

I/O TEST INDEX

MONITOR CONTROLLED I/O TEST NORMAL OPERATION SUMMARY

P/N 2191291

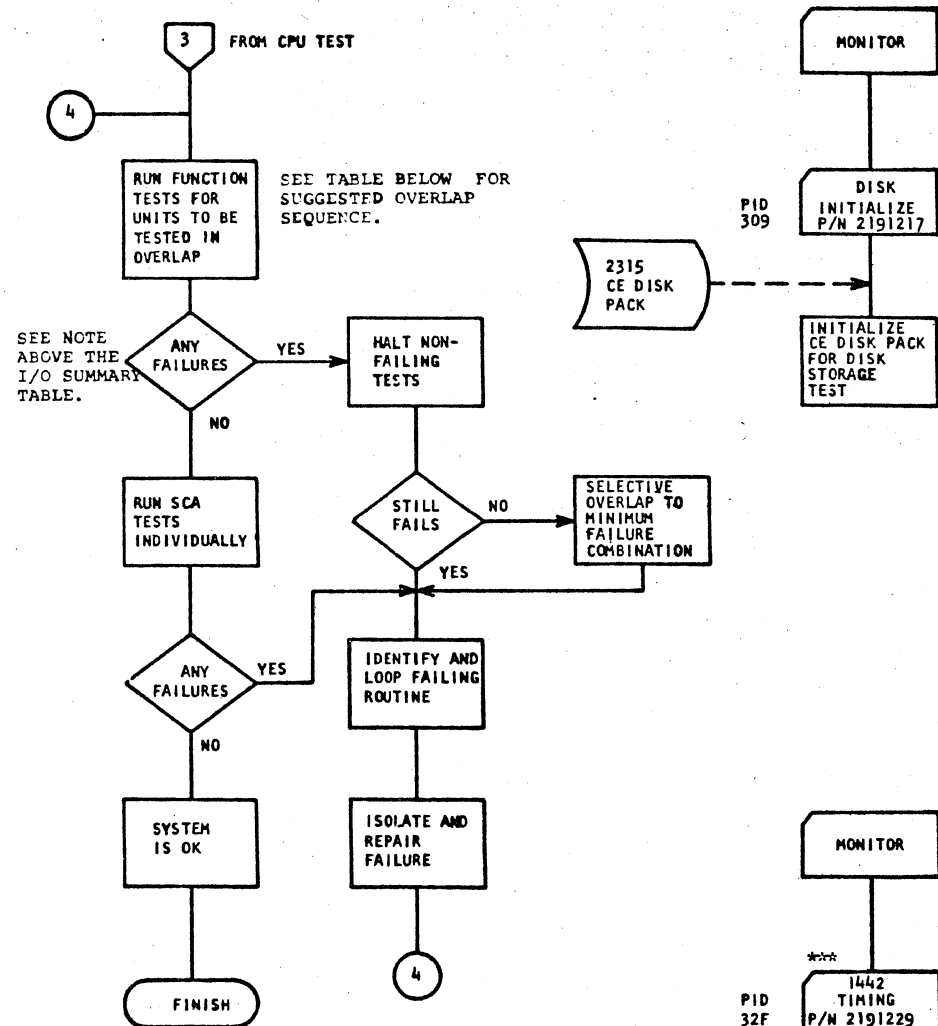
- FOR PAPER TAPE-MOUNT MONITOR TAPE IN TAPE READER AND MAKE READY. (THE RELOCATING LOADER IS AHEAD OF MONITOR ON THE MONITOR TAPE). PRESS STOP, RESET AND PROGRAM LOAD. MONITOR WILL LOAD AND STOP FOR LOADING THE NEXT I/O TEST TAPE. MOUNT NEXT I/O TAPE, SET SWS 0 AND 8 ON FOR SINGLE I/O TEST OR SWS 0,8,9 AND 15 ON FOR OVERLAP, AND THEN PRESS INTERRUPT REQUEST KEY. BRANCH TO ITEM 3.
- FOR CARD SYSTEM-PLACE RELOCATING LOADER, FOLLOWED BY MONITOR II AND I/O TESTS IN THE DESIRED RUN SEQUENCE IN THE CARD READER AND MAKE READER READY.
  - TO RUN TESTS ON AN INDIVIDUAL TEST BASIS, PLACE A BLANK CARD AFTER EACH I/O TEST DECK.
  - TO RUN TESTS IN OVERLAP MODE, PLACE A BLANK CARD AFTER THE LAST I/O TEST DECK.
  - TO STOP MONITOR AFTER ALL PROGRAMS ARE LOADED, PLACE BIT SW 15 ON BEFORE LOADING. PRESS STOP, RESET AND PROGRAM LOAD.
- A STATUS MESSAGE WILL PRINT OUT AFTER EACH I/O TEST IS LOADED. THIS MESSAGE WILL INDICATE LOADING SEQUENCE NUMBER OF THE TEST, THE PROGRAM ID (PID) NUMBER, THE STARTING ADDRESS IN STORAGE AND THE RELOCATION FACTOR.
- CONTROL OPTIONS MAY BE ENTERED AT ANY TIME. THIS IS FUNCTION 0 AND IS EFFECTIVE WHEN BIT SWITCH 0 AND 1 ARE OFF. IF BIT SWITCHES 4,5,6 AND 7 ARE OFF MONITOR IS ADDRESSED BUT WHEN THESE SWITCHES ARE SET TO A LOADING SEQUENCE NUMBER, THAT I/O TEST IS BEING ADDRESSED. CONTROL OPTIONS AND SWITCH SETTINGS ARE:
 

8 - RESTART	10 - LOCK ON FUNCTION	12 - LOOP ON ERROR (ONCE PER ERROR)	14 - HALT ON ERROR
9 - PRINT RTN ID	11 - LOOP ON PROGRAM	13 - BYPASS ERROR PRINT OUT	15 - HALT
- TO STOP ALL PROGRAMS, PRESS PROGRAM STOP, MONITOR WILL WAIT (3001) AT ADDRESS 040C.
- TO RESTART AFTER PROGRAM STOP, PRESS PROGRAM START. MONITOR HAS SAVED THE STATUS OF TESTS.
- TO STOP ANY ONE PROGRAM (WHILE OVERLAPPING), SET BIT SWS TO 0N01 AND PRESS INT. REQ. KEY. N=LOAD SEQUENCE NUMBER
- TO RESTART ANY ONE PROGRAM, SET BIT SWS TO 0N80 AND PRESS INTERRUPT REQUEST KEY. N = LOAD SEQUENCE NUMBER
- ERROR TYPE OUTS ARE EXPLAINED IN SECTION 4.2 OF EACH I/O DESCRIPTION.
- AS EACH TEST REACHES ITS NORMAL END A MESSAGE WILL BE TYPED OUT, FOR EXAMPLE XXXX END, WHERE XXXX = PID.

NOTE: THERE ARE TIME DEPENDENT DEVICES WHICH MUST BE SERVICED WITHIN A SPECIFIED TIME LIMIT AFTER AN INTERRUPT OCCURS. IF THESE LIMITS ARE NOT COMPLIED WITH, DATA WILL BE LOST BETWEEN THE PROCESSOR AND THE I/O DEVICE. THESE TYPES OF ERRORS OCCUR WHEN ANOTHER DEVICE'S PROGRAM KEEPS CONTROL IN MAINLINE OR INTERRUPT ROUTINE FOR EXTENDED PERIODS OF TIME.

PROGRAMS WITH A HIGH RATE OF INTERRUPTS MAY CAUSE OTHER PROGRAMS TO LOSE MAINLINE CONTROL. THIS LOSS OF CONTROL BETWEEN AN XIO READ, WRITE OR CONTROL AND AN XIO SENSE DEVICE MAY CAUSE FALSE DSW ERRORS. THIS TYPE OF ERROR SAYS THE DEVICE SHOULD HAVE BEEN BUSY AND NOT READY. THE PROBLEM IS THE DEVICE HAS COMPLETED IT'S OPERATION, DROPPED BUSY AND BROUGHT UP READY BEFORE CONTROL WAS REGAINED BY THE PROGRAM INDICATING THE ERROR.

IF ANY OF THE TYPES OF ERRORS ABOVE ARE SUSPECTED IN OVERLAP, RUN THE FAILING TEST ALONE TO CHECK FOR TRUE ERRORS.



PID	NAME	P/N	COMMENTS
300	MONITOR	2191201	
304	KB/PRINTER	2191241	
305	1627 PLOTTER	2191237	
309	DISK F.T.	2191213	
308	PAPER TAPE	2191233	
30C	1132 PRINTER	2191221	
314	1231 OMPR	2243554	
30D	1403 PRINTER	2243557	
318	SCA PROGRAM INSTRUCTION	2191297	ONLY SCA TEST FOR OVERLAP
30E	2501/1442 - V	2243551	EITHER PROGRAM BUT NOT BOTH
30F	1442 - VI, VII	2191225	EITHER PROGRAM BUT NOT BOTH

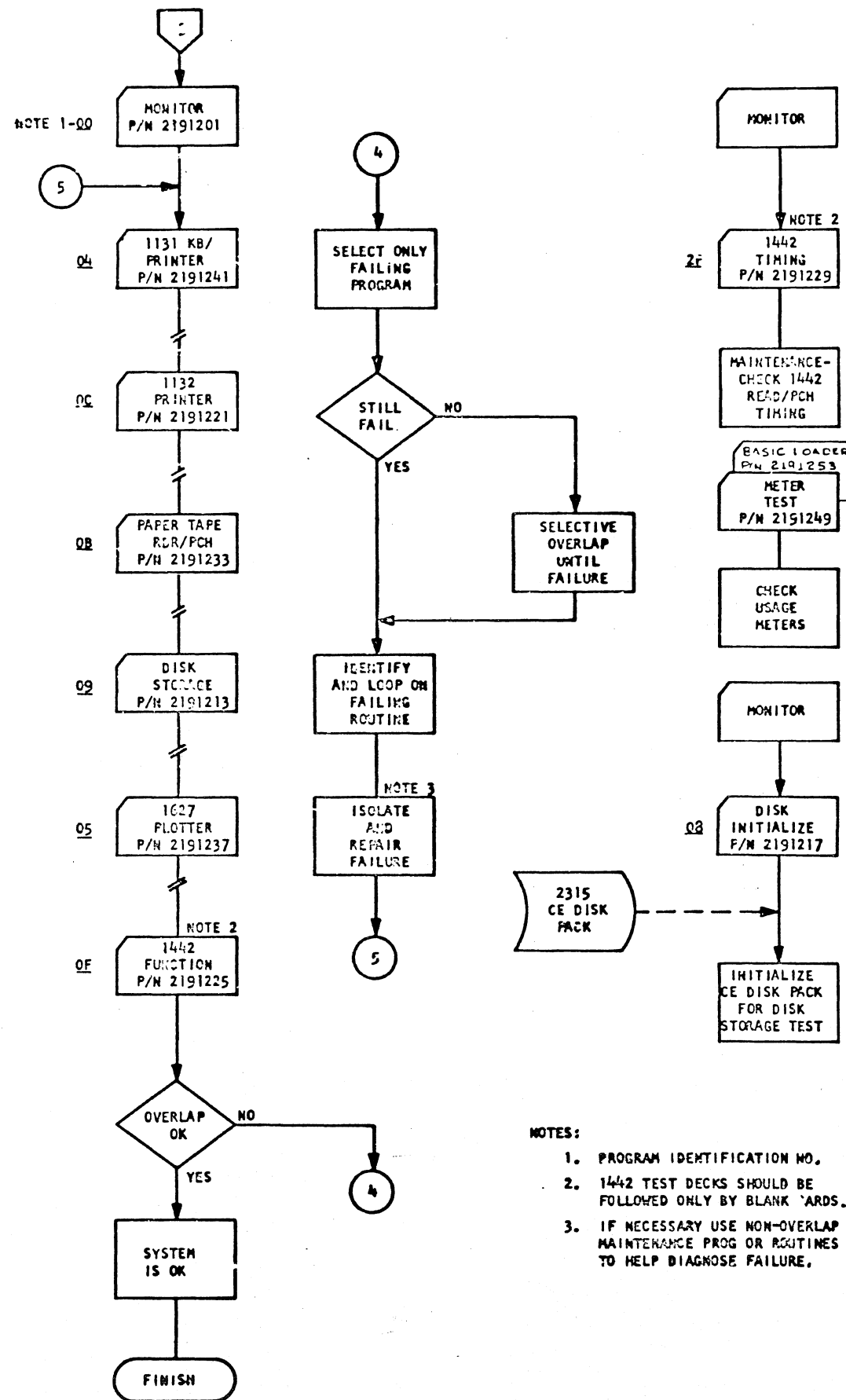
I/O SUMMARY TABLE

PROGRAM	PROG ID NO	CORE SIZE	OVER-LAP	RUN TIME	SPECIAL INSTRUCTIONS
RELOCATING LOADER - 1442	3AA	340	BASIC	-	CARD LOADER FOR MONITOR II
RELOCATING LOADER 2501	3AB	340	BASIC	-	CARD LOADER FOR MONITOR II
RELOCATING LOADER PAPER TAPE	3AC	340	BASIC	-	PAPER TAPE LOADER IN FRONT OF MONITOR II ON PAPER TAPE.
MONITOR II	300	1160	BASIC	-	PROVIDES BASIC CONTROL FOR ALL I/O PROGRAMS.
KEYBOARD PRINTER	304	1200	YES	3 MIN	FOR MANUAL KEYBOARD TESTS SEE SECTION 3.2.3.
1627 PLOTTER (SEE NOTE)	305	950	YES	6 MIN	FOR MANUAL CONTROL SEE SECTION 3.2.3.
DISK INITIALIZATION	308	2300	NO	5 MIN	USE ONCE TO INITIALIZE FE DISK PACK.
DISK FUNCTION TEST	309	2350	YES	3 MIN	USE ONLY DISKS WHICH HAVE BEEN INITIALIZED.
PAPER TAPE	308	750	YES	2 MIN	TAPE MAY BE REPRODUCED BY FUNCTION 1 AND RTN 4.
1132 PRINTER	30C	2200	YES	4 MIN	USE CARRIAGE TAPE WITH 16 OR FEWER PUNCHED HOLES.
1442 FUNCTION TEST	30F	1400	YES	NOT FIXED	ROUTINE 1-PUNCH AND FEED WILL PUNCH ONLY INTO BLANK CARDS.
1442 TIMING TEST	32F	1650	NO	NOT FIXED	ENTER 1442 MODEL-8105,8106 OR 8107.
SCA INSTRUCTION F.T.	318	1600	YES *	5 MIN	NO RESET FAST PASS OPTION RUNS ONE (1) MINUTE.
SCA WRT/RD BUF, LINE NOISE DET.	311	1550	YES *	11 MIN	BUFFER TEST 30 SECONDS AND LINE LISTENING 10.5 MINUTES
SCA WRAP-AROUND TEST	319	1800	NO	5 MIN	NO RESET, FAST PASS OPTION ONE MINUTE
SCA TRANSMIT/RECEIVE	3AE	2050	NO	NOT FIXED	TWO MINUTES IS MIN.OPTIONAL TIME INDEFINITE
SCA DISPLAY PROGRAM	3AF	3755	NO	NOT FIXED	LOAD WITH RELOCATING LOADER
2501/1442 - 5 F.T.	30E	1405	YES	NOT FIXED	
1231 OPTICAL MARK PAGE RDR F.T.	314	900	YES	NOT FIXED	USE CONTROL SHEET
1443 F.T.	30D	1400	YES	3 MIN	USE CARRIAGE TAPE CALLED OUT IN SECTION 5.3
SCA-BSC-PT-TO-PT	317	2500	NO	NOT FIXED	
SCA-BSC-MULTI-PT	31A	2500	NO	NOT FIXED	

\* WILL NOT OVERLAP WITH ANY OTHER SCA PROGRAM  
 \*\*\* AVAILABLE FOR CARD INPUT ONLY

DATE APR 66 SEP 66 JAN 67  
 EC NO. 4154908 415490C 419643

FEB 67 JUN 67 MAY 68  
 419656 RED 420317 420411  
 SN:VI6 5N:VI6 8APR69 15JUL 69 6 AUG 69  
 571005 571036 571013 571053  
 PROG ID PAGE 0A



- NOTES:
- PROGRAM IDENTIFICATION NO.
  - 1442 TEST DECKS SHOULD BE FOLLOWED ONLY BY BLANK CARDS.
  - IF NECESSARY USE NON-OVERLAP MAINTENANCE PROG OR ROUTINES TO HELP DIAGNOSE FAILURE.

MONITOR CONTROLLED I/O TEST NORMAL OPERATION SUMMARY

- PLACE MONITOR, FOLLOWED BY I/O TESTS TO BE RUN, IN 1442. DO NOT EXCEED TOTAL AVAILABLE CORE STORAGE, SEE TABLE BELOW.
- PLACE BLANK CARD AFTER LAST TEST DECK.
- PROGRAM LOAD. MONITOR WILL LOAD, TYPE OUT A0000 LD, THEN WAIT IN A LOOP FOR I/O PROGRAM LOADING INSTRUCTIONS.
- THE I/O TESTS MAY BE RUN IN ANY ONE OF THREE MODES SHOWN BELOW. SELECT THE DESIRED RUNNING MODE BY SETTING THE BIT SWITCHES AND PRESSING THE INTERRUPT REQUEST KEY WITH CONSOLE/KEYBD SWITCH SET TO CONSOLE.

0060 OVERLAP	00A0 SINGLE PROGRAM, LOAD AND GO	0020 SINGLE PROGRAM, WAIT BEFORE EXECUTE
THIS IS THE NORMAL MAINTENANCE TESTING MODE. ALL PROGRAMS WILL LOAD AND TYPE OUT THE STARTING CORE LOCATION OF EACH TEST.  SET BIT SW TO 0080 AND PRESS INT REQUEST TO BEGIN EXECUTION.	USE THIS MODE WHEN DEVICES ARE TO BE RUN SINGLY. EACH PROGRAM WILL LOAD AND BEGIN EXECUTION. WHEN THE PROGRAM REACHES ITS END AN END MESSAGE WILL BE TYPED OUT AND THE NEXT PROGRAM WILL LOAD AND BEGIN EXECUTION.	USE THIS MODE ONLY WHEN SPECIAL I/O TEST OPTIONS OR ROUTINES ARE TO BE SET UP PRIOR TO THE BEGINNING OF PROGRAM EXECUTION. PROGRAM WILL LOAD, THEN WAIT IN MONITOR LOOP FOR OPTIONS TO BE ENTERED: SEE I/O DESCRIPTIONS FOR OPTIONS.  AFTER OPTIONS HAVE BEEN ENTERED SET BIT SW TO: 0080 AND PRESS INT REQUEST TO BEGIN EXECUTION.

- CONTROL OPTIONS MAY BE ENTERED AT ANY TIME DURING PROGRAM EXECUTION: SEE DESCRIPTION. THE FOLLOWING OPTIONS WILL CONTROL ALL I/O PROGRAMS BEING RUN. TO CONTROL ONLY INDIVIDUAL PROGRAMS SEE SUMMARY TBL OR INDIVIDUAL DESCRIPTION.
  - 0090 LOOP ALL PROGRAMS
  - 0028 LOOP ANY FAILING PROGRAM ON ERROR
  - 0084 BYPASS ALL ERROR PRINTOUTS
  - 0052 HALT ALL PROGRAMS ON ERROR
  - 008C BYPASS ALL ERROR PRINTOUTS AND LOOP ANY FAILING PROGRAM ON ERROR
  - 0082 RESET ALL CONTROL OPTIONS

7. I/O SUMMARY TABLE

TEST	PROG ID NO	CORE WORDS	OVER-LAP	PUNNING TIME	LOOP PROGRAM	BYPASS PRINTOUT	LOOP ROUTINE	SPECIAL INSTRUCTIONS
MONITOR	0	1500	BASIC	-	-	-	-	PROVIDES BASIC CTRL FOR ALL I/O PROGRAMS.
KEYBOARD	4	1000	YES	3 MIN	-	-	0420	FOR MANUAL KEYED TESTS SEE SECT 3.2.1.
PLOTTER	5	1000	YES	6 MIN	0510	0508	0520	FOR MANUAL CONTROL SEE SECT 3.2.2.C.
PAPER TAPE	B	700	YES	2 MIN	-	-	0B20	READER ONLY= 0B02; PCH ONLY= 0B01.
PRINTER	C	1450	YES	2 MIN	0C10	-	0C20	INSTALL TEST CARRIAGE TAPE, P/N 2191276
DISK STORAGE	9	2300	YES	3 MIN	0910	-	0920	USE INITIALIZED FE DISK PACK ONLY.
1442 FUNCTION	F	1050	YES	10-20	0F10	0F08	0F20	LOAD ONLY BLANK CARDS AFTER DECK.
DISK INITIALIZE	8	-	NO	5 MIN	0810	-	0820	USE ONCE TO INITIALIZE FE DISK PACK.
1442 TIMING	2F	-	NO	NOT FIXED	-	2F04	-	ENTER 1442 MODEL: AFG6 OR AFG7.

- TO STOP ALL PROGRAMS PUSH PROGRAM STOP; MONITOR WILL WAIT (30FF) AT ADDRESS 0254.
- TO RESTART ALL PROGRAMS PUSH PROGRAM START.
- TO STOP ANY ONE PROGRAM (WHILE OVERLAPPING) SET BIT SW TO 400X. X = PROGRAM ID NUMBER.
- TO RESTART (RESELECT) ANY ONE PROGRAM SET BIT SW TO 408X. X = PROGRAM ID NUMBER. EXECUTE (0080) IF PROGRAM DOES NOT START.
- ERRORS ARE TYPED OUT AS FOLLOWS:
  - E0XNN 004R
  - X = PROG ID NUMBER - ERROR WAS DETECTED BY PROGRAM X.
  - NN = ERROR ID NUMBER - REFER TO PROG X DOCUMENTATION FOR ERROR DESCRIPTION.
  - RR = ROUTINE NUMBER (IN PROG X) IN WHICH FAILURE WAS DETECTED.
- AS EACH TEST REACHES ITS NORMAL END AN END MESSAGE WILL BE TYPED AS FOLLOWS:
  - A0XNN END

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	
2. REQUIREMENTS . . . . .	
2.1 PROGRAM REQUIREMENTS	
2.2 EQUIPMENT REQUIREMENTS	
3. OPERATING PROCEDURE . . . . .	
3.1 PROGRAM LOADING	
3.1.1 FROM CARDS	
3.1.2 FROM PAPER TAPE	
3.1.3 LOAD AND GO	
3.1.4 PAUSE BEFORE EXECUTE	
3.1.5 TO LOAD AFTER INITIAL LOAD	
3.2 PROGRAM OPERATION	
3.2.1 ORGANIZATION OF CONSOLE ENTRY SWITCHES	
3.2.2 SINGLE PROGRAM OPERATION	
3.2.3 OVERLAP OPERATION	
3.2.4 PROGRAM CONTROL OPTIONS - FUNCTION 0	
3.2.5 OPTION SELECTION - FUNCTIONS 1, 2, AND 3	
3.2.4 SPECIAL MONITOR CONTROL	
3.3 HALTS	
3.3.1 NORMAL HALTS	
3.3.2 ERROR HALTS	
3.4 TERMINATION	
3.5 PROGRAM RESTART	
3.6 PATCHING	
4. PRINTOUTS . . . . .	
4.1 STATUS MESSAGES	
4.2 ERROR MESSAGES	
5. COMMENTS . . . . .	
5.1 INTRODUCTION TO DIAGNOSTIC MONITOR OPERATION	
5.1.1 WHAT IS THE MONITOR	
5.1.2 WHY IS A MONITOR NECESSARY	
5.1.3 COMMON PROGRAM FUNCTIONS	
5.1.4 ADVANTAGES OF HAVING COMMON FUNCTIONS IN MONITOR	
5.1.5 HOW DOES THE MONITOR CONTROL A PROGRAM	
5.1.6 HOW DOES A TEST PROGRAM REQUEST CONTROL FROM THE MONITOR	
5.1.7 MAIN LINE SEQUENCE CONTROL	
5.1.8 HOW DOES MONITOR AGAIN OBTAIN CONTROL	
5.1.9 HOW ARE TEST MESSAGES PRINTED	
5.2 ROUTINE DESCRIPTION	
5.2.1 RELOCATING LOADER	
5.2.2 SUPERVISOR ROUTINE	
5.2.3 INTERRUPT ROUTINE	
5.2.4 BIT SWITCH ROUTINE	
5.2.5 MESSAGE LOG ROUTINE	
5.2.6 END ROUTINE	
5.2.7 HALT ROUTINE	

1. PURPOSE

THE DIAGNOSTIC MONITOR IS A CONTROL PROGRAM DESIGNED TO PROVIDE OVERLAP CAPABILITY AND TO HANDLE MANY PROGRAM FUNCTIONS WHICH ARE COMMON TO ALL OF THE TEST PROGRAMS.

