION OF THE INDICATED PROCEDURE. NOTE: IF THE CONDITION SWITCH IS SET TO 'FALSE' WHEN THE X$TALY PROCEDURE IS ENTERED, THE COUNTER IS NOT INCREMENTED AND RETURN IS MADE TO USER.

17.2.4. THE DUMP PROCEDURES

THE DUMP PROCEDURES GENERATE CALLING SEQUENCES WHICH OUTPUT THE INFORMATION COMPRISING THE DESIRED DUMPS. THE ACTUAL OUTPUT WHILE THE OBJECT PROGRAM IS RUNNING IS TO A SPECIAL DIAGNOSTIC FILE. WHEN A PROGRAM TERMINATES, THIS DUMP INFORMATION IS READ BACK, EDITED, AND PRINTED. IF NO CONDITIONAL PROCEDURES ARE USED, THE DUMP PROCEDURES WILL ALWAYS PRODUCE OUTPUT AS THE DUMP PROCEDURES STANDING ALONE ALWAYS PRODUCE OUTPUT.

17.2.4.1. THE X$MESG PROCEDURE

THE X$MESG WILL PRODUCE A LINE OF UP TO 120 CHARACTERS ON THE OUTPUT LISTINGIF THE CONDITION SWITCH IS SET TO 'ON'. THE ALPHANUMERIC LINE IT PRODUCES IMMEDIATELY FOLLOWS THE PROCEDURE REFERENCE. IT IS CALLED BY:

```
X$MESG      LENGTH
'DIAGNOSTIC MESSAGE'
```

THE FIELD 'LENGTH' SPECIFIES THE NUMBER OF WORDS IN THE MESSAGE THAT FOLLOWS; THE FIELD 'DIAGNOSTIC MESSAGE' CONSISTS OF ANY ALPHANUMERIC STRING ENCLOSED IN QUOTATION MARKS, AND WILL BE PRINTED EXACTLY AS ASSEMBLED. THIS PROCEDURE PERMITS A PROGRAMMER TO IDENTY THE CONDITIONS WHICH CAUSED A GIVEN DUMP BY HIS OWN SUITABLE MESSAGE.

17.2.4.2. THE X$CORE PROCEDURE

THE X$CORE PROCEDURE PRODUCES A PRINTOUT OF CORE MEMORY, PROVIDED THAT THE CONDITION SWITCH IS SET TO 'ON'. IT IS CALLED BY:

```
X$CORE      START, LENGTH, 'FORMAT', INDEX REGISTER
```

WHICH RESULTS IN 'LENGTH' NUMBER OF CORE LOCATIONS, BEGINNING AT CORE-LOCATION 'START+INDEX' BEING EDITED AND PRINTED ACCORDING TO 'FORMAT'. THE FORMAT AND INDEX FIELDS ARE OPTIONAL. THE FORMAT FIELD, IF PRESENT, CONTAINS A SINGLE LETTER ENCLOSED IN QUOTATION MARKS, SUCH AS A,D,E,F,I,O, OR S FOR STANDARD FORMATS; OR MAY CONTAIN SOME OTHER LETTER FOR A USER DEFINED FORMAT. IF A FORMAT IS NOT SPECIFIED 'O' FORMAT IS USED. SEE SECTION 17.2.5.1., FORMAT SPECIFICATION PROCEDURE. THE INDEX FIELD, IF PRESENT GIVES THE NUMBER OF AN INDEX REGISTER TO BE USED TO MODIFY THE 'START' ADDRESS TO OBTAIN THE EFFECTIVE DUMP ADDRESS. THE LEGAL VALUES FOR THE INDEX FIELD

ARE 1 THRU 15 (X1 THRU X15). THE MAXIMUM ALLOWABLE 'LENGTH' VALUE IS 037777.

17.2.4.3. THE X$DUMP PROCEDURE

THE X$DUMP PROCEDURE PRINTS OUT THE PROGRAM'S ENVIRONMENT; THE X, A, AND/OR R REGISTERS, IF DESIRED; AND A PRINT-OUT OF CORE MEMORY, PROVIDED THAT THE CONDITION SWITCH IS SET TO 'ON'. IT IS CALLED BY:

```
    X$DUMP      START, LENGTH, 'FORMAT', 'REGISTERS', INDEX REG
```

WHICH RESULTS IN A PRINT-OUT OF THE CARRY AND OVERFLOW CONDITIONS FOLLOWED BY CONTENTS OF THE SELECTED REGISTERS PRINTED IN OCTAL. THE 'REGISTERS' ARE SPECIFIED BY THE LETTERS 'X', 'A' AND 'R' IN ANY COMBINATION OR PERMUTATION. THE REGISTER DUMP, IF REQUESTED, IS FOLLOWED BY A CORE DUMP OF 'LENGTH' NUMBER OF CORE LOCATIONS, BEGINNING AT CORE-LOCATION 'START + INDEX' EDITED AND PRINTED ACCORDING TO THE 'FORMAT'. THE 'INDEX' FIELD, IF PRESENT, GIVES THE NUMBER OF AN INDEX REGISTER TO USED TO MODIFY THE 'START' ADDRESS TO OBTAIN THE EFFECTIVE DUMP ADDRESS. LEGAL INDEX FIELD VALUES ARE 1 THRU 15.

ALL FIELDS ARE OPTIONAL; IF 'START' IS ABSENT A START ADDRESS OF ZERO IS USED, IF 'LENGTH' IS ABSENT NO CORE DUMP WILL BE EXECUTED, AND IF 'FORMAT' IS ABSENT A STANDARD OCTAL CORE DUMP WILL RESULT.

THE MAXIMUM VALUE OF 'LENGTH' IS 037777 .

17.2.4.4. THE X$CW PROCEDURE

THE X$CW PROCEDURE PRODUCES THE CALLING SEQUENCE FOR A 'CHANGED WORD' PRINTOUT OF THE SPECIFIED AREA OF CORE MEMORY , PROVIDING THAT THE CONDITION SWITCH IS 'ON' . THE CALLING SEQUENCE IS :

```
       X$CW      START,LENGTH,'FORMAT'
```

THIS CAUSES THE NUMBER OF WORDS AS SPECIFIED BY THE LENGTH PARAMETER , BEGINNING AT THE LOCATION SPECIFIED BY THE START PARAMETER TO BE SEARCHED FOR WORDS THAT HAVE CHANGED SINCE THE PREVIOUS CALL TO X$CW . THE CHANGED WORDS , IF ANY , ARE THEN EDITED AND PRINTED ACCORDING TO THE SPECIFIED FORMAT . ONLY STANDARD FORMATS MAY BE SPECIFIED (SECTION 17.2.5.1) , AND IT IS NOT POSSIBLE TO REDEFINE FOR CHANGED WORD DYNAMIC DUMPS ANY OF THE STANDARD FORMATS . IF A FORMAT IS SPECIFIED AS OTHER THAN A STANDARD ONE AN OCTAL DUMP RESULTS .

THE FIRST REQUEST FOR A CHANGED WORD DYNAMIC DUMP OF A SPECIFIC AREA WILL RESULT IN THE EDITING AND PRINTING OF ALL WORDS IN THAT AREA . ALTHOUGH THERE IS NO LIMIT TO THE NUMBER OF CALLS ON X$CW , THE NUMBER OF SEPARATE AREAS WHICH MAY BE DUMPED IS RESTRICTED TO FIVE . IN THE EVENT THE USER SPECIFIES MORE THAN FIVE AREAS THOSE CALLS AFTER THE 5TH WILL BE IGNORED . THE 'START ' ADDRESS DETERMINES THE UNIQUENESS OF ONE AREA FROM THE NEXT .

17.2.4.5. THE X$TAPE PROCEDURE

THE X$TAPE PROCEDURE PRODUCES A PRINT-OUT OF THAT BLOCK OF MAGNETIC TAPE JUST PRIOR TO THE CURRENT POSITION OF THE SPECIFIED FILE-NAME, PROVIDED THAT THE CONDITION SWITCH IS SET TO 'ON'. THE UNISERVO DUMPED IS ALWAYS REPOSITIONED AFTER THE DUMP IS MADE; IF THE UNISERVO IS POSITIONED AT THE BEGINNING OF A TAPE, NO DUMP WILL

OCCUR. IT IS CALLED BY:

```
    X$TAPE     ADDRESS OF EXEC I/O PACKET,'FORMAT'
                         OR
    X$TAPE     ADDRESS OF FILE CONTROL TABLE,'FORMAT'
```

THE BLOCK OF DATA JUST PRIOR TO THE CURRENT POSITION OF THE TAPE IS EDITED AND PRINTED ACCORDING TO 'FORMAT'.

THE X$TAPE PROCEDURE REQUIRES THAT A MEMORY AREA OF SUFFICIENT SIZE BE AVAILABLE INTO WHICH THE BLOCK OF TAPE MAY BE READ. THIS BUFFER IS PROVIDED BY THE X$BUFR PROCEDURE DESCRIBED IN SECTION 17.2.5.2. IF THE TAPE BLOCK IS LONGER THAN THE BUFFER PROVIDED, ONLY THAT INFORMATION WHICH CAN BE HELD IN THE BUFFER WILL BE PRINTED. THERE WILL NOT BE ANY DIAGNOSTIC MESSAGE FOR TRUNCATED BLOCKS.

17.2.4.6. THE X$DRUM PROCEDURE

THE X$DRUM PROCEDURE PROVIDES A PRINT-OUT OF A PORTION OF MASS STORAGE MEMORY, PROVIDED THAT THE CONDITION SWITCH IS SET TO 'ON'. IT IS CALLED BY:

```
        X$DRUM     ADDR EXEC I/O PACKET,LOCN,LENGTH,'FORMAT'
```

WHICH RESULTS IN THE 'LENGTH' - SPECIFIED NUMBER OF WORDS FROM THE 'INTERNAL-FILE-NAME' IN EXEC I/O PACKET, BEGINNING WITH RELATIVE DRUM ADDRESS FOUND IN THE CORE LOCATION 'LOCN' BEING EDITED AND PRINTED ACCORDING TO 'FORMAT'. THE X$DRUM PROCEDURE REQUIRES A CORE BUFFER INTO WHICH INFORMATION FROM DRUM MAY BE READ. WHILE A BLOCK OF DRUM STORAGE LARGER THAN THE SIZE OF THE BUFFER MAY BE DUMPED, GREATER EFFICIENCY WILL RESULT FROM PROVIDING A BUFFER SUFFICIENTLY LARGE TO HOLD ALL OF THE DRUM INFORMATION REQUESTED AT ONE TIME.

IF NO BUFFER AREA AT ALL IS PROVIDED FOR THE X$TAPE AND X$DRUM PROCEDURES, THESE DUMPS WILL NOT BE CARRIED OUT; HOWEVER, THE SAME BUFFER AREA MAY BE USED FOR BOTH X$TAPE AND X$DRUM PROCEDURE CALLS.

WHEN INTERSPERSING CONDITIONAL PROCEDURES WITH DUMP PROCEDURES, THE SETTING OF THE CONDITION SWITCH REFLECTS ONLY THOSE CONDITIONALS WHICH HAVE BEEN PROCESSED AT THE TIME THE DUMP IS CALLED.

17.2.4.7. THE X$FILE PROCEDURE

THE DIAGNOSTIC SYSTEM CAN INTERFACE WITH THE ITEM BUFFERING PACKAGE, IN ORDER TO PROVIDE A SNAPSHOT WHENEVER AN ITEM IS READ FROM, OR WRITTEN TO, A PARTICULAR FILE. THIS FACILITY IS NOT AVAILABLE WHEN USING INDEFINITELY LONG-SPANNED ITEMS. THE PROCEDURE CALL:

```
        X$FILE       FCB, 'ON', 'FORMAT'
```

WILL CAUSE SUBSEQUENT ITEMS READ FROM, OR WRITTEN TO, THE FILE WHOSE FILE CONTROL BLOCK IS AT 'FCB' TO BE WRITTEN INTO THE DIAGNOSTIC FILE AS A SNAPSHOT DUMP. THE PROCEDURE CALL:

```
        X$FILE       FCB, 'OFF'
```

WILL TERMINATE SUCH SNAPSHOTS. FCB IS THE ADDRESS OF THE FILE CONTROL TABLE.

17.2.4.7 THE X$CREG PROCEDURE

THE X$CREG PROCEDURE CALLS FOR A DUMP OF USERS CONTROL REGISTERS (PROVIDED THAT THE CONDITION SWITCH IS SET TO 'ON'). IT IS CALLED BY:

```
        X$CREG     START,LENGTH,'FORMAT'
```

THE FIELD 'START' CONTAINS THE ADDRESS OF THE FIRST CONTROL REGISTER TO BE DUMPED. 'LENGTH' INDICATES THE NUMBER OF REGISTERS TO BE DUMPED. THE 'FORMAT' FIELD INDICATES HOW THE DUMPED INFORMATION IS TO BE EDITED.

17.2.4.8. EXAMPLES.

THE VALUES FOR X, Y, Z, A, B, AND C ARE THE SAME AS DEFINED IN THE EXAMPLE IN SECTION 17.2.3.5., CONDITIONAL PROCEDURES. ALSO ASSUME THAT THE FOLLOWING PROCEDURES ARE EXECUTED SEQUENTIALLY AND THAT THESE ARE THE FIRST GROUP ENCOUNTERED.

```
                                                         CONDITION   RESULT
           $(1)                                           SWITCH      DUMP DUMP
  PROCEDURE: (1)  X$MESG 7                                  TRUE      (1) YES
                  'BEGIN TEST OF DIAGNOSTICS'                -
                  X$IF  X 'EQ' A                          FALSE
             (2)  X$MESG 4                                FALSE      (2) NO
                  'TEST DATA GROUP A'                        -
                  X$IF  X 'EQ' A                          FALSE
                  X$OR  X 'LT' Z                          TRUE
                   X$BUFR  DUMPB,50  . CALL BUFFER SETUP
             (3)  X$CORE  TABLEX,100,'O'                  TRUE       (3) YES
             (4)  X$DUMP  TABLEY,200,'I','XA'             TRUE       (4) YES
             (5)  X$TAPE  FILEA,'O'                       TRUE       (5) YES
                  X$IF  Y 'GT' B                          FALSE
             (6)  X$TAPE  FILEB,'O'                       FALSE      (6) NO
                  X$OR  Y 'NE' Z                          TRUE
             (7)  X$TAPE  FILEC,'O'                       TRUE       (7) YES
             (8)  X$DRUM  FILED,DRDUMP,100,'A'            TRUE       (8) YES
             (9)  X$FILE  BETA,'ON','O'                   TRUE       (9) YES
            (10)  X$FILE  BETA,'OFF'                      TRUE      (10) NO
            (11)  X$CREG  1,12,'O'                        TRUE      (11) YES
                 $(2)  .
                 DRDUMP    +  0   .  VALUE SET DYNAMICALLY BY USER
                 ALPHA     RES  200  .
                 BETA     (FILE CONTROL BLOCK - READ MODE)
                 TABLEY    RES  200
                 TABLEX    RES  100
                   DUMPB    RES   50  . BUFFER FOR DUMPING FROM
                                      . TAPE AND DRUM
                 FILEA (EXEC I/O PACKET)
                 FILEB (EXEC I/O PACKET)
                 FILEC (EXEC I/O PACKET)

                 FILED (EXEC I/O PACKET)
```

RESULT: (1) THE MESSAGE 'BEGIN TEST FOR DIAGNOSTICS' WILL BE RECORDED IN DIAGNOSTIC FILE AS DUMP PROCEDURES STANDING ALONE ALWAYS PRODUCE OUTPUT.

(2) NO DUMP RECORDED CONDITION SWITCH FALSE.

(3) STARTING WITH THE LOCATION TABLEX, 100 CELLS OF MEMORY WILL BE DUMPED IN THE DIAGNOSTIC FILE. IF PRINTED, THE FORMAT WILL BE OCTAL AS 'O' IS A STANDARD SYSTEM DEFINED FORMAT. NOTE: IT IS POSSIBLE TO CHANGE STANDARD FORMATS IF SO DESIRED USING X$FRMT PROCEDURE PRIOR TO DUMP CALL.

(4) THE ENVIRONMENT DATA, CONTROL REGISTERS X AND A, AND CORE MEMORY STARTING WITH TABLEY THRU TABLEY+199 WILL BE RECORDED IN THE DIAGNOSTIC FILE. IF PRINTED, THE ENVIRONMENT DATA IS PRINTED AS TO STATUS, CONTROL REGISTERS ARE PRINTED ALWAYS IN OCTAL FORMAT, AND THE 200 CELLS OF CORE WILL BE IN DECIMAL FORMAT AS 'I' FORMAT SPECIFIED.

(5) THE BLOCK OF DATA JUST PRIOR TO PRESENT POSITION IS RECORDED IN DIAGNOSTIC FILE. IF EDITED AND PRINTED IT WILL BE IN OCTAL FORMAT AS 'O' FORMAT SPECIFIED

(6) NO DUMP RECORDED CONDITION SWITCH FALSE.

(7) THE TAPE WHOSE INTERNAL-FILE-NAME IS FILEC WILL BE MOVED BACKWARDS ONE BLOCK AND THEN READ FORWORD ONE BLOCK. THE BLOCK READ WILL GO INTO THE CORE LOCATION 'ALPHA', DEFINED IN THE X$BUFR PROCEDURE,AND WILL BE RECORDED IN THE DIAGNOSTIC FILE TO BE PRINTED IN OCTAL FORMAT WHEN EDITED.

(8) BEGINNING AT RELATIVE WORD ADDRESS 500 OF MASS STORAGE FILE 'FILED', 100 WORDS OF DATA ARE RECORDED IN THE DIAGNOSTIC FILE TO BE EDITED IN ALPHANUMERIC FORMAT. IT IS ASSUMED THAT THE CURRENT CONTENT OF THE LOCATION 'DRDUMP' HAS A VALUE OF 500. (THE COMMON BUFFER AREA 'ALPHA' IS USED TO READ DATA IN FROM THE DRUM FILE.

(9) THE FILE WHOSE FILE CONTROL BLOCK IS AT 'BETA' IS CONDITIONED TO RECORD IN THE DIAGNOSTIC FILE ALL SUBSEQUENT ACTIVITY AT THE ITEM LEVEL. THAT IS EVERY TIME A REQUEST IS MADE TO READ AN ITEM, THE ITEM THAT THE ITEM HANDLER POINTS TO IS RECORDED IN THE DIAGNOSTIC FILE.

(10) THE FILE CONTROL BLOCK 'BETA' IS CONDITIONED TO NOT RECORD ANY SUBSEQUENT ACTIVITY AT THE ITEM LEVEL.

(11) CONTROL REGISTERS X1 THRU X12 ARE RECORDED IN THE DIAGNOSTIC FILE AND WILL BE EDITED IN OCTAL FORMAT FOR PRINTING.

17.2.5. THE SPECIFICATION PROCEDURES

THE SPECIFICATION PROCEDURES, X$BUFR AND X$FRMT, PROVIDE A BUFFER AREA AND FORMAT SPECIFICATIONS, RESPECTIVELY.

17.2.5.1. THE X$FRMT PROCEDURE

THE PROCEDURE CONSISTS OF A STRING OF CHARACTERS SPECIFYING AN EDITING FORMAT EXACTLY LIKE THE FORTRAN FORMAT STATEMENT. IT IS CALLED BY:

```
    X$FRMT N,'M'
    '(A1A2A3...AN)'
```

WHERE N IS THE NUMBER OF WORDS COMPRISING THE FORMAT STRING AND 'M' IS THE ONE LETTER NAME OF THE FORMAT TO BE SUPPLIED. WHEN REQUESTING A DUMP, THE NAME OF THE FORMAT SPECIFICATION IS USED IN THE PROCEDURE REFERENCE. THE STRING OF ALPHANUMERIC CHARACTERS A1...AN REPRESENTS AN ENCODING OF THE FORMAT TO BE APPLIED TO THE INFORMATION PRINTED. THERE ARE NO INTERVENING SPACES WITHIN THE FORMAT STRING. THE FIRST NON-BLANK CHARACTER OF THE STRING MUST BE A LEFT PARENTHESIS: '('; AND THE LAST NON-BLANK CHARACTER MUST BE A RIGHT PARENTHESIS; ')'.

THERE ARE AVAILABLE A NUMBER OF STANDARD FORMATS WHICH ARE KNOWN TO THE SYSTEM WITHOUT EXPLICIT DEFINITION BY THE USER. THESE FORMATS ARE SUFFICIENTLY GENERALIZED IN NATURE TO BE CAPABLE OF HANDLING THE MAJORITY OF SITUATIONS WHICH WILL ARISE. ONLY IN PARTICULARLY INVOLVED SITUATIONS WILL IT BE NECESSARY TO DEFINE SPECIAL FORMATS.

THE STANDARD FORMATS ARE:

'F'	(8 F 14.8)	FIXED DECIMAL
'E'	(8 E 14.8)	FLOATING DECIMAL
'I'	(8 I 14)	INTEGER
'A'	(16 A 6)	ALPHANUMERIC
'O'	(8 O 14)	OCTAL
'S'	(4 S 30)	INSTRUCTION
'D'	(4 D 26.18)	DOUBLE PRECISION FLOATING POINT

THE USER CAN REDEFINE ANY FORMATS (INCLUDING THE STANDARD FORMATS). THE DYNAMICALLY MOST-RECENT DEFINITION WILL HOLD. MULTIPLE-LINE FORMATS ARE ALLOWABLE.