2. PREREQUISITES

2.1 PROGRAM PREREQUISITES

1. THIS PROGRAM MUST BE LOADED BY ONE OF THE DIAGNOSTIC LOADERS.
  - A. 03AA - 1442 RELOCATING LOADER
  - B. 03AC - 1134 RELOCATING LOADER
  - C. 03AB - 2501 RELOCATING LOADER
2. ALL PROGRAMS TO RUN UNDER CONTROL OF THIS MONITOR MUST BE IN DIAGNOSTIC MONITOR II FORMAT.

2.2 EQUIPMENT PREREQUISITES

1130 WITH INPUT FROM CARDS OR PAPER TAPE.

3. OPERATING PROCEDURE

3.1\*\*\* PROGRAM LOADING

3.1.1 FROM CARDS

PLACE THE MONITOR AND THE PROGRAMS TO BE LOADED UNDER MONITOR CONTROL IN THE READER. THE MONITOR MUST BE PRECEDED BY A RELOCATING LOADER. PLACE THE READER IN READY CONDITION. PRESS THE 1131 RESET THEN PROGRAM LOAD. THE CARDS WILL LOAD UNTIL A BLANK CARD IS ENCOUNTERED. OR THE READER GOES NOT READY.

IF A BLANK CARD IS PLACED AFTER EACH TEST PROGRAM DECK, EACH PROGRAM WILL LOAD, EXECUTE, THEN CALL IN THE NEXT PROGRAM UNTIL THE READER BECOMES NOT READY. IF TEST DECKS ARE NOT SEPARATED BY A BLANK CARD, ALL PROGRAMS PLACED IN THE READER WILL BE LOADED AND RUN IN OVERLAP. A MAXIMUM OF FIFTEEN (15) PROGRAMS CAN BE LOADED AND RUN IN OVERLAP.

NOTE - IF THE LAST CARD IN THE READ HOPPER IS NOT A BLANK CARD, THE READER WILL GO NOT READY BEFORE THE LAST PROGRAM CARD IS READ AND THE LOADER WILL STOP AT WAIT 30F8. TO CONTINUE, PRESS THE READER START AND THE 1131 START.

3.1.2 FROM PAPER TAPE

PLACE THE MONITOR TAPE ON THE 1134 PAPER TAPE READER (THE 1134 RELOCATING LOADER, 03AC, IS INCLUDED AS THE FIRST RECORD ON THE MONITOR TAPE). MAKE THE READER READY. PRESS THE 1131 RESET THEN PROGRAM LOAD. THE MONITOR WILL LOAD, THEN STOP TO ALLOW LOADING OF THE FIRST TEST PROGRAM TAPE.

WHEN THE NEXT TAPE IS READY TO LOAD, SET BIT SWITCHES 0 AND 8 ON IF LOADING A SINGLE PROGRAM, OR SWITCHES 0, 8, 9 AND 15 ON IF LOADING OVERLAP, THEN PRESS THE INTERRUPT REQUEST KEY. THIS WILL CAUSE THE NEXT TAPE TO LOAD. THE ABOVE PROCESS MUST BE REPEATED FOR EVERY TAPE TO BE LOADED IN OVERLAP. A MAXIMUM OF FIFTEEN (15) PROGRAMS CAN BE LOADED IN OVERLAP.

3.1.3 LOAD AND GO - LOADING IS NORMALLY LOAD AND GO WHETHER LOADING ONE PROGRAM OR SEVERAL PROGRAMS IN OVERLAP.

3.1.4 PAUSE BEFORE EXECUTE-TO PAUSE BEFORE EXECUTING THE PROGRAM(S) LOADED, TURN BIT SWITCH 15 ON BEFORE LOADING THE PROGRAM. THIS WILL TURN ON THE HALT SWITCH FOR EACH PROGRAM LOADED. TO START ALL PROGRAMS SET SWITCHES 0-7 AND 15 OFF, SWITCH 8 ON, SWITCHES 9-14 FOR DESIRED CON-

TROL THEN PRESS THE INTERRUPT REQUEST KEY ON THE 1131 CONSOLE. TO START ANY SINGLE PROGRAM FOLLOW ABOVE PROCEDURE WITH SWITCHES 4-7 SET TO THE LOAD SEQUENCE NUMBER OF THE PROGRAM TO BE STARTED (SEE SECTION 3.2.3 FOR EXPLANATION OF LOAD SEQUENCE NUMBER).

NOTE - THIS START PROCEDURE IS IDENTICAL WITH RESTART AS OUTLINED IN SECTION 3.5.

3.1.5 TO LOAD AFTER INITIAL LOAD - ONCE THE MONITOR PROGRAM HAS BEEN LOADED IT IS POSSIBLE TO REQUEST AT ANY TIME THE LOADING OF ANOTHER PROGRAM OR PROGRAMS. ADDITIONAL PROGRAMS CAN BE LOADED EITHER TO RUN INDIVIDUALLY OR IN OVERLAP WITH THE PROGRAM(S) PREVIOUSLY LOADED AND/OR IN OVERLAP WITH THEMSELVES. IF A SINGLE PROGRAM IS TO BE LOADED AND RUN ALONE, IT WILL TAKE THE PLACE IN CORE OF THE PROGRAM(S) PREVIOUSLY LOADED. THIS WILL BE TRUE EVEN WHEN THE PREVIOUS PROGRAMS WERE LOADED IN OVERLAP. IF ADDITIONAL PROGRAMS ARE LOADED IN OVERLAP, THEY CAN EITHER TAKE THE PLACE OF PROGRAMS PREVIOUSLY LOADED OR THEY CAN BE LOADED TO OVERLAP WITH THE PREVIOUSLY LOADED PROGRAMS.

1. THE LOAD REQUEST IS A FUNCTION 2 SWITCH ENTRY TO THE MONITOR. THIS MEANS THAT SWITCH 0 MUST BE ON AND SWITCHES 1-7 OFF TO GIVE A LOAD REQUEST. SWITCH 8 IS THE LOAD SWITCH. SWITCH 9 IS THE CONTINUE LOAD OR OVERLAP SWITCH.
2. RELOAD - THE SWITCH SETTING FOR RELOAD IS 8080. THIS WILL CLEAR THE MONITOR OF CONTROLS FOR ALL PREVIOUSLY LOADED PROGRAMS AND LOAD THE FIRST PROGRAM IN THE READER INTO CORE LOCATION 050C (1500 DECIMAL). IF THIS FIRST PROGRAM IS TERMINATED BY A BLANK CARD, THE LOADING WILL STOP AFTER THAT PROGRAM IS LOADED. IF THIS PROGRAM IS NOT TERMINATED BY A BLANK CARD THE NEXT PROGRAM IN THE READER WILL BE LOADED. IF THE FOLLOWING PROGRAMS ARE RELOCATABLE, THEY WILL BE RELOCATED STARTING WITH THE FIRST EVEN ADDRESS AFTER THE LAST ADDRESS USED BY THE PREVIOUSLY LOADED PROGRAM. FOR THE PURPOSE OF PROGRAM IDENTIFICATION FOR SWITCH ENTRIES, THE FIRST PROGRAM IN THIS RELOADED DECK NOW BECOMES PROGRAM NUMBER ONE (1), THE SECOND IS NUMBER TWO (2), ETC.
3. CONTINUE LOADING - THE SWITCH SETTING TO CONTINUE LOADING IS 80C0. THIS WILL CAUSE THE NEXT PROGRAM TO BE LOADED STARTING AT THE FIRST CORE POSITION FOLLOWING THE LAST ADDRESS IN THE PREVIOUSLY LOADED PROGRAM. THUS THE NEWLY LOADED PROGRAMS CAN BE RUN IN OVERLAP WITH THE PREVIOUSLY LOADED PROGRAMS.

3.2\*\*\* PROGRAM OPERATION

3.2.1 ORGANIZATION OF CONSOLE ENTRY SWITCHES

ALL CONTROL IS EXERTED ON THE TEST PROGRAMS AND THE MONITOR BY ENTRIES IN THE 1131 CONSOLE ENTRY SWITCHES. TO PROVIDE FLEXIBILITY, THESE SWITCHES ARE DIVIDED INTO THREE GROUPS. SWITCHES 0 AND 1 ARE THE FUNCTION SWITCHES (F), SWITCHES 4-7 ARE THE PROGRAM IDENTIFICATION SWITCHES (P), AND SWITCHES 8-15 ARE THE DATA SWITCHES (D). SWITCHES 2-3 ARE NOT USED.

1. FUNCTION SWITCHES (F) ARE USED TO IDENTIFY THE FUNCTION OF THE DATA ENTERED IN SWITCHES 8-15. THE SETTING OF THESE SWITCHES DETERMINES INTO WHICH OF THE 4 SWITCH WORDS IN THE PROGRAM CONTROL TABLE THE SETTING OF THE DATA SWITCHES WILL BE STORED. OF THE FOUR FUNCTIONS AVAILABLE, THE USE OF 2 HAS BEEN STANDARDIZED.
  - A. FUNCTION 00 IS USED TO SPECIFY THE PROGRAM CONTROL OPTIONS PROVIDED BY THE MONITOR. THESE ARE DISCUSSED IN DETAIL IN SECTION 3.2.4.
  - B. FUNCTION 01 IS USED BY EACH TEST PROGRAM TO SELECT A SPECIFIC ROUTINE. THIS IS DISCUSSED IN SECTION 3.2.5.

2. THE PROGRAM IDENTIFICATION SWITCHES (P) ARE USED TO IDENTIFY THE SPECIFIC PROGRAM FOR WHICH THE INFORMATION IN THE DATA SWITCHES (D) IS TO BE APPLICABLE. THE NUMBER TO BE USED TO IDENTIFY ANY PROGRAM IS DEPENDENT ON THE ORDER IN WHICH THAT PROGRAM WAS LOADED, THE FIRST PROGRAM LOADED IS PROGRAM ONE (1), THE FIFTH PROGRAM LOADED IS PROGRAM FIVE (5), ETC. SWITCHES 4-7 ARE SET TO THE BINARY EQUIVALENT OF THIS PROGRAM NUMBER. THUS, TO ENTER DATA TO PROGRAM 1, SWITCHES 4,5 AND 6 MUST BE TURNED OFF AND SWITCH 7 TURNED ON.
3. THE DATA SWITCHES (D) ARE USED TO ENTER SPECIFIC DATA INTO THE PROGRAM IDENTIFIED BY THE PROGRAM SWITCHES (P). THIS DATA IS STORED IN ONE OF THE 4 SWITCH WORDS IN THE PROGRAM CONTROL TABLE OF THE PROGRAM IDENTIFIED IN THE P SWITCHES. THE SPECIFIC WORD INTO WHICH THE DATA IS STORED IS DETERMINED BY THE SETTING OF THE FUNCTION SWITCHES (F).
4. TO ENTER A SWITCH SETTING, PRESS THE INTERRUPT REQUEST KEY ON THE 1131 CONSOLE. THE RESULTING LEVEL FOUR INTERRUPT WILL BE SERVICED BY THE MONITOR READ BIT SWITCH ROUTINE. THE SWITCHES WILL BE READ AND STORED BY THE INTERRUPT ROUTINE.

3.2.2 SINGLE PROGRAM OPERATION

1. LOADING - IF MORE THAN ONE TEST PROGRAM IS PLACED IN THE READER, EACH PROGRAM MUST BE FOLLOWED BY A BLANK CARD.
2. COMMUNICATION TO PROGRAM - SET SWITCHES 4-6 OFF AND 7 ON FOR ALL COMMUNICATION WITH THE PROGRAM. NO COMMUNICATION TO THE MONITOR IS REQUIRED.

3.2.3 OVERLAP OPERATION

1. LOADING - ALL PROGRAMS TO BE RUN IN OVERLAP MUST BE PLACED IN THE READER WITH NO BLANK CARDS BETWEEN THE PROGRAM. ALL PROGRAMS WILL BE LOADED INTO CORE STORAGE. A MAXIMUM OF 15 PROGRAMS MAY BE RUN IN OVERLAP.
2. HALT AFTER LOADING - IF THE HALT SWITCH (SWITCH 15) IS ON DURING LOADING OF A PROGRAM, THE HALT SWITCH IN THAT PROGRAM WILL BE SET. THUS AFTER THE PROGRAMS ARE LOADED, THEY MAY BE INDIVIDUALLY STARTED OR ALL STARTED TOGETHER BY FOLLOWING THE RESTART PROCEDURES OUTLINED IN SECTION 3.5.
3. COMMUNICATION TO PROGRAMS - TO COMMUNICATE TO AN INDIVIDUAL PROGRAM, ITS LOAD SEQUENCE NUMBER (PRINTED AS PART OF THE LOAD MESSAGE) MUST BE ENTERED IN SWITCHES 4-7. IF A CONTROL IS TO BE APPLICABLE TO ALL PROGRAMS, SWITCHES 4-7 MUST BE OFF.
4. EXAMPLE OF COMMUNICATION - IF THE DISC, CONSOLE KEYBOARD, AND 1132 FUNCTION TESTS ARE TO BE RUN IN OVERLAP AND THEY ARE LOADED IN THE ABOVE ORDER, THEIR LOAD SEQUENCE NUMBERS WOULD BE AS FOLLOWS.

TEST	NUMBER
DISC FT	1
CONS/KEYBOARD	2
1132 FT	3

IF SWITCH 15 WAS ON DURING LOADING OF ALL PROGRAMS, NO PROGRAMS WILL START UNTIL A RESTART COMMAND IS GIVEN.

ASSUME THE FOLLOWING CONTROL IS DESIRED, THE SWITCHES MUST BE SET AS SHOWN AND THE INTERRUPT REQUEST KEY PRESSED AFTER EACH SWITCH ENTRY.

CONTROL SWITCHES

SELECT ROUTINE 5 IN THE 1132 FT SET LOOP PROGRAM ON CONSOLE/KEYBOARD FT	4305
BYPASS ERROR MESSAGE PRINTOUT IN THE 1132 FT	0210
SET DISC FT TO PRINT ROUTINE START MESSAGE AND TO HALT ON ANY ERROR	0304
START ALL PROGRAMS AND SET TO LOOP ALL PROGRAMS	0142
	0090

F F	P P P P	D D D	D D D D
0 1	2 3 4 5 6 7	8 9 10	11 12 13 14 15
0 0	0 0 P P P P	X X X	X X X X X

8	RESTART
9	PRINT RTN ID
10	LOCK ON RTN
11	LOOP PROGRAM
12	LOOP ON ERROR
13	BYPASS ERROR PRINTOUT
14	HALT ON ERROR
15	HALT

3.2.4 PROGRAM CONTROL \_ FUNCTION 0

ALL FUNCTION 00 SWITCH ENTRIES REGARDLESS OF THE PROGRAM TO WHICH THEY REFER ARE SERVICED BY THE MONITOR. THUS, WITH THE EXCEPTION OF LOOP ON ERROR AND LOCK ON FUNCTION, WHICH ARE DEPENDENT ON SUPPORT ROUTINES IN THE INDIVIDUAL PROGRAMS, THE CONTROLS DISCUSSED HERE APPLY TO ALL MONITOR CONTROLLED PROGRAMS.

1. THE MONITOR PROVIDES THE FOLLOWING CONTROLS

- A. HALT - HALT THE PROGRAM AT THE COMPLETION OF THE PRESENT OPERATION. WHEN THE HALT SWITCH IS TURNED OFF, THE PROGRAM WILL CONTINUE FROM THE POINT AT WHICH IT WAS STOPPED.
- B. RESTART - REINITIALIZE THE PROGRAM AND RESTART. IF A ROUTINE BEEN PREVIOUSLY SELECTED, THE PROGRAM WILL BE RESTARTED AT THE BEGINNING OF THIS ROUTINE. IF NO ROUTINE IS SELECTED, THE PROGRAM WILL RESTART WITH ROUTINE ONE. THE PROGRAM HALT SWITCH IS NOT TURNED OFF BY A RESTART TO ALL PROGRAMS. THUS A HALTED PROGRAM WILL NOT RESTART UNTIL ITS HALT SWITCH IS TURNED OFF.
- C. PRINT ROUTINE START MESSAGE - BEFORE ANY ROUTINE IS STARTED, A MESSAGE MAY BE PRINTED IDENTIFYING THAT ROUTINE AND ITS STARTING ADDRESS.
- D. LOCK ON FUNCTION - MANY OF THE PROGRAMS ARE WRITTEN TO INCORPORATE A LOCK ON FUNCTION IN MOST ROUTINES. IN THESE PROGRAMS, LOCK ON FUNCTION WILL INHIBIT THE DATA ADVANCE AND LOOP ON THE MAJOR ROUTINE FUNCTION. SEE THE DOCUMENTATION OF EACH PROGRAM TO SEE EXACTLY HOW THIS FUNCTION IS IMPLEMENTED IN THAT PROGRAM.
- E. LOOP PROGRAM - WHEN A PROGRAM GOES TO END THIS CONTROL WILL AUTOMATICALLY RESTART IT FROM ROUTINE 1.
- F. LOOP ON ERROR - THE TEST DETECTING THE ERROR WILL BE LOOPED SO LONG AS THE ERROR PERSISTS. TO LOOP INTERMITTENT ERRORS, USE THE LOCK ON FUNCTION.
- G. BYPASS ERROR PRINTOUT
- H. HALT ON ERROR - THE PROGRAM WILL STOP AT WAIT 2. SINCE THIS WAIT IS FOLLOWED BY A BRANCH BACK TO THE WAIT, INTERRUPTS WILL NOT CAUSE THE PROGRAM TO CONTINUE. TO RECOVER FROM THIS HALT, TURN TO DISPLAY MODE, PRESS START ONCE THEN RETURN TO RUN AND PRESS START.

- 2. ALL OF THESE CONTROLS ARE ACCOMPLISHED BY A FUNCTION ZERO SWITCH ENTRY TO EITHER A SPECIFIC PROGRAM OR TO THE MONITOR. IF THE ENTRY IS TO BE TO THE MONITOR (SWITCHES 4-7 ALL OFF), THE CONTROL WILL APPLY TO ALL PROGRAMS. IF THE ENTRY IS TO A SPECIFIC PROGRAM THE CONTROL WILL APPLY TO THAT PROGRAM ONLY.

TO SPECIFY THE DESIRED CONTROL, SET THE SWITCHES AS FOLLOWS.

WHERE - P P P P IS 0000 IF THE CONTROL IS TO ALL PROGRAMS OR IS EQUAL TO THE PROGRAM SEQUENCE NUMBER IF THE ENTRY IS TO A SPECIFIC PROGRAM.

THE PROGRAM SEQUENCE NUMBER IS DETERMINED BY THE ORDER IN WHICH PROGRAMS ARE LOADED - THE FIRST PROGRAM IS NUMBER 1, THE SECOND NUMBER 2, ETC. THE BINARY EQUIVALENT OF THIS SEQUENCE NUMBER IS USED IN SWITCHES 4-7.

THUS THE SWITCH ENTRY

0588

WILL CAUSE THE FIFTH PROGRAM LOADED TO RESTART AND LOOP ON ANY DETECTED ERRORS. THIS ENTRY WILL ALSO TURN OFF THE LOCK ON ROUTINE, LOOP ON PROGRAM, BYPASS ERROR PRINTOUT, HALT ON ERROR, AND HALT CONTROLS SHOULD ANY OF THESE HAVE BEEN SET ON BY A PREVIOUS SWITCH ENTRY.

- 3. A CONTROL IS SET BY A SWITCH ENTRY HAVING THE SWITCH FOR THAT CONTROL ON. A CONTROL IS RESET BY A SWITCH ENTRY HAVING THE SWITCH FOR THAT CONTROL OFF. THUS, TO START AGAIN AFTER HAVING SET THE HALT SWITCH FOR A PROGRAM, A SWITCH ENTRY MUST BE MADE WITH SWITCH 15 OFF AND SWITCHES 0-7 SET AS THEY WERE WHEN THE HALT CONTROL WAS SET.
- 4. TO ENTER THE SWITCHES AFTER THEY HAVE BEEN SET, PRESS THE INTERRUPT REQUEST KEY ON THE 1131 CONSOLE. THE RESULTING LEVEL 4 INTERRUPT WILL BE SERVICED BY THE MONITOR READ BIT SWITCH ROUTINE. SINCE THE SWITCHES ARE READ IN INTERRUPT AND THE SETTINGS PLACED IN THE PROGRAM CONTROL TABLE IN INTERRUPT, THEY WILL GIVE IMMEDIATE CONTROL TO THE PROGRAM (SO LONG AS AN INTERRUPT IS NOT CONTINUALLY ON PREVENTING SERVICE OF THE CONSOLE INTERRUPT).

3.2.5 OPTION SELECTION - FUNCTIONS 1,2,3

FUNCTION SWITCH SETTINGS 01,10 AND 11 CAUSE THE DATA IN SWITCHES 8-15 TO BE STORED IN SW1, SW2, OR SW3 RESPECTIVELY IN THE PROGRAM CONTROL TABLE OF THE PROGRAM IDENTIFIED IN THE PROGRAM SWITCHES. THESE SWITCH ENTRIES ARE INTERROGATED BY EACH INDIVIDUAL TEST PROGRAM TO PROVIDE APPROPRIATE CONTROL FOR THAT PROGRAM.

- 1. ROUTINE SELECTION - FUNCTION 02 IS USED BY ALL PROGRAMS FOR ROUTINE SELECTION. IF THE STANDARD TEST CONTROL ROUTINES ARE USED BY THE PROGRAM, THE ROUTINE SELECTION WILL FUNCTION AS FOLLOWS.

- A. ANY ROUTINE IS SELECTED BY SETTING INTO SWITCHES 8-15 THE BINARY EQUIVALENT OF THE HEXIDECIMAL NUMBER OF THAT ROUTINE.
- B. ONCE A ROUTINE IS SELECTED, THE PROGRAM WILL LOOP IN THAT

ROUTINE UNTIL ANOTHER ROUTINE IS SELECTED.

- C. TO RETURN TO RUNNING ALL ROUTINES, THE SWITCHES SHOULD BE SET AS IF SELECTING ROUTINE ZERO. THE ROUTINE RUNNING AT THE TIME THIS ENTRY IS MADE WILL CONTINUE TO ITS NORMAL TERMINATION, THEN THE NEXT ROUTINE IN THE NORMAL SEQUENCE WILL BE RUN.

NOTE - THE ROUTINE NUMBERING STARTS WITH ROUTINE 1.

- D. IF AN INVALID ROUTINE NUMBER IS SET IN THE SWITCHES, THE CONTROL ROUTINE WILL RETURN TO RUNNING ALL ROUTINES, STARTING FROM THE FIRST ROUTINE.

- E. WHEN A NEW ROUTINE IS SELECTED, THE PROGRAM WILL GO TO THAT NEW ROUTINE IMMEDIATELY AFTER IT HAS SERVICED ALL PENDING INTERRUPTS.

- 2. OTHER TEST OPTIONS - WHILE THE USE OF FUNCTIONS 10 AND 11 HAS NOT BEEN STANDARDIZED, THESE FUNCTIONS HAVE MOST FREQUENTLY BEEN USED BY THE TEST PROGRAMS TO SPECIFY A FUNCTION WHICH IS TO BE PERFORMED REPEATEDLY AND TO ENTER NEW TEST DATA OR TEST PATTERNS.

3.2.6 SPECIAL MONITOR CONTROL

THREE SPECIAL CONTROL FEATURES ARE PROVIDED BY THE DIAGNOSTIC MONITOR THESE ARE PROGRAM STOP, ADDRESS STOP, AND PROGRAM DELAY.

- 1. PROGRAM STOP - THE PROGRAM STOP BUSHBUTTON SHOULD ALWAYS BE USED IN FAVOR OF THE IMMEDIATE STOP WHENEVER IT IS DESIRED TO TEMPORARILY STOP THE OPERATION OF THE PROGRAM. PRESSING PROGRAM STOP WILL CAUSE A LEVEL FIVE (5) INTERRUPT, WHICH IS SERVICED BY THE MONITOR HALT ROUTINE. THE PROGRAM WILL STOP AT A WAIT IN THE HALT ROUTINE AFTER ALL INTERRUPTS HAVE BEEN SERVICED. OPERATION OF THE PROGRAM CAN BE RESUMED BY EITHER PRESSING PROGRAM START OR BY PRESSING RESET THEN PROGRAM START. THIS SECOND START PROCEDURE ALLOWS THE CE TO USE LOAD AND DISP MODES (SET BY CONSOLE MODE SWITCH) TO LOOK AT AND/OR MODIFY ANY POSITION IN CORE BEFORE CONTINUING THE OPERATION OF THE PROGRAM. ALL CPU STATUS EXCEPT INDEX REGISTERS IS SAVED BY THE HALT ROUTINE AND RESTORED WHEN EITHER OF THE ABOVE START PROCEDURES IS USED.

- 2. ADDRESS STOP - THIS FEATURE WILL STOP THE PROGRAM ON ANY MAINLINE INSTRUCTION ADDRESS BEFORE IT IS EXECUTED. TO STOP AT A SELECTED ADDRESS -

- A. PRESS PROGRAM STOP
- B. SET THE BIT SWITCHES TO THE DESIRED STOP ADDRESS
- C. SET THE CONSOLE MODE SWITCH TO INT RUN.
- D. PRESS PROGRAM START - DO NOT PRESS RESET BEFORE PROGRAM START. INTERRUPT LEVEL FIVE MUST BE ON WHEN STARTING IN INTERRUPT RUN MODE FROM THE MONITOR HALT ROUTINE.

THE PROGRAM WILL STOP IN THE MONITOR HALT ROUTINE BEFORE THE INSTRUCTION AT THE SPECIFIED ADDRESS IS EXECUTED. TO RESTORE THE MACHINE STATUS TO THE NORMAL CONDITIONS BEFORE EXECUTION OF THE INSTRUCTION, SET THE CONSOLE MODE SWITCH TO SI, THEN STEP THRU THE PROGRAM UNTIL THE INTERRUPT LEVEL 5 IS TURNED OFF. THE NEXT INSTRUCTION TO BE EXECUTED WILL BE THE INSTRUCTION AT THE SELECTED STOP ADDRESS.

THE NEXT STOP ADDRESS MUST BE SET IN THE BIT SWITCHES ANY TIME THE PROGRAM HALTS AT A SELECTED ADDRESS OR HALTS AFTER PRESSING PROGRAM STOP.

- 3. PROGRAM DELAY - THIS FEATURE WILL PROVIDE A DELAY BETWEEN EACH INSTRUCTION EXECUTED IN MAINLINE, EFFECTIVELY SLOWING DOWN THE TOTAL OPERATION OF THE CPU. THIS FEATURE CAN BE USED TO SLOW DOWN THE OPERATION OF ANY I/O DEVICE BY DELAYING BETWEEN THE XIO COMMANDS.

THE BASE DELAY IS APPROXIMATELY 125 USEC. THIS CAN BE INCREASED IN INCREMENTS OF 25 USEC.

TO SET PROGRAM DELAY

- A. SET THE BIT SWITCHES TO 8020, THEN PRESS THE INTERRUPT REQUEST KEY THIS WILL DISABLE THE ADDRESS STOP FEATURE AND TRANSFER THE INTERRUPT RUN INTERRUPT TO THE PROGRAM DELAY ROUTINE.
- B. PRESS PROGRAM STOP.
- C. SET THE CONSOLE MODE SWITCH TO INT RUN.
- D. SET THE BIT SWITCHES FOR THE DESIRED DELAY. MINIMUM DELAY IS ALL SWITCHES OFF. THE TOTAL DELAY WILL BE APPROXIMATELY 125+25 WHERE D IS EQUAL TO THE DECIMAL EQUIVALENT OF THE BINARY SETTING OF THE SWITCHES.
- E. PRESS PROGRAM START - DO NOT PRESS RESET BEFORE PROGRAM START. INTERRUPT LEVEL 5 MUST BE ON WHEN STARTING IN INTERRUPT RUN MODE FROM THE MONITOR HALT ROUTINE.

3.3\*\*\* PROGRAM HALTS

3.3.1 NORMAL HALTS

HALT NO. (B REG)	DESCRIPTION	RESTART ACTION
3001	PROGRAM STOP OR ADDRESS STOP	PRESS START
3002	HALT ON ERROR	DISPLAY MODE-PRESS START. RUN MODE-PRESS START

\*\*

3.3.2 ERROR HALTS

HALT NO. (B REG)	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON FIRST CARD OF THE LOADER.	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F3	CARD 2 OF LOADER DID NOT LOAD	RELOAD
30F4	CAN NOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F5	READER CHECK WHEN LOADING MONITOR OR TEST PROGRAM	NPRO THEN PLACE CARDS RUN OUT IN FRONT OF REMAINING DECK AND PRESS. START.

30F6	MONITOR DID NOT LOAD	RELOAD
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START
30FA	CONSOLE PRINTER HANG UP - BUSY WILL NOT GO OFF	FIX THE CONSOLE PRINTER

\*\*\*\*\*

3.4\*\*\* PROGRAM TERMINATION

AT THE COMPLETION OF EACH PASS IN A TEST PROGRAM, THE PROGRAM WILL CALL ON THE MONITOR END ROUTINE. THIS ROUTINE WILL RESTART THE PROGRAM IF THE LOOP ON PROGRAM SWITCH IS ON. IF THE PROGRAM IS NOT TO BE LOOPED, A MESSAGE WILL PRINT IDENTIFYING THE PROGRAM AS TERMINATED. IF NOT RUNNING IN OVERLAP, THE MONITOR WILL BRANCH TO THE LOADER TO LOAD THE NEXT PROGRAM IN THE READER (UNLESS PROGRAM MUST BE EXECUTED).

IF A SWITCH ENTRY IS MADE REQUESTING LOADING OF A PROGRAM IN RELOAD MODE, THE PROGRAMS IN CORE WILL BE AUTOMATICALLY TERMINATED. IN THIS CASE NO END MESSAGE WILL BE PRINTED.

3.5\*\*\* PROGRAM RESTART

ANY PROGRAM MAY BE RESTARTED BY A FUNCTION 00 ENTRY TO THAT PROGRAM WITH SWITCH 8 ON. THIS CAN BE DONE AT ANY TIME WHEN THE PROGRAM IS RUNNING OR AFTER THE PROGRAM HAS TERMINATED.

A RESTART REQUEST TO THE MONITOR (SWITCHES 0-7 OFF AND SWITCH 8 ON) WILL RESTART ALL PROGRAMS WHICH ARE IN CORE. IF ANY PROGRAM HAS BEEN HALTED BY SETTING ITS HALT SWITCH, IT WILL REMAIN HALTED AFTER THIS RESTART (UNLESS FIRST START AFTER LOAD IN WHICH CASE ALL HALT SWITCHES ARE RESET).

IF THE MONITOR SHOULD HANG UP FOR ANY REASON (FOR EXAMPLE IF CONSOLE INTERRUPT FAILED TO RESET), IT CAN BE RESTARTED BY PRESSING IMMEDIATE STOP, RESET, AND PROGRAM START IN THAT ORDER.

3.6\*\*\* PROGRAM PATCHING

TO FACILITATE PROGRAM CORRECTION, MODIFICATION, AND EDITING, THE RELOCATING LOADER IS ABLE TO LOAD CARDS WHICH ARE KEYPUNCHED WITH HEX DATA. THESE CARDS ARE IDENTIFIED BY A 12 PUNCH IN COLUMN 1. COLUMNS 2-5 MUST CONTAIN, IN HEX, THE CORE ADDRESS INTO WHICH THE DATA ON THE CARD IS TO BE STORED. THE REMAINDER OF THE CARD IS USED FOR DATA, WHICH IS AGAIN IN HEX. EACH DATA WORD ON THE CARD MUST BE PRECEDED BY EITHER A BLANK OR AN R. THE BLANK WILL CAUSE THE LOADER TO INTERPRET THE DATA WORD AS AN INSTRUCTION OR CONSTANT. THE R WILL CAUSE THE LOADER TO INTERPRET THE DATA WORD AS AN ADDRESS WHICH MUST BE RELOCATED IF THE PROGRAM OF WHICH THIS PATCH IS A PART IS RELOCATED. IF NO DATA WORDS ARE INCLUDED ON THE CARD FOLLOWING THE ADDRESS, THE LOADER WILL INTERPRET THIS CARD AS A BRANCH CARD AND WILL BRANCH TO THE ADDRESS SPECIFIED IN COLUMNS 2-5. TWO BLANK COLUMNS IN A ROW WILL TERMINATE LOADING OF THE CARD. AFTER THESE TWO BLANKS, ANY IDENTIFYING INFORMATION MAY BE PLACED ON THE CARD. PATCH CARDS MUST BE PLACED JUST BEFORE THE END CARD IN THE PROGRAM TO BE PATCHED.

SOME EXAMPLES FOLLOW

A. +0645 C400R0857 D035

THIS WILL LOAD THREE WORDS STARTING AT ADDRESS 0645 PLUS THE RELOCATION FACTOR. THE FIRST WORD WILL BE C400, THE SECOND WORD WILL BE D035. IF THIS PATCH CARD IS PLACED IN THE PROGRAM DECK BEFORE THE END CARD OF THE FIRST PROGRAM LOAD, THE RELOCATION FACTOR WILL BE ZERO.

B. +0739 BR TO 0739

THIS CARD WILL CAUSE A BRANCH TO 0739 PLUS THE RELOCATION FACTOR. CARDS FOLLOWING THIS CARD WILL NOT BE LOADED UNLESS THE PROGRAM AT 0739 LOADS THEM OR RETURNS TO THE LOADER.

4. PRINTOUTS

4.1\*\*\* STATUS MESSAGES

A0000 NUM PID ADRS RELF LD  
XXXX XXXX XXXX XXXX

THIS MESSAGE IS PRINTED FOLLOWING THE LOADING OF ANY PROGRAM (EXCEPT MONITOR). THE MESSAGE GIVES THE LOAD SEQUENCE NUMBER, THE PROGRAM ID, THE ADDRESS INTO WHICH THE PROGRAM WAS LOADED, AND THE RELOCATION FACTOR.

A0001 SWS PID  
XXXX XXXX

THIS MESSAGE IS PRINTED EACH TIME A VALID SWITCH ENTRY IS READ BY THE MONITOR. THE MESSAGE CONTAINS THE SWITCH SETTING READ TOGETHER WITH THE PROGRAM ID OF THE PROGRAM INTO WHICH THE CONTENTS OF SWITCHES 8-15 WERE STORED. IF THE SWITCH ENTRY CALLED FOR HALT OF ANY PROGRAM THE WORD HALT WILL FOLLOW THE MESSAGE.

4.2\*\*\* ERROR MESSAGES

E0001 SWS INVLD  
XXXX

THE SETTING OF SWITCHES 5-7 DID NOT EQUAL THE LOAD SEQUENCE NUMBER OF ANY PROGRAM IN CORE.

E0003 OVR CORE

THE PROGRAM WHICH THE LOADER WAS ATTEMPTING TO LOAD EXCEEDED AVAILABLE CORE. LOADING WAS TERMINATED.

E0004 CKSUM

A CHECK SUM ERROR WAS DETECTED WHILE LOADING A TEST PROGRAM. THIS ERROR OCCURS UNDER ANY OF THE FOLLOWING CONDITIONS.

1. A CARD IS MISSING OR IS OUT OF SEQUENCE.
2. THERE IS AN EXTRA CARD IN THE DECK.
3. THE PUNCHED INFORMATION ON THE CARD IS NOT CORRECT.
4. DATA WAS LOST OR PICKED UP DUE TO A MACHINE MALFUNCTION.

5. DUE TO A CPU MALFUNCTION, THE CHECK SUM WAS NOT CORRECTLY CALCULATED.

WHEN THIS ERROR OCCURS ATTEMPT TO RELOAD THE PROGRAM.

E0005

OOON XXXX

THIS ERROR WILL OCCUR IF AN INTERRUPT OCCURS, BUT THE ILSW WAS NOT CORRECT. N IS THE INTERRUPT LEVEL AND XXXX IS THE ILSW. THIS PRINTOUT WILL ONLY OCCUR IF THE INTERRUPT IS RESET BY A BOSCC. NO ATTEMPT IS MADE BY THE ERROR ROUTINE TO RESET THE REQUEST BIT.

## 5. COMMENTS

## 5.1\*\*\* INTRODUCTION TO DIAGNOSTIC MONITOR OPERATION

## 5.1.1 WHAT IS THE DIAGNOSTIC MONITOR

THE DIAGNOSTIC MONITOR IS A CONTROL PROGRAM DESIGNED TO PROVIDE OVERLAP CAPABILITY AND TO HANDLE MANY PROGRAM FUNCTIONS WHICH ARE COMMON TO ALL OF THE TEST PROGRAMS.

THE MONITOR IS MADE UP OF THE FOLLOWING ROUTINES

1. RELOCATING LOADER INTERFACE
2. SUPERVISOR ROUTINE
3. INTERRUPT ROUTINE
4. BIT SWITCH ROUTINE
5. LOG ROUTINE
6. HALT ROUTINE
7. END ROUTINE

## 5.1.2 WHY IS A MONITOR NECESSARY TO ENABLE PROGRAMS TO RUN IN OVERLAP

THE MONITOR PROVIDES THE FOLLOWING FUNCTIONS WHICH ARE NECESSARY FOR OVERLAP OPERATION

1. RELOCATING LOADER - THIS LOADER ENABLES SEVERAL PROGRAMS WHICH ARE ASSEMBLED TO USE THE SAME CORE STORAGE LOCATIONS TO BE LOADED TOGETHER IN CORE. THIS IS ACCOMPLISHED BY LOADING THE FIRST PROGRAM IN THE CORE STORAGE FOR WHICH IT WAS ASSEMBLED, THEN LOADING THE NEXT PROGRAM STARTING AT THE FIRST AVAILABLE STORE LOCATION AFTER THE PREVIOUSLY LOADED PROGRAM. ALL ADDRESSES REFERENCED IN THE PROGRAM MUST ALSO BE MODIFIED BY A RELOCATION FACTOR, WHICH IS EQUAL TO THE DIFFERENCE BETWEEN THE ADDRESS AT WHICH THE PROGRAM IS ACTUALLY LOADED AND THE ADDRESS AT WHICH IT WAS ASSEMBLED.
2. INTERRUPT HANDLING -WHERE THERE ARE SEVERAL DEVICES ON ONE INTERRUPT LEVEL (LEVEL 4) OR WHERE TWO PROGRAMS ARE SHARING A SINGLE DEVICE (CONSOLE PRINTER IS USED BY ALL PROGRAMS), A CENTRAL INTERRUPT CONTROL IS REQUIRED TO IDENTIFY EACH INTERRUPT AND TO TRANSFER CONTROL TO THE APPROPRIATE INTERRUPT ROUTINE. THIS ROUTINE ALSO STORES THE CPU STATUS (ACCUMULATOR, EXTENSION, CARRY AND OVERFLOW, AND XR3) AND RESTORES STATUS UPON BRANCHING OUT OF INTERRUPT.
3. TIME SHARING -THIS IS THE BASIC OVERLAP FUNCTION PERFORMED BY THE

MONITOR. THIS FUNCTION IS ACCOMPLISHED BY THE MONITOR SUPERVISOR ROUTINE. THE SUPERVISOR CONTINUALLY CHECKS EACH PROGRAM LOADED FOR A REQUEST TO RUN A ROUTINE IN THAT PROGRAM. ONCE SUCH A REQUEST IS FOUND, THE SUPERVISOR WILL BRANCH TO THAT ROUTINE. AFTER THE ROUTINE IS RUN, THE TEST PROGRAM MUST BRANCH BACK TO THE SUPERVISOR. ALL OTHER PROGRAMS ARE THEN CHECKED BEFORE ANOTHER ROUTINE IN THE FIRST PROGRAM CAN BE RUN.

4. ONE INTERFACE WITH INPUT AND OUTPUT DEVICE FOR ALL PROGRAMS -EACH PROGRAM HAS NEED FOR CONTROL INPUT FROM THE BIT SWITCHES AND MESSAGE OUTPUT TO THE CONSOLE PRINTER. SINCE ONLY ONE PROGRAM CAN USE THESE DEVICES AT A TIME, THE MONITOR SERVES TO DIRECT TRAFFIC, SENDING ANY BIT SWITCH ENTRY TO THE APPROPRIATE PROGRAM AND OUTPUTTING EACH PROGRAM MESSAGE IN ITS TURN.

## 5.1.3 WHAT COMMON PROGRAM FUNCTIONS ARE HANDLED BY THE DIAGNOSTIC MONITOR

1. MESSAGE OUTPUT - COMMUNICATION TO THE CONSOLE PRINTER IS SERIAL AND MUST BE IN ROTATE AND TILT CODE. ANY MESSAGE TO THE PRINTER MUST, THEREFORE, FIRST BE CONVERTED, THEN TRANSFERRED TO THE PRINTER A CHARACTER AT A TIME. PUTTING THIS FACILITY IN THE MONITOR ALLOWS EACH TEST PROGRAM TO OUTPUT ITS MESSAGES BY SIMPLY SETTING UP A TABLE WHICH DEFINES THE MESSAGE THEN BRANCHING TO THE APPROPRIATE (ERROR OR STATUS) MONITOR LOG ROUTINE.
2. BIT SWITCH ENTRY-TO SIMPLIFY THE USE OF BIT SWITCHES BY EACH PROGRAM, THE MONITOR ON REQUEST READS THE BIT SWITCHES AND PLACES THE READING IN CORE STORAGE IN THE APPROPRIATE PROGRAM. THUS THE PROGRAM HAS AT ALL TIMES THE LATEST SWITCH READING. ALSO TO INCREASE THE FLEXIBILITY AND THE AMOUNT OF INFORMATION THAT CAN BE CONVEYED BY THE BIT SWITCHES, THE SWITCHES ARE DIVIDED INTO FOUR FUNCTIONS THE SWITCH SETTING CORRESPONDING TO EACH FUNCTION IS SET IN A SEPARATE CORE POSITION IN THE PROGRAM.
3. PRINT LOAD MESSAGE - THIS MESSAGE IS PRINTED AFTER A PROGRAM IS SUCCESSFULLY LOADED. IT IDENTIFIES THE PROGRAM, THE ADDRESS AT WHICH IT WAS LOADED AND GIVES THE RELOCATION FACTOR FOR THAT PROGRAM.
4. CONTROL PROGRAM END - WHEN A PROGRAM HAS COMPLETED ONE PASS THE MONITOR DETERMINES WHETHER OR NOT IT IS TO BE LOOPED. IF NOT, IT PRINTS AN END MESSAGE AND TERMINATES THE PROGRAM.
5. RESTART - THE MONITOR WILL REINITIALIZE AND RESTART ANY SPECIFIED PROGRAMS OR ALL PROGRAMS. A MESSAGE IS PRINTED IDENTIFYING THE PROGRAM RESTARTED.
6. HALT -THE MONITOR WILL HALT EXECUTION OF ANY SPECIFIED PROGRAM OR ALL PROGRAMS. A MESSAGE IS PRINTED IDENTIFYING THE PROGRAM HALTED.
7. ERROR CONTROL - HALT ON ERROR, LOOP ON ERROR, AND BYPASS ERROR PRINTOUT ARE ALL HANDLED BY THE MONITOR.
  - A. HALT ON ERROR WILL HALT AFTER THE ERROR MESSAGE IS PRINTED. IF HALT ON ERROR IS ON, BYPASS ERROR PRINTOUT WILL BE IGNORED THE PROGRAM WILL HALT AT A NON INTERRUPTABLE WAIT IN THE MONITOR.
  - B. LOOP ON ERROR  
SO LONG AS THE ERROR PERSISTS.
  - C. BYPASS ERROR PRINTOUTS ALLOWS THE PROGRAM TO LOOP ON AN ERROR CONDITION WITHOUT PRINTING THE ERROR MESSAGE.
8. ROUTINE NUMBER LOG - UNDER SWITCH CONTROL THE MONITOR WILL LOG THE ROUTINE NUMBER AND ADDRESS OF EACH TEST ROUTINE BEFORE THAT



ROUTINE IS RUN.

9. PROGRAM STOP - THE CONSOLE PROGRAM STOP PUSHBUTTON IS SUPPORTED BY A MONITOR ROUTINE WHICH WILL STOP ALL PROGRAMS WITHOUT LOSING ANY INTERRUPTS. THIS ROUTINE ALSO SAVES THE C.P.U. STATUS AND PROVIDES A RE-ENTRY FROM LOCATION ZERO. THIS FEATURE ALLOWS THE C.E. TO MANUALLY CHECK OR MODIFY ANY CORE LOCATION AFTER THE STOP, THEN CONVENIENTLY RESTART BY PRESSING RESET AND PROGRAM START.
10. ADDRESS STOP - THIS ROUTINE USES THE INTERRUPT RUN MODE. IN THIS MODE AN INTERRUPT OCCURS AFTER EACH MAINLINE INSTRUCTION. THE DESIRED STOP ADDRESS, SET BY THE BIT SWITCHES, IS THEN COMPARED TO THE IAR ADDRESS. THE ROUTINE HALTS ALL PROGRAMS WHEN THESE ARE EQUAL.
11. PROGRAM DELAY - THIS ROUTINE ALSO USES THE INTERRUPT RUN MODE. IN THIS MODE IT PROVIDES A MINIMUM DELAY OF 125 USECS BETWEEN THE EXECUTION OF EACH MAINLINE INSTRUCTION. THIS DELAY CAN BE INCREASED IN INCREMENTS OF 25 USECS BY BIT SWITCH CONTROL.