17.2.5.2. THE X$BUFR PROCEDURE

THE X$BUFR PROCEDURE IS USED TO DEFINE AN AREA OF CORE INTO WHICH INFORMATION FROM TAPE AND DRUM IS READ. IT IS CALLED BY:

```
    X$BUFR      START, LENGTH
```

WHERE 'START' IS THE STARTING CORE ADDRESS OF THE BUFFER, AND 'LENGTH' IS THE LENGTH OF THE BUFFER.

17.2.5.3. THE X$MARK AND X$BACK PROCEDURES

TWO PROCEDURES ARE AVAILABLE FOR SAVING DUMPS UP TO A CERTAIN POINT IN EXECUTION AND THEN DELETING THEM AT THE USER'S DISCRETION. THE X$MARK PROCEDURE DENOTES THE POINT IN PROCESSING WHOSE ATTAINMENT PERMITS THE USER TO IGNORE SUBSEQUENT DUMPS. THIS POINT IS A CERTAIN SECTION OF HIS PROGRAM. THE X$BACK PROCEDURE WILL DELETE ALL DUMPS PREVIOUS TO THE X$BACK REFERENCE AND SUBSEQUENT TO THE LAST X$MARK FOR WHICH AN X$BACK HAS NOT YET OCCURRED.

THUS, THE X$MARK AND X$BACK PROCEDURES BEHAVE MUCH AS LEFT AND RIGHT PARENTHESES SURROUNDING PORTIONS OF A PROGRAM WHICH ARE TO BE DUMPED ONLY IF TERMINATION OCCURS BETWEEN THEM.

X$MARK AND X$BACK PAIRS MAY BE NESTED TO A DEPTH OF FIVE. THE TOTAL NUMBER OF OCCURRENCES OF X$MARK AND X$BACK IS UNRESTRICTED. THE PROCEDURE CALLS ARE:

```
X$MARK

X$BACK
```

17.2.5.4. THE X$ON AND X$OFF PROCEDURES

TWO PROCEDURES, X$ON AND X$OFF, ARE AVAILABLE FOR OVERALL CONTROL OF CALLS ON DEBUGGING PROCEDURES. WHENEVER A X$OFF IS IN CONTROL, ALL REFERENCES TO DEBUGGING PROCEDURES (EXCEPT X$ON) ARE NULLIFIED, REGARDLESS OF THE SETTING OF THE CONDITION SWITCH, NO OUTPUT WILL BE MADE TO THE DIAGNOSTIC FILE. AFTER A X$ON IS ENCOUNTERED, THE DEBUGGING PROCEDURES AGAIN BECOME ACTIVE. THE PROCEDURE CALLS ARE:

```
X$ON

X$OFF
```

THE CONDITION SWITCH IS INTERNAL TO THE DUMP ROUTINES .

17.2.5.5. EXAMPLES.

```
PROCEDURE:    $(2)   .
              ALPHA    RES  200     .

              $(1)   .
          (1) X$FRMT  1,'O'
              '(6014)'
          (2) X$BUFR  ALPHA,200
          (3) X$MARK
              X$CORE    ALPHA,200,'A'
          (4) X$BACK
          (5) X$OFF
              X$CORE    ALPHA,200,'E'
          (6) X$ON
              X$CORE    ALPHA,200,'I'
```

RESULTS: (1) THE STANDARD OCTAL FORMAT 'O' HAS BEEN REDEFINED TO PRINT 6 OCTAL FIELDS PER LINE INSTEAD OF 8. THE APPROPIATE DATA IS WRITTEN ON THE DIAGNOSTIC FILE SO THE ABOVE ACTION IS EFFECTIVE WHEN THE DIAGNOSTIC EDITOR PROCESSES THE RECORDED DYNAMIC DATA.

(2) PROVIDES THE DIAGNOSTIC SYSTEM WITH A 200 WORD BUFFER FOR TAPE AND DRUM FILE DUMPS. FOR DRUM FILES IT IS RECOMMENDED THAT THE BUFFER BE SOME MULTIPLE OF 28,THE LENGTH OF A FASTRAND SECTOR.

(3) SAVES THE CURRENT LOCATION WHERE THE NEXT WRITE IS TO BE MADE IN THE DIAGNOSTIC FILE.

(4) RESETS THE CURRENT LOCATION POINTER TO THE VALUE SAVED BY THE MOST RECENT X$MARK REFERENCE. RESULT IS THAT ALL INTERVENING DUMP INFORMATION IS ERASED. E.G. THE DATA RECORDED BY X$CORE ALPHA,200,'A' HAS BEEN DELETED.

(5) DISABLES ALL DIAGNOSTIC SYSTEM PROCEDURES EXCEPT X$ON SO THE X$CORE ALPHA,200,'E' CALL IS IGNORED.

(6) ENABLES DIAGNOSTIC PROCEDURES, X$CORE ALPHA,200,'I' CALL IS IN EFFECT AND REQUESTED DUMP RECORDED.

17.2.6. ASYNCRONOUS ACTIVITY DIAGNOSTIC CALLS

EACH DIAGNOSTIC ROUTINE THAT IS PART OF THE USER'S PROGRAM IS PROCESSED SERIALLY. CARE SHOULD BE OBSERVED IN USING THE X$ON AND SUBPROGRAM ACTIVITY SHOULD SET THE SWITCH OFF, IT WILL BE OFF FOR ALL SUBPROGRAM ACTIVITIES.

A COMMON EXIT IN THE DYNAMIC DIAGNOSTIC LIBRARY ROUTINE WILL LOOK AHEAD TO SEE IF THE NEXT INSTRUCTION IS A CALL TO THE DIAGOSTIC SYSTEM. IF TRUE, A LOCK-OUT CONDITION WILL REMAIN IN EFFECT UNTIL THE SERIES OF CALLS ON THE SYSTEM IS BROKEN. THIS ASSURES THE USER THAT A SERIES OF CALLS WILL NOT BE INTERRUPTED INADVERTENTLY BY ONE OF THE OTHER SUBPROGRAM ACTIVITIES.

17.3. THE PMD STATEMENT

17.3.1. GENERAL

A POST-MORTEM DUMP EXECUTIVE CONTROL STATEMENT MAY BE USED TO DUMP CORE MEMORY FOLLOWING THE EXECUTION OF A TASK. DUMPS MAY BE MADE OF OVERLAY SEGMENTS, ELEMENTS, OR SPECIFIED PARTS OF ELEMENTS, AS LONG AS THEY WERE CURRENTLY IN CORE AT THE TIME THE ROUTINE TERMINATED. SEVERAL OPTIONS ARE AVAILABLE FOR OUTPUT FORMATTING AND FOR SELECTING THE CORE AREAS TO BE DUMPED.

THE GENERAL FORM OF THE CONTROL STATEMENT IS:

```
@PMD,OPTIONS     SPECIFICATIONS
```

IF NO INFORMATION WAS SAVED BY THE SYSTEM WHEN THE PREVIOUS EXECUTION TERMINATED, NO DUMPS ARE POSSIBLE. THIS CONDITION MAY BE CAUSED BY A 'Z' OPTION GIVEN TO THE COLLECTOR WHEN THE PROGRAM WAS CONSTRUCTED, BY A MISPLACED PMD CARD, OR BY CERTAIN RARE ERROR CONDITIONS. IN THE EVENT THAT NO DUMP IS AVAILABLE, A MESSAGE IS PRODUCED.

THE @PMD STATEMENTS MUST FOLLOW THE @XQT STATEMENT OF THE PROGRAM THAT HAS TERMINATED IN ORDER TO BE HONORED. ONLY PURE DATA AND THE CONDITIONAL STATEMENTS; @SETC, @JUMP, AND @TEST INTERVENE. THE FOLLOWING IS AN EXAMPLE:

```
 NO.       STATEMENT
----   ------------------
  1    @XQT   PROGX
  2     DATA
  :       :
  :       :
 10     DATA
 11    @TEST    TE/6/S3
 12    @JUMP    3
```

```
13    @SETC   6/S4
14    @PMD    ELEMENT-1,ELEMENT-2
15    @XQT    PROGY
 :      :
 :      :
```

IF PROGX TERMINATES BEFORE PROCESSING ALL OF THE DATA STATEMENTS THAT FOLLOW THE @XQT AND S3 OF THE CONDITION WORD HAS A VALUE OF 6, S4 OF THE CONDITION WORD WILL BE SET TO 6 AND STATEMENT NUMBERS 14,15 WILL BE HONORED FOR PROCESSING. WHEN STATEMENT 16 IS ENCOUNTERED, THE RUN WILL BE TERMINATED IF IT IS NOT DEMAND.

17.3.2. OPTIONS

OPTIONS ARE SELECTED THROUGH USE OF OPTION LETTERS PUNCHED INTO THE @PMD CARD. THE OPTIONS FALL INTO THE FOLLOWING CLASSES: (1) GENERAL, (2) SPECIAL, (3) OPTIONS WITH SPECIALS, AND (4) BLANK

17.3.2.1. GENERAL OPTIONS

THE GENERAL OPTIONS MAY BE USED WITH ANY OTHERS IN A @PMD STATEMENT. THEY ARE:

(1) 'E' OPTION: IF THE LETTER E IS PLACED IN THE OPTIONS FIELD, THE @PMD STATEMENT WILL BE PROCESSED ONLY WHEN THE PREVIOUS ROUTINE TERMINATED IN ERROR.

(2) 'C' OPTION: THE 'C' OPTION WILL CAUSE A DUMP OF THE WORDS THAT WERE CHANGED DURING THE EXECUTION OF THE ALLOCATED PROGRAM FOR THE AREA OF CORE PRESCRIBED BY THE SPECIFICATIONS PORTION OF THE PMD STATEMENT.

(3) 'B' OPTION: AFTER PROCESSING THE REST OF THE PMD STATEMENT, THIS OPTION WILL CAUSE AN OCTAL DUMP OF ALL OF BLANK COMMON STORAGE AREA. IF USED WITH THE 'C' OPTION, THE 'C' OPTION WILL BE IGNORED FOR BLANK COMMON STORAGE.

(4) 'P' OPTION: THE LETTER 'P' USED IN CONJUNCTION WITH ANY OF THE OTHER OPTIONS KNOWN TO PMD WILL CAUSE AN OCTAL DUMP OF THE PCT BLOCK(S) USED BY THE RUN TO BE PRINTED PRECEEDING THE DUMP OF THE PROGRAM . THE BLOCKS ARE DUMPED IN OCTAL FORMAT . THE SEGMENT LOAD TABLES (IF ANY EXIST) ARE ALSO DUMPED IN OCTAL FORMAT IF THE 'P' OPTION IS SPECIFIED .

17.3.2.2. SPECIAL OPTIONS

ONLY ONE SPECIAL OPTION SHOULD BE USED ON A SINGLE PMD STATEMENT. IF MORE THAN THAN ONE SPECIAL OPTION IS USED, THE SPECIAL 'A' OPTION 'A' IS ASSUMED. ALL SPECIAL OPTIONS REQUIRE THE SPECIFICATIONS FIELD DESCRIBED BELOW. IF NO SPECIAL OPTION IS SUPPLIED, THE 'BLANK' OPTION RULES WILL BE APPLIED. THE SPECIFICATIONS FIELD FOR THE SPECIAL OPTIONS TAKES THE FORM OF A LIST OF ELEMENT OR SEGMENT NAMES:

NAME 1, NAME 2, NAME 3, (ETC.)

EACH NAMED ENTRY WILL BE DUMPED IN OCTAL FORMAT AND IN ORDER OF ALLOCATION. IF THE SPECIFICATION FIELD IS BLANK, ALL ELEMENTS IN MEMORY AT TERMINATION OF THE PREVIOUS ROUTINE WILL BE DUMPED. THESE SPECIAL OPTIONS ARE:

(1) 'A' OPTION: AN 'A' OPTION WILL PRODUCE A DUMP OF ALL MEMORY SPECIFIED IN EACH ELEMENT OR SEGMENT NAMED IN THE SPECIFICATION LIST.

(2) 'D' OPTION: A 'D' OPTION WILL PRODUCE A DUMP OF THE D-BANK PORTION OF EACH ELEMENT OR SEGMENT NAMED IN THE

SPECIFICATION LIST.

(3) 'I' OPTION: AN 'I' OPTION WILL PRODUCE A DUMP OF ALL I-BANK PORTIONS OF EACH ELEMENT OR SEGMENT NAMED IN THE SPECIFICATION LIST.

17.3.2.3. OPTIONS USED WITH SPECIAL OPTIONS

(4) 'X' OPTION: WHEN USED IN CONJUNCTION WITH THE 'A', 'I', OR 'D' OPTIONS, THE 'X' OPTION HAS AN EXCEPT EFFECT. ALL ACTIVE ELEMENTS WILL BE DUMPED EXCEPT THOSE NAMED IN THE SPECIFICATION LIST, AND THOSE BELONGING TO THE SEGMENTS NAMED IN THE SPECIFICATION LIST.

(5) 'L' OPTION: WHEN USED WITH THE 'A', 'I', OR 'D' OPTIONS, THE L OPTION, PRESENT, CAUSES A DUMP TO BE TAKEN OF ANY ACTIVE ELEMENTS FROM THE SYSTEM LIBRARY. THE 'L' OPTION WHEN USED ALONE WILL CAUSE THE ACTIVE LIBRARY ELEMENTS TO BE DUMPED.

17.3.2.4. THE 'BLANK' OPTION

IF NO SPECIAL OPTIONS ARE NAMED ON THE PMD CARD, THE SPECIFICATION FIELD MUST FOLLOW THE FORM:

```
NAME, START, LENGTH, FORMAT
```

THIS OPTION ALLOWS THE USER TO DUMP INFORMATION UNDER A SPECIFIC FORMAT WITHOUT OUTPUTTING EXCESSIVE AMOUNTS OF UNNECESSARY MATERIAL.

THE 'NAME' FIELD IS THAT OF AN ELEMENT AND MUST BE PRESENT.
THE 'START' FIELD MUST BE OF THE FORM:

```
N/M .
```

WHERE 'M' REPRESENTS THE LOCATION COUNTER OF THE ELEMENT TO BE DUMPED, AND 'N' REPRESENTS AN ADDRESS, RELATIVE TO THE BEGINNING OF 'M', AT WHICH DUMPING SHOULD BEGIN. IF 'M' OR 'N' IS OMITTED, A ZERO IS ASSUMED TO BE ITS RESPECTIVE VALUE.

THE 'LENGTH' FIELD MUST BE THE NUMBER OF WORDS TO BE DUMPED. IF OMITTED, THE LENGTH WILL BE ASSUMED TO BE ALL OF LOCATION COUNTER 'M' OF THE SPECIFIED ELEMENT.

STANDARD 'D','S' AND USER DEFINED FORMATS ARE NOT APPLICABLE FOR CHANGED WORD DUMPS. FOR ALL OTHER CASES THE USER MAY SUPPLY HIS OWN FORTRAN TYPE FORMATS (ENCLOSED IN PARENTHESES) OR USE THE SYSTEM DEFINED FORMATS PREVIOUSLY MENTIONED. THE

17.3.3 EXAMPLES

```
@PMD
```

RESULTS IN AN OCTAL DUMP OF ALL ACTIVE (ALLOCATED IN CORE) SEGMENTS OF A USERS PROGRAM . NO BLANK COMMON WILL BE DUMPED .

```
@PMD,EAXL ELEMENT-NAME-1, ELEMENT-NAME-2
```

RESULTS IN AN OCTAL DUMP OF ALL ACTIVE ELEMENTS EXCEPT ELEMENTNAME-1, ELEMENT-NAME-2, AND SYSTEM LIBRARY ELEMENTS ON AN ERROR TERMINATION.

```
@PMD,BDI  SEGMENT-NAME
```

RESULTS IN AN OCTAL DUMP OF SEGMENT-NAME (IF ACTIVE) AND BLANK COMMON AREA OF CORE STORAGE.

```
        @PMD,EBCD ELEMENT-NAME
```

RESULTS IN AN OCTAL DUMP OF CHANGED WORDS IN DBANK OF ELEMENTNAME (IF ACTIVE) AND BLANK COMMON.

```
        @PMD      ALPHA,100/3,50,A
```

RESULTS IN A 50 WORD ALPHANUMERIC FORMAT DUMP OF ELEMENT ALPHA (IF ACTIVE) UNDER CONTROL OF LOCATION COUNTER 3 BEGINNING WITH RELATIVE ADDRESS 100 OF LOCATION COUNTER 3.

```
        @PMD,B      . DUMP ALL OF PROGRAM INCLUDING BLANK COMMON.
```

17.4. PMD AND DYNAMIC DUMP CORRELATION

BECAUSE PMD AND THE DYNAMIC DUMPS WRITE INFORMATION INTO THE SAME FILE (WHICH IS KNOWN TO THE SYSTEM AND DYNAMICALLY ASSIGNED)) IT IS NECESSARY THEY BE AWARE OF EACH OTHERS EXISTENCE . ABSENCE OF A PARTICULAR FORMAT IMPLIES AN OCTAL DUMP . DYNAMIC DUMPS WILL PERFORM SEVERAL CHECKS TO ENSURE THEY NOT BEEN DESTROYED AND DO NOT PRODUCE FALSE DATA .

IN ORDER TO ACCOMPLISH THIS IT MAY REQUIRE A CERTAIN AMOUNT OF COOPERATION FROM THE USER . IT IS SUGGESTED THAT THE USER SET UP AN ERROR INTERUPT ANALYSIS ROUTINE TO DETERMINE IF A REQUEST FOR A DUMP RESULTED IN ACCESSING A CORE LOCATION OUTSIDE THE USERS AREA . IN THE EVENT THIS ACCESS OCCURS A GUARD MODE INTERUPT RESULTS AND THE DUMP ROUTINE CANNOT DETERMINE THE USERS CORE LIMITS AT ALL TIMES . THIS IS NOT A REQUIREMENT FOR SUCCESSFUL UTILIZATION OF DYNAMIC DUMPS , BUT IS INTENDED TO ASSIST IN SATISFACTORY PROGRAM TESTING .

THE DYNAMIC DUMPS MAKE SEVERAL CHECKS FOR 'REASONABLENESS', (E.G. DUMPING INTO THE CORRECT FILE , AND NOT DUMPING PAST THE FILE LIMIT) .

THE FILE LIMIT IS AN INSTALLATION VARIABLE AND IS SPECIFIED AT SYSTEM GENERATION TIME . THE PROGRAM TERMINATION ROUTINE WILL (IF A @PMD IMAGE OCCURS) DUMP THE COMPLETED PROGRAM AT A GIVEN SECTOR ADDRESS WHICH IS THE ADDRESS SPECIFIED AT GENERATION TIME . THE CONSTANT THEN IS THE LENGTH OF THE DYNAMIC DUMP FILE . IN THE EVENT THE DYNAMIC PORTION OF THE DIAGNOSTIC FILE IS FULL A MESSAGE FROM THE DUMP ROUTINES IS PRODUCED AND PRINTED WHEN THE DUMPS ARE EDITED , NO MORE DUMPS WILL BE TRANSFERRED TO THE DIAGNOSTIC FILE, HOWEVER, THE CALLS WILL BE HONORED AS WILL A CALL ON PMD .

18. SYSTEM SETUP

18.1. 1108 EXECUTIVE SYSTEM GENERATION

18.1.1. GENERAL

THE SYSTEM GENERATION PROGRAM PROVIDES EACH INSTALLATION WITH A MEANS OF DESCRIBING THE MACHINE CONFIGURATION AND OPERATING SYSTEM FEATURES IT DESIRES IN SUCH A MANNER THAT AN OPERATING SYSTEM IS PRODUCED TAILORED SPECIFICALLY TO THE INSTALLATION'S PARTICULAR HARDWARE CONFIGURATION AND DATA PROCESSING NEEDS. THE SYSTEM GENERATION ROUTINE ACCEPTS A CONTROL LANGUAGE WHICH DESCRIBES THE CONFIGURATION OF THE SITE. IT ALSO ACCEPTS INSTALLATION MODIFICATIONS AND ADDITIONS, UNIVAC SUPPLIED UPDATES TO THE SYSTEM, AND PARAMETERS INDICATING THE DESIRED OUTPUTS OF THE GENERATION RUN. THE ROUTINE THEN GENERATES A RUN WHICH PERFORMS THE REQUIRED SYSTEM GENERATION AND UPDATING. INPUT TO THE RUN ARE THE MASTER PROGRAM FILES OF THE SYSTEM COMPONENTS. THE OUTPUTS ARE UPDATED MASTER PROGRAM FILES, AN OPERABLE SYSTEM TAPE, AND A LISTING OF THE UPDATING RUN.

MAINTENANCE OF THE 1108 SOFTWARE SYSTEM IS EFFECTED THROUGH THE USE OF THE SYSTEM GENERATION PROGRAM. NORMALLY, ALL CHANGES ARE PREPARED BY UNIVAC AND FORWARDED TO EACH INSTALLATION. THESE CHANGES WILL INCLUDE THE NECESSARY UPDATE STATEMENTS AND SYMBOLIC CORRECTIONS. ALL PROGRAMS AFFECTED BY THE CHANGES ARE RE-ASSEMBLED AND RE-ALLOCATED. OUTPUT OPTIONS ARE THE SAME AS FOR NORMAL SYSTEM GENERATION. ALL HISTORICAL INFORMATION (FILE DIRECTORIES, ETC.) IS RETAINED WHENEVER AN UPDATE OF THE SYSTEM IS PERFORMED; HENCE, AN UPDATE IS NOT A COMPLETE REGENERATION OF THE SYSTEM. THE SYSTEM GENERATION PROGRAM WILL ALLOW FOR MAINTENANCE OF THE SYSTEM IN THE FACE OF ADDITIONS MADE TO THE SYSTEM BY A PARTICULAR INSTALLATION. FOR ALL INSTALLATIONS THE SAME DECK DISTRIBUTED BY UNIVAC WILL SUFFICE REGARDLESS OF CURRENT SYSTEM CONFIGURATION.