5.1.4. WHAT IS THE ADVANTAGE OF HAVING THESE COMMON FUNCTIONS IN A MONITOR

EACH PROGRAM IS SIMPLER AND IS FORCED TO FOLLOW A STANDARD ORGANIZATION. EACH PROGRAM IS BY ITSELF SHORTER - THAT IS OF GREATEST IMPORTANCE WHEN RUNNING IN OVERLAP SINCE IT ALLOWS MORE PROGRAMS TO BE LOADED IN THE SAME CORE.

5.1.5 HOW DOES THE MONITOR CONTROL A PROGRAM

NO TEST ROUTINES ARE RUN UNTIL A REQUEST FOR CONTROL IS MADE TO THE MONITOR BY THE TEST PROGRAM. THE MONITOR SUPERVISOR ROUTINE WILL THEN DETERMINE WHEN CONTROL IS TO BE TRANSFERRED, AND WILL TRANSFER CONTROL TO THE TEST ROUTINE. AT THE END OF THE ROUTINE OR WHENEVER WAITING FOR AN INTERRUPT, THE TEST PROGRAM MUST RETURN CONTROL TO THE MONITOR SUPERVISOR.

5.1.6 HOW DOES A TEST PROGRAM REQUEST CONTROL FROM THE MONITOR

CONTROL WILL BE TRANSFERRED TO THE TEST PROGRAM IF

1. AN INTERRUPT IS RECEIVED WHICH IS TO BE SERVICED BY THAT PROGRAM.
2. THE MONITOR SUPERVISOR FINDS AN ADDRESS IN THE MAIN LINE SEQUENCE CONTROL FIELD (MLSCF) FOR THAT PROGRAM.

THUS, BEFORE A PROGRAM GIVES UP CONTROL TO THE MONITOR, IT MUST PLACE IN ITS MLSCF AN ADDRESS TO WHICH THE MONITOR IS TO RETURN, OR IT MUST INITIATE AN OPERATION WHICH WILL RESULT IN AN INTERRUPT.

5.1.7 WHAT IS THE MAIN LINE SEQUENCE CONTROL FIELD (MLSCF)

THE MLSCF IS PART OF THE PROGRAM CONTROL TABLE WHICH IS THE MAJOR INTERFACE BETWEEN THE MONITOR AND THE TEST PROGRAM. EACH PROGRAM HAS ITS OWN PROGRAM CONTROL TABLE WHICH IS LOCATED AT THE BEGINNING PROGRAM. IT LOOKS LIKE THIS

* * PROGRAM CONTROL TABLE *			
PID	DC	/OXXX	PROGRAM ID
RID	DC	0	ROUTINE ID
RAD	DC	0	ROUTINE ADDR
SW0	DC	0	PROGRAM CONTROL
SW1	DC	0	ROUTINE SELECTION
SW2	DC	0	
SW3	DC	0	

	DC	STRT	LOOP PROGRAM ADDRESS
	DC	STRT	RESTART ADDRESS
MLSCF	DC	*-*	ENTRY ONE
	DC	*-*	ENTRY TWO
	DC	/FFFF	TERMINATOR

THE MONITOR OBTAINS FROM THIS TABLE THE PROGRAM ID, ROUTINE NUMBER AND ROUTINE ADDRESS WHEN PRINTING ERROR AND STATUS MESSAGES. IT FINDS THE RESTART ADDRESS HERE ALSO. WHEN BIT SWITCH DATA IS ENTERED FOR A PROGRAM, THE MONITOR WILL PLACE THE DATA IN SWITCHES 8 THROUGH 15 INTO ONE OF FOUR SWITCH STORAGE WORDS IN THIS TABLE. THE STORAGE WORD USED WILL DEPEND ON THE SETTING OF SWITCHES 0 AND 1 (FUNCTION THE MAIN LINE SEQUENCE CONTROL FIELD (MLSCF) ALSO CONTAINED IN THIS TABLE IS THE HEART OF THE MONITOR CONTROL. ALL COMMUNICATION BETWEEN THE TEST PROGRAM AND THE MONITOR SUPERVISOR IS THROUGH THE MLSCF. THE SUPERVISOR SEQUENTIALLY SCANS THE MLSCF OF EACH TEST PROGRAM. WHEN AN ADDRESS IS FOUND IN ANY MLSCF THE SUPERVISOR WILL TRANSFER TO THAT ADDRESS.

5.1.8 ONCE A TEST PROGRAM HAS OBTAINED CONTROL TO THE MONITOR

TO RETURN CONTROL TO THE MONITOR SUPERVISOR, THE TEST ROUTINE MUST EXECUTE THE FOLLOWING INSTRUCTION.

BSI I START

HOWEVER, BEFORE TRANSFERRING CONTROL TO THE MONITOR A RETURN TO THE TEST IN THE MLSCF.

5.1.9 HOW DOES A TEST PROGRAM PRINT A MESSAGE

ALL MESSAGES ARE PRINTED BY THE MONITOR LOG ROUTINE. COMMUNICATION WITH THIS ROUTINE IS THROUGH A CALLING SEQUENCE WHICH TRANSFERS TO THE LOG ROUTINE THE NECESSARY MESSAGE DATA.

TWO TYPES OF MESSAGES ARE PRINTED, STATUS MESSAGES AND ERROR MESSAGES. A STATUS MESSAGE IS IDENTIFIED BY AN A AS THE FIRST CHARACTER IN THE MESSAGE AND IS PRINTED IN BLACK. AN ERROR MESSAGE IS IDENTIFIED BY AN E AND IS PRINTED IN RED.

5.2\*\*\* ROUTINE DESCRIPTIONS

5.2.1 THE RELOCATING LOADER

THE DIAGNOSTIC MONITOR AND ALL TEST PROGRAMS ARE LOADED BY A RELOCATING LOADER. TWO VERSIONS OF THIS LOADER ARE AVAILABLE--1442, AND PAPER TAPE. EACH OF THESE LOADERS IS INITIALLY LOADED IN CORE LOCATIONS 0-160. THE RESIDENT PORTION IS IN LOCATIONS 140-160.

1. FORMATS

THESE LOADERS WILL LOAD ABSOLUTE PROGRAMS, RELOCATABLE PROGRAMS, OR HEX PATCH CARDS. THE MONITOR AND ALL NONOVERLAP PROGRAMS ARE ASSEMBLED AND LOADED IN ABSOLUTE FORMAT. ANY OVERLAP PROGRAM IS NOT ORG'D. AT 1500 MUST BE ASSEMBLED AND LOADED IN ABSOLUTE.

2. RELOCATION - THE RELOCATING LOADER ASSUMES ALL PROGRAMS WHICH ARE TO BE RELOCATED ARE ASSEMBLED AT A STARTING ADDRESS OF 05DC (1500 DECIMAL). THE FIRST PROGRAM LOADED WILL BE STORED, STARTING FROM THIS ADDRESS. ADDITIONAL PROGRAMS LOADED WILL BE RELOCATED STARTING FROM THE FIRST EVEN ADDRESS AFTER THE LAST ADDRESS IN THE PREVIOUSLY LOADED PROGRAM. THUS THE RELOCATION FACTOR IS EQUAL TO THE ADDRESS AT WHICH A PROGRAM IS LOADED MINUS 05DC (1500 DECIMAL). THIS FACTOR, TOGETHER WITH THE LOAD ADDRESS AND THE PROGRAM ID, IS PRINTED AFTER EACH PROGRAM IS LOADED.

3. CHECK SUM - ALL DATA LOADED BY THE RELOCATING LOADER IS CHECK SUMMED. IF A CHECK SUM ERROR IS DETECTED AN ERROR MESSAGE WILL BE PRINTED AND LOADING WILL BE TERMINATED. THIS CHECK SUM TAKES INTO CONSIDERATION BOTH THE CORRECTNESS OF THE DATA AND THE SEQUENCE IN WHICH THE CARDS ARE LOADED. THUS EXTRA OR MISSING CARDS WILL BE DETECTED BY THE CHECK SUM. THE CHECK WILL BE BYPASSED ON ANY CARD WHICH HAS A CHECK SUM OF ZERO. THE ADVANCE OF THE CARD COUNT WILL NOT BE BYPASSED. THUS, IF A CARD WITHIN A PROGRAM DECK MUST BE MODIFIED, IT WILL STILL LOAD IF THE CHECK SUM WORD (BITS 2-9 OF CARD COLUMN 2 AND 12-5 OF CARD COLUMN 3) IS MADE EQUAL TO ZERO. IF ADDITIONAL CARDS ARE ADDED TO THE DECK, THESE ALSO WILL LOAD IF THEIR CHECK SUM WORD IS EQUAL TO ZERO. HOWEVER, SINCE THE CARD COUNT IS ADVANCED, ANY CARDS HAVING NON-ZERO CHECK SUMS WHICH FOLLOW CARDS ADDED TO THE DECK WILL NOT LOAD. THEREFORE, ANY CARDS ADDED SHOULD BE PLACED AT THE END OF THE DECK - JUST BEFORE THE END CARD AND THE CHECK SUM OF THE END CARD CHANGED TO ZERO. TO MAKE CHANGES IN PROGRAMS WITHOUT CONCERN FOR CHECK SUM, SEE PROGRAM PATCHING (SECTION 3.6).
4. OVER CORE CHECK - IF DURING LOADING ALL OF CORE BECOMES FILLED BEFORE ALL PROGRAMS ARE LOADED, THE LOADING PROCESS WILL TERMINATE AND A MESSAGE WILL BE PRINTED (OVER CORE).

#### 5.2.2 SUPERVISOR ROUTINE

1. THE SUPERVISOR SEQUENTIALLY SCANS THE MLSCF OF EACH PROGRAM - INCLUDING THE MONITOR. IF AN MLSCF WORD IS FOUND TO BE POSITIVE AND NON ZERO, THE SUPERVISOR WILL ASSUME THAT THE WORD CONTAINS AN ENTRY ADDRESS AND WILL EXECUTE A BRANCH TO THAT ADDRESS. THE ADDRESS OF THE MLSCF IS SAVED BY THE SUPERVISOR. THIS ADDRESS IS USED BY THE MONITOR TO IDENTIFY THE TEST PROGRAM WHEN THAT PROGRAM TRANSFERS BACK TO THE MONITOR WITH A LOG CALL, AN END CALL, OR A RETURN TO START.
2. ROUTINE START LOG - WHEN THE PROGRAM RETURNS CONTROL TO THE MONITOR (BSI 1 START), THE SUPERVISOR WILL CHECK THE ROUTINE START SWITCH (RTNSW). THIS SWITCH SHOULD BE SET IN THE TEST PROGRAM CONTROL ROUTINE BEFORE STARTING A NEW TEST ROUTINE. IF THIS SWITCH IS ON (NON ZERO) AND ROUTINE START LOG HAS BEEN REQUESTED (SWITCH 9 FUNCTION 0), THE SUPERVISOR WILL PRINT A MESSAGE IDENTIFYING THE PROGRAM, THE ROUTINE NUMBER AND THE ROUTINE START ADDRESS.
3. RESTART AND HALT - ARE ALSO FUNCTIONS OF THE SUPERVISOR. THE HALT BIT FOR A PROGRAM IS ONE, THE SUPERVISOR WILL BYPASS SCANNING OF THE MLSCF OF THAT PROGRAM. IF THE MONITOR HALT BIT IS A ONE, THE SUPERVISOR WILL SCAN ONLY THE MONITOR MLSCF - THUS STOPPING ALL TEST PROGRAMS.
4. EXECUTE PAUSE - THE SUPERVISOR WILL ALSO STOP PROGRAM EXECUTION WHEN PRINTING ANY MESSAGE OR WHEN LOADING A PROGRAM.

#### 5.2.3 INTERRUPT ROUTINE

1. ALL INTERRUPTS ARE HANDLED BY THE MONITOR INTERRUPT ROUTINE. THIS ROUTINE SAVES THE STATUS OF THE CPU - EXCEPT INDEX REGISTERS 1 AND 2, TRANSFERS TO THE TEST PROGRAM INTERRUPT SERVICE ROUTINE, THEN RESTORES THE CPU STATUS BEFORE BRANCHING OUT OF INTERRUPT.
2. A TRANSFER VECTOR IS USED TO SPECIFY THE ADDRESS OF THE TEST PROGRAM INTERRUPT ROUTINE. THIS ADDRESS MUST BE SET IN THE TRANSFER VECTOR BY THE TEST PROGRAM.
3. INVALID INTERRUPTS ARE HANDLED BY THE INTERRUPT ERROR ROUTINE. AN INVALID INTERRUPT IS AN INTERRUPT FOR WHICH AN ADDRESS HAS NOT BEEN SET IN THE TRANSFER VECTOR. THE INTERRUPT ERROR ROUTINE WILL SET UP AN ERROR MESSAGE CALL AND WILL THEN TRY TO BRANCH OUT OF

THE INTERRUPT. IF THE BRANCH OUT TURNS OFF THE INTERRUPT, AN ERROR MESSAGE WILL BE PRINTED WHICH IDENTIFIES THE INTERRUPT LEVEL AND THE ILSW. IF THE INTERRUPT FAILS TO TURN OFF, THE PROGRAM WILL STOP AT THE INTERRUPT ERROR WAIT. PRESSING IMMEDIATE STOP, RESET AND PROGRAM START IN THAT ORDER WILL RESET THE INTERRUPT AND ALLOW THE ERROR MESSAGE TO BE PRINTED.

4. IF A VALID INTERRUPT WILL NOT RESET, THE PROGRAM WILL LOOP IN THE INTERRUPT ROUTINE. TO RECOVER, PRESS IMMEDIATE STOP, RESET AND PROGRAM START - IN THAT ORDER.

#### 5.2.4 BIT SWITCH ROUTINE

1. THE BIT SWITCHES ARE READ IN INTERRUPT WHEN THE INTERRUPT REQUEST KEY IS PRESSED. UNLESS EXECUTING THE KEYBOARD TEST, THE POSITION OF THE CONSOLE/KEYBOARD SWITCH IS IGNORED. WHEN EXECUTING THE KEYBOARD TEST THE BIT SWITCH IS READ ONLY IF THE CONSOLE/KEYBOARD SWITCH IS IN CONSOLE POSITION.
2. THE BIT SWITCH ROUTINE STORES THE CONTENTS OF SWITCHES 8 THROUGH 15 IN THE PROGRAM CONTROL TABLE OF THE PROGRAM IDENTIFIED BY SWITCHES 4-7. IF SWITCHES 4-7 ARE ALL OFF, THE BIT SWITCH READING WILL BE STORED IN THE MONITOR CONTROL TABLE.
3. THE PROGRAM IDENTIFICATION USED IN SWITCHES 4-7 IS A BINARY NUMBER DETERMINED BY THE ORDER IN WHICH THE PROGRAMS WERE LOADED. THE FIRST PROGRAM LOADED IS PROGRAM NO. 1 (SWITCH 7 ONLY), THE THIRD IS PROGRAM NO. 3 (SWITCH 6 AND 7), ETC. THUS IF ONLY ONE PROGRAM IS LOADED SWITCH 7 IS ALWAYS USED TO IDENTIFY THAT PROGRAM.
4. THE HALT SWITCH (SWITCH 15 FUNCTION 0) FOR ALL PROGRAMS IS INTERROGATED BY THE BIT SWITCH ROUTINE. IF THIS SWITCH IS ON THE ROUTINE WILL SET THE HALT BIT FOR THE PROGRAM IDENTIFIED BY SWITCHES 4-7. A MESSAGE WILL BE PRINTED SPECIFYING THAT THE PROGRAM HAS BEEN HALTED.
5. INVALID SWITCH SETTINGS ARE NOT STORED AND A MESSAGE IS PRINTED IDENTIFYING THE SWITCH SETTING AS INVALID. THE SWITCH SETTING IS INVALID IF THERE IS NO PROGRAM IN CORE CORRESPONDING TO THE PROGRAM IDENTIFICATION SET IN SWITCHES 5-7.

#### 5.2.5 MESSAGE LOG ROUTINE

1. THE LOG ROUTINE WILL PRINT EITHER STATUS MESSAGES OR ERROR MESSAGE DEPENDING ON THE ENTRY TO THE ROUTINE. ERROR MESSAGES ARE DISTINGUISHED FROM STATUS MESSAGES BY AN E IN THE MESSAGE ID WHILE THE STATUS MESSAGE CONTAINS AN A. ALSO THE ERROR MESSAGE IS PRINTED IN RED, THE STATUS MESSAGE IN BLACK.
2. MESSAGE FORMAT - THE MESSAGE FORMAT IS THE SAME FOR EITHER STATUS OR ERROR MESSAGES.

THE FIRST THREE WORDS ARE STANDARD MESSAGE IDENTIFICATION WORDS WHICH ARE PLACED IN THE MESSAGE BY THE LOG ROUTINE.

APPNN OORR AAAA  
OR  
EPPNN OORR AAAA

WHERE - A OR E IDENTIFIES THE MESSAGE AS EITHER A STATUS OR ERROR MESSAGE RESPECTIVELY.

PP IS THE LAST TWO DIGITS OF THE PROGRAM ID NUMBER. THIS IDENTIFIES THE PROGRAM TO WHICH THE MESSAGE REFERS.

NN IS THE MESSAGE NUMBER IN THE PROGRAM IDENTIFIED BY PP.

RR IS THE NUMBER OF THE ROUTINE IN THE PROGRAM IDENTIFIED BY PP WHICH WAS EXECUTING AT THE TIME CALL ON LOG WAS MADE.

AAAA IS THE ADDRESS IN CORE OF THE ROUTINE IDENTIFIED BY RR.

NOTE - FOR MONITOR MESSAGE THE ROUTINE NUMBER AND ADDRESS ARE NOT PRINTED.

3. AN ALPHA MESSAGE CAN NEXT BE PRINTED. THIS MESSAGE IS USED EITHER TO HEAD THE MESSAGE DATA (WHICH WILL BE PRINTED ON THE NEXT LINE) OR TO COMMUNICATE MACHINE STATUS AND/OR ERROR DESCRIPTION TO THE CE. IF DATA IS TO BE PRINTED FOLLOWING THE ALPHA MESSAGE A CARRIAGE RETURN AND TAB WILL BE CODED INTO THE MESSAGE FOLLOWING THE ALPHA.
4. DATA IS PRINTED IN EITHER NEXIDECIMAL OR DECIMAL. IF NO ALPHA IS INCLUDED IN THE MESSAGE, THE DATA WILL BE PRINTED ON THE SAME LINE AS THE THREE WORDS OF MESSAGE IDENTIFICATION. IF ALPHA IS INCLUDED IN THE MESSAGE, THE ALPHA PHRASES WILL BE PRINTED, THEN THE DATA WILL BE PRINTED DIRECTLY BELOW THE ALPHA.
5. DECIMAL DATA WILL ALWAYS BE 5 DIGITS WHILE HEX IS 4. IF THE DECIMAL DATA IS NEGATIVE, A MINUS (-) WILL BE PRINTED IN FRONT OF THE DECIMAL WORD.
6. MULTIPLE LINE MESSAGES WILL INCLUDE THE MESSAGE ID ON THE FIRST LINE ONLY.
7. ERROR CONTROL IS PROVIDED BY THE ERROR LOG ROUTINE. HALT ON ERROR BYPASS ERROR PRINTOUT, OR LOOP ON ERROR ARE ALL FUNCTIONS OF THIS ROUTINE.
8. IF THE LOG ROUTINE IS BUSY, THE LOG BUSY SUBROUTINE WILL SET THE ADDRESS OF THE USER LOG CALL IN THE MLSCF OF THE USER PROGRAM. THUS THE MONITOR WILL FORCE A LOOP ON THE LOG CALL UNTIL THE LOG ROUTINE IS NO LONGER BUSY.

#### 5.2.6 END ROUTINE

THIS ROUTINE IS ENTERED ON THE COMPLETION OF EACH COMPLETE PASS BY A TEST PROGRAM. THE ROUTINE WILL RESTART THE PROGRAM CAUSING IT TO LOOP IF THE LOOP PROGRAM SWITCH FOR THE MONITOR OR FOR THE TEST PROGRAM IS ON.

IF THE PROGRAM IS NOT LOOPED, THE MONITOR WILL PRINT AN END MESSAGE WHICH WILL IDENTIFY THE OPERATION OF THAT PROGRAM AS TERMINATED.

#### 5.2.7 HALT ROUTINE

1. THE HALT ROUTINE IS ENTERED BY
  - A. PRESSING THE PROGRAM STOP PUSHBUTTON ON THE 1131 CONSOLE.
  - B. ADDRESS STOP WHEN THE IAR ADDRESS IS EQUAL TO THE ADDRESS SET IN THE BIT SWITCHES DURING THE LAST HALT.
2. THIS ROUTINE CYCLES THE PROGRAM DOWN, I.E., ALLOWS ALL INTERRUPTS TO BE SERVICED BEFORE STOPPING IN A WAIT INSTRUCTION. TO CONTINUE, PRESS PROGRAM START.
3. THE CPU STATUS IS SAVED ANYTIME A HALT OCCURS IN THE HALT ROUTINE THUS, AFTER A HALT, THE CE CAN GO MANUALLY TO ANY PLACE IN CORE, READ OR MODIFY ANY STORAGE LOCATION, THEN RETURN TO THIS ROUTINE AND CONTINUE. TO SIMPLIFY THIS RESTART PROCEDURE, A BRANCH EXISTS

AT ADDRESS ZERO WHICH WILL TRANSFER TO THE HALT ROUTINE AFTER A HALT HAS OCCURRED. THUS, TO RESTART AFTER A HALT WHEN THE CONTENTS OF IAR HAVE BEEN CHANGED, PRESS RESET, THEN PROGRAM START.

4. ADDRESS STOP IS ALSO A FUNCTION OF THE HALT ROUTINE. THE FIRST INSTRUCTION FOLLOWING THE WAIT IN THE HALT ROUTINE IS AN XIO TO READ THE BIT SWITCHES. ANY ADDRESS SET IN THE BIT SWITCHES AT THIS TIME WILL BE STORED AS AN ADDRESS STOP ADDRESS. THE ADDRESS STOP FEATURE USES THE INTERRUPT RUN MODE TO PROVIDE AN INTERRUPT FOLLOWING EACH MAINLINE INSTRUCTION. THUS TO ENABLE ADDRESS STOP THE CPU MUST BE PLACED IN THE INTERRUPT RUN MODE. WHEN IN THIS MODE ADDRESS STOP WILL CAUSE THE PROGRAM TO HALT ANY TIME THE IAR IS EQUAL TO THE SETTING OF THE BIT SWITCHES ON THE LAST START FROM THE HALT ROUTINE. SINCE ADDRESS STOP ALSO HALTS IN THIS SAME HALT ROUTINE, THE NEXT STOP ADDRESS MUST BE SET IN THE SWITCHES FOLLOWING EACH ADDRESS STOP.