18.1.2. PROCEDURES

THE SYSTEM GENERATION ROUTINE (SGR) CONSTRUCTS AN ORDERED RUN FOR THE UPDATING OF A SYSTEM'S SYMBOLIC ELEMENTS (SYSTEM LIBRARY) AND FOR THE CONSTRUCTION OF AN OPERABLE SYSTEM TAPE, FROM A SET OF PARAMETER CARDS (SYSTEM GENERATION STATEMENTS - SGS'S), A SKELETON CORRECTION DECK, UNIVAC CORRECTIONS, AND USER'S CORRECTIONS. UNIVAC PROVIDES THE SGR WITH THE SKELETON CORRECTION DECK. THIS CORRECTION DECK CONTAINS INFORMATION RELATING THE SGS'S TO THE CORRECTIONS NECESSARY TO AFFECT THOSE CHANGES DESCRIBED ON THE SGS'S. UNIVAC CORRECTIONS WILL BE ENTERED IN THE SYSTEM LIBRARY, AS OPPOSED TO USER CORRECTIONS WHICH ONLY AFFECT THE OPERABLE SYSTEM TAPE.

SYSTEM GENERATION STATEMENTS (SGS'S) ARE COMPRISED OF ANY NUMBER OF FIELDS WHICH MAY BE COMPRISED OF ANY NUMBER OF SUB-FIELDS. A SUB-FIELD IS A STRING OF 18 OR FEWER CHARACTERS NOT CONTAINING A SPACE, COMMA, RIGHT BRACKET, LEFT BRACKET, RIGHT PARENTHESIS, LEFT PARENTHESIS, PLUS SIGN, MINUS SIGN, OR EQUAL SIGN. NOTE, THE FIRST FIELD NEED NOT START IN COLUMN 1.

A STATEMENT MAY BE CONTINUED ON THE FOLLOWING CARD BY PLACING A SEMI-COLON FOLLOWING THE LAST SPECIFICATION FIELD.

THE SGS CARD FILE MAY BE FOLLOWED BY A FILE OF UNIVAC CORRECTIONS (PERMANENT CORRECTIONS) AND A CARD FILE OF USER CORRECTIONS (TEMPORARY CORRECTIONS). THE CORRECTIONS ARE OF THE FORM DESCRIBED IN SECTION 5.6.3. ALL CORRECTIONS TO AN ELEMENT MUST BE GROUPED, ORDERED, AND PRECEDED BY A CARD WITH AN ASTERISK IN COLUMN ONE AND THE NAME AND VERSION OF THE ELEMENT TO WHICH THE CORRECTIONS APPLY, IN THE SECOND AND THIRD FIELDS. USER CORRECTIONS WILL BE REPRODUCED WITH UPDATED LINE NUMBERS RESULTING FROM UNIVAC ADDITIONS OR DELETIONS.

18.1.3. SYSTEM GENERATION STATEMENTS

ACCOUNT PRIORITY AND DEADLINE

THE ACCOUNT STATEMENT IS USED TO SPECIFY THE LIMITATIONS IMPOSED ON A GIVEN ACCOUNT NUMBER. THIS STATEMENT IS OF THE FORM:

```
ACCOUNT N PRIORITY P,Q DEADLINE A REAL-TIME A

  WHERE -N IS THE ACCOUNT NUMBER
        -P IS THE HIGHEST PRIORITY A RUN WITH THE ACCOUNT
           NUMBER N MAY TAKE.
           'A'-'Z'
        -Q IS THE PRIORITY USED WHEN NONE HAS BEEN
           SPECIFIED ON THE @RUN STATEMENT.
           'A'-'Z'
  NOTE:  THE 'PRIORITY' AND 'P,Q' FIELDS MAY BE OMITTED
           WHEN THE STANDARD PRIORITY VALUES ARE TO BE
           USED.
        -A IS TO SPECIFY WHETHER DEADLINE OR REAL-TIME
           PROGRAMS ARE TO BE ALLOWED FOR THE GIVEN
           ACCOUNT NUMBER.  A MAY BE EITHER
           'A' (ALLOWED) OR
           'N' (NOT ALLOWED)
  NOTE:  THE 'DEADLINE' AND 'A' FIELDS MAY BE
           OMITTED WHEN THE SYSTEM STANDARD IS TO BE
           USED.
  NOTE:  THE 'REAL-TIME' AND 'A' FIELDS MAY BE
           OMITTED WHEN THE SYSTEM STANDARD IS TO BE
           USED.
  NOTE:  SEE DELETE ACCOUNTS.
  SEE   -EXECUTIVE CONTROL LANGUAGE
           ORGANIZATIONAL STATEMENTS
             THE @RUN STATEMENT
               PRIORITY SUB-FIELD
```

ADD PROCESSOR

THIS STATEMENT IS TO SPECIFY PROCESSORS TO BE ADDED TO THE SYSTEM. ONE ADD STATEMENT IS NECESSARY FOR EACH PROCESSOR TO BE ADDED. THE ADD STATEMENT IS OF THE FORM

```
ADD PROCESSOR  N  F  PRODUCES  A

  WHERE -N IS THE PROCESSOR NAME.
        -F IS THE NAME OF THE FILE WHERE THE PROCESSOR IS
           LOCATED.
        -A IS ONE OF THE FOLLOWING.
           'A SYMBOLIC ELEMENT'
           'A RELOCATABLE ELEMENT'
           'BOTH A SYMBOLIC AND A RELOCATABLE ELEMENT'
  NOTE:  SEE DELETE PROCESSORS.
  SEE   -EXECUTIVE CONTROL LANGUAGE
           PROCESSOR CALL STATEMENTS
             THE ADDITION OF NEW PROCESSORS
```

ASSOCIATE SYMBIONTS

THIS STATEMENT SPECIFIES SYMBIONT ASSOCIATIONS. THE STATEMENT IS OF THE FORM:

```
ASSOCIATED SYMBIONTS I  PRINT O1,...,ON  PUNCH P1,...,PM
```

```
WHERE -I IS THE INPUT SYMBIONT NAME.
      -O1,...,ON ARE THE ASSOCIATED PRINT SYMBIONT
         NAMES.
NOTE:  THE 'PRINT' AND 'O1,...,ON' FIELDS MAY BE
         OMITTED WHEN NO PRINT SYMBIONTS ARE
         ASSOCIATED.
      -P1,...,PM ARE THE ASSOCIATED PUNCH SYMBIONT
         NAMES.
NOTE:  THE 'PUNCH' AND 'P1,...,PM' FIELDS MAY BE
         OMITTED WHEN NO PUNCH SYMBIONTS ARE
         ASSOCIATED.
NOTE:  ONE STATEMENT IS NECESSARY FOR EACH DEVICE
         ASSOCIATION DESIRED.
NOTE:  SEE DELETE SYMBIONTS.
SEE   -EXECUTIVE CONTROL LANGUAGE
         ORGANIZATIONAL STATEMENTS
           THE @SYM STATEMENT
             USE OF @SYM WITH PRINT$ AND PUNCH$
SEE   -SYMBIONTS
         GENERAL DESCRIPTION
         SYSTEM GENERATION PARAMETERS
           DEVICE ASSOCIATION
```

AUTOMATIC SCHEDULING

THIS STATEMENT SPECIFIES ONE USER RUN TO BE AUTOMATICALLY SCHEDULED WHEN THE SYSTEM IS BOOTSTRAPPED. THE STATEMENT HAS THE FORM:

```
AUTOMATICALLY SCHEDULE  P

    WHERE -P IS THE NAME OF THE FILE OR PROGRAM FILE
             ELEMENT TO BE STARTED.

    SEE   -EXECUTIVE CONTROL LANGUAGE
             THE @START STATEMENT
```

AVERAGE PROGRAM SIZE

THIS STATEMENT IS USED TO SPECIFY THE AVERAGE SIZE OF FORTRAN OR COBOL PROGRAMS. THE STATEMENT IS OF THE FORM:

```
AVERAGE  SIZE  OF . P  PROGRAMS  S

  WHERE -P IS THE PROCESSOR NAME
           'FORTRAN'
           'COBOL'
        -S IS THE NUMBER OF STATEMENTS
```

CATALOGUED FILES ESTIMATE

THIS VALUE ESTIMATES THE NUMBER OF FILES WHICH THE SYSTEM WILL BE REQUESTED TO CATALOGUE. AN ACCURATE ESTIMATE WILL AID IN THE EFFICIENT OPERATION OF THE MASTER DIRECTORY SEARCH TECHNIQUE. THE FORM IS:

```
CATALOGUED FILES  N

  WHERE -N IS THE NUMBER OF FILES EXPECTED.
```

COMMUNICATIONS OPTIONS

MAXIMUM COMMUNICATIONS BUFFER LENGTH

THIS STATEMENT IS USED TO ESTABLISH A MAXIMUM PERMISSABLE BUFFER LENGTH TO BE ACCEPTED FOR BOTH INPUT AND OUTPUT COMMUNICATIONS BUFFERS FOR ALL REAL-TIME PROGRAMS AND SYSTEM SYMBIONTS FOR REMOTE TERMINALS. THIS STATEMENT IS OF THE FORM:

```
MAXIMUM COMMUNICATIONS BUFFER LENGTH  N  CHARACTERS

  WHERE THE VALUE OF N IS EXPRESSED IN CHARACTERS.
```

MAXIMUM ESI COMPLETION ACTIVITY TIME

THIS STATEMENT IS USED BY THE DISPATCHER AS THE LIMIT FOR THE AMOUNT OF TIME AN ESI COMPLETION ACTIVITY MAY HAVE CONTROL IN ORDER TO DETECT CLOSED LOOPS AND EXCESSIVE PROCESSING BY A REAL-TIME PROGRAM WHICH MIGHT HAVE AN ADVERSE EFFECT ON OTHER REAL-TIME PROGRAMS AND OTHER PARTS OF THE OPERATING SYSTEM. THIS STATEMENT IS OF THE FORM:

```
MAXIMUM ESI COMPLETION ACTIVITY TIME  N  MILLISECONDS

  WHERE THE VALUE OF N IS EXPRESSED IN MILLISECONDS
```

REVERSE BUFFERING

THIS STATEMENT IS USED TO DEFINE WHETHER OR NOT THE CPU HARDWARE FEATURE OF THE REVERSE BUFFERING CAPABILITY IS TO BE EMPLOYED BY THE SOFTWARE. THIS STATEMENT IS OF THE FORM:

```
REVERSE BUFFERING FOR COMMUNICATIONS  A

  WHERE -A IS EITHER
           'ALLOWED' OR
           'NOT ALLOWED'
```

COMMUNICATIONS BUFFER TIMING

THIS STATEMENT DETERMINES WHETHER OR NOT THE TIMING OF COMMUNICATIONS BUFFERS IS TO BE PROVIDED. THIS STATEMENT IS OF THE FORM:

```
COMMUNICATIONS BUFFER TIMING  A

  WHERE -A IS EITHER
           'PERFORMED' OR
           'OMITTED'
```

AUTOMATIC DIALING OPERATIONS

THIS STATEMENT IS USED TO SPECIFY WHETHER OR NOT ANY AUTOMATIC DIALING CAPABILITIES ARE PRESENT AT THE CENTRAL SITE. THIS STATEMENT IS OF THE FORM:

```
DIALING OPERATIONS  A

  WHERE -A IS EITHER
           'ACCEPT' OR
           'REJECT'
```

SYMBIONT SUPPORT

THIS STATEMENT DETERMINES WHETHER OR NOT SECOND LEVEL

SUPPORT IS TO BE PROVIDED. SECOND LEVEL SUPPORT IS USED PRIMARILY BY THE SYMBIONTS. THIS STATEMENT IS OF THE FORM:

```
SECOND LEVEL COMMUNICATIONS SUPPORT  A

  WHERE -A IS EITHER
           'PROVIDED' OR
           'OMITTED'
```

COMMUNICATIONS BUFFER POOLS

THIS STATEMENT DETERMINES WHETHER OR NOT THE USE OF BUFFER POOLS BY ANY REAL-TIME PROGRAM IS TO BE PERMITTED. THIS STATEMENT IS OF THE FORM:

```
COMMUNICATIONS BUFFER POOLS  A

  WHERE -A IS EITHER
           'ALLOWED' OR
           'NOT ALLOWED'
```

COMMUNICATIONS LOG ENTRIES

THIS STATEMENT DETERMINES WHETHER OR NOT SYSTEM LOG ENTRIES ARE TO BE MADE WITH INFORMATION SUCH AS TIME OF INITIATION AND VOLUME OF USAGE. THIS STATEMENT IS OF THE FORM:

```
COMMUNICATIONS LOG ENTRIES  A

  WHERE -A IS EITHER
           'PROVIDED' OR
           'OMITTED'
```

REAL-TIME OPERATIONS VALIDATION

THIS STATEMENT IS USED TO DETERMINE WHETHER COMPLETE VALIDATION IS TO BE PERFORMED FOR UNDEBUGGED REAL-TIME PROGRAMS OR IF ONLY PARTIAL VALIDATION IS TO BE PERFORMED ON STABLE AND THOROUGHLY DEBUGGED REAL-TIME PROGRAMS. THIS STATEMENT IS EXPRESSED AS EITHER:

```
PERFORM COMPLETE REAL-TIME VALIDATION
          OR
PERFORM PARTIAL REAL-TIME VALIDATION
```

CONNECT CHANNEL & DISCONNECT CHANNEL

THESE STATEMENTS ARE TO SPECIFY CHANNEL CONNECTIONS, CHANNEL TYPE, EQUIPMENT TYPE, AND THE NUMBER OF UNITS ATTACHED. THE CONNECT AND DISCONNECT STATEMENTS ARE NOT TO BE USED FOR THE ALTERNATE CHANNEL OF A DUAL PAIR. THESE STATEMENTS ARE OF THE FORM:

```
CONNECT CHANNEL C TO D, T U
DISCONNECT D FROM CHANNEL C, T U

  WHERE -C IS THE CHANNEL NUMBER
        -D IS THE EQUIPMENT TYPE GIVEN BY ANY ONE OF THE
         FOLLOWING.
         'VIIIC'    (UNISERVO VIII C)
         'VIC'      (UNISERVO VI C)
         'IVC'      (UNISERVO IV C)
```

```
            'IIIA'      (UNISERVO III A)
            'IIA'       (UNISERVO II A)
            'FH432'     (FLYING HEAD 432 MAGNETIC DRUM)
            'FH880'     (FLYING HEAD 880 MAGNETIC DRUM)
            'FASTRAND'  (FASTRAND II)
            'HSP755'    (HIGH SPEED PRINTER)
            'RP'        (CARD READER AND PUNCH)
            'PT'        (PAPER TAPE)
            '1004II'    (1004 II)
            '1004III'   (1004 III)
            'CTS'       (COMMUNICATIONS)
            'WTS'       (WORD TERMINAL)
            'CTMC'      (COMMUNICATIONS TERMINAL CONTROL)
            'NONSTD'    (NONSTANDARD DEVICE TYPE)
         -T IS THE CHANNEL TYPE GIVEN BY ONE OF THE
            FOLLOWING.
            'S' (SINGLE CHANNEL)
            'P' (PARTIAL DUAL - R/W,R/R)
            'D' (FULL DUAL - R/W,R/W)
   NOTE:  THE ALTERNATE CHANNEL IS ASSUMED TO BE C+1.
         -U IS THE NUMBER OF UNITS.
   SEE   -SUPERVISOR
            SCHEDULING
              FACILITIES INVENTORY
```

FOR NONSTANDARD DEVICES IT IS NECESSARY TO USE THE NAME ASSOCIATION SYSTEM GENERATION PARAMETER TO SPECIFY SYMBOLIC NAMES TO USE FOR ASSIGNING THESE DEVICES. THE NUMBER OF UNITS FIELD IS DISREGARDED FOR NONSTANDARD DEVICE CHANNELS. THESE CHANNELS ARE ASSIGNED AS AN ENTITY.

OTHER, MORE SPECIFIC STATEMENTS MAY BE INCLUDED TO MORE FULLY DESCRIBE THE CHANNEL EQUIPMENT. THESE ARE OF THE FORM:

```
IVC 800 PPI C
```

THIS STATEMENT IS TO SPECIFY THAT A DENSITY OF 880 PPI IS AVAILABLE ON THE IV C TAPE UNITS ON CHANNEL C.

```
IVC TRANSLATE C
VIC  TRANSLATE C
VIIIC TRANSLATE C
```

THIS STATEMENT SPECIFIES THE TRANSLATE FEATURE IS AVAILABLE ON THE TAPE UNITS ON CHANNEL C.

```
FASTRAND FASTBAND C
```

THIS STATEMENT SPECIFIES FASTBAND IS AVAILABLE ON THE FASTRAND II UNITS ON CHANNEL C.

```
FASTRAND SECTOR BUFFER C
```

THIS STATEMENT SPECIFIES THAT A SECTOR BUFFER IS PRESENT IN THE CONTROL UNIT OF THIS FASTRAND CHANNEL, AND IN THE AJOINING CONTROL FOR DUAL-CHANNEL SUBSYSTEMS.

```
1004 CODE IMAGE READER C
1004 PAPER TAPE READER C
1004 PAPER TAPE PUNCH C
```

```
    1004 VIC N C '

      THESE STATEMENTS SPECIFY ADDITIONAL 1004 II AND
      1004 III FEATURES.
      N GIVES THE NUMBER OF UNITS ON THE 1004 III.

      SEE    -EXECUTIVE CONTROL LANGUAGE
                ORGANIZATIONAL STATEMENTS
                  THE @COL STATEMENT

    CTS AUTOMATIC DIALING C
    CTS UNATTENDED        C
    CTS LEASED LINE       C
    WTS AUTOMATIC DIALING C
    WTS UNATTENDED        C
    WTS LEASED LINE       C

      THESE STATEMENTS ARE INCLUDED TO DESCRIBE MORE SPECIFTC
      FEATURES OF THE CTS AND WTS. CARDS REFERRING TO A
      LEASED LINE ARE USED FOR LINE SPEEDS OTHER THAN 2400
      BPS.


CONTINUE OR TERMINATE RUN
  THESE STATEMENTS ARE TO SPECIFY RUN CONTINUATION OR
  TERMINATION WHEN PRINT AND PUNCH FILES AREAS ARE EXCEEDED.
  THE STATEMENTS ARE OF THE FORM:

    CONTINUE   RUN   WHEN   P   FILE   EXCEEDED
    TERMINATE  RUN   WHEN   P   FILE   EXCEEDED

      WHERE -P IS EITHER
                'PRINT' OR
                'PUNCH'
      SEE    -EXECUTIVE CONTROL LANGUAGE
                ORGANIZATIONAL STATEMENTS
                  THE @RUN STATEMENT
                    PAGES SUB-FIELD
      SEE    -EXECUTIVE CONTROL LANGUAGE
                ORGANIZATIONAL STATEMENTS
                  THE @RUN STATEMENT
                    CARDS SUB-FIELD

  ANOTHER FORM OF THESE STATEMENTS ARE USED TO SPECIFY RUN
  CONTINUATION OR TERMINATION WHEN THE RUN ESTIMATED TIME
  HAS BEEN EXCEEDED.

    CONTINUE    RUN   WHEN   TIME   LIMIT   EXCEEDED
    TERMINATE RUN WHEN TIME LIMIT EXCEEDED

      SEE    -EXECUTIVE CONTROL LANGUAGE
                ORGANIZATIONAL STATEMENTS
                  THE @RUN STATEMENT
                    RUNNING TIME/DEADLINE FIELD


CORE SIZE
  THIS STATEMENT IS TO SPECIFY THE SIZE OF CORE STORAGE
  AVAILABLE. THE STATEMENT IS OF THE FORM:

    CORE SIZE N K

      WHERE -N IS THE NUMBER OF WORDS (IN THOUSANDS) IN CORE
                STORAGE.  THE OPTIONS ARE
```

```
            '65'
            '131'
            '196'
            '262'
   SEE   -SUPERVISOR
            SCHEDULING
              FACILITIES INVENTORY
```

DELETE LISTS

THE FOLLOWING STATEMENTS ARE USED TO DELETE ENTIRE LISTS PREVIOUSLY CONSTRUCTED BY THE SYSTEM GENERATION ROUTINE. A LIST MAY BE DELETED AND RECONSTRUCTED WITH ONE SYSTEM GENERATION. THE STATEMENTS ARE OF THE FORM:

```
DELETE  ACCOUNTS
DELETE  PROCESSORS
DELETE  SYMBIONTS
DELETE  CHANNEL UNIT NAMES
DELETE  RE-ENTRANT ROUTINES
DELETE  REMOTE TERMINALS
REMOVE  PROCESSORS
```

DEMAND FACILITY

THIS STATEMENT INCLUDES OR DELETES THE DEMAND FACILITY OF THE EXECUTIVE. THE FORM IS:

```
DEMAND FACILITY S

  WHERE -S IS EITHER:
          'INCLUDED', OR
          'DELETED'
```

DIVIDE DRUM

THE DIVIDE STATEMENT IS TO DIVIDE FH432 AND FH880 DRUM INTO CONTIGUOUS AREAS AND SIMULATED FASTRAND. ALL REMAINING DRUM IS USED AS SIMULATED FASTRAND. THE STATEMENT IS OF THE FORM:

```
DIVIDE  C  U  N

  WHERE -C IS THE CHANNEL NUMBER.
        -U IS THE UNIT NUMBER.
        -N IS THE NUMBER OF LOCATIONS TO BE ALLOCATED AS
           CONTIGUOUS DRUM.
  SEE   -SUPERVISOR
           SCHEDULING
             FACILITIES INVENTORY
  SEE   -FILE CONTROL SYSTEM
           FILE ORGANIZATION
             MASS STORAGE ALLOCATION
               GENERAL
```

LOG MEDIUM

THIS STATEMENT IS TO SPECIFY THE LOG MEDIUM, THE STATEMENT IS OF THE FORM:

```
LOG MEDIUM  TYPE

    WHERE -TYPE IS THE EQUIPMENT TYPE.
```

MAXIMUM DELAY TIME

THIS STATEMENT SETS THE MAXIMUM TIME AN EXTERNAL FUNCTION CAN BE OUTSTANDING FOR THE ARBITRARY DEVICE HANDLER. THE STATEMENT IS OF THE FORM:

```
MAXIMUM DELAY TIME T SECONDS

  WHERE -T IS THE TIME IN SECONDS.
```

MAXIMUM FILE SIZE

THIS STATEMENT IS USED TO SET THE SYSTEM MAXIMUM FILE SIZE TO BE USED WHEN A MAXIMUM FILE SIZE HAS NOT BEEN SPECIFIED ON AN ASSIGN STATEMENT. THE STATEMENT IS OF THE FORM:

```
MAXIMUM FILE SIZE  N

  WHERE -N IS THE NUMBER OF TRACKS
  SEE   -EXECUTIVE CONTROL LANGUAGE
           INPUT/OUTPUT SPECIFICATION STATEMENTS
             THE @ASG STATEMENT
               THE FASTRAND @ASG STATEMENT
```

MAXIMUM NUMBER OF LOG ENTRIES

THIS STATEMENT IS TO SPECIFY THE MAXIMUM NUMBER OF LOG ENTRIES ANY ONE RUN MAY MAKE. THIS STATEMENT IS OF THE FORM:

```
LOG MAXIMUM  N

  WHERE -N IS THE MAXIMUM NUMBER.
  SEE   -EXECUTIVE REQUEST FUNCTIONS
           THE SYSTEM LOG
```

MINIMUM DBANK ADDRESS

THIS STATEMENT SETS THE MINIMUM DBANK ADDRESS FOR EACH PROGRAM COLLECTED. ITS EFFECT IS TO RESERVE IBANK ADDRESS SPACE FOR LINKING RE-ENTRANT PROCESSORS. THE FORM IS:

```
MINIMUM DBANK ADDRESS  A

  WHERE A IS THE OCTAL OR DECIMAL ADDRESS.
```

MAXIMUM NUMBER OF OPENED RUNS

THIS STATEMENT IS USED TO SPECIFY THE MAXIMUM NUMBER OF RUNS THAT MAY BE OPENED AT ANY ONE TIME. THE STATEMENT IS OF THE FORM:

```
MAXIMUM OPENED RUNS  N

  WHERE -N IS THE NUMBER OF RUNS.
  SEE   -FILE CONTROL SYSTEM
           FILE ORGANIZATION
             MASS STORAGE ALLOCATION
               ROLLOUT OF FILES TO MAGNETIC TAPE
```

NAME AND C/U ASSOCIATION

THIS STATEMENT IS TO IDENTIFY CHANNELS AND SPECIFIC UNITS WITH SYMBOLIC NAMES. THESE NAMES ARE TO BE USED IN THE 'TYPE' FIELD TO OBTAIN AN ASSIGNMENT. THIS STATEMENT IS OF THE FORM:

```
      NAME N IS C U

    WHERE -N IS A ONE TO SIX CHARACTER NAME.
          -C IS THE CHANNEL NUMBER.
          -U IS THE UNIT NUMBER.
    NOTE:  IF A NAME IS TO BE ASSOCIATED WITH A CHANNEL
             THE U FIELD IS NOT NECESSARY.
    NOTE:  ONE STATEMENT IS NECESSARY FOR EACH ASSOCIATION
             DESIRED.
    NOTE:  SEE DELETE CHANNEL UNIT NAMES.
    SEE   -EXECUTIVE CONTROL LANGUAGE
             INPUT/OUTPUT SPECIFICATION STATEMENTS
               THE @ASG STATEMENT
                 THE ARBITRARY DEVICE @ASG STATEMENT
```

NOISE CONSTANT

THIS STATEMENT IS USED TO SET THE STANDARD TAPE NOISE CONSTANT. THE STATEMENT IS OF THE FORM:

```
  NOISE CONSTANT  N

    WHERE -N IS THE NOISE CONSTANT VALUE.
```

PROBE

THIS STATEMENT IS USED TO SPECIFY THE INPUT SYMBIONT, OR SYMBIONT CLASS, WHICH IS TO BE PLACED IN THE PROBE MODE. THE STATEMENT IS OF THE FORM:

```
  PROBE  S

    WHERE -S IS THE SYMBIONT NAME, OR SYMBIONT CLASS,
             OF THE CORRESPONDING UNIT TO BE PROBED.
    SEE   -SYMBIONTS
             ON-SITE DEVICE HANDLERS
               INPUT OPERATION
    SEE   -SYMBIONTS
             SYSTEM GENERATION PARAMETERS
               PROBE
```

PROCESSOR DESCRIPTION

THIS STATEMENT DEFINES THE TYPE AND CHARACTERISTICS OF THE CENTRAL PROCESSING UNIT. THE FORM IS:

```
      PROCESSOR P CHANNELS C

        WHERE -P IS '1108' OR '1108A', AND
              -C IS THE NUMBER OF CHANNELS PRESENT.
```

RE-ENTRANT ROUTINES

THIS STATEMENT IS TO IDENTIFY RE-ENTRANT ROUTINES. THE STATEMENT IS OF THE FORM:

```
  REENTRANT  ROUTINE  R
```

```
    WHERE -R IS ROUTINE NAME.
    NOTE:  SEE DELETE RE-ENTRANT ROUTINES.
    SEE   -BASIC OPERATIONAL DESCRIPTIONS
             DEMAND PROCESSING
               CONVERSATIONAL COMPONENTS
                 RE-ENTRANT CONTROL


  REAL TIME FACILITY

    THIS STATEMENT INCLUDES OR DELETES THE REAL TIME
    CAPABILITY OF THE EXECUTIVE.  THE FORM IS:

      REAL TIME S

        WHERE -S IS 'INCLUDED' OR 'DELETED'.


REMOTE TERMINALS
  THIS STATEMENT IS USED TO IDENTIFY THE EQUIPMENT AT THE
  REMOTE TERMINALS. THE STATEMENT IS OF THE FORM:

    REMOTE TERMINAL      I  C  R  P  T  Q

      WHERE -I IS A SIX CHARACTER FIELD WHICH
               IDENTIFIES THE REMOTE LOCATION.
            -C IS THE REMOTE CONFIGURATION AND MAY BE ONE OF
               THE FOLLOWING.
               '1004II'  (1004 II)
               '1004III' (1004 III)
               'KSR35'*   (MODEL 35 TELETYPE*OR EQUIVALENT)
               'ASR35'*   (AUTOMATIC SEND RECEIVE MODEL 35)*
            -R IS THE NAME OF A SYMBIONT (FOR RUN INPUT) OR
               THE NAME OF A FILE (TO AUTOMATICALLY START
               A PREDEFINED RUN) FOR THE INPUT STANDBY IDLE
               STATE CONDITION.  IF R IS A NAME OF A FILE,
               THIS FIELD HAS SUBFIELDS AS FOLLOWS:
                 Q,F,K1,K2,E,V

                 WHERE -Q IS THE QUALIFIER (OPTIONAL).
                       -F IS THE FILE NAME PORTION OF THE
                          EXTERNAL NAME.
                       -K1 IS THE READ KEY (OPTIONAL).
                       -K2 IS THE WRITE KEY (OPTIONAL).
                       -E IS THE NAME OF THE ELEMENT WITHIN
                          THE FILE IF MORE THAN ONE.
                       -V IS THE ELEMENT VERSION (APPLIES
                          ONLY IF E IS SPECIFIED).

            -P IS A ONE TO SIX CHARACTER PRINT SITE-ID.
            -T IS THE PROGRAM TYPE (IF NOT OTHERWISE
               SPECIFIED) OF ALL RUNS ENTERED AT THIS
               TERMINAL.  T IS EITHER
               'BATCH' OR
               'DEMAND'.
            -Q IS A ONE TO SIX CHARACTER PUNCH SITE-ID.
      NOTE:  THE Q FIELD MAY BE OMITTED.
      NOTE:  ONE STATEMENT IS NECESSARY FOR EACH INPUT OUTPUT
               ASSOCIATION DESIRED.
      NOTE: SEE DELETE REMOTE TERMINALS.
      SEE   -EXECUTIVE CONTROL LANGUAGE
               ORGANIZATIONAL STATEMENTS
                 THE @RUN STATEMENT
                   PRIORITY SUB-FIELD
```

*TRADEMARK OF TELETYPE CORPORTION

```
        SEE   -EXECUTIVE CONTROL LANGUAGE
                 INPUT/OUTPUT SPECIFICATION STATEMENTS
                   THE @ASG STATEMENT
                     THE ARBITRARY DEVICE @ASG STATEMENT
        SEE   -SYMBIONTS
                 SYSTEM GENERATION PARAMETERS
                   DEVICE ASSOCIATION


  RESERVE MASS STORAGE
    THE PURPOSE OF THIS STATEMENT IS TO RESERVE A MASS STORAGE
    AREA AS A NAMED PERMANENT FILE. THE STATEMENT IS OF THE
    FORM:

      RESERVE  F  C  U  A  D  G
        WHERE -F IS THE FILE NAME BY WHICH THE FILE IS CATA-
                 LOGUED AND RETRIEVED.  THE FILE NAME FIELD
                 HAS THE FORMAT:
                   Q,F,K1,K2
                   WHERE -Q IS THE QUALIFIER (OPTIONAL).
                         -F IS THE FILE NAME PORTION OF THE
                            EXTERNAL NAME.
                         -K1 IS THE READ KEY (OPTIONAL).
                         -K2 IS THE WRITE KEY (OPTIONAL).
                   NOTE:  THE F-CYCLE SUBFIELD OF THE FILE
                          NAME AS DISCUSSED ELSEWHERE IN THIS
                          DOCUMENT IS ALWAYS TAKEN AS ZERO
                          FOR THE RESERVE STATEMENT.

              -C IS THE CHANNEL NUMBER.
              -U IS THE UNIT NUMBER.
              -A IS THE STARTING ADDRESS
              -D IS TO INDICATE WHETHER POSITIONS OR TRACKS
                 ARE TO BE ASSIGNED. THE D FIELD MAY CONTAIN
                 'POSITIONS' OR
                 'TRACKS'.
              -G IS THE NUMBER OF GRANULES TO ASSIGN. THE
                 STARTING ADDRESS IS THE LOGICAL SECTOR ADDRESS
                 0 TO N FOR EACH UNIT AND MUST BE THE START OF
                 A TRACK IF D CONTAINS 'TRACKS' OR POSITION IF
                 D CONTAINS 'POSITIONS'.
        SEE   -EXECUTIVE CONTROL LANGUAGE
                 INPUT/OUTPUT SPECIFICATION STATEMENTS
                   THE @ASG STATEMENT
                     THE FASTRAND @ASG STATEMENT


  RETAIN CYCLES AND F-CYCLES.
    THIS STATEMENT IS TO SPECIFY THE NUMBER OF CYCLES TO RETAIN
    WHEN A NUMBER HAS NOT BEEN SPECIFIED. THE STATEMENT IS OF
    THE FORM:

      RETAIN  N  C

        WHERE -N IS THE NUMBER OF CYCLES.
              -C IS EITHER
                 'CYCLES' OR
                 'FCYCLES'


  ROLLOUT CONTROL POINTS
    THIS STATEMENT IS USED TO SPECIFY THE FASTRAND AVAILABILITY
    POINTS AT WHICH ROLL OUT OF FILES IS AUTOMATICALLY STARTED
    AND STOPPED:
```

```
    ROLLOUT POINTS  N  M

      WHERE -N IS THE NUMBER OF TRACKS AVAILABLE WHEN ROLLOUT
                STARTS.
            -M IS THE NUMBER OF TRACKS AVAILABLE WHEN ROLLOUT
                IS STOPPED.
      SEE   -FILE CONTROL SYSTEM
               FILE ORGANIZATION
                 MASS STORAGE ALLOCATION
                   ROLLOUT OF FILES TO MAGNETIC TAPE
      SEE   -EXECUTIVE SYSTEM DESIGN CRITERIA
               PROGRAM FILES
                 'CYCLE' PARAMETER
      SEE   -EXECUTIVE CONTROL LANGUAGE
               INPUT/OUTPUT SPECIFICATION STATEMENTS
                 THE @ASG STATEMENT
                   NOTATION FOR FILE NAMES
      SEE   -EXECUTIVE CONTROL LANGUAGE
               PROCESSOR CALL STATEMENTS
                 STATEMENT FORMAT


CTMC
  THIS STATEMENT IS TO DEFINE THE UNITS ON THE CTMC. THE
  STATEMENT IS OF THE FORM:


    CTMC      G  C  T  S  B  N  L  M  I,O  D

      WHERE -G IS THE LT GROUP IDENTITY.
            -C IS THE CHANNEL NUMBER.
            -T IS THE UNIT TYPE AND MAY CONTAIN:
               'STANDARD'
               'NASA'
               'GSA3EI'
            -S IS THE UNIT SPEED AND MAY CONTAIN:
               'LOW'
               'MEDIUM'
               'SYNC'
               'PARALLEL'
            -B IS THE LINE SPEED IN BITS PER SECOND.
            -N IS THE NUMBER OF BITS PER CHARACTER.
            -L IS THE LINE TYPE AND MAY CONTAIN:
               'COMMONCARRIER'
               'LEASED'
               'TELEGRAPH'
            -M IS THE IDLE STATE FOR AN INPUT UNIT.
               'OFF' SPECIFIES NO IDLE STATE MONITOR.
               'STANDBY' CALL FOR AUTOMATIC STARTING
               OF A RUN DEFINED IN THE REMOTE STATEMENT.
            -I IS THE INPUT UNIT INDICATOR NUMBER.
            -O IS THE OUTPUT UNIT INDICATOR NUMBER.
            -D IS THE DIAL UNIT INDICATOR NUMBER
               WHEN PRESENT.
      SEE   -INPUT/OUTPUT DEVICE HANDLERS
               COMMUNICATIONS MULTIPLEXOR HANDLER
                 ASSIGNING LT DEVICES


SHARING PERCENTAGES
  THESE STATEMENTS ARE USED TO SPECIFY HOW CPU TIME IS TO BE
  SHARED AMONG THE FOUR TYPES OF PROGRAMS.  THE STATEMENTS
  ARE OF THE FORM:
```

```
DEMAND MINIMUM PERCENTAGE  N
DEMAND MAXIMUM PERCENTAGE  N

  WHERE -N IS THE PERCENTAGE.
  SEE   -SUPERVISOR
           SCHEDULING
             THE DYNAMIC ALLOCATOR
               SHARING BETWEEN PROGRAM TYPES
```

STANDARD DEADLINE SPECIFICATION

THIS STATEMENT IS TO SPECIFY THE ASSUMED DEADLINE SPECIFICATION (THE DEADLINE SPECIFICATION USED IN THE ABSENCE OF A DEADLINE SPECIFICATION ON AN ACCOUNT STATEMENT). THIS STATEMENT IS OF THE FORM:

```
STANDARD DEADLINES  A

  WHERE -A IS EITHER
           'ALLOWED' OR
           'NOT ALLOWED'
  SEE   -EXECUTIVE CONTROL LANGUAGE
           ORGANIZATIONAL STATEMENTS
             THE @RUN STATEMENT
               PRIORITY SUB-FIELD
```

STANDARD DENSITY

THIS STATEMENT IS USED TO RESET THE STANDARD TAPE DENSITY. THIS IS THE DENSITY TO BE USED IF NOT SPECIFIED ON THE @ASG STATEMENT OR SET MODE I/O FUNCTION. THIS STATEMENT IS OF THE FORM:

```
STANDARD DENSITY N

  WHERE -N IS THE NUMBER OF FRAMES PER INCH.
```

STANDARD PAGE AND CARD LIMIT

THESE STATEMENTS SET THE ASSUMED PAGE AND CARD LIMITS (THE LIMITS USED WHEN NONE HAVE BEEN SPECIFIED ON THE RUN CARD). THESE STATEMENTS ARE OF THE FORM:

```
STANDARD PAGE LIMIT  N
STANDARD CARD LIMIT  M

  WHERE -N IS THE MAXIMUM NUMBER OF PAGES.
        -M IS THE MAXIMUM NUMBER OF CARDS.
  SEE   -EXECUTIVE CONTROL LANGUAGE
           ORGANIZATIONAL STATEMENTS
             THE @RUN STATEMENT
               PAGES SUB-FIELD
  SEE   -SYMBIONTS
           SYSTEM GENERATION PARAMETERS
             MASS STORAGE REQUIREMENTS
```

STANDARD RUN TIME

THE RUN TIME STATEMENT IS USED TO SET THE ASSUMED RUN ESTIMATED TIME (THE ESTIMATED TIME USED WHEN NONE HAS BEEN SPECIFIED ON THE @RUN CARD). THIS STATEMENT IS OF THE FORM:

```
STANDARD RUN TIME  T  MINUTES
```

```
    WHERE -T IS THE TIME IN MINUTES.
    SEE   -EXECUTIVE CONTROL LANGUAGE
              ORGANIZATIONAL STATEMENTS
                THE @RUN STATEMENT
                  THE RUN-OPTIONS SUB-FIELD
```

STARTING UNIT NUMBER

THIS STATEMENT IS TO SPECIFY THE STARTING UNIT NUMBER ON A GIVEN CHANNEL WHEN THE STARTING UNIT NUMBER IS OTHER THAN ZERO. THE STATEMENT HAS THE FORM:

```
  STARTING UNIT NUMBER ON CHANNEL  C  U

    WHERE -C IS THE CHANNEL NUMBER.
          -U IS THE STARTING UNIT NUMBER.
```

SYSTEM RESIDENCE

THIS STATEMENT SPECIFIES THE CHOICE OF RESIDENCE FOR THE 1108 EXECUTIVE SYSTEM. THE STATEMENT IS OF THE FORM:

```
  SYSTEM  RESIDENCE  C

    WHERE -C IS THE CHANNEL NUMBER.
    SEE   -SUPERVISOR
              SCHEDULING
                FACILITIES INVENTORY
    SEE   -FILE CONTROL SYSTEM
              FILE ORGANIZATION
                MASS STORAGE ALLOCATION
                  GENERAL
```

TIME ALLOCATION

THESE STATEMENTS DESCRIBE THE TIME VALUES USED IN THE CALCULATION OF THE QUANTUMS OF TIME GIVEN TO BATCH AND DEMAND PROGRAMS.

```
  BATCH ALLOCATION  A
  DEMAND ALLOCATION  B
  BATCH PRIORITY FACTOR  C
  DEMAND PRIORITY FACTOR  D
```

TIMES A AND B ARE GIVEN IN MILLISECONDS. IT IS THE AMOUNT OF TIME GIVEN TO A PROGRAM AT THE HIGHEST PRIORITY LEVEL.
C AND D ARE MULTIPLIERS OF THE RUN PRIORITY WHICH CONTROL THE EFFECT OF PRIORITY UPON TIME ALLOCATION. THE PRIORITY LETTERS YIELD THE VALUES: 26-A,25-B,... 1-Z. THE EXPRESSION USED TO FIND THE QUANTUM (Q) IS:

$$Q = A * (1 + P/F) * T$$

WHERE 'P' IS PRIORITY, 'F' THE PRIORITY FACTOR, AND 'T' THE LEVEL FACTOR.