----- LAST PAGE -----

```

*****
*
* THIS ENGINEERING CHANGE REFLECTS MAJOR
* CHANGES TO THE DIAGNOSTIC MONITOR. PREVIOUS
* TESTS WHICH RAN WITH THE EARLIER MONITOR WILL
* NOT RUN WITH THIS MONITOR.
*
* DIAGNOSTIC MONITOR II WILL NOT RUN TESTS OF
* ENGINEERING LEVEL PRIOR TO EC 419643 DATED
* NOV 15,1966.
*****
* THE FOLLOWING ADDRESSES MUST BE MAINTAINED FOR
* COMPATIBILITY WITH DIMAL- PID 0302.
*
* ADRS LABEL ADRS LABEL ADRS LABEL
* /044A ILT0 /048E ILT4 /0414 ASTOP
* /045B ILT1 /03F9 ILT5 /0410 AQ5
* /046C ILT2 /0280 MRSRT /058D LOAD2
* /047D ILT3 /054F MRBSD
*****
ABS
DC /0300
*****
-----
* 1130 DIAGNOSTIC MONITOR
-----
* MONITOR RESTART
*****
* RESFT AND START WILL FORCE ENTRY TO
* THIS ROUTINE WHICH WILL REINITIALIZE
* THE MONITOR CONTROL AND BRANCH TO
* THE SUPERVISOR.
-----
*
* ORG 0
* LDX 4 BR FOR RESTART
* BSS 3 SKIP INDEX REGS
* BSC L MRSRT GO RESTART
*****
* INTERRUPT TRANSFER VECTORS
*****
*
* ORG 8
* XFVC BSS 1 SPACE FOR ILT0
* DC ILT1
* DC ILT2
* DC ILT3
* BSS 1 SPACE FOR ILT4
* DC ILT5
* BSS 6 NOT USED
*****
* UTILITY EQUATE TABLE
*****
*
* UTLTY DC MRBSD MONITOR UTILITY CONTROL
* ILIR DC ASTOP INTERRUPT RUN
* AQ5A DC AQ5 A AND Q FOR TRACE INT
* DC /0000 NOT USED
*****
* INITIALIZE RESTART AND TRANSFER VEC
*****

```

0000 0 0300

0000 0 6004  
0001 0003  
0004 0 4C00 0280

0008  
0008 0001  
0009 0 045B  
000A 0 046C  
000B 0 047D  
000C 0001  
000D 0 03F9  
000E 0006

0014 0 054F  
0015 0 0414  
0016 0 0410  
0017 0 0000

0018 0 6004  
0019 0 044A  
001A 0 045B  
001B 0 046C  
001C 0 047D  
001D 0 048E

0160  
0160 0 01CF  
0161 0 0264  
0162 0 02A4  
0163 0 02F8  
0164 0 0426

0165 0 0000  
0166 0 020D  
0167 0 02F5  
0168 0 023E  
0169 0 0000  
016A 0 0000

016B 0 0480  
016C 0 0480  
016D 0 0480  
016E 0 0480  
016F 0 0480  
0170 0 0480  
0171 0 0480  
0172 0 0480  
0173 0 0480  
0174 0 0480  
0175 0 0480  
0176 0 0480  
0177 0 0480  
0178 0 0480  
0179 0 0480  
017A 0 0480  
017B 0 0480  
017C 0 0480  
017D 0 0480  
017E 0 0480  
017F 0 0480  
0180 0 0480  
0181 0 0480  
0182 0 0480  
0183 0 0480  
0184 0 0480  
0185 0 0480

```

*
* ORGO LDX 4
* IXFVC DC ILT0
* DC ILT1
* DC ILT2
* DC ILT3
* DC ILT4
*
* *****
* MONITOR EQUATE TABLE
* *****
* THIS IS A TABLE OF COMMON LABELS
* USED BY THE TEST PROGRAMS TO REFER
* TO DESIRED ADDRESSES IN THE MONITOR.
*-----
*
* ORG /160
* BEGIN DC BGIN THESE ARE CALL
* START DC STRT LABELS USED BY TEST
* ERROR DC ERRI PROGRAMS TO CALL
* LOG DC LOGI MONITOR ROUTINES
* END DC MEND
*
* RTNSW DC *-* ROUTINE START SW
* ERLCK DC MSW0 LOCK ON ERROR CONTROL
* LOGBY DC IOBSY I/O BUSY SW ADDR
* ADDR DC MXEQ2&1
* DC /0000 NOT USED
* DC /0000 NOT USED
*
*-----
* THE TEST PROGRAM MUST ENTER ITS
* INTERRUPT ROUTINE ADDRESS IN THE
* APPROPRIATE PLACE IN THIS TABLE
* BEFORE EXECUTING ANY XIO COMMAND
* WHICH WILL RESULT IN AN INTERRUPT.
*-----
*
* DC IERR 15
* DC IERR 14
* DC IERR 13
* DC IERR 12
* DC IERR 11
* DC IERR 10
* DC IERR 9
* DC IERR 8
* DC IERR 7
* DC IERR 6
* DC IFRR 5
* DC IFRR 4
* DC IFRR 3
* DC IERR 2
* DC IERR 1
* ILO DC IERR 0 1442 COL RQST
* DC IERR 15
* DC IFRR 14
* DC IERR 13
* DC IFRR 12
* DC IERR 11
* DC IERR 10
* DC IERR 9
* DC IERR 8
* DC IERR 7
* DC IERR 6
* DC IFRR 5

```

0186	0	0480	DC	IERR	4		30001380
0187	0	0480	DC	IERR	3		30001390
0188	0	0480	DC	IERR	2		30001400
0189	0	0480	DC	IERR	1		30001410
018A	0	0480	IL1	DC	IERR	0	1132 SERVICE RQST
018B	0	0480	DC	IERR	15		30001420
018C	0	0480	DC	IERR	14		30001430
018D	0	0480	DC	IERR	13		30001440
018E	0	0480	DC	IERR	12		30001450
018F	0	0480	DC	IERR	11		30001460
0190	0	0480	DC	IERR	10		30001470
0191	0	0480	DC	IERR	9		30001480
0192	0	0480	DC	IERR	8		30001490
0193	0	0480	DC	IERR	7		30001500
0194	0	0480	DC	IERR	6		30001510
0195	0	0480	DC	IERR	5		30001520
0196	0	0480	DC	IERR	4		30001530
0197	0	0480	DC	IERR	3		30001540
0198	0	0480	DC	IERR	2		30001550
0199	0	0480	DC	IERR	1		30001560
019A	0	0480	IL2	DC	IERR	0	DISC SERVICE RQST
019B	0	0480	DC	IERR	15		30001570
019C	0	0480	DC	IERR	14		30001580
019D	0	0480	DC	IERR	13		30001590
019E	0	0480	DC	IERR	12		30001600
019F	0	0480	DC	IERR	11		30001610
01A0	0	0480	DC	IERR	10		30001620
01A1	0	0480	DC	IERR	9		30001630
01A2	0	0480	DC	IERR	8		30001640
01A3	0	0480	DC	IERR	7		30001650
01A4	0	0480	DC	IERR	6		30001660
01A5	0	0480	DC	IERR	5		30001670
01A6	0	0480	DC	IERR	4		30001680
01A7	0	0480	DC	IERR	3		30001690
01A8	0	0480	DC	IERR	2		30001700
01A9	0	0480	DC	IERR	1		30001710
01AA	0	0480	IL3	DC	IERR	0	1627 SERVICE RQST
01AB	0	0480	DC	IERR	15		30001720
01AC	0	0480	DC	IERR	14		30001730
01AD	0	0480	DC	IERR	13		30001740
01AE	0	0480	DC	IERR	12		30001750
01AF	0	0480	DC	IERR	11		30001760
01B0	0	0480	DC	IERR	10		30001770
01B1	0	0480	DC	IERR	9		30001780
01B2	0	0480	DC	IERR	8		30001790
01B3	0	0480	DC	IERR	7		30001800
01B4	0	0480	DC	IERR	6		30001810
01B5	0	0480	DC	IERR	5		30001820
01B6	0	0480	DC	IERR	4		30001830
01B7	0	0480	DC	IERR	3		30001840
01B8	0	0480	DC	IERR	2	1442 OP COMP	30001850
01B9	0	04C6	DC	MCINT	1	1053	30001860
01BA	0	0480	IL4	DC	IERR	0	PAPER TAPE
01BB	0	0480	RQTY	DC	IERR		CON PRINTER REQUEST
01BC	0	04EA	RQKB	DC	MMRBS-1		KEYBOARD REQUEST
01BD	0	0480	SVKB	DC	IERR		KB SERVE REQUEST
01BE	0	0480	IL5	DC	IERR		INTERRUPT LEVEL 5

\*\*\*\*\*  
PROGRAM ID TABLE  
\*\*\*\*\*  
\* THIS TABLE IS USED BY THE MONITOR  
\* TO IDENTIFY THE PROGRAM OR PROGRAMS IN  
\* CORE AND TO PROVIDE AN ADDRESS FOR  
\* COMMUNICATION WITH THESE PROGRAMS.  
\* AS EACH PROGRAM IS LOADED ITS MLSCF

01BF	0	0000	DC	*-*	.	.	OF	30002060
01C0	0	0000	DC	*-*	.	.	OE	30002070
01C1	0	0000	DC	*-*	.	.	OD	30002080
01C2	0	0000	DC	*-*	.	.	OC	30002090
01C3	0	0000	DC	*-*	.	.	OB	30002100
01C4	0	0000	DC	*-*	.	.	OA	30002110
01C5	0	0000	DC	*-*	.	.	O9	30002120
01C6	0	0000	DC	*-*	.	.	O8	30002130
01C7	0	0000	DC	*-*	.	.	O7	30002140
01C8	0	0000	DC	*-*	.	.	O6	30002150
01C9	0	0000	DC	*-*	.	.	O5	30002160
01CA	0	0000	DC	*-*	.	.	O4	30002170
01CB	0	0000	DC	*-*	.	.	O3	30002180
01CC	0	0000	DC	*-*	.	.	O2	30002190
01CD	0	0000	DC	*-*	.	.	O1	30002200
01CE	0	0213	MPIDT	DC	MLSCF	.	O0	30002210

\*\*\*\*\*  
BEGIN ROUTINE  
\*\*\*\*\*  
\* THIS ROUTINE IS ENTERED BY ALL  
\* PROGRAMS FOLLOWING PROGRAM LOAD.  
\* THE ROUTINE WILL SET THE MLSCF  
\* ADDRESS OF THE PROGRAM IN THE  
\* PROGRAM ID TABLE, GO  
\* INITIALIZE THE PROGRAM, TYPE  
\* OUT A MESSAGE IDENTIFYING THE  
\* PROGRAM LOADED, AND START THE  
\* PROGRAM IF LOAD AND GO.  
\*\*\*\*\*  
CALLING SEQUENCE  
\* BSI I BEGIN  
\* DC ADDRESS OF PROGRAM CONTROL TABLE  
\*\*\*\*\*  
BGIN DC \*-\*  
MDX L PGMN,1  
LD RELF LOAD THE RELOCATION  
STO L MTABL&5 FACTOR AND SET IN MSG  
\* LD I BGIN LD ADDR OF PCT  
STO L MTABL&3 STORE IN MESSAGE  
STO BGIN1&1  
A K0009 CALCULATE MLSCF ADRS  
STO TEMPB SAVE MLSCF ADRS  
\* BGIN1 LDX L1 \*-\* XRI#PCT ADRS  
LD 1 0 LD PID  
STO L MTABL&2 SET PID IN MESSAGE  
LDX I3 PGMN XR3#PRGM NO.  
LDX 2 15  
LD K8000  
SRA 3 0  
STO XLECT  
SLCA 2 0  
LD TEMPB  
STO L2 MPIDT-15 STO MLSCF ADRS  
\* XIO L RDBS READ BIT SWS  
LD L MTABL  
SLA 15  
SRA 15

1130 DIAGNOSTIC MONITOR II

1130 DIAGNOSTIC MONITOR II

```

01F1 0 E814      OR      K0080      30002740
01F2 0 D103      STD      1 3          30002750
*                               30002760
*                               30002770
01F3 0 C010      LD       XLFCT      30002780
01F4 0 73FF      MDX     3 -1        30002790
01F5 0 F80D      OR       SELCT      SKIP IF NOT OVERLAP
01F6 0 D00C      STD     SELCT      STORE SELECT CONTROL
*                               30002800
*                               30002810
*-----*
* THIS SUBROUTINE WILL LOG THE LOAD
* MESSAGE.
*-----*
*                               30002820
*                               30002830
*                               30002840
*                               30002850
*                               30002860
01F7 0 622E      LDX     2 /002E     SFT WORD CONTROL
01F8 0 6300      LDX     3 0         SET MSG ID
01F9 0 C806      LDD     MMESO      LD ALPHA ADDR
01FA 0 4400 04F3 BSI     L  ML0G     GO LOG MESSAGE
*                               30002900
*                               30002910
01FC 0 6700 017A LDX     L3 ILO     LOAD NEXT PRGM
01FE 0 6078      LDX     LOADR      30002920
*                               30002930
*                               30002940
*                               30002950
0200 0000      BSS     E 0         30002960
0200 0 0589      MMESO  DC  ADRLD    30002970
0201 0 059F      DC     ALD         30002980
0202 0 0000      TEMPB  DC  *-#     30002990
0203 0 0000      SELCT  DC  *-#     30003000
0204 0 0000      XLECT  DC  *-#     30003010
0205 0 8000      K8000  DC  /8000   30003020
0206 0 0080      K0080  DC  /0080   30003030
0207 0 007F      K007F  DC  /007F   30003040
*                               30003050
*****
* MONITOR CONTROL TABLE
*****
*                               30003060
*                               30003070
*                               30003080
*                               30003090
0208 0000      BSS     F 0         30003100
0208 0 0000      ZEROS  DC  0000    FIVE ZEROS USED AS MSG
0209 0 0009      K0009  DC  /0009   30003110
*                               30003120
*                               30003130
020A 0 0000      MPID   DC  0000    PID
020B 0 0000      DC     0000       30003140
020C 0 0000      DC     0000       30003150
020D 0 0000      MSW0   DC  *-#     SWITCH FUNCTION 0
020E 0 0000      MSW1   DC  *-#     1
020F 0 0000      MSW2   DC  *-#     2
0210 0 0000      MSW3   DC  *-#     3
0211 0 0000      DC     0000       30003170
0212 0 0000      DC     0000       30003180
*                               30003190
*                               30003200
*                               30003210
*                               30003220
0213 0 0000      MLSCF  DC  *-#     READ BIT SWS
0214 0 0000      TYPI   DC  *-#     TYPE INTERRUPT
0215 0 0000      ERBY   DC  *-#     ERROR BUSY
0216 0 0000      MMBSY  DC  *-#     MONITOR MESSAGE BUSY
0217 0 0000      IERM   DC  *-#     INTERRUPT ERROR MESSAGE
0218 0 FFFF      TERMX  DC  /FFFF   30003280
*                               30003290
*                               30003300
*****
* MONITOR INITIALIZATION
*****
*                               30003310
*                               30003320
*                               30003330
*                               30003340
0219 0 6305      INLIZ  LDX  3 5     SET RESTART FROM ZERO
021A 0 C318      LD     3 IXFVC-1   AND INTERRUPT TRANSFER
021B 0 D307      STD   3 XFVC-1     VECTOR ADDR IN LOW CORE
021C 0 73FF      MDX   3 -1        30003350
021D 0 70FC      MDX   INLIZ&1     30003360
021E 0 C318      LD     3 ORG0      30003370
021F 0 D300      STD   3 0         30003380
*                               30003390
*                               30003400
*                               30003410

```

```

*****
* SUPERVISOR ROUTINE
*****
*                               30003420
*                               30003430
*                               30003440
*                               30003450
*                               30003460
* THIS ROUTINE POLLS ALL SELECTED
* PROGRAMS LOOKING FOR AN ENTRY IN THE
* MLSCF TABLE OF EACH PROGRAM. IF AN
* ENTRY IS FOUND THIS ROUTINE WILL
* BRANCH TO THE ADDRESS SPECIFIED BY
* THE ENTRY. ONLY THE FIRST NON ZERO
* ENTRY IN EACH MLSCF IS EXECUTED. UPON
* RETURN FROM ANY PROGRAM THE SUPERVISOR
* WILL POLL ALL OTHER PROGRAMS BEFORE
* EXECUTING ANOTHER ROUTINE IN
* THAT PROGRAM.
*-----*
* CALLING SEQUENCE
* BSI I START
*-----*
*                               30003500
*                               30003510
*                               30003520
*                               30003530
*                               30003540
*                               30003550
*                               30003560
*                               30003570
*                               30003580
*                               30003590
*                               30003600
*                               30003610
*                               30003620
0220 0 C0EC      MXEQ   LD     MSW0   30003630
0221 0 D05B      STO    MSWOX      30003640
0222 0 E0E4      AND    K007F      30003650
0223 0 7400 027B MDX   L  STRSW     SKIP IF NOT START
0225 0 D0E7      STO    MSW0       30003660
*                               30003670
*                               30003680
0226 0 10A0      SLT    32         30003690
0227 0 D400 0165 STO   L  RTNSW     RESET RTN START SW
0229 0 D051      STO    STRSW      RESET START SW
*                               30003700
*                               30003710
*                               30003720
022A 0 0C00 03D0 XIO   L  LOGT      SENSE CONSOLE DSW
022C 0 1002      SLA    2           CK FOR INTRPT
022D 0 4428 04C6 BSI   L  MCINT,Z& BR IF RESPONSE BIT ON
*                               30003730
*                               30003740
*                               30003750
*                               30003760
022F 0 6110      LDX   1 16        30003770
0230 0 C0D2      LD     SELCT      30003780
0231 0 E8D3      OR    K8000      DO NOT HALT MONITOR
0232 0 D0D1      STO   XLECT      30003790
*                               30003800
*                               30003810
0233 0 C0D0      MXFQ1 LD  XLECT    30003820
0234 0 1140      SLCA  1 0         30003830
0235 0 F0CF      ENR   K8000      30003840
0236 0 D0CD      STO   XLECT      30003850
0237 0 C500 01BE LD   L1 MPIDT-16 LD MLSCF ADDRESS
0239 0 D004      STO   MXEQ2&1    30003860
023A 0 4C02 023D BSC   L  MXEQ2,C  30003870
023C 0 70E3      MDX   MXEQ        30003880
*                               30003890
*                               30003900
023D 0 6600 0213 MXEQ2 LDX  L2 MLSCF XR2#ADDRS OF MLSCF
023F 0 6932      STX   1 MXEQ7&1  SAVE XR1
*                               30003910
*                               30003920
*                               30003930
0240 0 C2F7      LD     2 -9       LD PID
0241 0 4C18 0258 BSC   L  MXEQ5,&- BR IF MONITOR
*                               30003940
*                               30003950
*                               30003960
0243 0 C2FA      LD     2 -6       NOT MONITOR- LD SWO
0244 0 F838      OR    MSWOX      COMBINE WITH MON SWO
0245 0 4804      BSC   E          SKIP IF NOT HALT
0246 0 1810      SRA   16         30003990
0247 0 E835      OR    MSWOX      COMBINE WITH MON SWO
0248 0 18D8      RTE   24         CK START BIT
0249 0 4C10 0254 BSC   L  MXEQ4,-  BR IF NOT START
*                               30004000
*                               30004010
*                               30004020
*                               30004030
024B 0 682F      STX   STRSW      SET START SW
024C 0 2000      LDS   0          30004040
024D 0 C2FA      LD     2 -6       30004050
024E 0 F025      AND   K007E      RESET RESTART
024F 0 D2FA      STO   2 -6       30004070
0250 0 C024      LD     SMSGA      SET TO PRINT RESTART MSG

```

1130 DIAGNOSTIC MONITOR II

```

0251 0 D200          STO 2 0          SET MLSCF ENTRY          30004100
0252 0 7201          MXEQ3 MDX 2 1          30004110
0253 0 7004          MDX          MXEQ5          30004120
*                   30004130
0254 0 C2FA          MXEQ4 LD 2 -6          IF NOT MONITOR- LD SWO  30004140
0255 0 E8B7          OR          MSWO          COMBINE WITH MON SWO  30004150
0256 0 4C04 0271     BSC L MXEQ7,E          BR IF HALT SW ON      30004160
*                   30004170
0258 0 C200          MXEQ5 LD 2 0          LD MLSCF ENTRY          30004180
0259 0 D008          STO          MXEQ6&E1          SET UP FOR EXIT        30004190
025A 0 4C18 0252     BSC L MXEQ3,&-          BR IF ZERO              30004200
025C 0 FORB          EOR          TERMX          30004210
025D 0 4C18 0271     BSC L MXEQ7,&-          BR IF TERMINATOR       30004220
025F 0 1810          SRA          16          ZERO MLSCF ENTRY       30004230
0260 0 D200          STO 2 0          30004240
0261 0 4C02 0000     MXEQ6 BSC L *-*,C          EXIT TO USER           30004250
0263 0 70EE          MDX          MXEQ3          30004260
*                   30004270
0264 0 0000          STRT DC *-*          SUPERVISOR ENTRY       30004280
0265 0 6680 023E     LDX I2 MXEQ2&I          30004290
0267 0 C0A5          LD          MSWO          30004300
0268 0 EAF8          OR 2 -6          COMBINE WITH USER SWO 30004310
0269 0 1009          SLA 9          CK RTN START LOG SW    30004320
026A 0 7400 0165     MDX L RTNSW          CK RTN START SW        30004330
026C 0 4810          BSC -          30004340
026D 0 7003          MDX          MXEQ7          BR IF NOT BOTH          30004350
026E 0 4400 02F8     BSI L LOGI          LOG RTN START MSG      30004360
0270 0 0208          DC          ZEROS          30004370
*                   30004380
*                   30004390
0271 0 6500 0000     MXEQ7 LDX L1 *-*          RESET XRI TO SELECT CNTRL 30004400
0273 0 70BF          MDX          MXEQ1          30004410
*                   30004420
0274 0 007E          K007E DC /007E          CHANGE TO 007F         30004430
0275 0 0293          SMSGA DC SMSG          30004440
0276 0 0000          DC /0000          CAN BE USED FOR EVEN K 30004450
0277 0 8001          DC /8001          30004460
0278 0 05AD          DC ASTRT          30004470
0279 0 0000          DC /0000          30004480
027A 0 0000          STPID DC *-*          30004490
027B 0 0000          STRSW DC *-*          START SW                30004500
027C 0 0001          K0001 DC /0001          30004510
027D 0 0000          MSWOX DC *-*          30004520
027E 0002          BSS 2 WARNING DON'T REMOVE USED FOR DIMAL 30004530
*                   30004540
*****
*                   30004550
*                   30004560
*                   30004570
* THIS SUBROUTINE RESTARTS THE MONITOR. IF THE 30004580
* RESTART IS FROM PROGRAM HALT THE SUBROUTINE WILL 30004590
* BRANCH TO GO. IF THE RESTART IS FROM THE INVALID 30004600
* INTERRUPT TRAP, THE SUBROUTINE WILL BRANCH 30004610
* INDIRECT ON THE INTERRUPT ENTRY. IF NONE OF THE 30004620
* ABOVE THE SUBROUTINE WILL CONTINUE EXECUTION 30004630
* OF ALL PROGRAMS NOT HALTED. 30004640
*-----
*                   30004650
*                   30004660
*                   30004670
*                   30004680
0280 0 C400 0412     MRSRT LD L STPSW          KEEP MRSRT AT /0280    30004690
0282 0 4C20 040B     BSC L GO,Z          RESTART FROM HALT      30004700
*                   30004710
*                   30004720
0284 0 C092          LD IERM          CK FOR RESTART FROM    30004730
0285 0 4CA0 0480     BSC I IERR,Z          INVALID INTRPT TRAP    30004740
*                   30004750
*                   30004760
0287 0 D06D          STO IOBSY          RESET IOBSY             30004770
0288 0 D06B          STO ERC          RESET ERROR CALL        30004780
0289 0 6700 0414     LDX L3 ASTOP          30004790