18.1.4. INITIAL SYSTEM

THE FOLLOWING STATEMENTS ARE USED TO GENERATE THE INITIAL SYSTEM.

```
  AVERAGE  SIZE  OF  COBOL    PROGRAMS  500
  AVERAGE  SIZE  OF  FORTRAN  PROGRAMS  500
  CONTINUE   RUN  WHEN  PRINT  FILE  EXCEEDED
  CONTINUE   RUN  WHEN  PUNCH  FILE  EXCEEDED
```

```
CONTINUE  RUN  WHEN  TIME  LIMIT  EXCEEDED
CORE  SIZE  65 K
DEMAND  MAXIMUM  PERCENTAGE  70
DEMAND  MINIMUM  PERCENTAGE  20
LOG MAXIMUM  10
MAXIMUM  DELAY  TIME  300  SECONDS
MAXIMUM  FILE  SIZE  1000
MAXIMUM OPENED RUNS  5
NOISE CONSTANT 18
RETAIN  5  CYCLES
RETAIN  5  FCYCLES
STANDARD CARD LIMIT  500
STANDARD DEADLINES ALLOWED
STANDARD  DENSITY  800
STANDARD PAGE LIMIT  100
STANDARD RUN TIME  6  MINUTES
```

18.2. SYSTEM LOADING

AS A PART OF THE ORDERED RUN CONSTRUCTED BY THE SYSTEMS GENERATION ROUTINE, AN OUTPUT TAPE IS PRODUCED WHICH CONTAINS THE 1108 EXECUTIVE SYSTEM AND THE SYSTEMS LIBRARY. THIS TAPE IS INITIALLY LOADED BY THE MANUAL HARDWARE BOOTSTRAP METHOD. THE BOOTSTRAP CAN BE PERFORMED ON THE VIIIC, VIC OR IVC TAPE CHANNELS. FOUR 'INITIAL LOAD' TOGGLE SWITCHES ON THE CPU MAINTENANCE PANEL MUST BE SET TO DETERMINE THE INPUT CHANNEL.

THE BOOTSTRAP BLOCK IS LOADED AT THE FIRST LOCATION IN THE MEMORY MODULE SELECTED BY THE MEMORY SELECT REGISTER.

IMMEDIATELY AFTER THE BOOTSTRAP ROUTINE RECEIVES CONTROL THE AREAS AT 2000 OCTAL AND ABOVE WHERE THE EXEC IS TO BE STORED IN CORE IS COPIED TO MASS STORAGE. IF A PREVIOUS EXEC WAS STILL IN CORE, EXCEPT FOR THE FIRST 2000 OCTAL WORDS, ITS CONTENTS ARE SAVED FOR INFORMATION PURPOSES.

IF JUMP KEY TWO (2) IS SET, A SERIES OF SELECTED CORE AREAS MAY BE SAVED ON MASS STORAGE. THE PARAMETERS ARE ENTERED THROUGH THE PROGRAM ADDRESS COUNTER (P).

AFTER THE REQUESTED AREAS ARE SAVED, THE ROUTINE IN CORE READS THE TAPE TO BRING IN EACH BLOCK OF THE SYSTEM AND COPY EACH TO THE PROPER DRUM LOCATIONS. THIS COPY OF THE EXECUTIVE SYSTEM IS LOCATED AS SPECIFIED AT SYSTEM GENERATION TIME. A FIXED AREA ON THIS DRUM BEGINNING AT ADDRESS ZERO IS RESERVED FOR EXECUTIVE SYSTEM USAGE ONLY. THIS FIXED AREA IS DETERMINED AT SYSTEM GENERATION TIME, DEPENDING ON THE CONFIGURATION, I.E. NUMBER OF EQUIPMENT HANDLERS REQUIRED AND NUMBER OF PROCESSORS TO BE LOCATED ON DRUM.

THE USER HAS THE OPTION OF PLACING ANY OF THE PROCESSORS ON A SLOWER MASS STORAGE DEVICE IF IT EXISTS. AN INITIALIZATION ROUTINE FINALLY PLACES A COPY OF THE RESIDENT PORTION OF THE EXECUTIVE SYSTEM IN CORE, INITIALIZES AND/OR RE-INSTATES FILE DIRECTORIES AND GIVES THE SYSTEM CONTROL.

18.2.1. INITIAL LOAD CONSOLE COMMUNICATIONS

DURING INITIAL LOAD THERE EXISTS A MEANS OF TEMPORARILY MODIFYING THE SITE TABLE ASSIGNMENTS. IF INITIAL LOAD IS MADE VIA A TAPE CHANNEL AND JUMP KEY 5 IS SET, THE FOLLOWING TYPE-OUT WILL APPEAR :

```
D,P  MOD
```

THE CONSOLE WILL BE PLACED IN THE INPUT MODE TO ACCEPT DRUM AND/OR PRINTER CHANNEL

MODIFICATIONS. THE DRUM CHANNEL IS THE CHANNEL USED TO OPERATE AND STORE THE EXEC. THE PRINTER CHANNEL IS USED BY THE PANIC DUMP ROUTINES TO PRINT DATA. IF A PRINTER CHANNEL IS NOT SPECIFIED THE DUMP WILL BE STORED IN A STORAGE AREA. THESE CHANNELS MAY BE TEMPORARILY MODIFIED VIA THE FOLLOWING TYPE-INS:

```
DCC
PCC
DCC PCC   OR   PCC DCC
```

CC - DESIGNATES THE CHANNEL NUMBER (IN DECIMAL)
CR - DESIGNATES THE END OF MESSAGE.

IF AT ANY TIME A TYPE-IN ERROR OCCURS PRIOR TO TYPING THE 'CR' THE ENTIRE MESSAGE MAY BE IGNORED BY TYPING THE LOZENGE.

IF INITIAL LOAD IS MADE VIA A DRUM CHANNEL AND JUMP KEY 5 IS SET, THE FOLLOWING TYPE-OUT WILL OCCUR:

```
P,T  MOD
```

THE CONSOLE WILL BE PLACED IN THE INPUT MODE TO ACCEPT TEMPORARY MODIFICATIONS TO THE PRINTER AND/OR TAPE BOOTSTRAP CHANNEL. THE ROUTINE ACCEPTS DATA UNTIL A PERIOD IS ENCOUNTERED. THE ITEMS MUST BE SEPARATED BY A SPACE BUT THEY MAY APPEAR IN ANY ORDER AND ANY COMBINATION OF ONE OR ALL. THE TYPE-IN HAS THE FOLLOWING FORMAT:

```
TCC
PCC
TCC PCC
```

CC - DESIGNATES THE CHANNEL NUMBER (IN DECIMAL).

AS IN THE TAPE BOOTSTRAP CASE, IF A TYPE-IN ERROR OCCURS BEFORE THE CR IS TYPED THE MESSAGE MAY BE CANCELLED BY TYPING IN A LOZENGE.

ALSO ASSOCIATED WITH BOTH THE TAPE AND DRUM INITIAL LOAD ROUTINES ARE CERTAIN ERROR MESSAGES THAT MAY BE TYPED OUT DURING INITIAL LOAD. THESE ERROR MESSAGES MAY BE DIVIDED INTO TWO SEPARATE GROUPS. THE FIRST IS THE I/O GROUP. THIS GROUP CONSISTS OF ERROR MESSAGES THAT MAY BE TYPED OUT UPON THE OCCURRENCE OF AN ERROR CONDITION THAT ARISES IN CONJUNCTION WITH AN I/O OPERATION. IF THIS ERROR WAS IN THE FORM OF AN EXTERNAL INTERRUPT THE APPROPRIATE STATUS CODE WILL BE INSERTED INTO THE MESSAGE. THUS THERE EXISTS TWO FORMS OF AN I/O ERROR MESSAGE. THE FORMAT AND EXPLANATION OF THESE TWO TYPES IS AS FOLLOWS:

```
      ERR : CH CC SSSSSS DDDDDD AAAAAA
 AND
      ERR : CH CC DDDDDD AAAAAA
   CC - CHANNEL NUMBER(OCTAL) FROM WHICH THE ERROR AROSE
    S - STATUS CODE
    D - DIAGNOSTIC MESSAGE EXPLAINING TYPE OF INTERRUPT.
        THIS MAY BE ANY OF THE FOLLOWING:
            ISIIN  - ISI INPUT MONITOR
            ISIOUT - ISI OUTPUT MONITOR
            ISIFNC - ISI FUNCTION MONITOR
            ESIIN  - ESI INPUT MONITOR
            ESIOUT - ESI OUTPUT MONITOR
            ESICWP - ESI CONTROL WORD PARITY
            ISICWP - ISI CONTROL WORD PARITY
```

```
          DATAP  - ESI OR ISI DATA PARITY
          ESIEI  - ESI EXTERNAL INTERRUPT
          ISIEI  - ISI EXTERNAL INTERRUPT
  A - ADDRESS OF THE MAIN CODE THAT WAS INTERRUPTED.
```

THE OTHER GROUP OF ERROR MESSAGES MAY BE CALLED COMPUTER MALFUNCTION AND MISCELLANEOUS ERRORS. THIS GROUP TAKES INTO ACCOUNT ALL THE INTERRUPTS THAT ARE NOT ASSOCIATED WITH I/O. THE FORM AND EXPLANATION OF THIS TYPE OF MESSAGE IS AS FOLLOWS:

```
      ERR : DDDDDD AAAAAA
  D - DIAGNOSTIC MESSAGE EXPLAINING TYPE OF INTERRUPT.
      THIS MAY BE ANY OF THE FOLLOWING:
          POWLOS  - POWER LOSS
          MEM1P   - MEMORY  1 INSTRUCTION PARITY
          MEM2P   - MEMORY  2 INSTRUCTION PARITY
          MEM3P   - MEMORY  3 INSTRUCTION PARITY
          MEM4P   - MEMORY  4 INSTRUCTION PARITY
          ICRP    - ICR PARITY
          ILINS   - ILLEGAL INSTRUCTION
          EXRET   - EXECUTIVE RETURN
          GRDMOD  - GUARD MODE
          TSTSET  - TEST AND SET
          FPUNDF  - FLOATING POINT UNDERFLOW
          FPOVF   - FLOATING POINT OVERFLOW
          DIVFLT  - DIVIDE FAULT.
  A - ADDRESS OF THE MAIN CODE THAT WAS INTERRUPTED
```

18.3. SYSTEM GENERATION

THE SYSTEM GENERATION ROUTINE MAY BE USED TO AID IN THE GENERATION AND RE-GENERATION OF ANY SYSTEM (OR RUN DECK). THIS SECTION GIVES A DETAILED DESCRIPTION OF HOW THE SYSTEM GENERATION ROUTINE CAN BE APPLIED TO ANY SYSTEM OR RUN.

18.3.1. INTRODUCTION

THE OBJECT OF THE SYSTEM GENERATION ROUTINE IS TO SUPPLY THE SYSTEMS PROGRAMMER WITH A MEANS TO CONTROL THE CONTENTS OF ALL SYMBOLIC ELEMENTS OF A SYSTEM. THE PURPOSE BEHIND THIS IS TO ALLOW THE SYSTEMS PROGRAMMER TO UPDATE ALL SYMBOLIC ELEMENTS IN THE FIELD WITH THE SAME CORRECTION DECK. AT THE SAME TIME, THE SYSTEM USER IS GIVEN A CONVENIENT AND SIMPLE MEANS TO ALTER THE RELOCATABLE ELEMENTS OF THE SYSTEM AND GENERATE AN OPERABLE SYSTEM TO MEET THE REQUIREMENTS OF HIS CHANGING NEEDS AND HARDWARE CONFIGURATION.

THE SYSTEMS PROGRAMMER MUST PROVIDE THE SGR WITH A SKELETON RUN DECK WHICH CONTAINS A DESCRIPTION OF THE SYSTEM, ALL ALLOWABLE SYSTEM MODIFICATIONS, AND THE CORRECTIONS NECESSARY TO ACCOMPLISH THESE MODIFICATIONS. THE USER THEN, NEED ONLY SPECIFY, WITH SIMPLE STATEMENTS (SYSTEM GENERATION STATEMENTS, SGS'S) , THOSE MODIFICATIONS HE DESIRES AND THE SGR WILL CONSTRUCT AND EXECUTE A RUN TO INCLUDE THE DESIRED MODIFICATIONS AND GENERATE AN OPERABLE SYSTEM.

THE SYSTEMS PROGRAMMER AND THE USER MAY ALSO ENTER CORRECTIONS TO THE SYSTEM'S ELEMENTS THROUGH THE SGR AND THESE CORRECTIONS WILL BE INCLUDED IN THE RUN. CORRECTIONS TO AN ELEMENT ARE OF TWO KINDS - SYSTEMS PROGRAMMER CORRECTIONS WHICH ARE ENTERED IN THE SYMBOLIC ELEMENTS AND USER CORRECTIONS WHICH ONLY ENTER SYMBOLIC ELEMENTS WHEN AUTHORIZED BY THE SYSTEMS PROGRAMMER. THUS, THE SYSTEMS PROGRAMMER CAN PREVENT THE USER FROM ALTERING THE SYSTEM'S SYMBOLIC ELEMENTS AND YET ALLOW HIM TO ALTER THE SYSTEM'S RELOCATABLE ELEMENTS. THE SGR WILL ALSO REPRODUCE THE USER'S CORRECTION DECK WITH CORRECTED LINE NUMBERS WHICH HAVE RESULTED FROM SYSTEMS PROGRAMMER CORRECTIONS AND THE CORRECTIONS GENERATED FROM SGS'S.

18.3.2. CALLING SEQUENCE FOR THE SGR

A DESCRIPTION OF EACH OF THE DATA FILES AND OPTIONS WILL FOLLOW.

```
@XQT,OPTIONS  SGR
     SKELETON ELEMENT NAME
@EOF
-
-    SYSTEM GENERATION STATEMENTS
-
@EOF
-
-    USER'S CORRECTIONS
-
@EOF
-
-    SYSTEMS PROGRAMMER CORRECTIONS
-
@EOF
-
-    SKELETON CORRECTIONS
-
@END
```

THE SKELETON 'FILENAME' (12 CHARACTERS MAXIMUM) MUST HAVE BEEN PREVIOUSLY ASSIGNED. THE FILE INDICATED AS SKELETON CORRECTIONS WILL APPLY TO THIS ELEMENT. THE @EOF AND @END CARDS MUST BE PRESENT AS SHOWN, EVEN IF THE SECTION DOES NOT CONTAIN DATA.

18.3.3. SYSTEM GENERATION STATEMENTS

SYSTEM GENERATION STATEMENTS (SGS'S) ARE COMPRISED OF ANY NUMBER OF FIELDS WHICH MAY BE COMPRISED OF ANY NUMBER OF SUB-FIELDS. A SUB-FIELD IS A STRING OF 18 OR FEWER CHARACTERS NOT CONTAINING A SPACE, COMMA, RIGHT BRACKET, LEFT BRACKET, RIGHT PARENTHESIS, LEFT PARENTHESIS, PLUS SIGN, MINUS SIGN, OR EQUAL SIGN. SUB-FIELDS ARE SEPARATED BY A COMMA OR A COMMA FOLLOWED BY ONE OR MORE BLANKS. FIELDS ARE SEPARATED BY ONE OR MORE BLANKS. THE FIRST FIELD MUST HAVE ONLY ONE SUB-FIELD; THIS FIELD IS CALLED THE SGS LABEL. SGS'S HAVE THE FOLLOWING FORM.

```
LABEL  S11,S12,...,S1K  S21,S22,...,S2L  ...  SN1,SN2,...,SNM
```

A STATEMENT MAY BE CONTINUED ON THE FOLLOWING CARD BY PLACING A SEMI-COLON FOLLOWING THE LAST FIELD CODED ON THE LINE. NOTE THAT THE LABEL NEED NOT START IN COLUMN 1.

18.3.4. CORRECTIONS

THE SGS FILE MAY BE FOLLOWED BY A FILE OF USER CORRECTIONS (TEMPORARY CORRECTIONS) AND A FILE OF SYSTEMS PROGRAMMER CORRECTIONS (PERMANENT CORRECT IONS). THE CORRECTIONS ARE OF THE FORM DESCRIBED IN THE SECTION 'FORMAT OF CORRECTION LINES' (5.6.3). SPECIAL CARE MUST BE TAKEN WHEN CHANGING CORRECT ION IDENTIFICATION CHARACTERS. SINCE SKELETON, PERMANENT, AND USER CORRECTION WILL BE MERGED, THEY MUST HAVE THE SAME IDENTIFICATION CHARACTERS ACTIVE AT ANY GIVEN POINT. IN THE SKELETON FILE, THE FIRST CHARACTER OF A CORRECTION IDENTIFICATION MAY NOT BE A POUND SIGN (#) OR A PLUS SIGN (+); AND IN THE PERMANENT OR TEMPORARY FILE IT MAY NOT BE AN ASTERISK(*), OR A LEFT BRACKET ([).

ALL CORRECTIONS TO AN ELEMENT MUST BE GROUPED, ORDERED, AND PRECEDED BY CARD WITH AN ASTERISK IN COLUMN ONE, FOLLOWED BY THE NAME OF THE ELEMENT AND VERSION TO WHICH THE CORRECTIONS APPLY. THE ELEMENT NAME AND VERSION MUST BE SEPARATED BY A SLASH(/). PERMANENT AND TEMPORARY CORRECTIONS WILL BE INTERNALLY STORED AND MAY BE REFERENCED BY A 'CORRECT' DIRECTIVE.

18.3.5. SKELETON

THE SGR WILL PRODUCE THE RUN DECK FROM A SKELETON (FURNISHED BY THE SYSTEMS PROGRAMMER WITH THE SYSTEM'S SYMBOLIC ELEMENTS) BY REFERENCING THE SGS'S FOR INFORMATION. THE SKELETON IS AN ELEMENT OF CARDS TO BE EDITED AND INSERTED IN THE RUN DECK, AND CARDS TO DIRECT THE SGR. THE SKELETON IS PROCESSED SEQUENTIALLY AND THE CARDS ARE INTERPRETED, EDITED AND ADDED TO THE RUN DECK AS THEY ARE ENCOUNTERED. PERMANENT AND TEMPORARY CORRECTIONS ARE MERGED WITH THE RUN DECK AS THE SKELETON IS PROCESSED. CONFLICTS IN LINE NUMBERS AND CORRECTIONS ARE DETECTED AND NOTED.
THE SKELETON IS COMPRISED OF THREE TYPES OF CARDS.

1. CARDS TO DIRECT THE SGR.
2. CARDS TO BE EDITED AND ADDED TO THE RUN DECK.
3. CARDS TO ENTER THE RUN DECK DIRECTLY.

DIRECTIVE CARDS IN THE SKELETON ARE RECOGNIZED BY AN ASTERISK IN COLUMN 1 AND A SECOND FIELD CONTAINING ONE OF THE FOLLOWING:

CORRECT
END
IF
ELSE
INCREMENT
LOOP
DEFINE
PROCESS

18.3.5.1. SYSTEM GENERATION STATEMENT REFERENCES

DIRECTIVE CARDS AND CARDS TO BE EDITED REFERENCE INFORMATION ON SGS'S IN ANY OF THE FOUR WAYS GIVEN BELOW. L INDICATES A SGS LABEL; AND N, F, AND S REPRESENT INTEGERS.

1. [L] REPRESENTS THE NUMBER OF SGS'S WITH THE LABEL 'L'.
2. [L,N] REPRESENTS THE NUMBER OF FIELDS ON THE N'TH SGS WITH THE LABEL 'L'
3. [L,N,F] REPRESENTS THE NUMBER OF SUB-FIELDS IN THE F'TH FIELD ON THE N'TH SGS WITH THE LABEL 'L'.
4. [L,N,F,S] REPRESENTS THE CONTENTS OF THE S'TH SUB-FIELD IN THE F'TH FIELD ON THE N'TH SGS WITH THE LABEL 'L'.

ANY CARD NOT RECOGNIZED BY THE SGR IS SEARCHED FOR ANY OF THE ABOVE REFERENCES. IF NONE IS FOUND THE CARD IS ADDED TO THE RUN DECK. IF A REFERENCE IS FOUND, THE INFORMATION IS RETRIEVED FROM THE SGS AND SUBSTITUTED FOR THE REFERENCE. WHEN ALL REFERENCES ON THE SKELETON CARD HAVE BEEN RETRIEVED AND SUBSTITUTED, THE CARD IS ADDED TO THE RUN DECK. IF THE INFORMATION CANNOT BE FOUND ON ANY OF THE SGS'S, THE CARD IS NOT ADDED TO THE RUN DECK AND PROCESSING IS CONTINUED WITH THE NEXT SKELETON CARD.

18.3.5.2. CORRECT DIRECTIVE CARD

THE CORRECT DIRECTIVE CARD IS NECESSARY FOR THE INSERTION OF SYSTEM PROGRAMMER AND USER CORRECTIONS INTO THE RUN DECK. THE END DIRECTIVE CARD IS USED IN CONJUNCTION WITH THE CORRECT CARD AS SHOWN BELOW.

```
* CORRECT NAME VERSION
-
-
-
* END
```

ALL CORRECTIONS TO THE ELEMENT NAME/VERSION WILL BE INSERTED AT THE PROPER POINT IN THE RUN DECK BETWEEN THE DIRECTIVE CARDS. USER, SYSTEM PROGRAMMER, AND SKELETON CORRECTION CARDS WILL BE MERGED AND CHECKED FOR CONFLICTING CORRECTIONS. OTHER FORMS OF THE CORRECT DIRECTIVE CARD ARE:

```
* CORRECT  NAME VERSION PERMANENTLY
* CORRECT  NAME         PERMANENTLY
```

THESE CARDS ARE USED IN CONJUNCTION WITH THE END DIRECTIVE AS ABOVE, BUT ONLY SYSTEMS PROGRAMMER CORRECTIONS ARE ADDED TO THE RUN DECK.

THE NAME AND VERSION ON A CORRECT DIRECTIVE MUST BE SEPARATED BY AT LEAST ONE SPACE. WHEN A CORRECT DIRECTIVE IS ENCOUNTERED, SGR IS CONDITIONED TO MERGE THE CORRESPONDING TEMPORARY AND/OR PERMANENT CORRECTIONS INTO THE RUN STREAM. A THREE-WAY MERGE WILL TAKE PLACE WHEN A CORRECTION APPEARS IN THE SKELETON FILE. THE *END DIRECTIVE ASSOCIATED WITH THE *CORRECT DIRECTIVE WILL CAUSE ALL TEMPORARY AND/OR PERMANENT CORRECTIONS WHICH ARE NOT ALREADY ACCESSED TO BE MERGED AND INSERTED IN THE RUN STREAM. CONFLICTS IN LINE NUMBERS WILL BE NOTED AND INSERTED OR BYPASSED ACCORDING TO THE FOLLOWING RULE: TEMPORARY CORRECTIONS WILL BE GIVEN THE HIGHEST ORDER OF PRIORITY, THEN PERMANENT CORRECTIONS, AND FINALLY SKELETON CORRECTIONS.

ANY TEMPORARY CORRECTIONS APPLIED TO A SOURCE ELEMENT BY THE SKELETON MUST OCCUR PRIOR TO ANY 'PERMANENT ONLY' CORRECTIONS. PERMANENT FILES ARE DELETED FROM STORAGE AFTER BEING ACCESSED AS 'PERMANENT ONLY'.

18.3.5.3. IF DIRECTIVE CARD

THE IF DIRECTIVE CARD IS USED TO CONDITIONALLY SKIP A SEQUENCE OF SKELETON CARDS. THE END DIRECTIVE CARD IS ALSO USED IN CONJUNCTION WITH THE IF CARD AS SHOWN BELOW. THE IF DIRECTIVE CARD CAN BE USED IN ONE OF TWO WAYS. THE FIRST IS:

```
* IF  <BOOLEAN EXPRESSION>   THEN
-
-
-
* END
```

THE CARD OR CARDS BETWEEN THESE DIRECTIVE CARDS WILL BE SKIPPED IF THE BOOLEAN EXPRESSION IS FALSE. THE INTERVENING CARDS WILL BE PROCESSED IF THE EXPRESSION IS TRUE.

THE SECOND FORM IS:

```
* IF  <BOOLEAN EXPRESSION>   THEN
-
-
-
* ELSE
-
-
-
* END
```

IF THE BOOLEAN EXPRESSION IS TRUE THE CARDS BETWEEN THE IF AND THE ELSE DIRECTIVE CARDS WILL BE PROCESSED; AND THE CARDS BETWEEN THE ELSE AND THE END DIRECTIVE CARDS WILL BE SKIPPED. IF THE BOOLEAN EXPRESSION IS FALSE THE CARDS BETWEEN THE IF AND ELSE DIRECTIVE CARDS WILL BE SKIPPED; AND THE CARDS BETWEEN THE ELSE AND THE END DIRECTIVE CARDS WILL BE PROCESSED.

THE OPERANDS ALLOWED IN THE BOOLEAN EXPRESSION ARE OF FOUR FORMS.

1. <SGS REFERENCE> = <SGS SUB-FIELD>
2. <SGS REFERENCE> = <NUMBER>
3. <SGS REFERENCE> = <SGS REFERENCE>

4. <SGS REFERENCE>

IF, IN ANY CASE, THE REFERENCE CANNOT BE FOUND, THE OPERAND IS FALSE. IF THE REFERENCE IS FOUND ON THE SGS'S BUT DOES NOT EQUAL THE CORRESPONDING SUB-FIELD, NUMBER, OR SGS REFERENCE THE OPERAND IS FALSE. THE OPERAND IS TRUE WHEN THE REFERENCE IS FOUND AND IS EQUAL TO THE SUB-FIELD, NUMBER, OR SGS REFERENCE. TWO OPERATORS ARE ALLOWED: OR AND NOT. A COMPLETE SYNTACTICAL DESCRIPTION FOLLOWS.

WHEN <NUMBER> IS USED AS THE SECOND OPERAND IN AN EQUIVALENCE; IT WILL BE DIRECTLY COMPARED IF THE FIRST OPERAND WAS AN SGS SUBFIELD REFERENCE; IT WILL BE CONVERTED IF THE FIRST OPERAND WAS AN SGS INTEGER REFERENCE. <NUMBER MUST ALWAYS BE IN DECIMAL FORM.

18.3.5.4. INCREMENT DIRECTIVE CARD

THE INCREMENT AND LOOP DIRECTIVE CARDS ARE USED, AS SHOWN BELOW, TO REPEATEDLY PROCESS A SERIES OF CARDS.

```
* INCREMENT <ALPHABETIC> TO <FINAL VALUE>
-
-
-
* LOOP
```

AN INDEX IS SPECIFIED ON THE INCREMENT DIRECTIVE CARD, WHICH IS INCREMENTED EACH TIME THE INTERVENING CARDS ARE PROCESSED. THE INDEX WILL START WITH THE VALUE ONE AND BE INCREMENTED BY ONE UNTIL IT IS GREATER THAN THE FINAL VALUE. THE INDEX, SPECIFIED BY ANY SINGLE ALPHABETIC, MAY BE USED IN ANY OF THE INTEGER FIELDS IN A SGS REFERENCE.

THE FINAL VALUE MAY BE GIVEN BY A NUMBER, A SGS REFERENCE, OR AN INDEX. A COMPLETE SYNTACTICAL DESCRIPTION FOLLOWS.

18.3.5.5. DEFINE PROCESS DIRECTIVE CARD

THE PURPOSE OF THE DEFINE AND PROCESS DIRECTIVE CARDS IS TO ADD TO THE LANGUAGE OF SKELETON DIRECTIVE CARDS THE ABILITY TO CALL A SUBROUTINE OF COMMON SKELETON DIRECTIVE CARDS. THE OBJECT IS TO ALLOW THE SYSTEMS PROGRAMMER TO CONSTRUCT THE SKELETON WITH A MINIMUM OF CARDS AND EFFORT. A PROCESS MUST BE DEFINED, AS SHOWN BELOW, BEFORE IT IS CALLED.

```
* DEFINE PROCESS NAME
-
-
-
* END
```

THE INTERVENING CARDS WILL BE ASSOCIATED WITH THE GIVEN NAME. THE CARDS WILL NOT BE INTERPRETED WHEN THE PROCESS IS DEFINED.

18.3.5.6. PROCESS DIRECTIVE CARD

A PROCESS IS CALLED WITH THE FOLLOWING DIRECTIVE CARD.

```
* PROCESS NAME  P1 P2 ... PN
```

WHEN A PROCESS DIRECTIVE CARD IS ENCOUNTERED, THE PARAMETERS P1 THROUGH PN ARE PASSED TO THE DEFINED PROCESS I.E., PM IS SUBSTITUTED FOR EACH [#M] ON THE CARDS DEFINING THE PROCESS 'NAME'. PROCESSES MAY BE CALLED RECURSIVELY.

18.3.5.7. SYNTACTICAL DESCRIPTION

THE PROPER FORMS OF THE IF DIRECTIVE CARD AND THE INCREMENT DIRECTIVE CARD ARE GIVEN BELOW IN BACKUS NORMAL FORM:

```
<NUMBER>                   := 0/1/2/.../99/100
<LABEL>                    := SGS LABEL FIELD
<SUB-FIELD>                := SGS SUB-FIELD
<ALPHABETIC>               := A/B/C/.../Y/Z
<SIMPLE EXPRESSION>        := <NUMBER>/<ALPHABETIC>
<INTEGER EXPRESSION>       := <SIMPLE EXPRESSION>/
                              <SIMPLE EXPRESSION> +
                              <INTEGER EXPRESSION>/
                              <SIMPLE EXPRESSION> -
                              <INTEGER EXPRESSION>
<SGS INTEGER REFERENCE>:= [<LABEL>]/
                              [<LABEL>,<INTEGER EXPRESSION>]/
                              [<LABEL>,<INTEGER EXPRESSION>,
                              <INTEGER EXPRESSION>]
<SGS REFERENCE>            := <SGS INTEGER REFERENCE>/
                              [<LABEL>,<INTEGER EXPRESSION>,
                              <INTEGER EXPRESSION>,
                              <INTEGER EXPRESSION>]
<BOOLEAN OPERAND>          := <SGS REFERENCE>=<SUB-FIELD>/
                              <SGS REFERENCE>=<NUMBER>/
                              <SGS REFERENCE>=<SGS REFERENCE>/
                              <SGS REFERENCE>
<BOOLEAN EXPRESSION>       := <BOOLEAN OPERAND>/
                              <BOOLEAN OPERAND> OR
                              <BOOLEAN EXPRESSION>/
                              NOT <BOOLEAN OPERAND>
<FINAL VALUE>              := <NUMBER>/<ALPHABETIC>/
                              <SGS INTEGER REFERENCE>
```

THE PROPER FORMS ARE THUS,

```
    * IF <BOOLEAN EXPRESSION> THEN
```

AND

```
    * INCREMENT <ALPHABETIC> TO <FINAL VALUE>
```

18.3.5.8 NESTING

NESTING OF DIRECTIVES IN SGR IS LIMITED ONLY BY SIZE OF CORE, AND SGR WILL REQUEST CORE DYNAMICALLY AFTER THE INITIAL AREA IS USED. BECAUSE OF THE NATURE OF THE NEST CAPABILITY, RULES FOR *END DIRECTIVES MUST BE ADHERED TO STRICTLY. THE RESULTS WHEN THESE RULES ARE BROKEN ARE INDETERMINATE.

18.3.6. SKELETON CORRECTIONS

THE SKELETON IS AN ELEMENT COMPRISED OF DATA ITEMS, CORRECTION ITEMS, AN CONTROL ITEMS. CONTROL ITEMS WILL BEGIN WITH A POUND SIGN (#) INSTEAD OF THE CONVENTIONAL MASTER SPACE (@). CORRECTION ITEMS FOR UPDATING THE SKELETON MUST CONTAIN A PLUS (+) SIGN IN COLUMN ONE.

IT SHOULD BE NOTED THAT CORRECTIONS AND ADDITIONS IN THE SKELETON SHOULD CANCEL EACH OTHER OUT. THIS IS NECESSARY SINCE THE SKELETON ELEMENT IS USED FOR EACH GENERATION, AND ADDITIONS OR DELETIONS WOULD THEREFORE BE CUMULATIVE

18.3.7. OPTIONS

THE FOLLOWING OPTIONS WILL BE RECOGNIZED BY SGR;

A - FORM OUTPUT ELEMENT RUN STREAM EVEN IN THE CASE OF ERRORS.
B - INHIBIT EXECUTION OF THE RUN STREAM.
C - SINGLE SPACE ALL PRINTING.
D - INHIBIT PUNCHING OF REVISED TEMPORARY (USER) CORRECTIONS.
E - PRINT INPUT SKELETON.
F - PRINT CORRECTION TO THE SKELETON.
G - PRINT REVISED SKELETON.
H - PRINT PERMANENT (SYSTEM PROGRAMMER) CORRECTIONS.

```
I - PRINT TEMPORARY (USER) CORRECTIONS.
J - PRINT REVISED TEMPORARY (USER) CORRECTIONS.
K - PRINT SGS'S.
L - PUNCH THE RUN STREAM.
M - PRINT THE RUN STREAM.  18.3.8  EXAMPLE
A. SKELETON ELEMENT 'SKELNAME'
#RUN
*DEFINE  ASM
*CORRECT    [#1]
#ASM  [#1],[#2]
-55,56
*INCREMENT  A TO 2
*IF  [DEMAND,A,1,1]=MAXIMUM
       L      A0,[DEMAND,A,3,1],,U
*END
*IF    [DEMAND,A,1,1]=MINIMUM
         L  A1,[DEMAND,A,3,1],,U
*END
*LOOP
*END
*END
#ASG,X     SYSTEM
*PROCESS     ASM   SOURCE   OBJECT
#END
B.  CALLING SEQUENCE
@XQT,A,L      SGR
SKELETON    SKELNAME
@EOF
DEMAND MAXIMUM PERCENTAGE 70
DEMAND MINIMUM PERCENTAGE  20
@EOF
*SOURCE
-8,9
              J    START
@EOF
-5,6
@EOF
@END
C.  OUTPUT RUN STREAM
@RUN
@ASG,X   SYSTEM
@ASM    SOURCE,OBJECT
-5,6
-8,9
              J    START
-55,56
              L    A0,70,,U
              L    A1,20,,U
@END
D.  UPDATED USER CORRECTIONS
-6,7
    J     START
```

19. UTILITY ROUTINES

19.1. CONVERSION AIDS

TO AID INSTALLATIONS IN THE TRANSITION FROM OTHER COMPUTERS TO THE UNIVAC 1108, A SET OF CONVERSION AID ROUTINES ARE INCORPORATED INTO THE 1108 EXECUTIVE SYSTEM. THOSE CONVERSION ROUTINES DESIRED BY AN INSTALLATION WILL BE INCLUDED IN THE SYSTEMS LIBRARY, AND DEFINED AS SYSTEMS PROCESSORS AT SYSTEM SET UP TIME. THE ROUTINES MAY BE CALLED AS DESIRED BY USE OF THE EXECUTIVE PROCESSOR CALL STATEMENT (CHAPTER 5.6)

FOR THE INITIAL 1108 SOFTWARE SYSTEM, THE FOLLOWING CONVERSION AID ROUTINES WILL BE PROVIDED:

19.1.1. UNIVAC 1107 TO UNIVAC 1108

THIS PROCESSOR WILL CONVERT MAGNETIC TAPES CREATED BY THE 1107 COMPLEX UTILITY ROUTINE (CUR) TO MAGNETIC TAPES ACCEPTABLE AS INPUT TO AN 1108 PROGRAM-FILE. THE PROCESSOR WILL ACCEPT 1107 SYMBOLIC ELEMENTS, COBOL LIBRARY ELEMENTS, AND PROCEDURE ELEMENTS AND CONVERT THEM TO 1108 SYMBOLIC ELEMENTS, COBOL PROCEDURE ELEMENTS, AND ASSEMBLER PROCEDURE ELEMENTS RESPECTIVELY. ALL OTHER 1107 ELEMENT TYPES WILL BE IGNORED.

THE PROCESSOR RESIDES IN THE SYSTEMS LIBRARY AND IS INITIATED BY THE FOLLOWING PROCESSOR CALL COMMAND:

```
@CON78,OPTIONS  FILE1.,FILE2.
```

THE AVAILABLE OPTIONS WHICH INDICATE ELEMENT TYPE ARE AS FOLLOWS:

S	SYMBOLIC ELEMENTS
C	COBOL LIBRARY ELEMENTS
P	PROCEDURE ELEMENTS

IF NONE OF THE OPTIONS ARE SPECIFIED, ALL ELEMENTS OF TYPE 'S', 'C', AND 'P' ARE CONVERTED.

'FILE1.' IS THE INPUT TAPE FILE OF 1107 ELEMENTS AND 'FILE2.' IS THE OUTPUT TAPE FILE OF 1108 ELEMENTS . BOTH 'FILE1.' AND 'FILE2' MUST HAVE BEEN PREVIOUSLY ASSIGNED TO THE RUN WITH THE 'ASG' CONTROL COMMAND. ONE CALL TO @CON78 CONVERTS ONE FILE, TWO CALLS CONVERTS TWO FILES, ETC. IT SHOULD BE NOTED THAT A PERIOD IS REQUIRED TO DEFINE THE FILE NAME. THE FILES ARE NEVER REWOUND AND END OF FILE MARKS ARE WRITTEN AFTER EACH FILE CONVERTED.

COBOL PROCEDURES ARE HANDLED SOMEWHAT DISTINCTLY. EACH PROCEDURE IS SURROUNDED BY A PROC LINE AND AN END LINE.THE LABEL ON THE PROC LINE IS THAT OF THE ELEMENT NAME AND IT IS NOT EXTERNALLY DEFINED.THEREFORE , IN ORDER TO USE THE PROCEDURE IN A COMPILATION IT WILL BE NECESSARY TO PDP IT.

19.2. OUTPUT EDITING ROUTINE

19.2.1. ORGANIZATION

THE OUTPUT EDITING ROUTINE IS AN INTERPRETIVE ROUTINE WHICH WILL PERFORM EDITING FUNCTIONS FOR OUTPUT PRODUCED ON THE LINE PRINTER, THE CARD PUNCH, AND THE CONSOLE. THE INTERPRETIVE INSTRUCTIONS PERFORMED BY THE ROUTINE ARE CONSTRUCTED ALONG MUCH THE SAME LINES AS ARE MACHINE LANGUAGE INSTRUCTIONS:

```
--------------------------------------------
: F  :   T   :  D   : X :        M         :
--------------------------------------------
  5      7      6     2          16
```

(F) -- FUNCTION CODE
(T) -- TYPE WHEEL, ETC.
(D) -- DECIMAL POINT LOCATION, ETC.
(X) -- SPECIFIES INDIRECT ADDRESS AND USE OF THE SIMULATED INDEX REGISTER
(M) -- ADDRESS (MEMORY LOCATION OF DATA, ETC.)

THE AVAILABLE FUNCTIONS ARE LISTED BELOW WITH THEIR FUNCTION CODES IN OCTAL AND THE NAME OF THE CORRESPONDING PROCEDURE CELL.

EDITING FUNCTIONS

E$D	01--DECIMAL
E$O	02--OCTAL
E$B	03--BINARY
E$C	04--ALPHANUMERIC CHARACTERS
E$A	05--ALPHANUMERIC WORDS
E$E	06--FLOATING-POINT (FORTRAN E)
E$F	07--FLOATING TO FIXED (FORTRAN F)
E$DE	26--DOUBLE PRECISION FLOATING POINT
E$DF	27--DOUBLE PRECISION FLOATING TO FIXED

OUTPUT FUNCTIONS

E$WT	10--WRITE AND TERMINATE
E$W	11--WRITE
E$WS	12--WRITE AND SAVE

MODAL FUNCTIONS

E$SCL	13--SET SCALE
E$PNT	14--SET POINT
E$FLD	15--SET FIELD
E$INDX	16--SET INDEX
E$OVRP	17--OVERPUNCH

CONTROL FUNCTIONS

E$TERM	20--TERMINATE
E$LINK	21--LINK
E$JUMP	22--JUMP
E$RPT	23--REPEAT
E$CLR	24--CLEAR

THE ROUTINE IS CALLED BY THE FOLLOWING INSTRUCTION:

```
               LMJ    11, EOUT$
```

THERE ARE TWO ENTRY POINTS TO THIS SUBROUTINE. THE NORMAL ENTRY POINT IS EOUT$. THE OTHER, EOUTR$, IS THE POINT FOR RE-ENTRY AFTER E$TERM (TERMINATE) FUNCTION AND IS DISCUSSED UNDER TERMINATE.

THE ADDRESSED WORD IN THE M DESIGNATOR MAY BE EITHER IN CONTROL OR CORE MEMORY. ANY WORD, EVEN A VOLATILE REGISTER, IS PERMISSIBLE; BUT IF REGISTER 11 IS ADDRESSED, THE LOCATION OF THE INTERPRETIVE WORD WHICH REFERENCES 11 WILL BE PUT OUT. ALL REGISTERS, INCLUDING VOLATILE ONES, ARE SAVED AND RESTORED. THE X DESIGNATOR IS USED TO SPECIFY INDIRECT ADDRESSING AND THE USE OF THE SINGLE SIMULATED INDEX REGISTER. ITS PERMISSIBLE VALUES ARE

0	NO ACTION
1	USE ADDRESS INDIRECTLY
2	APPLY SIMULATED INDEX REGISTER
3	APPLY SIMULATED INDEX REGISTER THEN USE ADDRESS INDIRECTLY

INDIRECT ADDRESSING IS PERMITTED TO ONE LEVEL ONLY AND THE B, H, AND I DESIGNATORS OF THE INDIRECTLY ADDRESSED WORD ARE IGNORED. HOWEVER, IT IS POSSIBLE TO INDIRECTLY ADDRESS CONTROL MEMORY. ALL MODES MAY BE USED WITH INDIRECT ADDRESSING.

THE VARIOUS FUNCTIONS ARE DESCRIBED IN DETAIL BELOW. THEY ARE ALL CALLABLE AS PROCEDURES. EACH OF THE PROCEDURE CALLS WILL GENERATE ONE WORD IN THE PROPER FORMAT. THE PARAMETERS OF THESE PROCEDURES ARE INTERPRETED DIFFERENTLY DEPENDING ON THE NUMBER WRITTEN. A SINGLE PARAMETER IS TAKEN AS M; TWO PARAMETERS AS M AND X; THREE PARAMETERS AS T, D, AND M; AND FOUR AS T, D, M, AND X. ANY MISSING PARAMETERS WILL BE ASSUMED TO BE ZERO.

ENTRY TO THE GENERALIZED EDITOR MAY BE OBTAINED BY THE PROCEDURES E$OUT OR E$OUTR, DEPENDING ON THE ENTRY POINT DESIRED. NO PARAMETERS ARE REQUIRED.

19.2.2. THE EDITING FUNCTIONS

THESE FUNCTIONS ACTUALLY CONVERT THE INFORMATION TO BE PUT OUT. IN ALL CASES EXCEPT E$A, ALPHANUMERIC WORDS (SEE E$A BELOW), THE T FIELD SPECIFIES THE TYPE WHEEL AT WHICH THE RIGHT-MOST DIGIT, BIT, OR CHARACTER IS TO BE PRINTED.