```

1130 DIAGNOSTIC MONITOR II

```

0288 0 6F00 0015     STX L3 ILIR          INITIALIZE FOR ADRS STOP 30004780
028D 0 6700 04EA     LDX L3 MMRBS-1        30004790
028F 0 6F00 01BC     STX L3 RQKB          RESTORE RQKB            30004800
*                   30004810
0291 0 4C00 0219     BSC L INLIZ          CONTINUE EXECUTION      30004820
*                   30004830
*-----
*                   30004840
* COME TO HERE TO PRINT START MESSAGE
*-----
*                   30004850
*                   30004860
*                   30004870
0293 0 7400 02F5     SMSG MDX L IOBSY,0          CK I/O BUSY SW          30004880
0295 0 700B          MDX          SMSG1          BR IF BUSY              30004890
*                   30004900
*                   30004910
0296 0 C0DD          LD K007E          30004920
0297 0 EC00 027C     OR L K0001          30004930
0299 0 D0DA          STO K007E          30004940
029A 0 C2F7          LD 2 -9          LD PID                  30004950
029B 0 D0DE          STO STPID          STO IN MSG              30004960
029C 0 405B          BSI LOGI          PRINT START MSG        30004970
029D 0 0275          DC SMSGA          30004980
*                   30004990
*                   30005000
029E 0 C1FF          LD 1 -1          LOAD RESTART ADRES     30005010
029F 0 D100          STO 1 0          SET IN MLSCF            30005020
02A0 0 70D0          MDX MXEQ7          30005030
*                   30005040
SMSG1 LD SMSGA          SET TO TRY AGAIN IF    30005050
STO 2 0          PRINTOUT RTN IS BUSY  30005060
MDX MXEQ7          30005070
*                   30005080
*****
*                   30005090
*                   30005100
*                   30005110
*                   30005120
*                   30005130
*                   30005140
*                   30005150
*-----
*                   30005160
*                   30005170
*                   30005180
*                   30005190
*                   30005200
*                   30005210
*                   30005220
*                   30005230
*                   30005240
*                   30005250
*                   30005260
*                   30005270
*                   30005280
*-----
*                   30005290
*                   30005300
*                   30005310
ERRI DC *-*          30005320
LDX I1 MXEQ2&E1          XRI#MLSCF ADRES        30005330
LDX L2 TYPI          30005340
LDX I3 ERRI          X3#CALL SEQ ADDR     30005350
LD ERC          CK FOR ERROR BUSY  30005360
BSC L BUSY,Z          BUSY RETURN              30005370
STX 3 ERC          SAVE CALL SEQ ADDR  30005380
*                   30005390
*                   30005400
*                   30005410
*                   30005420
*                   30005430
*                   30005440
*                   30005450
02A4 0 0000          30005460
02A5 0 6580 023E     LDX I1 MXEQ2&E1          XRI#MLSCF ADRES        30005470
02A7 0 6600 0214     LDX L2 TYPI          30005480
02A9 0 6780 02A4     LDX I3 ERRI          X3#CALL SEQ ADDR     30005490
02AB 0 C048          LD ERC          CK FOR ERROR BUSY  30005500
02AC 0 4C20 02EB     BSC L BUSY,Z          BUSY RETURN              30005510
02AE 0 6B45          STX 3 ERC          SAVE CALL SEQ ADDR  30005520
*                   30005530
*                   30005540
*                   30005550
*                   30005560
*                   30005570
*                   30005580
*                   30005590
02AF 0 6B43          STX 3 FRSW          SET ERSW                30005600
02B0 0 C1FA          LD 1 -6          LD FUNC 0 SWS           30005610
02B1 0 EAF9          OR 2 MSWO-TYPI          OR WITH MON SWS         30005620
02B2 0 1881          SRT 1          CHECK FOR HALT ON ERR  30005630
02B3 0 4C04 02B8     BSC L ERRI1,E          BR IF HALT ON ERROR     30005640
02B5 0 1881          SRT 1          CK BYPASS ERR MESS     30005650
02B6 0 4C04 02C1     BSC L RSTX&2,E          BR IF BYPASS ERR MESS  30005660

```

1130 DIAGNOSTIC MONITOR II

1130 DIAGNOSTIC MONITOR II

```

02B8 0 7048 *
ERRI1 MDX ELOG GO TO LOG RTN 30005460
* 30005470
* 30005480
*-----*
* LOG FINISHED - RESTORE AND RETURN 30005490
* TO USER. 30005500
*-----*
* 30005510
* 30005520
* 30005530
* 30005540
02B9 0 10A0 LOGR SLT 32 30005550
02BA 0 D03A STO IOBSY RESET IOBSY 30005560
02BB 0 D400 0204 STO L XLECT RESET SUP SCAN CONTROL 30005570
02BD 0 6780 030B LDX I3 LOG5&1 SET XR3 TO CALL ADDR 30005580
02BF 0 6500 0000 RSTX LDX L1 *-* XI#MLSCF ADRS 30005590
02C1 0 6D00 023E STX L1 MXEQ2&1 30005600
02C3 0 C02F LD ERSW 30005610
02C4 0 4F18 0001 BSC L3 1,E- RETURN IF NOT ERR 30005620
* 30005630
02C6 0 1810 ERRI2 SRA 16 30005640
02C7 0 D02B STO ERSW RESET ERROR 30005650
02C8 0 D02B STO FRC SWITCHES 30005660
* 30005670
02C9 0 C02D LD LINF BYPASS SW CK AFTER 1ST 30005680
02CA 0 4F28 0002 BSC L3 2,Z& LINE OF MESSAGE 30005690
* 30005700
02CC 0 C1FA LD 1 -6 LD FUNC 0 SWS 30005710
02CD 0 EAF9 OR 2 MSWO-TYPI OR WITH MCN SWS 30005720
02CE 0 1801 SRA 1 CHECK FOR HALT ON ERR 30005730
02CF 0 4C04 02D9 BSC L HALT,E 30005740
* 30005750
02D1 0 100D FRR13 SLA 13 CHECK FOR LOOP 30005760
02D2 0 4F10 0002 BSC L3 2,- ON ERROR 30005770
* 30005780
02D4 0 C301 LD 3 1 LOOP UNLESS 30005790
02D5 0 4F08 0002 BSC L3 2,& LOOP ADRS ZERO 30005800
02D7 0 4F80 0001 BSC I3 1 LOOP 30005810
* 30005820
02D9 0 3002 HALT WAIT 2 HALT ON ERROR 30005830
02DA 0 70FE MDX HALT 30005840
02DB 0 70F5 MDX ERRI3 30005850
*-----*
* ERROR BUSY SUBROUTINE 30005860
*-----*
* 30005870
* 30005880
02DC 0 1810 ERRI7 SRA 16 30005890
02DD 0 D015 STO ERSW RESET ERR SW 30005900
02DE 0 6700 02E5 LDX L3 ERRI8 SET REENTRY 30005910
02E0 0 6F00 0215 STX L3 ERRI8 IN MLSCF 30005920
02E2 0 6903 STX 1 FRR18&1 30005930
02E3 0 4C00 0271 BSC L MXEQ7 GO TO SUPERVISOR 30005940
* 30005950
02E5 0 6500 0000 ERRI8 LDX L1 *-* RESTORE XR1 30005960
02E7 0 6780 02F4 LDX I3 ERC RESTORE XR3 30005970
02E9 0 6809 STX 3 ERSW AND RESET ER SW 30005980
02EA 0 701A MDX LOG4 GO TO LOG AGAIN 30005990
* 30006000
* 30006010
*-----*
* LOG BUSY SUBROUTINE 30006020
*-----*
* 30006030
* 30006040
02EB 0 73FE BUSY MDX 3 -2 30006050
02EC 0 6F00 0216 STX L3 MMBSY STORE BUSY RETURN 30006060
02EE 0 4C00 0220 BSC L MXEQ GO TO SUPERVISOR 30006070
* 30006080
* 30006090
*-----*
* CONSTANTS FOR ERROR AND LOG 30006100
*-----*
* 30006110
* 30006120
02F0 0000 BSS F 0 30006130

```

```

02F0 0 0000 PIDWS DC *-* MID STORAGE 30006140
02F1 0 0000 DC *-* ROUTINE STORE 30006150
02F2 0 0000 DC *-* RTN ADRS STORE 30006160
02F3 0 0000 ERSW DC *-* ERROR SW 30006170
02F4 0 0000 ERC DC *-* ERROR CALL SW 30006180
02F5 0 0000 IOBSY DC *-* I/O BUFR BUSY SW 30006190
02F6 0 8100 CR DC /8100 CARRIER RETURN 30006200
02F7 0 0000 LINE DC *-* 30006210
* 30006220
***** 30006230
* LOG ROUTINE 30006240
***** 30006250
* THIS ROUTINE LOGS MONITOR AND TEST 30006260
* PROGRAM MESSAGES TO THE CONSOLE 30006270
* TYPEWRITER. CONTROL IS RETURNED TO THE 30006280
* CALLING PROGRAM WHEN LOG IS COMPLETED. 30006290
*-----*
* CALLING SEQUENCE 30006300
* BSI I LOG 30006310
* DC ADRS OF MESSAGE 30006320
* 30006330
* 30006340
* MESSAGE FORMAT 30006350
* DC MESSAGE NUMBER 30006360
* DC HEX/DECIMAL SWITCH 30006370
* DC DATA WORD ID 30006380
* DC ALPHA ADRS 1 30006390
* DC ALPHA ADRS 2 30006400
* 0 TO 15 MODIFIERS 30006410
*-----*
* 30006420
LOGI DC *-* 30006430
LDX I1 MXEQ2&1 XR1#MLSCF ADRS 30006440
LDX I3 LOGI XR3# CALL SEQ ADRS 30006450
LD IOBSY CHECK IF I/O 30006460
BSC L BUSY,Z BUFFER BUSY 30006470
STO ERSW RESET ERROR SW 30006480
* 30006490
ELOG LDD 1 -9 LOAD AND STORE PID 30006500
STO PIDWS AND ROUTINE NUMBER 30006510
LD 1 -7 LD RTN ADRS 30006520
STO PIDWS&2 STORE RTN ADRS 30006530
* 30006540
LOG4 MDX L IOBSY,0 ERROR LOG CHECK 30006550
MDX ERRI7 FOR BUSY- IF BUSY 30006560
RETURN TO ERROR RTN 30006570
SAVE XR1 30006580
* 30006590
STX 1 RSTX&1 30006600
STX 3 LOG5&1 30006610
LOG5 LDX I3 *-* XR3# MESSAGE ADRS 30006620
LD 3 0 FETCH MESSAGE ID 30006630
SRT 8 30006640
LD PIDWS OR MID WITH PID 30006650
SLT 8 30006660
STO PIDWS STO TOTAL MESS ID 30006670
* 30006680
LDX L1 TABL XR1# CONVERSION 30006690
TABLE ADDRESS 30006700
* 30006710
LDX 2 -78 LD SPACE 30006720
LD 1 16 SET I/O BUFFER TO 30006730
LOG7 STO 2 MES0&79 SPACES 30006740
MDX 2 1 30006750
MDX LOG7 30006760
* 30006770
LDX 2 MES0&1 XR2#I/O BUFR ADRS 30006780
LD CR SET CARRIER RETURN IN OUT 30006790
STO 2 -1 30006800
LD 1 21 LD SHRD CODE 30006810
MDX L ERSW,0 SKIP IF NOT ERROR 30006820
LD 1 20 LD SHRD CODE 30006830
02F8 0 0000 30006840
02F9 0 6580 023F 30006850
02FB 0 6780 02F8 30006860
02FD 0 C0F7 30006870
02FE 0 4C20 02EB 30006880
0300 0 D0F2 30006890
* 30006900
0301 0 C9F7 30006910
0302 0 D8ED 30006920
0303 0 C1F9 30006930
0304 0 D0ED 30006940
* 30006950
0305 0 7400 02F5 30006960
0307 0 70D4 30006970
* 30006980
0308 0 69B7 30006990
0309 0 6801 30007000
030A 0 6780 0000 30007010
030C 0 C300 30007020
030D 0 1888 30007030
030F 0 C0E1 30007040
030F 0 1088 30007050
0310 0 D0DF 30007060
* 30007070
0311 0 6500 03DA 30007080
* 30007090
0313 0 62B2 30007100
0314 0 C110 30007110
0315 0 D277 30007120
0316 0 7201 30007130
0317 0 70FD 30007140
* 30007150
0318 0 6229 30007160
0319 0 C0DC 30007170
031A 0 D2FF 30007180
031B 0 C115 30007190
031C 0 7400 02F3 30007200
031E 0 C114 30007210

```



1130 DIAGNOSTIC MONITOR II

1130 DIAGNOSTIC MONITOR II

```

031F 0 D200          *   STO  2 0          STO SHIFT RED/BLACK          30006820
                                IN I/O BUFFER          30006830
0320 0 C301          *   LD   3 1          LD HEX/DEC SW          30006840
0321 0 1800          *   RTE  16          30006850
0322 0 C302          *   LD   3 2          LD WORD CONTROL          30006860
0323 0 D003          *   STO  LINE          STO MULT LINE CONTROL          30006870
0324 0 1001          *   SLA  1          30006880
0325 0 1801          *   SRA  1          30006890
0326 0 D918          *   STD  1 24          STO WC AND H/D SW          30006900
0327 0 4C02 0334    *   BSC  L LOG8,C          BR IF NOT 1ST LINE          30006910
                                30006920
0329 0 C0C6          *   LD   PIDWS          LD MSG ID          30006930
032A 0 18D0          *   RTE  16          30006940
032B 0 407A          *   BSI  HEX          CONV AND PUT IN MSG          30006950
032C 0 C0C4          *   LD   PIDWS&1          LD RTN ID          30006960
032D 0 4C08 0334    *   BSC  L LOG8,&          BR IF NO RTN ID          30006970
032F 0 18D0          *   RTE  16          30006980
0330 0 4075          *   BSI  HEX          CONV AND PUT IN MSG          30006990
0331 0 C0C0          *   LD   PIDWS&2          LD RTN ADDR          30007000
0332 0 18D0          *   RTE  16          30007010
0333 0 4072          *   BSI  HEX          CONV AND PUT IN MSG          30007020
                                30007030
0334 0 C112          * LOG8 LD  1 18          LD TAB          30007040
0335 0 D200          *   STO  2 0          STO TAB IN I/O BUFF          30007050
0336 0 C303          *   LD   3 3          LD FIRST ALPHA ADDR          30007060
0337 0 D05B          *   STO  ALPHA          SET ALPHA SW          30007070
0338 0 4420 0393    *   BSI  L ALPHA,Z          IF NOT ZERO-GET MESS          30007080
033A 0 C304          *   LD   3 4          LD SEC ALPHA ADDR          30007090
033B 0 4420 0393    *   BSI  L ALPHA,Z          IF NOT ZERO-GET MESS          30007100
033D 0 C055          *   LD   ALPHA          TEST ALPHA SW          30007110
033E 0 4C18 0345    *   BSC  L LOG9,&-          BR IF NO ALPHA          30007120
0340 0 C11A          *   LD   1 26          30007130
0341 0 7400 03F2    *   MDX  L WC,0          SKIP IF NO DATA WORDS          30007140
0343 0 D200          *   STO  2 0          STO CR AND TAB IN MSG          30007150
0344 0 7201          *   MDX  2 1          30007160
                                30007170
0345 0 7314          * LOG9 MDX  3 20          XR3#DATA TABLE ADDR          30007180
0346 0 680B          *   STX  3 LOG12&1          STO DATA TABLE ADDR          30007190
0347 0 63F1          *   LDX  3 -15          SET TO CK DATA ID          30007200
0348 0 C918          *   LDD  1 24          LD WC AND H/D SW          30007210
0349 0 4C04 0351    * LOG10 BSC  L LOG12,E          BR IF PRINT DATA          30007220
034B 0 C918          * LOG11 LDD  1 24          LD WC AND H/D SW          30007230
034C 0 18C1          *   RTE  1          SET FOR NEXT BIT          30007240
034D 0 D918          *   STD  1 24          30007250
034E 0 7301          *   MDX  3 1          SET XR3 FOR NEXT BIT          30007260
034F 0 70F9          *   MDX  LOG10          LOOP UNTIL LAST BIT          30007270
0350 0 700B          *   MDX  LEND          30007280
                                30007290
0351 0 C700 0000    * LOG12 LD  L3 *-*          LD NEXT DATA WORD          30007300
                                30007310
                                30007320
0353 0 7289          *   MDX  2 -/77          30007330
0354 0 0004          * FOUR DC  4          THIS WORD SKIPPED          30007340
0355 0 7277          *   MDX  2 /77          30007350
0356 0 7005          *   MDX  LEND          END IF MSG BUF FULL          30007360
                                30007370
0357 0 18D0          *   RTE  16          30007380
0358 0 4C04 03B6    *   BSC  L DEC,E          CK HEX/DEC SW          30007390
035A 0 404B          *   BSI  HEX          CONV HEX IF SW OFF          30007400
035B 0 70EF          *   MDX  LOG11          RETURN FOR MORE          30007410
                                30007420
035C 0 C117          * LEND LD  1 23          30007430
035D 0 D200          *   STO  2 0          SET FORM FEED AND TERM          30007440
035E 0 6700 0028    *   LDX  L3 MESO          SET MESS START ADDR          30007450
0360 0 6B6F          *   STX  3 LOGT          30007460
0361 0 6893          *   STX  IOBSY          SET I/O BUSY          30007470
                                30007480
                                30007490
*-----*
* TYPE THE MESSAGE- THE FIRST TYPE

```

```

* COMMAND IS GIVEN THEN THE PROGRAM          30007500
* RETURNS TO THE SUPERVISOR TO RUN          30007510
* OTHER PROGRAMS UNTIL THE TYPE          30007520
* COMPLETE INTERRUPT OCCURS.          30007530
*-----*
* TYP XIO LOGT CHECK TYPE BUSY          30007540
* TYP XIO LOGT CHECK TYPE BUSY          30007550
* TYP XIO LOGT CHECK TYPE BUSY          30007560
* TYP XIO LOGT CHECK TYPE BUSY          30007570
* TYP XIO LOGT CHECK TYPE BUSY          30007580
* TYP XIO LOGT CHECK TYPE BUSY          30007590
* TYP XIO LOGT CHECK TYPE BUSY          30007600
* TYP XIO LOGT CHECK TYPE BUSY          30007610
* TYP XIO LOGT CHECK TYPE BUSY          30007620
* TYP XIO LOGT CHECK TYPE BUSY          30007630
* TYP XIO LOGT CHECK TYPE BUSY          30007640
* TYP XIO LOGT CHECK TYPE BUSY          30007650
* TYP XIO LOGT CHECK TYPE BUSY          30007660
* TYP XIO LOGT CHECK TYPE BUSY          30007670
* TYP XIO LOGT CHECK TYPE BUSY          30007680
* TYP XIO LOGT CHECK TYPE BUSY          30007690
* TYP XIO LOGT CHECK TYPE BUSY          30007700
* TYP XIO LOGT CHECK TYPE BUSY          30007710
* TYP XIO LOGT CHECK TYPE BUSY          30007720
* TYP XIO LOGT CHECK TYPE BUSY          30007730
* TYP XIO LOGT CHECK TYPE BUSY          30007740
* TYP XIO LOGT CHECK TYPE BUSY          30007750
* TYP XIO LOGT CHECK TYPE BUSY          30007760
* TYP XIO LOGT CHECK TYPE BUSY          30007770
* TYP XIO LOGT CHECK TYPE BUSY          30007780
* TYP XIO LOGT CHECK TYPE BUSY          30007790
* TYP XIO LOGT CHECK TYPE BUSY          30007800
* TYP XIO LOGT CHECK TYPE BUSY          30007810
* TYP XIO LOGT CHECK TYPE BUSY          30007820
* TYP XIO LOGT CHECK TYPE BUSY          30007830
* TYP XIO LOGT CHECK TYPE BUSY          30007840
* TYP XIO LOGT CHECK TYPE BUSY          30007850
* TYP XIO LOGT CHECK TYPE BUSY          30007860
* TYP XIO LOGT CHECK TYPE BUSY          30007870
* TYP XIO LOGT CHECK TYPE BUSY          30007880
* TYP XIO LOGT CHECK TYPE BUSY          30007890
* TYP XIO LOGT CHECK TYPE BUSY          30007900
* TYP XIO LOGT CHECK TYPE BUSY          30007910
* TYP XIO LOGT CHECK TYPE BUSY          30007920
* TYP XIO LOGT CHECK TYPE BUSY          30007930
* TYP XIO LOGT CHECK TYPE BUSY          30007940
* TYP XIO LOGT CHECK TYPE BUSY          30007950
* TYP XIO LOGT CHECK TYPE BUSY          30007960
* TYP XIO LOGT CHECK TYPE BUSY          30007970
* TYP XIO LOGT CHECK TYPE BUSY          30007980
* TYP XIO LOGT CHECK TYPE BUSY          30007990
* TYP XIO LOGT CHECK TYPE BUSY          30008000
* TYP XIO LOGT CHECK TYPE BUSY          30008010
* TYP XIO LOGT CHECK TYPE BUSY          30008020
* TYP XIO LOGT CHECK TYPE BUSY          30008030
* TYP XIO LOGT CHECK TYPE BUSY          30008040
* TYP XIO LOGT CHECK TYPE BUSY          30008050
* TYP XIO LOGT CHECK TYPE BUSY          30008060
* TYP XIO LOGT CHECK TYPE BUSY          30008070
* TYP XIO LOGT CHECK TYPE BUSY          30008080
* TYP XIO LOGT CHECK TYPE BUSY          30008090
* TYP XIO LOGT CHECK TYPE BUSY          30008100
* TYP XIO LOGT CHECK TYPE BUSY          30008110
* TYP XIO LOGT CHECK TYPE BUSY          30008120
* TYP XIO LOGT CHECK TYPE BUSY          30008130
* TYP XIO LOGT CHECK TYPE BUSY          30008140
* TYP XIO LOGT CHECK TYPE BUSY          30008150
* TYP XIO LOGT CHECK TYPE BUSY          30008160
* TYP XIO LOGT CHECK TYPE BUSY          30008170