E$D--DECIMAL: THE ADDRESSED WORD IS TREATED AS IF IT WERE A SIGNED DECIMAL INTEGER, AND IS EDITED WITHOUT A DECIMAL POINT UNLESS A SET POINT (14) IS IN EFFECT. LEADING ZEROS TO THE LEFT ARE SUPPRESSED AND A MINUS SIGN, IF ANY, IS PRINTED IMMEDIATELY TO THE LEFT OF THE NUMBER (ALSO SEE OVERPUNCH (17)). IF THE VALUE IS ZERO, A SINGLE ZERO WILL BE PRINTED. IF A SET POINT IS IN EFFECT, THE DECIMAL NUMBER IS ASSUMED TO HAVE THE STATED POINT SPECIFIED BY THE SET POINT AND THE D FIELD SPECIFIES THE NUMBER OF DECIMAL DIGITS TO BE PRINTED TO THE RIGHT OF THE DECIMAL POINT. IF A SET FIELD (15) WITH D=0 IS IN EFFECT, THE SPECIFIED FIELD IS TREATED AS AN UNSIGNED DECIMAL INTEGER.

E$O--OCTAL: THE D LOW-ORDER BITS OF THE ADDRESSED WORD ARE EDITED AND PRINTED AS (D+2)/3 OCTAL DIGITS, UNSIGNED. FOR A FULL OCTAL, BINARY, OR ALPHANUMERIC CHARACTER WORD, D MUST ALWAYS BE GIVEN AS 36.

E$B--BINARY: THE D LOW-ORDER BITS OF THE ADDRESSED WORD ARE EDITED AS D BINARY DIGITS UNSIGNED.

E$C--ALPHANUMERIC CHARACTERS: THE D LOW-ORDER BITS OF THE ADDRESSED WORD ARE EDITED AND PRINTED AS (D+5)/6 ALPHANUMERIC CHARACTERS IN FIELD DATA CODE.

E$A--ALPHANUMERIC WORDS: THE D WORDS BEGINNING WITH THE ADDRESSED WORD ARE EDITED AS 6*D CHARACTERS IN FIELD DATA CODE. FOR THIS EDITING FUNCTION ONLY, THE T FIELD SPECIFIES THE PRINT POSITION AT WHICH THE LEFT-MOST CHARACTER IS PRINTED.

E$E--FLOATING-POINT (FORTRAN E): THE ADDRESSED WORD IS EDITED AS A FLOATING-POINT NUMBER WITH D SIGNIFICANT DIGITS. NORMALLY THESE WILL ALL PRINT TO THE RIGHT OF THE DECIMAL POINT (ALSO SEE SET SCALE). A DECIMAL EXPONENT CONSISTING OF A SIGN AND TWO DIGITS WILL BE INSERTED IMMEDIATELY TO THE RIGHT OF THE SIGNIFICANT PORTION. IF THE FLOATING-POINT NUMBER IS NEGATIVE, A MINUS SIGN WILL BE INSERTED IMMEDIATELY TO THE LEFT OF THE NUMBER (ALSO SEE OVERPUNCH). IF THE ADDRESSED WORD IS MINUS ZERO, NO EFFECT WILL OCCUR, AND THE FIELD WILL BE LEFT BLANK.

E$F--FLOATING TO FIXED (FORTRAN F): THE ADDRESSED WORD IS ASSUMED TO BE A FLOATING-POINT NUMBER AND IS EDITED TO FIXED-POINT WITH D PLACES FOLLOWING THE DECIMAL POINT. NEGATIVE NUMBERS, INCLUDING MINUS ZERO, ARE TREATED AS ABOVE.

E$DE--DOUBLE PRECISION FLOATING POINT: THIS EDITING FUNCTION IS THE SAME AS THE FLOATING POINT (06) FUNCTION WITH THE ADDRESSED WORD AND THE ADDRESSED WORD PLUS ONE EDITED AS A DOUBLE PRECISION FLOATING POINT NUMBER. A DECIMAL EXPONENT CONSISTING OF A SIGN AND THREE DIGITS WILL BE INSERTED IMMEDIATELY TO THE RIGHT OF THE SIGNIFICANT PORTION.

E$DF--DOUBLE PRECISION FLOATING TO FIXED: THIS EDITING FUNCTION IS THE SAME AS THE FLOATING TO FIXED (07) FUNCTION WITH THE ADDRESSED WORD AND ADDRESSED WORD PLUS ONE EDITED AS A DOUBLE PRECISION FLOATING POINT NUMBER.

19.2.2.1. EXAMPLE OF EDITING FUNCTIONS

THE EDITING FUNCTIONS PERFORM THE GENERATION OF A FORM WORD WHICH WAS DISCUSSED EARLIER. IT IS NECESSARY TO CALL EOUT$ PRIOR TO THE FUNCTIONS THEMSELVES. AFTER CALLING EOUT$ THE EDITORS ARE INVOKED TO INDICATE THE TYPE OF TRANSLATION.

SUPPOSE IT IS DESIRED TO EDIT TWO WORDS INTO DIFFERENT FORMATS AND STORE THEM AS PART OF A LINE TO BE OUTPUT. ONE METHOD IS SHOWN BELOW:

```
        E$OUT
        E$A     1,1,M  . EDIT ALPHA WORD AT M TO POSITION 1
        E$D     18,1,M1  . EDIT DECIMAL VALUE AT M1 TO POSITION 18

                        .(M)=060710111213 & (M1)=0606061626364
```

UPON RETURN TO THE NEXT INSTRUCTION THE FIRST THREE WORDS OF THE PRINT BUFFER ARE:

```
ABCDEF        1234
```

19.2.3. THE OUTPUT FUNCTIONS

THE OUTPUT FUNCTIONS SERVE TO TRANSMIT THE EDITED LINE TO AN OUTPUT DEVICE; THE PRINTER, THE CARD PUNCH, OR THE CONSOLE. THE DEVICE TO BE USED IS DETERMINED BY THE D FIELD:

```
        PRINTER    D=0
        CARD PUNCH D=1
        CONSOLE    D=2
```

THE WORD OR CHARACTER COUNT IS GIVEN IN THE T FIELD. THIS COUNT MUST BE GIVEN (IT IS NOT ASSUMED MAXIMUM IF IT IS GIVEN AS ZERO). FOR PRINTER, THE WORD COUNT IS NORMALLY 22; FOR CARD PUNCH, NORMALLY 14. FOR THE CONSOLE T IS A CHARACTER COUNT. FOR THE PRINTER, THE M DESIGNATOR SERVES TO SPECIFY THE NUMBER OF LINES TO BE SPACED. A VALUE GREATER THAN THE LENGTH OF A LOGICAL PAGE WILL RESULT IN PRINTING ON THE FIRST LINE OF THE NEXT PAGE. FOR THE PUNCH AND CONSOLE THE M DESIGNATOR IS IGNORED.

E$WT--WRITE AND TERMINATE: THE EDITED IMAGE IS TRANSMITTED TO THE SPECIFIED DEVICE AND THE ROUTINE RETURNS TO THE NEXT INSTRUCTION IN MACHINE LANGUAGE MODE. THE IMAGE IS RESET TO BLANKS.

E$WS--WRITE AND SAVE: THE EDITED IMAGE IS TRANSMITTED TO THE SPECIFIED DEVICE AND THE ROUTINE CONTINUES TO THE NEXT INSTRUCTION IN THE INTERPRETIVE MODE. THE IMAGE IS LEFT AVAILABLE FOR USE BY FURTHER OUTPUT FUNCTIONS OR FURTHER EDITING.

19.2.4. THE MODAL FUNCTIONS

THE MODAL FUNCTIONS SERVE TO ENTER INFORMATION WHICH AFFECTS THE INTERPRETATION OF ONE OR MORE OF THE INSTRUCTIONS WHICH FOLLOW. FIVE MODAL INSTRUCTIONS ARE AVAILABLE:

E$SCL--SET SCALE: THE CONTENTS OF THE ADDRESS FIELD ARE TREATED AS A SIGNED POWER OF 10 TO BE APPLIED TO ANY FLOATING-POINT OR FLOATING TO FIXED FUNCTION WHICH FOLLOWS THE SET SCALE. FOR FLOATING-POINT, THE SCALE IS THE NUMBER OF DIGITS TO BE PRINTED TO THE LEFT OF THE DECIMAL POINT. THE EXPONENT FIELD IS REDUCED ACCORDINGLY, SO THAT THE RESULTING VALUE IS THE SAME AS IF NO SET SCALE MODAL WERE IN EFFECT. NEGATIVE VALUES OF THE ADDRESS (THE 16-BIT ONES' COMPLEMENT) WILL INTRODUCE LEADING ZEROS AFTER THE DECIMAL POINT AND INCREASE THE EXPONENT FIELD ACCORDINGLY.

FOR FLOATING TO FIXED CONVERSION, THE ACTUAL VALUE OF THE RESULTING NUMBER IS ALTERED BY MULTIPLYING IT BY THE POWER OF 10 INDICATED BY THE ADDRESS. THE SET SCALE MODAL REMAINS IN EFFECT UNTIL IT IS COUNTERMANDED BY A NEW SET SCALE. UPON INITIAL ENTRY TO EOUT$, THE SCALE IS ASSUMED TO BE 0.

E$PNT--SET POINT: THE SET POINT MODAL SPECIFIES THE POSITION OF THE BINARY POINT FOR THE NEXT EDITING FUNCTION TO BE ENCOUNTERED (PRESUMABLY A DECIMAL EDITING FUNCTION). IT REMAINS IN EFFECT ONLY FOR THE SINGLE EDIT. THE ADDRESS OF THE SET

POINT GIVES THE NUMBER OF BITS FOLLOWING THE BINARY POINT. NEGATIVE VALUES ARE PERMITTED (SEE SET FIELD).

E$FLD--SET FIELD: THE SET FIELD MODAL IS USED TO SPECIFY A SUBFIELD OF THE NEXT WORD TO OCCUR (PRESUMABLY A DECIMAL, OCTAL, BINARY, OR ALPHANUMERIC CHARACTERS FUNCTION). THE T FIELD SPECIFIES THE LEFT-HAND MARGIN AND THE M FIELD THE RIGHT-HAND MARGIN. THE BITS OF THE 1108 WORD ARE NUMBERED, FOR THE PURPOSES OF THIS FUNCTION, FROM LEFT (00) TO RIGHT (35). THE D FIELD SPECIFIES EXTENSION OF SIGN; IF IT IS NON-ZERO, THE FIELD IS TREATED AS SIGNED. A SET FIELD WITH D = 0 AND T = 0 MAY BE USED TO TREAT FIELDS, INCLUDING THE SIGN BIT, AS UNSIGNED, UNLESS M = 35 (I.E., A WHOLE WORD MUST ALWAYS BE SIGNED IN THE EVENT A SIGN IS APPLIED).

THE SET FIELD MODAL REMAINS IN EFFECT ONLY FOR THE NEXT FUNCTION ENCOUNTERED. IF BOTH A SET FIELD MODAL AND A SET POINT MODAL ARE IN EFFECT WHEN EDITING OCCURS, THE SET FIELD MODAL WILL BE APPLIED FIRST. IN THIS CASE, THE SET POINT SPECIFIES THE BINARY POINT COUNTING FROM THE RIGHT-HAND END OF THE SPECIFIED FIELD.

E$INDX--SET INDEX: THE SET INDEX IS USED TO ADDRESS A QUANTITY IN MEMORY WHICH IS TO BE LOADED INTO THE SINGLE SIMULATED INDEX REGISTER. FOR ANY FUNCTION WHICH ADDRESSES MEMORY (INCLUDING THIS ONE), THE PRESENCE OF A 1 BIT IN THE INCREMENT (H) PORTION OF THE ADDRESS WILL CAUSE THE SIMULATED INDEX TO BE ADDED TO THE SPECIFIED ADDRESS BEFORE ACCESS IS MADE. THE LEFT HALF OF THE INDEX-REGISTER WORD IS IGNORED. IF THE D FIELD IS NONZERO, THE CONTENTS OF THE M FIELD (WITH SIGN EXTENSION) ARE LOADED INTO THE SIMULATED INDEX REGISTER. THE SET INDEX MODAL REMAINS IN EFFECT UNTIL IT IS COUNTERMANDED BY ANOTHER SET INDEX.

E$OVRP--OVERPUNCH: THE OVERPUNCH MODAL SPECIFIES THAT ANY MINUS SIGNS PRODUCED BY THE EDITING FUNCTIONS ARE TO BE REMOVED FROM THEIR POSITIONS IN FRONT OF THE EDITED NUMBERS AND PLACED AS 11-PUNCHES OVER THE LOW-ORDER DIGITS. IN THE CASE OF FLOATING POINT EDITING, THE SIGN OF THE MANTISSA IS PLACED OVER THE LOW-ORDER DIGIT OF THE MANTISSA AND THE SIGN OF THE EXPONENT OVER ITS LOW-ORDER DIGIT. THE SPACE THAT WOULD NORMALLY CONTAIN THE SIGN OF THE EXPONENT IS OMITTED. THE OVERPUNCH MODE IS INITIATED BY ITS OCCURRENCE WITH ADDRESS 1. IT IS COUNTERMANDED BY ITS OCCURRENCE WITH ADDRESS 0. UPON INITIAL ENTRY TO EOUT$, THE OVERPUNCH MODE IS ASSUMED TO BE OFF.

19.2.5. THE CONTROL FUNCTIONS

THE CONTROL FUNCTIONS SERVE TO INTRODUCE INTO THE INTERPRETIVE LANGUAGE SOME OF THE CONTROL OPERATIONS AVAILABLE IN MACHINE LANGUAGE.

E$TERM--TERMINATE: THE TERMINATE CONTROL CAUSES THE ROUTINE TO RETURN TO THE NEXT INSTRUCTION IN MACHINE LANGUAGE. UPON RE-ENTRY TO THE ROUTINE AT THE POINT EOTR$, ALL COUNTERS, MODES IN EFFECT, INTERPRETIVE SUBROUTINES, AND ANY PARTIAL IMAGE ARE LEFT UNDISTURBED. IF RE-ENTRY IS MADE AT EOUT$, THESE ARE ALL CLEARED. ENTRY AT EOUTR$ IS MADE BY THE INSTRUCTION

```
        LMJ       11, EOUTR$
```

E$LINK--LINK: THE LINK CONTROL IS USED TO FORM SUBROUTINES IN THE EDITING LANGUAGE. ITS EFFECTIVE ADDRESS SPECIFIES THE LOCATION OF THE ENTRY TO A SUBROUTINE. SUBROUTINES MAY BE NESTED TO A DEPTH OF 10.

E$JUMP--JUMP: THE JUMP CONTROL WITH A NON ZERO EFFECTIVE ADDRESS CAUSES AN INTERPRETIVE TRANSFER OF CONTROL TO THE DESIGNATED LOCATION. IF THE ADDRESS IS ZERO, THE JUMP CONTROL SERVES AS A SUBROUTINE EXIT. TRANSFER IS TO THE INTERPRETIVE INSTRUCTION FOLLOWING THAT LINK CONTROL MOST RECENTLY EXECUTED FOR WHICH NO EXIT HAS BEEN PERFORMED.

E$RPT--REPEAT: THE REPEAT CONTROL CAUSES THE NEXT SINGLE INTERPRETIVE INSTRUCTION TO BE REPEATED A NUMBER OF TIMES SPECIFIED IN THE D FIELD OF THE REPEAT WORD. A REPEAT CONTROL PRECEDING A LINK WILL BE MEANINGLESS; FOR MULTIPLE EXECUTION OF A LINK, THE ROUTINE EOUT$ ITSELF SHOULD BE CALLED WITHIN A MACHINE LANGUAGE LOOP. THE T AND M FIELDS CONTAIN INCREMENTS TO THE T AND M FIELDS OF THE INSTRUCTION TO BE

REPEATED FOR EACH EXECUTION. ANY MODES SET BY THE MODAL INSTRUCTIONS WHICH WOULD BE IN EFFECT FOR THE FIRST EXECUTION OF A REPEATED INSTRUCTION REMAIN IN EFFECT FOR ALL EXECUTIONS.

E$CLR--CLEAR: THE CLEAR CONTROL SETS THE IMAGE TO BLANKS.

19.2.6. EXAMPLES

SEVERAL EXAMPLES OF TYPICAL CALLING SEQUENCES TO EOUT$ FOLLOW:

1. THE FORTRAN INSTRUCTION

```
          PRINT 100, A, I, N, B, C
100       FORMAT (6X, E20.7, I20, O20, 1P.2F20.6)
```

IS EQUIVALENT TO THE INTERPRETIVE SEQUENCE

```
E$OUT
E$E       26,7,A
E$D       46,0,I
E$O       66,0, N
E$SCL     1
E$F       86, 6, B
E$F       106, 6, C
E$WT      22, 0, 1
NEXT MACHINE LANGUAGE INSTRUCTION
```

2. IF THIS LINE WERE TO BE PUT OUT ALSO ON THE CARD PUNCH, WHOSE OUTPUT CODE IS 1, THEN THE LAST INTERPRETIVE INSTRUCTION WOULD BE REPLACED BY

```
E$WS      14, 1, 0
E$WT      22, 0, 1
```

ONLY THE FIRST 80 COLUMNS OF THE IMAGE WOULD BE PUNCHED.

3. THE FORTRAN INSTRUCTION

```
PRINT 100, (J (I), K (I), L(I), M(I), I=1, 4)
100 FORMAT  (20I6)
```

IS EQUIVALENT TO THE FOLLOWING INTERPRETIVE SEQUENCES:

```
E$RPT     30, 4, 1
E$D       6, 0, J, 2
E$RPT     30, 4, 1
E$D       12, 0, K, 2
E$RPT     30, 4, 1
E$D       18, 0, L, 2
E$RPT     30, 4, 1
E$D       24, 0, M, 2
E$WT      22, 0, 1
```

APPENDIX A--DIAGNOSTIC MESSAGES

BAD RUN STATEMENT

THE RUN STATEMENT IS IMPROPERLY FORMATTED. TWO COMMON ERRORS ARE TO HAVE A MISSING RUN-ID OR ACCOUNTING FIELD. THIS MESSAGE WILL ONLY BE PRINTED FOR DEMAND RUNS; FOR BATCH RUNS THE DEVICE IS PLACED IN THE RUN SEARCH MODE. IF THE STATEMENT WAS SUBMITTED FROM A DEMAND TERMINAL, IT MAY BE IMMEDIATELY RETYPED IN THE PROPER FORMAT.

COARSE SCHEDULER COULD NOT FIND PROGRAM

THE REQUESTED PROGRAM OR PROCESSOR IS NOT IN THE GIVEN FILE, LIB$, OR TPF$ (DEPENDING ON THE STATEMENT). IF THE RUN IS NOT DEMAND, IT IS TERMINATED.

COARSE SCHEDULER ENCOUNTERED FILE ERROR

THE FILE REQUESTED ON A @XQT OR PROCESSOR STATEMENT COULD NOT BE ASSIGNED. IF THE RUN IS NOT DEMAND, IT IS TERMINATED.

DATA STATEMENTS ENCOUNTERED--IGNORED

DATA STATEMENTS WERE ENCOUNTERED WHEN THE COARSE SCHEDULER WAS ATTEMPTING TO READ CONTROL STATEMENTS; I.E., A PROGRAM OR PROCESSOR WAS NOT IN CONTROL OF THE RUN AT THE TIME THESE STATEMENTS WERE ENCOUNTERED.

@END STATEMENT ENCOUNTERED BY SYSTEM--IGNORED

AN @END STATEMENT WAS ENCOUNTERED WHEN THE COARSE SCHEDULER WAS ATTEMPTING TO READ CONTROL STATEMENTS; I.E., THE @DATA PROCESSOR WAS NOT IN CONTROL OF THE RUN AT THE TIME THIS STATEMENT WAS ENCOUNTERED.

@EOF STATEMENT ENCOUNTERED BY SYSTEM--IGNORED

AN @EOF STATEMENT WAS ENCOUNTERED WHEN THE COARSE SCHEDULER WAS ATTEMPTING TO READ CONTROL STATEMENTS; I.E., A PROGRAM OR PROCESSOR WAS NOT IN CONTROL OF THE RUN AT THE TIME THIS STATEMENT WAS ENCOUNTERED.

ILLEGAL ACCOUNT NUMBER

THE ACCOUNT NUMBER ON THE RUN STATEMENT IS NOT IN THE INSTALLATION'S SUMMARY ACCOUNT FILE AND THE OPERATOR HAS CHOSEN TO REJECT IT. THIS MESSAGE WILL ONLY BE PRINTED FOR DEMAND RUNS. THE RUN STATEMENT MAY BE IMMEDIATELY RETYPED WITH AN ACCEPTABLE ACCOUNT NUMBER.

INTERVENING STATEMENTS SKIPPED

A CONDITIONAL STATEMENT HAS BEEN ENCOUNTERED AND HAS CAUSED ONE OR MORE CONTROL STATEMENTS TO BE BYPASSED.

OPERATOR TERMINATED RUN

1. THE OPERATOR REPLIED WITH AN X TO @MSG STATEMENT WITH A W OPTION; BOTH BATCH AND DEMAND RUNS ARE TERMINATED IN THIS CASE.
2. THE OPERATOR TYPED AN UNSOLICITED X KEYIN FOR THIS RUN. THE MESSAGE IS PRINTED ONLY FOR BATCH RUNS. AN UNSOLICITED X KEYIN FOR A DEMAND RUN SIMPLY TERMINATES THE CURRENTLY EXECUTING PROGRAM.

REMAINING CONTROL STATEMENTS IGNORED

THE RUN HAS BEEN TERMINATED BECAUSE OF AN ERROR CONDITION AND THE REMAINING CONTROL STATEMENT ARE NOT PROCESSED.

UNRECOVERABLE I/O ERROR WHEN READING FILE FFFFFFFFFFFF

THE COARSE SCHEDULER ENCOUNTERED AN UNRECOVERABLE I/O ERROR WHEN SEARCHING FILE FFFFFFFFFFFF FOR A PROGRAM OR PROCESSOR. IF THE RUN IS NOT DEMAND, IT IS TERMINATED.

APPENDIX B--ERROR CODES

LIST OF ERROR CODES

TYPE	CODE	DESCRIPTION
2	01	ATTEMPT TO READ PAST END OF RUN FILE
2	02	TWO ERROR RETURNS IN A ROW TO A USER
2	03	UNRECOVERABLE I/O (TAPE OR DRUM) ERROR
2	04	IMAGE LENGTH IN A READ FILE IS GREATER THAN 15
2	040	PACKET ERROR (NOT INSIDE STORAGE LIMITS OR ACROSS I AND D BANK)
2	041	MAXIMUM PAGES WITH OPTION TO KILL AT MAX. PAGES
3	01	PAST END OF FILE
3	02	SECOND ABNORMAL RETURN FROM READ$
3	03	I/O (READ$)
3	04	IMLENGTH (READ$)
3	05	ADD FILE NOT ASG'D
3	06	READ ACCESS WORD FAILURE
3	020	ALT. FILE NOT ASSIGNED ON DEMAND OR RT RUN
3	021	CANNOT ASSIGN ALT. FILE
3	022	TYPE OF CALL DOES NOT MATCH TYPE OF FILE
3	023	PACKET OUT OF LIMITS
3	024	READ$ FILE NOT ASSIGNED
3	025	FIRST ERROR ON READ IN READ$ FILE
3	034	EXCEEDED PRA FILE
3	035	EXCEEDED PRA FILE
3	036	EXCEED PU$ FILE
3	037	EXCEED PR$ FILE
3	040	BUFFER OUT OF LIMITS
3	041	MAX PAGES
3	042	MAX CARDS
3	077	NO ADD NESTING
4	01	ER INDEX VALUE GREATER THAN DEFINED. THE ACTIVITY IS TERMINATED
4	02	PACKET NOT WITHIN PROGRAM LIMITS
4	03	ILLEGAL ER INDEX VALUE WITHIN THOSE DEFINED. THE ACTIVITY IS TERMINATED
4	04	NO IDENTITY UPON AWAIT$ REQUEST
4	05	DUPLICATE OR INVALID ACTIVITY ID REQUESTED WHEN REFERENCING FORKING
4	06	REAL TIME PRIORITY IS NOT PERMITTED UNDER THE ACCOUNT NUMBER USED
4	07	EXCEEDED RUN TIME
4	010	PACKET NOT IN USERS AREA (ER FACIL$)
4	011	DRUM READ ERROR WHILE PROCESSING ER FACIL$
4	012	PACKET SUPPLIED IN ER TO BPOOL$ IS NOT COMPLETELY IN USER AREA
4	013	BUFFER AREA SUPPLIED IN ER TO BPOOL$ IS NOT COMPLETELY IN USER AREA
4	014	BUFFER AREA SUPPLIED IN ER TO BPOOL$ IS NOT LARGE ENOUGH FOR 1 BUFFER OF LENGTH SPECIFIED
4	015	BUFFER SIZE SUPPLIED IN ER TO BPOOL$ IS LESS THAN 2 WORDS
4	016	PACKET SUPPLIED IN ER TO BJOIN$ IS NOT

		COMPLETELY IN USER AREA
4	017	BUFFER AREA SUPPLIED IN ER TO BJOIN$ IS NOT COMPLETELY IN USER AREA
4	020	FILE CONTROL TABLE NOT IN USERS AREA
4	021	NO ABNORMAL-, FILE ERROR-, OR DEVICE-EXIT SPECIFIED IN CONTROL TABLE
4	022	BUFFER SIZE SUPPLIED BY ER TO BJOIN$ IS LESS THAN 2 WORDS
4	023	CONTROL PACKET REFERENCED IN BJOIN$ CALLING PACKET IS NOT COMPLETELY IN USER AREA
4	024	BUFFER AREA SUPPLIED IN ER TO BJOIN$ IS NOT LARGE ENOUGH FOR 1 BUFFER AS SPECIFIED IN CONTROL PACKET
4	030	REQUESTED NRT$ FROM AN ACTIVITY WHICH IS NON REAL TIME
4	031	INTTERRUPT ACTIVITY LEVEL REQUESTED WHICH IS NOT PERMITTED.
4	032	NO NAME COULD BE FOUND OR HAS BEEN DEFINED.
4	033	ALL ACTIVITIES OF THE PROGRAM HAVE REQUESTED DEACTIVATION.
4	034	REQUESTED QUARTER-WORD MODE. HOWEVER, NO QUARTER WORD HARDWARE AVAILABLE.
4	035	REQUESTED PROC. STATE REG. BE CHANGED. HOWEVER, I/O WAS OUTSTANDING FOR THE ACTIVITY.
4	036	ILLEGAL STORAGE LIMITS REQUESTED.(PSRSI$, PSRSD$).
4	037	ALL REMAINING ACTIVITIES W/IDENTITIES WENT TO AWAIT$.
4	040	SYNTAX ERROR DISCOVERED IN CONTROL STATEMENT IMAGE
4	041	IMAGE LENGTH OVER 40 WORDS LONG
4	042	CONTROL STATEMENT IMAGE CONTAINS A COMMAND WHICH CANNOT BE SERVICED BY -CSF-
4	043	BUFFER ADDRESS OF CONTROL STATEMENT IMAGE LIES OUTSIDE OF USERS CORE LIMITS
4	050	ATTEMPT MADE TO LOAD OUTSIDE PROGRAM AREA.
4	051	ATTEMPT MADE TO LOAD AN UNDEFINED SEGMENT.
4	052	PROGRAMS SEGMENT LOAD TABLE CONTAINS ILLEGAL SEGMENT.
4	053	ILLEGAL MCORE$ REQUEST.
4	054	ILLEGAL LCORE$ REQUEST.
4	055	BAD I/O STATUS.
4	060	UNACCEPTABLE RECOGNITION CODE FOR PRIVILEGED EXECUTIVE REQUEST (ER DLOC$, DIW$).
4	077	ENTRANCE TO -FCUI- WITH AN ILLEGAL ER INDEX WAS DETECTED. CURRENTLY LEGAL ER INDICES ARE FACIL$, BPOOL$, AND BJOIN$
6	01	A CMO$ REQUEST WITH POOL MODE SPECIFIED AND LT'S START OF OUTPUT BUFFER LOCATION IS ZERO.
6	02	A CMO$ REQUEST WITH POOL MODE SPECIFIED AND LT'S BUFFER POOL CONTROL WORD IS INVALID.
6	03	A CMO$ REQUEST WITH POOL MODE SPECIFIED AND BUFFER LOCATION SPECIFIED IS INVALID.
6	34	A CMO$ REQUEST WITH POOL MODE SPECIFIED AND THE BUFFER CHARACTER COUNT EXCEEDS THE SYSTEM'S MAXIMUM CHARACTER COUNT.
6	05	A CADD$ REQUEST WITH INVALID BUFFER LOCATIONS.
6	06	A CADD$ REQUEST AND FIRST BUFFER LOCATION IS ZERO.
6	07	A ROUTE$ REQUEST WITH NO FACILITIES ASSIGNED FOR THE SPECIFIED LT.
6	10	A ROUTE$ REQUEST WITH INVALID POINTER WHEN REQUESTING AN ALTERNATE PATH. THE POINTER IS GREATER HTAN THE NUMBER OF ALTERNATES

		SPECIFIED AT SYSTEMS GENERATION TUME.
6	11	A ROUTE$ REQUEST AND USERS SPECIFIED MODE CODE ALTERNATE IS GREATER THAN 3 INDICATING A POSSIBLE ERROR.
6	12	A ROUTE$ REQUEST WITH INVALID PROGRAM TYPE.
6	13	A ROUTE$ REQUEST WITH INVALID LT LOCATION.
6	14	A ROUTE$ REQUEST REQUESTING OUTPUT BUT OUTPUT NOT ALLOWED FOR THE ALTERNATE ROUTE.
6	16	A CGET$ OR CADD$ REQUESTING A ZERO BUFFER COUNT.
6	20	A ROUTE$ REQUEST AND USERS LT HAS NOT BEEN INITIALIZED.
6	21	A CGET$ OR CADD$ REQUEST WITH INVALID EXECUTIVE LINK PARAMETER.
6	22	A CGET$ OR CADD$ REQUEST WITH INVALID POOL TYPE. MUST BE OPEN CHAINS TYPE.
6	23	A CGET$ REQUEST REQUESTING BUFFERS AND NO BUFFERS ARE AVAILABLE. BLE.
6	24	A ROUTE$ REQUEST REQUESTING AN ALTERNATE PATH WHICH HAS ALREADY BEEN ASSIGNED.
7	1	AN ER CMS$ REFERENCE HAS BEEN MADE FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.
7	2	AN ER CMS$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 WHICH IS BEYOND THE LIMITS OF THE PROGRAM.
7	03	THE INITIALIZATION REQUEST SPECIFIES A LT TABLE WHICH IS CURRENTLY IN USE OR HAS BEEN INITIALIZED BUT HAS NEVER BEEN TERMINATED.
7	04	THE INITIALIZATION REQUEST SPECIFIED AN INVALID FILE NAME, AN INCORRECT FILE NAME OR THE FILE NAME CONTAINED AN ERROR.
7	05	AN INVALID VALUE WAS SPECIFIED FOR THE INPUT COMPLETION ACTIVITY USAGE CODE WITHIN THE REQUESTING PACKET FOR AN ER CMS$.
7	06	THE FILE NAME FOR AN INITIALIZATION REQUEST DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE (PCT).
7	07	THE END OF INPUT ACTIVITY CODE IN THE PACKET FOR AN INITIALIZATION REQUEST IS NOT VALID.
7	010	THE SINGLE BUFFER MODE INPUT CHARACTER COUNT WITHIN THE PACKET FOR AN INITIALIZATION REQUEST EXCEEDS THE MAXIMUM PERMISSABLE VALUE ESTABLISHED AT SYSTEMS GENERATION TIME.
7	011	THE CONTROL GROUP ADDRESS SPECIFIED IN A PACKET TO BE USED FOR POOL MODE OF OPERATION IS INVALID.
7	012	POOL MODE OF OPERATION HAS BEEN SPECIFIED, BUT NO BUFFERS PRESENTLY EXIST IN THE POOL SPECIFIED AS THE ONE TO BE USED.
7	013	THE ADDRESS WITHIN THE PACKET SPECIFIED AS THE INPUT COMPLETION ACTIVITY WAS NOT VALID.

7 014 THE OUTPUT USAGE CODE SPECIFIED WITHIN THE PACKET FOR AN INITIALIZATION REQUEST IS NOT VALID.

7 015 THE SINGLE BUFFER MODE OUTPUT CHARACTER COUNT WITHIN THE PACKET FOR AN INITIALIZATION REQUEST EXCEEDS THE MAXIMUM PERMISSABLE VALUE ESTABLISHED AT SYSTEMS GENERATION TIME.

7 016 AN ER CMT$ REFERENCE HAS SPECIFIED A LT GROUP WHICH HAS BEEN TERMINATED PREVIOUSLY OR HAS NEVER BEEN INITIALIZED.

7 017 AN ER CMT$ REFERENCE HAS BEEN MADE FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.

7 020 AN ER CMT$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 BEYOND THE BOUNDS OF THE PROGRAM.

7 021 A REQUEST FOR AN INPUT OPERATION HAS SPECIFIED A FILE NAME FOR WHICH INPUT IS NOT PERMISSABLE (NO INPUT FACILITY FOR THE LT GROUP WAS DEFINED AT SYSTEMS GENERATION TIME).

7 022 AN INVALID VALUE WAS SPECIFIED FOR THE INPUT COMPLETION USAGE CODE WITHIN THE REQUESTING PACKET FOR AN ER CMI$.

7 023 THE NUMBER OF CHARACTERS SPECIFIED FOR A SINGLE BUFFER MODE INPUT OPERATION EXCEEDS THE MAXIMUM PERMISSABLE VALUE ESTABLISHED AT SYSTEMS GENERATION TIME.

7 024 AN ADDRESS BEYOND THE BOUNDS OF THE PROGRAM WAS SPECIFIED IN H2 OF A0 FOR AN ER DIRECTED TO THE COMMUNICATIONS HANDLER.

7 025 A NON-MULTIPLE OF 6 WAS SPECIFIED AS THE CHARACTER COUNT FOR AN INPUT OPERATION ON THE WTS SUBSYSTEM.

7 026 AN INPUT OPERATION HAS SPECIFIED A CHARACTER COUNT AND BUFFER STARTING ADDRESS SUCH THAT A PORTION OF THE INPUT BUFFER EXISTS BEYOND THE LIMITS OF THE PROGRAM.

7 027 AN INPUT OPERATION HAS SPECIFIED POOL MODE OF OPERATION FOR A FILE NAME ASSOCIATED WITH EQUIPMENT WHICH USES THE ISI METHOD OF BUFFERING. BUFFER SWAPPING FOR ISI DEVICES IS NOT PERMITTED DUE TO THE DESIGN OF THE HARDWARE WHICH OPERATES ON DATA BY BLOCKS RATHER THAN AS A CONTINUOUS INPUT STREAM.

7 030 A REQUEST TO THE COMMUNICATIONS HANDLER HAS SPECIFIED AN INVALID FILE NAME.

7 031 A REQUEST FOR AN OUTPUT OPERATION HAS SPECIFIED A FILE NAME FOR WHICH OUTPUT IS NOT PERMISSABLE (NO OUTPUT FACILITY FOR THE LT GROUP WAS DEFINED AT SYSTEMS GENERATION TIME).

7 032 AN INVALID VALUE WAS SPECIFIED FOR THE OUTPUT COMPLETION USAGE CODE WITHIN THE REQUESTING PACKET FOR

AN ER CMO$.

7 033 THE NUMBER OF CHARACTERS SPECIFIED FOR A SINGLE BUFFER MODE OUTPUT OPERATION EXCEEDS THE MAXIMUM PERMISSABLE VALUE ESTABLISHED AT SYSTEMS GENERATION TIME.

7 034 AN OUTPUT OPERATION HAS SPECIFIED A CHARACTER COUNT AND A BUFFER STARTING ADDRESS SUCH THAT A PORTION OF THE OUTPUT BUFFER EXISTS BEYOND THE LIMITS OF THE PROGRAM.

7 036 A CORE AREA SPECIFIED BY AN ER CPOOL$ IS NOT VALID AS IT IS EITHER BEYOND THE BOUNDS OF THE PROGRAM OR SPLIT BETWEEN THE I-BANK AND D-BANK.

7 037 THE CORE AREA SPECIFIED BY AN ER CPOOL$ IS LOCATED IN THE I-BANK. POOL BUFFERING USING THE I-BANK IS NOT PERMITTED SINCE A PROGRAM IS SUPPOSED TO BE SO POSITIONED IN CORE SUCH THAT THE HARDWARE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS WILL BE UTILIZED IN ORDER TO REDUCE THE NUMBER OF CORE MEMORY ACCESSES FROM 3 TO 2 FOR EACH DATA TRANSFER.

7 040 A CHARACTER COUNT EXCEEDING THE MAXIMUM PERMISSABLE VALUE HAS BEEN SPECIFIED BY AN ER CPOOL$.

7 041 AN INSUFFICIENT CORE AREA HAS BEEN SPECIFIED BY AN ER CPOOL$ SUCH THAT IT IS NOT POSSIBLE TO FORM A CLOSED CHAIN WITH THE NUMBER OF SEGMENTS SPECIFIED TO BE USED.

7 042 AN ER TO THE COMMUNICATIONS HANDLER HAS BEEN EXECUTED FROM THE WRONG TYPE OF CODING. USE OF THE COMMUNICATIONS HANDLER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.

7 043 THE SINGLE MODE OF BUFFERING HAS SPECIFIED A COMMUNICATIONS BUFFER WHICH IS IN THE I-BANK. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE I-BANK BECAUSE REAL-TIME PROGRAMS ARE PROPERLY POSITIONED IN CORE SO THAT THE D-BANK CAN UTILIZE THE MEMORY OVERLAP FEATURE FOR EACH CHARACTER TRANSFER.

7 044 AN ER HAS BEEN MADE TO THE COMMUNICATIONS HANDLER PRIOR TO THE ESTABLISHMENT OF AN 'ERR MODE' ROUTINE.

7 45 THE ADDRESS WITHIN THE PACKET SPECIFIED AS THE OUTPUT COMPLETION ACTIVITY WAS NOT VALID FOR AN ER CMS$.

7 46 AN ER HAS BEEN MADE REQUESTING USE OF A LT GROUP BY THE COMMUNICATIONS HANDLER BEFORE THAT LT GROUP HAS BEEN INITIALIZED. A LT GROUP MUST BE INITIALIZED VIA AN ER CMS$ BEFORE ANY INPUT OR OUTPUT CAN BE PERFORMED.

7 47 AN ER CMO$ HAS DESIGNATED THAT POOL MODE SHOULD BE USED FOR OUTPUT, BUT THE BUFFER POOL CONTROL WORD SPECIFIED IN THE LT TABLE IS INVALID.

7 50 AN ER CMI$ FOR SINGLE BUFFER MODE HAS SPECIFIED AN INPUT BUFFER AREA WHICH CONFLICTS WITH THE MAIN STORAGE MODULE LOCATIONS OF THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED. THE REAL-TIME PROGRAM MUST BE PROPERLY POSITIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.

7 51 AN ER CMO$ FOR SINGLE BUFFER MODE HAS SPECIFIED AN OUTPUT BUFFER AREA WHICH CONFLICTS WITH THE MAIN STORAGE MODULE LOCATIONS OF THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED. THE REAL-TIME PROGRAM MUST BE PROPERLY POSITIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.

7 52 THE CORE AREA SPECIFIED BY AN ER CPOOL$ IS LOCATED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACCESS CONTROL WORDS. COMMUNICATIONS BUFFERS ARE NOT PERMITTED IN THE SAME MAIN STORAGE MODULE AS THE ESI ACW'S BECAUSE OF THE EXTENDED TIME REQUIRED FOR EACH CHARACTER TRANSFER WHEN THE MEMORY OVERLAP CAPABILITY IS NOT UTILIZED. THE REFERENCE TO CPOOL$ MUST BE MADE BY A PROPERLY POSITIONED IN MAIN STORAGE SO AS TO TAKE ADVANTAGE OF THE MEMORY OVERLAP FEATURE FOR ESI DATA TRANSFERS.

7 53 AN ER CMS$ HAS SPECIFIED A LT TABLE WHICH EXISTS IN THE ED TO BE SOHPISTICATED ENOUGH TO UTILIZE THE HARDWARE FEATURE OF MAIN STORAGE OVERLAP BY BEING DIVIDED INTO APPROPRIATE I-BANK AND D-BANK PORTIONS. THE LT TABLE MUST BE PLACED BY THE REAL-TIME PROGRAM IN ITS D-BANK.

7 54 AN ER CMD$ REFERENCE HAS BEEN MADE FROM THE WRONG TYPE OF CODING. USE OF THIS ER IS NOT PERMITTED FOR DEMAND, DEADLINE BATCH, OR BATCH ACTIVITIES.

7 55 AN ER CMD$ REFERENCE HAS SPECIFIED AN ADDRESS IN H2 OF A0 WHICH IS BEYOND THE LIMITS OF THE PROGRAM.

7 56 AN ER CMD$ REFERENCE HAS SPECIFIED A LT GROUP WHICH HAS BEEN TERMINATED PREVIOUSLY OR HAS NEVER BEEN INITIALIZED.

7 57 THE ADDRESS WITHIN THE PACKET SPECIFIED AS THE DIAL COMPLETION ACTIVITY ADDRESS WAS NOT VALID FOR AN ER CMD$.

7 64 AN ER CMD$ REFERENCE HAS SPECIFIED A DIAL ACCESS CONTROL WORD WHOSE CHARACTER COUNT AND BUFFER STARTING ADDRESS ARE SUCH THAT A PORTION OF THE BUFFER AREA EXISTS BEYOND THE LIMITS OF THE PROGRAM.

7 65 AN ER CMT$ REFERENCE HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE. THIS INDICATES THAT THE ER WAS NOT EXECUTED IN A PROPER SEQUENCE OF THE FILE HAS ALREADY BEEN RELEASED.

7 66 AN ER CMD$ REFERENCE HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE. THIS INDICATES THAT THE ER WAS NOT EXECUTED IN A PROPER SEQUENCE OR THE FILE HAS ALREADY BEEN

RELEASED.

7 67 AN ER CMH$ REFERENCE HAS SPECIFIED A FILE NAME WHICH DOES NOT APPEAR IN THE FACILITY ASSIGNMENT SECTION OF THE USER'S PROGRAM CONTROL TABLE. THIS INDICATES THAT THE ER WAS NOT EXECUTED IN A PROPER SEQUENCE OR THE FILE HAS ALREADY BEEN RELEASED.

UNIVAC

SPERRY RAND