```

```

03A2 0 0005      FIVE DC      5      THIS WORD SKIPPED      30008180
03A3 0 7277      MDX  2 /77      30008190
03A4 0 70B7      MDX      LFND      END IF MSG BUF FULL  30008200
*
03A5 0 70F0      *      MDX      ALPH1      GET ANOTHER WORD      30008210
*
*-----*
*      HEXADECIMAL CONVERSION SUBROUTINE
*-----*
03A6 0 0000      HEX  DC      *-*
03A7 0 7201      MDX  2 1      LEAVE SPACE      30008280
03A8 0 C0AB      LD      FOUR      SET CHAR CONVERT  30008290
03A9 0 D02E      STO      WCNT      COUNT      30008300
*
03AA 0 1810      HEX1 SRA      16      CLEAR ACC TO ZERO  30008310
03AB 0 1084      SLT      4      BRING IN NEXT CHAR 30008320
03AC 0 D001      STO      HEX2&1    30008330
03AD 0 C500 0000  HEX2 LD      L1 *-*      GET CHAR CODE      30008340
03AE 0 D200      STO      2 0      SET IN I/O BUFFER  30008350
03AF 0 7201      MDX  2 1      ADV I/O BUFFER ADDRS 30008360
03B0 0 74FF 03D8 MDX  L WCNT,-1    CHECK IF WORD HAS   30008370
03B1 0 70F6      MDX      HEX1      BEEN COMPLETELY CNVTD 30008380
*
03B4 0 4C80 03A6 *      BSC  I  HEX      RETURN      30008390
*
*-----*
*      DECIMAL CONVERSION SUBROUTINE
*-----*
03B6 0 C0EB      DEC  LD      FIVE      SET CHAR CONVERT  30008400
03B7 0 D820      STD      WCNT      COUNT      30008410
03B8 0 C020      LD      WCNT&1    CHECK IF WORD TO BE 30008420
03B9 0 4C10 03BF BSC  L  DEC1,-    CONVERTED IS NEGATIVE 30008430
*
03BB 0 C031      LD      NEGS      SFT A NEGATIVE SIGN 30008440
03BC 0 D200      STO      2 0      IN MESSAGE      30008450
03BD 0 10A0      SLT      32
03BE 0 901A      S      WCNT&1    GENERATE 2S COMPLEMT 30008460
*
03BF 0 7205      DEC1 MDX  2 5      SET TO STD LOW ORDR 1ST 30008470
03C0 0 18D0      RTE      16      PUT WORD IN EXTEN  30008480
*
03C1 0 1810      DEC2 SRA      16      CLEAR ACC      30008490
03C2 0 A814      D      TFN      CONVERT VALUE      30008500
03C3 0 18D0      RTE      16      BRING IN REMAINDER 30008510
03C4 0 D001      STO      DEC3&1    STO NXT DIGIT VALUE 30008520
03C5 0 C500 0000  DEC3 LD      L1 *-*      GET CHAR CODE      30008530
03C6 0 D200      STO      2 0      SET CHAR CODE IN OUT 30008540
03C7 0 72FF      MDX  2 -1      PUT AREA      30008550
03C8 0 74FF 03D8 MDX  L WCNT,-1    CHECK IF END OF CONV 30008560
03C9 0 70F5      MDX      DEC2
*
03CC 0 7206      MDX  2 6      FINISHED      30008570
03CD 0 C000      LD      *      SET ACC NON ZERO  30008580
03CE 0 4C00 0349 BSC  L  LOG10     RETURN      30008590
*
*-----*
*      LOG ROUTINE CONSTANTS AND STORAGE
*-----*
03D0 0000      BSS  F  0
03D0 0 0028      LOGT DC      MESA      0      MESSAGE ADDR      30008600
03D1 0 0F00      DC      /0F00      1      1053 SENSE IOCC  30008610
03D2 0 03D4      DC      LOGT&4     2      CHAR ADDR      30008620
03D3 0 0900      DC      /0900      3      1053 WRITE IOCC 30008630
03D4 0 0000      DC      *-*      4      CHAR      30008640
03D5 0 0000      DC      *-*      5      HALF WORD SW     30008650

```

```

03D6 0 0000      DC      *-*      6      TYP DLY CNTR      30008860
03D7 0 000A      TFN  DC      /000A      30008870
*
03D8 0000      BSS  E  0      30008880
03D8 0 0000      WCNT DC      *-*      30008890
03D9 0 0000      DC      *-*      30008900
03DA 0 C400      TABL DC      /C400      0      TABL&0      30008910
03DB 0 FC00      DC      /FC00      1      1      30008920
03DC 0 D800      DC      /D800      2      2      30008930
03DD 0 DC00      DC      /DC00      3      3      30008940
03DE 0 F000      DC      /F000      4      4      30008950
03DF 0 F400      DC      /F400      5      5      30008960
03E0 0 D000      DC      /D000      6      6      30008970
03E1 0 D400      DC      /D400      7      7      30008980
03E2 0 E400      DC      /E400      8      8      30008990
03E3 0 E000      DC      /E000      9      9      30009000
03E4 0 3E00      DC      /3E00      A      10     30009010
03E5 0 1A00      DC      /1A00      B      11     30009020
03E6 0 1E00      DC      /1E00      C      12     30009030
03E7 0 3200      DC      /3200      D      13     30009040
03E8 0 3600      DC      /3600      E      14     30009050
03E9 0 1200      DC      /1200      F      15     30009060
03FA 0 2100      DC      /2100      SPACE 16     30009070
03EB 0 6200      DC      /6200      R      17     30009080
03EC 0 4121      DC      /4121      TAB    18     30009090
03ED 0 8400      DC      /8400      MINUS 19     30009100
03EE 0 0936      DC      /0936      SFT RED 20    30009110
03EF 0 053E      DC      /053E      SFT BLK 21    30009120
03F0 0 FFFF      DC      /FFFF      TERM   22    30009130
03F1 0 03FF      DC      /03FF      FORM FD 23    30009140
03F2 0 0000      WC   DC      *-*      WD CNTL 24    30009150
03F3 0 0000      DC      *-*      HEX/DEC 25   30009160
03F4 0 8141      DC      /8141      CR & TAB 26  30009170
03F5 0004      BSS  4 WARNING DON'T REMOVE USED FOR DIMAL 30009180
*
*****
*      HALT ROUTINE
*****
*      THIS ROUTINE IS ENTERED FROM
*      ADDRESS STOP BY PRESSING PROGRAM
*      STOP. THE ROUTINE WILL SAVE THE STATUS
*      OF THE MACHINE THE ROUTINE WILL
*      RESTORE THIS STATUS WHEN STARTED FROM
*      THE WAIT OR BY NORMAL MONITOR RESTART.
*      PRESS RESET AND THEN START.
*-----*
*      KEEP ILT5 AT /03F9
*
03F9 0 0000      ILT5 DC      *-*
03FA 0 1000      NOP
03FB 0 D814      STD      AQ5      *****
03FC 0 0815      XIO      DSW5-1    SAVE AQ      30009330
03FD 0 4C80 0015 BSC  I  ILIR,-Z    SENSE AND RESET DSW 30009340
*
*      GO SERVICE INTR OR
*      GO TO STOP
*
03FF 0 4C20 0404 BSC  L  STOP,Z
0401 0 4480 01BE BSI  I  IL5
0403 0 7014      MDX      SAQ
*
*-----*
*      STOP SUBROUTINE -SAVES STATUS THEN WAIT
*      START BY PRESSING START OR RESET AND
*      START.
*-----*
0404 0 C0FF      STOP LD      STOP
0405 0 D00C      STO      STPSW      SET DELAY CNT
0406 0 74FF 0412 STOP1 MDX  L  STPSW,-1  DELAY TO CYCLE
0408 0 70FD      MDX      STOP1      DOWN INTERRUPTS
0409 0 6808      STX      STPSW      SET STOP SWITCH
040A 0 3001      WAIT     1      WAIT 3001      30009470

```

```
*
040B 0 1810      GD   SRA   16          30009540
040C 0 0005      STN   STPSW  RESET STOP SW  30009550
040D 0 080E      XIO   RDBSW  READ BIT SWS  30009560
040E 0 7009      MDX   SAQ    RESTORE AQ   30009570
*
0410 0000        BSS   F 0          30009580
0410 0 0000      AQ5   DC   *-*      STG FOR INT5 AND HALT 30009590
0411 0 0000      DC     *-*      ACC AND EXT REGS   30009600
0412 0 0000      STPSW DC   *-*      STOP SW           30009610
0413 0 3F01      DSW5  DC   /3F01     INT 5 SENSE DSW COM 30009620
*
0414 0 C0E4      ASTCP LD   ILT5     LD IAR ADDRS      30009630
0415 0 F008      EOR   SADRS    COMPARE WITH SWS  30009640
0416 0 4C18 0404 BSC   L  STOP,&-  RETURN IF NOT EQUAL 30009650
0418 0 C8F7      SAQ   LDD   AQ5    RESTORE AQ       30009660
0419 0 4CC0 03F9 BOSC  I  ILT5     30009670
*
041C 0000        BSS   F 0          30009680
041C 0 041E      RDBSW DC   SADRS    30009690
041D 0 3A00      DC     /3A00     READ BIT SW IOCC  30009700
041E 0 0000      SADRS DC   *-*      30009710
*
*-----*
* DELAY SUBROUTINE - ENTERED WHEN RUNNING 30009720
* IN INTERRUPT MODE. IT PROVIDES A DELAY 30009730
* BETWEEN EXECUTION OF EACH MAIN LINE 30009740
* INSTRUCTION. THIS DELAY CAN BE INCREASED 30009750
* BY SETTING A COUNT IN MONITOR FUNCTION 2 30009760
* SWITCHES. 30009770
*-----*
041F 0 C0FE      DELAY LD   SADRS    LD DELAY COUNT 30009780
0420 0 D004      STO   DELYC     SAVE 30009790
0421 0 74FF 0425 DELY1 MDX  L  DELYC,-1  DECR DELAY COUNT 30009800
0423 0 70FD      MDX   DELY1     LOOP 30009810
0424 0 70F3      MDX   SAQ      RETURN TO MAINLINE 30009820
0425 0 0000      DELYC DC   *-*    DELAY COUNT 30009830
*
*-----*
*****
* MONITOR END ROUTINE 30009840
*-----*
* THIS ROUTINE IS ENTERED BY AN END 30009850
* CALL FROM ANY TEST PROGRAM. IT WILL 30009860
* FIRST CHECK TO SEE IF THE MONITOR 30009870
* LOOP PROGRAM SWITCH IS ON. 30009880
* IF NOT, AN END MESSAGE 30009890
* IS TYPED. IF NOT OVERLAP MODE, 30009900
* THE ROUTINE WILL BRANCH TO LOAD TO 30009910
* ATTEMPT TO LOAD ANOTHER PROGRAM. 30009920
* IF LOOP, THE ROUTINE WILL BRANCH 30009930
* TO THE LOOP ADDRESS GIVEN IN THE 30009940
* PROGRAM CONTROL TABLE. 30009950
*-----*
* CALLING SEQUENCE 30009960
* BSI I END 30009970
*-----*
0426 0 0000      MEND  DC   *-*      30009980
0427 0 6780 023E LDX   I3  MXEQ2&1  X3#MLSCF ADDR 30009990
0429 0 C3FA      LD    3 -6        LD CONTROL SWS  30010000
042A 0 EC00 020D OR    L  MSWD     OR WITH MON SWS 30010010
042C 0 1804      SRA   4          CK FOR LOOP PROGRAM 30010020
042D 0 4F84 FFFE BSC   I3 -2,&F   BR IF LOOP 30010030
*
042F 0 10A0      SLT   32         30010040
0430 0 C3F7      LD    3 -9        LD PID 30010050
0431 0 D400 050E STO   L  MTABL&2 SET IN MSG 30010060
```

```
0433 0 C015      LD    MMESF      LD END MESSAGE 30010220
0434 0 6600 8004 LDX   L2 /8004   SET WD CNTL 30010230
0436 0 4400 04F3 BSI   L  MLOG    LOG MESSAGE 30010240
*
0438 0 C400 050D LD    L  PGMN    CK IF OVER LAP 30010250
043A 0 1801      SRA   1          30010260
043B 0 4C20 0271 BSC   L  MXEQ7,Z BR IF OVLAP 30010270
043D 0 C400 0155 LD    L  /155    30010280
043F 0 4C04 0271 BSC   L  MXEQ7,E BR IF LD FROM TAPE 30010290
*
0441 0 D400 050D STO   L  PGMN    IF NOT OVLAP 30010300
0443 0 6700 05DC LDX   L3 /05DC   GO LOAD NEXT PRGM 30010310
0445 0 6F00 015C STX   L3 NLOC    30010320
0447 0 4C00 0581 BSC   L  LOAD    30010330
*
MMESF DC   AEND 30010340
*
*****
* MONITOR INTERRUPT ROUTINES 30010350
*****
*-----*
* INTERRUPT LEVEL ZERO 30010360
*-----*
* KEEP ILT0 AT /044A 30010370
*
044A 0 0000      ILT0 DC   *-*      30010380
044B 0 1000      NOP  30010390
044C 0 D855      STD   AQO        SAVE AQ 30010400
044D 0 6B09      STX   3 X30&1   SAVE XR3 30010410
044E 0 2809      STS   SSO        STORE STATUS 30010420
044F 0 085C      XIO   ILSW-1    SENSE AND 30010430
0450 0 630F      LDX   3 15      30010440
0451 0 4820      BSC   Z          SKIP IF NO ILSW 30010450
0452 0 1340      SLCA  3          FIND BIT IN ILSW 30010460
0453 0 4780 016B BSI   I3 ILO-15 GO TO USER ROUTINE 30010470
0455 0 C84C      LDD   AQO        RESTORE AQ 30010480
0456 0 6700 0000 X30   LDX  L3 *-* RSTORE XR3 30010490
0458 0 2000      SSO   LDS  *-*   LOAD STATUS 30010500
0459 0 4CC0 044A BOSC  I  ILT0    30010510
*
*-----*
* INTERRUPT LEVEL ONE 30010520
*-----*
* KEEP ILT1 AT /045B 30010530
*
045B 0 0000      ILT1 DC   *-*      30010540
045C 0 1000      NOP  30010550
045D 0 D846      STD   AQ1        SAVE AQ 30010560
045E 0 6B09      STX   3 X31&1   SAVE XR3 30010570
045F 0 2809      STS   SS1        STORE STATUS 30010580
0460 0 084B      XIO   ILSW-1    SENSE AND 30010590
0461 0 630F      LDX   3 15      30010600
0462 0 4820      BSC   Z          SKIP IF NO ILSW 30010610
0463 0 1340      SLCA  3          FIND BIT IN ILSW 30010620
0464 0 4780 017B BSI   I3 IL1-15 GO TO USER ROUTINE 30010630
0466 0 C83D      LDD   AQ1        RESTORE AQ 30010640
0467 0 6700 0000 X31   LDX  L3 *-* RSTORE XR3 30010650
0469 0 2000      SSI   LDS  *-*   LOAD STATUS 30010660
046A 0 4CC0 045B BOSC  I  ILT1    30010670
*
*-----*
* INTERRUPT LEVEL TWO 30010680
*-----*
* KEEP ILT2 AT /046C 30010690
*
046C 0 0000      ILT2 DC   *-*      30010700
046D 0 1000      NOP  30010710
046E 0 D837      STD   AQ2        SAVE AQ 30010720
046F 0 6B09      STX   3 X32&1   SAVE XR3 30010730
0470 0 2809      STS   SS2        STORE STATUS 30010740
```

1130 DIAGNOSTIC MONITOR II

1130 DIAGNOSTIC MONITOR II

```

0471 0 083A      XIO  ILSW-1  SENSE AND      30010900
0472 0 630F      LDX  3 15      30010910
0473 0 4820      BSC  Z        SKIP IF NO ILSW  30010920
0474 0 1340      SLCA 3        FIND BIT IN ILSW 30010930
0475 0 4780 018B BSI  I3 IL2-15 GO TO USER ROUTINE 30010940
0477 0 C82F      LDD  AQ2      RESTORE AQ        30010950
0478 0 6700 0000 X32  LDX  L3 *-* RESTORE XR3      30010960
047A 0 2000      SS2  LDS  *-*  RESTORE STATUS  30010970
047B 0 4CC0 046C BOSC I  ILT2  30010980
*
*-----*
*          INTERRUPT LEVEL THREE
*-----*
*          KEEP ILT3 AT /047D
*
ILT3 DC  *-*
NOP
STO  AQ3      SAVE AQ        30011060
STX  3 X33&1 SAVE XR3      30011070
STS  SS3      STORE STATUS  30011080
XIO  ILSW-1   SENSE AND      30011090
LDX  3 15     30011100
BSC  Z        SKIP IF NO ILSW  30011110
SLCA 3        FIND BIT IN ILSW 30011120
BSI  I3 IL3-15 GO TO USER ROUTINE 30011130
LDD  AQ3      RESTORE AQ        30011140
X33  LDX  L3 *-* RESTORE XR3      30011150
SS3  LDS  *-*  RESTORE STATUS  30011160
BOSC I  ILT3  30011170
*
*-----*
*          INTERRUPT LEVEL FOUR
*-----*
*          KEEP ILT4 AT /048E
*
ILT4 DC  *-*
NOP
STO  AQ4      SAVE AQ        30011240
STX  3 X34&1 SAVE XR3      30011260
STS  SS4      STORE STATUS  30011270
XIO  ILSW-1   SENSE AND      30011280
LDX  3 15     30011290
SLCA 3        FIND BIT IN ILSW 30011300
BSI  I3 IL4-15 GO TO USER ROUTINE 30011310
LDD  AQ4      RESTORE AQ        30011320
X34  LDX  L3 *-* RESTORE XR3      30011330
SS4  LDS  *-*  RESTORE STATUS  30011340
BOSC I  ILT4  30011350
*
*-----*
*          KEYBOARD RESPONSE BIT 1
*-----*
*
MKBR BSI  I  SVKB GO TO KB USER RTN 30011410
MDX  MC12  30011420
*
*-----*
*          INTERRUPT CONSTANTS AND STORAGE
*-----*
*
BSS  E  0      30011480
AQ0  DC  *-*   30011490
AQ1  DC  *-*   30011500
AQ2  DC  *-*   30011510
AQ3  DC  *-*   30011520
AQ4  DC  *-*   30011530
      DC  *-*   30011540
      DC  *-*   30011550
      DC  *-*   30011560
      DC  *-*   30011570

```

```

04AB 0 0000
04AC 0 0455
04AD 0 0300
04AE 0 0000
04AF 0 0000
04B0 0 0000
04B1 0 08FA
04B2 0 7400 0217
04B4 0 30F9
04B5 0 D05B
04B6 0 C0F9
04B7 0 90F4
04B8 0 1804
04B9 0 D056
04BA 0 6700 04C0
04BC 0 6F00 0217
04BE 0 4C80 04B0
04C0 0 6305
04C1 0 6230
04C2 0 10A0
04C3 0 4038
04C4 0 4C00 0271
04C6 0 0000
04C7 0 0812
04C8 0 D011
04C9 0 6303
04CA 0 1340
04CB 0 F400 0205
04CD 0 D0F0
04CE 0 68E0
04CF 0 C00A
04D0 0 4F82 04DB
04D2 0 C0DB
04D3 0 6780 04AF
04D5 0 4C02 04CA
04D7 0 4C80 04C6
04DA 0000
04DA 0 0000
04DB 0 0F01
04DC 0 04F2
04DD 0 049E
04DE 0 04DF
04DF 0 4480 01BB
04E1 0 70F0

```

```

DC  *-*
*
XSC1 DC  XS1
ILSW DC  /0300 INT 4 SENSE ILSW
MCIST DC *-*  CONSOLE INTER STOR
MCX2 DC  *-*  CON INT RTN XR2 STORE
*
*****
*          INTERRUPT ERROR
*****
*
IERR DC  *-*
XIO  ILSW-1  SENSE ILSW
MDX  L  IERM,0 IF INTERRUPT WILL NOT
WAIT  -7      TURN OFF- WAIT HERE
*
STO  MTABLE5 STO ILSW
LD  IERR
S  XSC1  CALCULATE INTERRUPT
SRA  4    LEVEL
STO  MTABLE4 STO INTERRUPT LEVEL
LDX  L3 IERR2
STX  L3 IERM
BSC  I  IERR
*
IERR2 LDX  3 5 SET MSG NO
LDS  2 /0030 SET DATA WORD ID
SLT  32
BSI  MERR LOG MSG
BSC  L  MXFQ7 GO TO SUPERVISOR
*
*****
*          CONSOLE INTERRUPT ROUTINE
*****
* THIS ROUTINE WILL SERVICE THE CONSOLE
* INTERRUPT. IF THE INTERRUPT IS DUE TO A
* USER XIO IT WILL BRANCH TO THE USER
* INTERRUPT ROUTINE.
*-----*
*
MCINT DC  *-*
XIO  MCDSW SENSE, RESET
STO  MCDSW AND STORE DSW
LDX  3 3
MC11 SLCA 3 LOOK FOR DSW BIT
EDR  L  K8000 TAKE OUT ZERO BIT
STO  MCIST SAVE REMAINING DSW
STX  3 MCX2 SAVE DSW POSITION CNT
LD  MCDSW SET DSW IN ACCUM
BSC  I3 MCDSW&1,C GO TO SERVICE RTN
MC12 LD  MCIST RETURN - LD DSW
LDX  I3 MCX2 RESTORE POSITION CNT
BSC  L  MC11,C IF NOT 0-CONTINUE CK
BSC  I  MCINT RETURN TO MAINLINE
*
BSS  E  0
MCDSW DC  *-* CONSOLE DSW
DC  /0F01
DC  MMR MANUAL REQ 2
DC  MKBR KB RESPONSE 1
DC  MTWR TW RESPONSE 0
*
*-----*
*          CONSOLE PRINTER RESPONSE BIT 0
*-----*
*
MTWR BSI  I  RQTY GO TO TW USER RTN
MDX  MC12

```

```
*-----*
* MANUAL REQUEST BIT 2
*-----*
04E2 0 1003 MMR SLA 3 CK CON/KB SW
04E3 0 4C10 04EF BSC L MMR1,- BR IF KB
04E5 0 6700 04EA LDX L3 MMRBS-1 RESTORE RQKB
04E7 0 6F00 01BC STX L3 RQKB
04E9 0 7001 MDX MMRBS
*
04EA 0 0000 DC *-* ENTRY FROM RQKB
04EB 0 402D MMRBS BSI MRBS
04EC 0 6F00 0213 STX L3 MLSCF SET RD SW RETURN
04EE 0 70E3 MDX MCI2
*
04EF 0 C0EA MMR1 LD MCDSW
04F0 0 4480 01BC BSI I RQKB
04F2 0 70DF MDX MCI2
*
*****
* MONITOR LOG ROUTINE
*****
* THIS ROUTINE SETS THE MONITOR MESSAGE
* ID NUMBER, WORD COUNT AND ALPHA MESSAGE
* ADDRESS INTO THE MESSAGE TABLE THEN
* CALLS ON LOG OR ERROR.
*-----*
* CALLING SEQUENCE
* BSI MLOG
* ACC # ALPHA ADDRS 1
* EXT # ALPHA ADDRS 2
* XR2 # WORD CNT AND MESS ID
*-----*
04F3 0 0000 MLOG DC *-*
04F4 0 D815 STD MALPH STO ALPHA ADDRESSES
04F5 0 6A13 STX 2 MLOG2&2 STO WD CNTL
04F6 0 6B10 STX 3 MLOG2 STO MSG NO.
04F7 0 4480 0163 BSI I LOG
04F9 0 0507 DC MLOG2 MESS ADDR
*
04FA 0 4C80 04F3 BSC I MLOG
*
04FC 0 0000 MERR DC *-*
04FD 0 D80C STD MALPH STO ALPHA ADDRESSES
04FE 0 6A0A STX 2 MLOG2&2 STO WD CNTL
04FF 0 6B07 STX 3 MLOG2 STO MSG NO.
0500 0 4480 0162 BSI I ERROR
0502 0 0507 DC MLOG2 MESS ADDR
0503 0 0000 DC /0000
*
0504 0 4C80 04FC BSC I MERR
*-----*
* MONITOR MESSAGE TABLE
*-----*
0506 0001 BSS E 1
0507 0 0000 MLOG2 DC *-* MESSAGE NUMBER
0508 0 0000 DC *-* HEX/DEC
0509 0 0000 DC *-* WORD CONTROL
050A 0 0000 MALPH DC *-* ALPHA ADDRS 1
050B 0 0000 DC *-* ALPHA ADDRS 2
050C 0 0000 MTABL DC *-* BIT SWITCH READINGS
050D 0 0000 PGMM DC *-* PROGRAM NUMBER
050E 0 0000 DC *-* PID
050F 0 0000 DC *-* STORAGE ADDR
```

```
30012260
30012270
30012280
30012290
30012300
30012310
30012320
30012330
30012340
30012350
30012360
30012370
30012380
30012390
30012400
30012410
30012420
30012430
30012440
30012450
30012460
30012470
30012480
30012490
30012500
30012510
30012520
30012530
30012540
30012550
30012560
30012570
30012580
30012590
30012600
30012610
30012620
30012630
30012640
30012650
30012660
30012670
30012680
30012690
30012700
30012710
30012720
30012730
30012740
30012750
30012760
30012770
30012780
30012790
30012800
30012810
30012820
30012830
30012840
30012850
30012860
30012870
30012880
30012890
30012900
30012910
30012920
30012930
```

```
0510 0 0000
0511 0 0000
0512 0 8000
0513 0 059F
0514 0 0000
0515 0004
*
0519 0 0000
051A 0 6300
051B 0 6F00 0204
*
051D 0 0834
051E 0 C0ED
051F 0 1004
0520 0 180C
0521 0 D001
0522 0 6700 0000
*
0524 0 C400 0205
0526 0 1800
0527 0 630F
0528 0 1340
0529 0 C700 01BF
052B 0 6700 0579
052D 0 4C88 0519
*
052F 0 D007
0530 0 C0DB
0531 0 180F
0532 0 D007
*
0533 0 C0DB
0534 0 1008
0535 0 1808
0536 0 6700 0000
0538 0 73FA
0539 0 D700 0000
*
053B 0 C3FD
053C 0 D0D1
*
053D 0 C400 020F
053F 0 100B
0540 0 4C10 054B
*
0542 0 C400 048E
0544 0 D480 0014
0546 0 6780 0014
0548 0 7301
0549 0 6F00 048E
054B 0 6700 055A
054D 0 4C80 0519
*
054F 0 0000
0550 0 4C80 054F
```

```
DC *-* INTERRUPT LEVEL
DC *-* ILSW/SW
X8000 DC /8000
DC ALD
DC /0000
BSS 4 WARNING DON'T REMOVE USED FOR DIMAL
*
*****
* MONITOR READ BIT SWITCH
* ROUTINE
*****
* THIS ROUTINE READS THE BIT SWITCHES IN
* INTERRUPT AND PLACES THE DATA IN SWITCHES
* 8-15 IN THE PCT OF THE PROGRAM IDENTIFIED
* BY SWITCHES 4-7. FOLLOWING THE INTERRUPT
* THE SUPERVISER WILL RETURN TO THIS ROUTINE
* IN MAINLINE TO CHECK THE FUNCTION ZERO
* SWITCHES FOR HALT AND LOAD.
*-----*
MRBS DC *-*
LDX 3 0 SET FOR NO MLSCF ENTRY
STX L3 XLECT SET SCAN TO START AT MON
*
XIO RDBS READ BIT SWITCHES
LD MTABL LOAD BIT SW
SLA 4 CLEAR ALL BUT PRGM
SRA 12 NUMBER BITS
STO *&1 STO NUM TO BE VERIFIED
LDX L3 *-* XR3#PROGRAM NUMBER
*
LD L X8000
SRA 3 0
LDX 3 15
SLCA 3 0
LD L3 MPIDT-15 LD MLSCF ADDR
LDX L3 MRBSE LD ERROR RETURN
BSC I MRBS,& BR IF INVALID PGM NO.
*
STO MRBS3&1 STORE SW ADDR
LD MTABL LOAD BIT SW
SRA 14 GET FUNCTION
STO MRBS4&1 STORE FUNCTION
*
LD MTABL LOAD BIT SW
SLA 8 STRIP OFF
SRA 8 ALL BUT DATA
MRBS3 LDX L3 *-* X1 # MLSCF ADRS
MDX 3 -6 ADV TO SW ADDR
MRBS4 STO L3 *-* PUT SW DATA IN PROG
*
LD 3 -3 AT PROPER FUNC LEVEL
STO MTABL&2 LD PID
*
LD L MSW2 LD MONITOR SW 2
SLA 11
BSC L MRBSC,- BR OUT IF NO UTILITY
*
MRBSB LD L ILT4
STO I UTLY SET UTILITY RETURN
LDX I3 UTLY
MDX 3 1
STX L3 ILT4 SET BR TO UTILITY
MRBSC LDX L3 MRBS5 LD NORMAL RETURN
BSC I MRBS BR OUT OF INTRPT
*
MRBSD DC *-* KEEP MRBSD AT /054F
BSC I MRBSD DUMMY RETURN IF
UTILITY NOT IN CORE
```

```
30012940
30012950
30012960
30012970
30012980
30012990
30013000
30013010
30013020
30013030
30013040
30013050
30013060
30013070
30013080
30013090
30013100
30013110
30013120
30013130
30013140
30013150
30013160
30013170
30013180
30013190
30013200
30013210
30013220
30013230
30013240
30013250
30013260
30013270
30013280
30013290
30013300
30013310
30013320
30013330
30013340
30013350
30013360
30013370
30013380
30013390
30013400
30013410
30013420
30013430
30013440
30013450
30013460
30013470
30013480
30013490
30013500
30013510
30013520
30013530
30013540
30013550
30013560
30013570
30013580
30013590
30013600
30013610
```

```

*-----*
*   CONSTANTS AND STORAGE FOR   30013620
*   READ BIT SW ROUTINE         30013630
*-----*
0552 0000          BSS F 0          30013640
0552 0 050C      RDBS DC          MTABL IOCC TO READ BIT SWS 30013650
0553 0 3A00          DC          /3A00          30013660
0554 0 05B1      MMES1 DC          ASWS          MESSAGE ADDRESSES 30013670
0555 0 059B      DC          ASIVD          SWS INVLD          30013680
0556 0 05B6      MMES3 DC          AHALT          30013690
0557 0 05A4      MMES4 DC          ACKSM          30013700
0558 0 05A8      MMES7 DC          ADVCR          30013710
0559 0001          BSS          I WARNING DON'T REMOVE USED FOR DIMAL 30013720
*-----*
*   RETURN HERE AFTER INTERRUPT AND CHECK THE 30013730
*   SWITCHES FOR HALT AND LOAD              30013740
*-----*
055A 0 10A0      MRBS5 SLT          32          30013750
055B 0 C0B0          LD          MTABL          LD SW ENTRY          30013760
055C 0 180E          SRA          14          SET TO LOOK AT FUNC SWS 30013770
055D 0 4C20 0564  BSC L          MRBS6,Z      BR UNLESS FUNC 0    30013780
*-----*
055F 0 C0AC          LD          MTABL          LD SW ENTRY          30013790
0560 0 100F          SLA          15          LOOK AT HALT SW          30013800
0561 0 4820          BSC          Z          30013810
0562 0 C0F3          LD          MMES3          PRINT MSG IF HALT 30013820
0563 0 18D0          RTE          16          30013830
*-----*
0564 0 6301      MRBS6 LDX          3 1          SET MSG NO.          30013840
0565 0 6205          LDX          2 /0005      SET WD CNTL          30013850
0566 0 C0ED          LD          MMES1          30013860
0567 0 4400 04F3  BSI L          MLOG          30013870
*-----*
0569 0 C400 020F  LD L          MSW2          LD MONITOR SW 2    30013880
056B 0 1890          SRT          16          30013890
056C 0 D400 020F  STO L          MSW2          RESET SW2          30013900
*-----*
056F 0 1099          SLT          25          30013910
*-----*
056F 0 4C02 0581  BSC L          LOAD,C      GO RELOAD          30013920
*-----*
0571 0 1001          SLA          1          30013930
0572 0 4C10 0220  BSC L          MXEQ,-      BR IF NOT DELAY      30013940
*-----*
0574 0 6700 041F  LDX L3          DELAY          30013950
0576 0 6F00 0015  STX L3          ILIR          SET FOR DELAY          30013960
0578 0 7006          MDX          MRBSF          30013970
*-----*
*   COME HERE IF THE PROGRAM IDENTIFIED IN   30013980
*   SWITCHES 4-7 IS NOT IN CORE. PRINT SWITCHES 30013990
*   INVALID MESSAGE.                        30014000
*-----*
0579 0 10A0      MRBS6 SLT          32          30014010
057A 0 CODA          LD          MMES1&1      PID NO GOOD          30014020
057B 0 6201          LDX          2 /0001      SET WD CNTL          30014030
057C 0 6301          LDX          3 1          SET MSG NO.          30014040
057D 0 4400 04FC  BSI L          MERR          LOG MES - SWS INVLD 30014050
*-----*
057F 0 4C00 0220  MRBSF BSC L          MXFQ          GO TO SUPERVISOR 30014060
*-----*
*****
*   LOAD NEXT PROGRAM                      30014070
*-----*

```

```

0581 0 4C28 058A
0583 0 1810
0584 0 D400 050D
0586 0 6700 050C
0588 0 6F00 015C

```

```

058A 0 4400 02F8
058C 0 0510
058D 0 6700 017A
058F 0 6078

```

```

0590 0 6304
0591 0 C0C5
0592 0 7002

```

```

0593 0 6303
0594 0 C0C3
0595 0 1890
0596 0 6200
0597 0 4400 04FC
0599 0 4C00 0220

```

```

059B 0 9A92
059C 0 9A21
059D 0 2276
059E 0 B600
059F 0 5E32
05A0 0 FFFF
05A1 0 0536
05A2 0 7632
05A3 0 FFFF
05A4 0 1E5A
05A5 0 9AB2
05A6 0 7200
05A7 0 FFFF
05A8 0 52B6
05A9 0 6221
05AA 0 1E52
05AB 0 6236
05AC 0 FFFF
05AD 0 059A
05AF 0 9E3E
05AF 0 629E
05B0 0 FFFF
05B1 0 9A92
05B2 0 9A21
05B3 0 2156
05B4 0 2232
05B5 0 FFFF
05B6 0 263E
05B7 0 5E9E
05B8 0 FFFF
05B9 0 76B2
05BA 0 7221
05BB 0 2156
05BC 0 2232
05BD 0 2121

```

```

*****
*
LOAD RSC L LOAD1,8Z BR IF CONTINUE LOAD
SRA 16
STO L PGMN RESET PRGM NUMBER
LDX L3 /05DC
STX L3 NLOC RESFT RELOCATION

```

```

*
LOAD1 BSI L LOGI GO PRINT CD MSG
DC MTABL&4
*
LOAD2 LDX L3 ILO KFEF LOAD2 AT /058D
LDX LOADR REQUIRED BY LOADER
LOAD NEXT PRGM

```

```

*-----*
*   PRINT LOADER ERROR MESSAGES
*-----*

```

```

*
CKSUM LDX 3 4 SET MSG NUMBER
LD MMES4
MDX LDERM PRINT CKSUM MSG

```

```

*
OVCR LDX 3 3 SET MSG NUMBER
LD MMFS7
LDERM SRT 16
LDX 2 0
BSI L MFRR GO PRINT MSG
BSC L MXFQ

```

```

*****
*   MONITOR ALPHA PHRASES
*****

```

```

ASIVD DC /9A92 SW SWS INVLD
DC /9A21 S
AIVD DC /2276 IN INVLD
DC /B600 V
ALD DC /5F32 LD LD
DC /FFFF
AEND DC /0536 F END
DC /7632 ND
DC /FFFF
ACKSM DC /1F5A CK CKSUM
DC /9AB2 SU
DC /7200 M
DC /FFFF
AOVCR DC /52B6 OV OVR CORE
DC /6221 R
DC /1E52 CO
DC /6236 RE
DC /FFFF
ASTRT DC /059A S START
DC /9E3E TA
DC /629E RT
DC /FFFF
ASWS DC /9A92 SW SWS
DC /9A21 S
APID DC /2156 P
DC /2232 ID
DC /FFFF
AHALT DC /263F HA HALT
DC /5E9E LT
DC /FFFF
ADRLD DC /76B2 NU NUM PID ADRS RELF
DC /7221 M
DC /2156 P
DC /2232 ID
DC /2121

```

```

30014300
30014310
30014320
30014330
30014340
30014350
30014360
30014370
30014380
30014390
30014400
30014410
30014420
30014430
30014440
30014450
30014460
30014470
30014480
30014490
30014500
30014510
30014520
30014530
30014540
30014550
30014560
30014570
30014580
30014590
30014600
30014610
30014620
30014630
30014640
30014650
30014660
30014670
30014680
30014690
30014700
30014710
30014720
30014730
30014740
30014750
30014760
30014770
30014780
30014790
30014800
30014810
30014820
30014830
30014840
30014850
30014860
30014870
30014880
30014890
30014900
30014910
30014920
30014930
30014940
30014950
30014960
30014970

```

```

05BF 0 3E32      DC      /3E32      AD      30014980
05BF 0 629A      DC      /629A      RS      30014990
05C0 0 2162      DC      /2162      R       30015000
05C1 0 365E      DC      /365E      EL      30015010
05C2 0 1221      DC      /1221      F       30015020
05C3 0 FFFF      DC      /FFFF
*
*****
*          LOADER/MONITOR INTERFACE
*****
*
0156          ORG      /156          30015040
0156 0 4C00 0219 BSC L INLIZ      BLANK CD RET TO MONITOR 30015050
0158 0 4C00 0590 BSC L CKSUM      30015060
015A 0 4C00 0593 BSC L OVCR       30015070
015C 0 05DC      NLOC DC /5DC      NEXT AVAILABLE LOCATION 30015080
015D 0 0000      RELF DC /0000     RELOCATION FACTOR        30015090
015E 0 05DC      DC      /5DC      SET BASE ADDRS         30015100
*
0078 0          LOADR EQU /0078     START LOAD ADDRS       30015110
0028 0          MESO EQU /0028     MESO # LOADER INPUT AREA 30015120
05DC          ORG      /5DC          30015130
05DC 058D      END      LOAD2        30015140
NO STATEMENTS FLAGGED IN THE ABOVE ASSEMBLY

```

```

ACKSM 05A4 0557
ADDRS 0168
ADRLD 05B9 0200
AFND 05A1 0449
AHALT 05B6 0556
AIVD 059D
ALD 059F 0201 0513
ALPHA 0393 0337 0338 033B 033D 0399
ALPH1 0396 0394 039B 039E 03A5
AOVCR 05A8 0558
APID 05B3
AQ0 04A2 044C 0455
AQ1 04A4 045D 0466
AQ2 04A6 046E 0477
AQ3 04A8 047F 0488
AQ4 04AA 0490 0498
AQ5 0410 0016 03FB 0418
AQ5A 0016
ASIVD 059B 0555
ASTOP 0414 0015 0289
ASTRT 05AD 0278
ASWS 05B1 0554
BEGIN 0160
BGIN 01CF 0160 01D5
BGIN1 01DC 01D9
BUSY 02EB 02AC 02FE
CKSUM 0590 0158
CR 02F6 0319
DEC 03B6 0358
DEC1 03BF 03B9
DEC2 03C1 03CB
DEC3 03C5 03C4
DELAY 041F 0574
DELYC 0425 0420 0421
DELY1 0421 0423
DSW5 0413 03FC
ELOG 0301 0288
FND 0164
FRBY 0215 02E0
ERC 02F4 0288 02AB 02AE 02C8 02F7
ERLCK 0166
ERR1 02A4 0162 02A9
ERR11 02B8 02B3
ERR12 02C6
ERR13 02D1 02DB
ERR17 02DC 0307
ERR18 02E5 02DE 02E2
ERROR 0162 0500
FRSW 02F3 02AF 02C3 02C7 02DD 02E9 0300 031C
FIVE 03A2 03B6
FOUR 0354 03A8
GO 040B 02B2
HALT 02D9 02CF 02DA
HEX 03A6 032B 0330 0333 035A 03B4
HEX1 03AA 03B3
HEX2 03AD 03AC
IERM 0217 0284 04B2 04BC
IFRR 04B0 016B 016C 016D 016E 016F 0170 0171 0172 0173 0174 0175 0176 0177
0178 0179 017A 017B 017C 017D 017E 017F 0180 0181 0182 0183 0184
0185 0186 0187 0188 0189 018A 018B 018C 018D 018E 018F 0190 0191
0192 0193 0194 0195 0196 0197 0198 0199 019A 019B 019C 019D 019E
019F 01A0 01A1 01A2 01A3 01A4 01A5 01A6 01A7 01A8 01A9 01AA 01AB
01AC 01AD 01AE 01AF 01B0 01B1 01B2 01B3 01B4 01B5 01B6 01B7 01B8
01BA 01BB 01BD 01BE 0285 04B6 04BE
IFRR2 04C0 04BA
ILIR 0015 0288 03FD 0576
ILSW 04AD 044F 0460 0471 0482 0493 04B1
ILTO 044A 0019 0459

```

1130 DIAGNOSTIC MONITOR II

ILT1 045B 0009 001A 046A  
 ILT2 046C 000A 001B 047B  
 ILT3 047D 000B 001C 048C  
 ILT4 048E 001D 049C 0542 0549  
 ILT5 03F9 000D 0414 0419  
 ILO 017A 01FC 0453 058D  
 IL1 018A 0464  
 IL2 019A 0475  
 IL3 01AA 0486  
 IL4 01BA 0496  
 IL5 01BF 0401  
 INLIZ 0219 0156 021D 0291 0383  
 IOBSY 02F5 0167 0287 0293 028A 02FD 0305 0361  
 IXFVC 0019 021A  
 K0001 027C 0297  
 K0009 0209 01DA  
 K007E 0274 024E 0296 0299  
 K007F 0207 0222  
 K0080 0206 01F1  
 K8000 0205 01E4 0231 0235 04CB 0524  
 LDERM 0595 0592  
 LEND 035C 0350 0356 03A4  
 LINE 02F7 02C9 0323  
 LOAD 0581 0447 056F  
 LOADR 0078 01FE 058F  
 LOAD1 058A 0581  
 LOAD2 058D 05DC  
 LOG 0163 04F7  
 LOGBY 0167  
 LOGI 02F8 0163 026E 029C 02FB 058A  
 LOGR 0289 037B  
 LOGT 03D0 022A 0360 0362 036B 036C 036F 0371 0374 0376 037D 037E 038A 03D2  
 LOG10 0349 034F 03CE  
 LOG11 034B 035B  
 LOG12 0351 0346 0349  
 LOG4 0305 02EA  
 LOG5 030A 02BD 0309  
 LOG7 0315 0317  
 LOG8 0334 0327 032D  
 LOG9 0345 033E  
 MALPH 050A 04F4 04FD  
 MCDSW 04DA 0385 04C7 04C8 04CF 04D0 04EF  
 MCINT 04C6 01B9 022D 04D7  
 MCIST 04AE 04CD 04D2  
 MCII 04CA 04D5  
 MC12 04D2 04A0 04E1 04EE 04F2  
 MCX2 04AF 04CE 04D3  
 MEND 0426 0164  
 MERR 04FC 04C3 0504 057D 0597  
 MESD 0028 0315 0318 035E 03D0  
 MKBR 049E 04DD  
 MLOG 04F3 01FA 0436 04FA 0567  
 MLOG2 0507 04F5 04F6 04F9 04FE 04FF 0502  
 MLSCF 0213 01CE 023D 04EC  
 MMBSY 0216 02EC  
 MMESF 0449 0433  
 MMFS0 0200 01F9  
 MMFS1 0554 0566 057A  
 MMFS3 0556 0562  
 MMFS4 0557 0591  
 MMFS7 0558 0594  
 MMR 04E2 04DC  
 YMRBS 04EB 01BC 028D 04E5 04F9  
 MMR1 04EF 04F3  
 MPID 020A  
 MPIDT 01CE 01E9 0237 0529  
 MRBS 0519 04EB 052D 054D  
 MRBSB 0542

1130 DIAGNOSTIC MONITOR II

MRBSC 054B 0540  
 MRBSD 054F 0014 0550  
 MRBSE 0579 052B  
 MRBSF 057F 0578  
 MRBS3 0536 052F  
 MRBS4 0539 0532  
 MRBS5 055A 054B  
 MRBS6 0564 055D  
 MRSRT 0280 0004  
 MSWO 020D 0166 0220 0225 0255 0267 02B1 02CD 042A  
 MSWOX 027D 0221 0244 0247  
 MSW1 020E  
 MSW2 020F 053D 0569 056C  
 MSW3 0210  
 MTABL 050C 01D3 01D7 01DF 01ED 0431 04B5 04B9 051E 0530 0533 053C 0552 055B  
 055F 058C  
 MTWR 04DF 04DE  
 MXEQ 0220 023C 02EE 0572 057F 0599  
 MXEQ1 0233 0273  
 MXEQ2 023D 0168 0239 023A 0265 02A5 02C1 02F9 0427  
 MXEQ3 0252 025A 0263  
 MXEQ4 0254 0249  
 MXEQ5 0258 0241 0253  
 MXEQ6 0261 0259  
 MXEQ7 0271 023F 0256 025D 026D 02A0 02A3 02E3 043B 043F 04C4  
 NFGS 03ED 03BB  
 NLOC 015C 0445 0588  
 ORGO 0018 021F  
 OVCR 0593 015A  
 PGMN 050D 01D0 01E1 0438 0441 0584  
 PIDWS 02F0 0302 0304 030E 0310 0329 032C 0331  
 RDBS 0552 01EB 051D  
 RDBSW 041C 040D  
 RFLF 015D 01D2  
 RQKB 01BC 028F 04E7 04F0  
 RQTY 01BB 0368 04DF  
 RSTX 02BF 02B6 0308  
 RTNSW 0165 0227 026A  
 SADR5 041E 0415 041C 041F  
 SAQ 0418 0403 040F 0424  
 SELECT 0203 01F5 01F6 0230  
 SMSG 0293 0275  
 SMSGA 0275 0250 029D 02A1  
 SMSG1 02A1 0295  
 SSO 0458 044E  
 SS1 0469 045F  
 SS2 047A 0470  
 SS3 048B 0481  
 SS4 049B 0492  
 START 0161  
 STOP 0404 03FF 0404 0416  
 STOP1 0406 0408  
 STPID 027A 029B  
 STPSW 0412 0280 0405 0406 0409 040C  
 STRSW 027B 0223 0229 024B  
 STRT 0264 0161  
 SVKB 01BD 049E  
 TABL 03DA 0311  
 TEMPR 0202 01DB 01E8  
 TFN 03D7 03C2  
 TERM 03F0 0379  
 TERMX 0218 025C  
 TLOPP 037D 038F  
 TYP 0362 0364  
 TYPI 0214 02A7 02B1 02CD 0381  
 TYP1 036C 0377 0388 038F  
 TYP2 0376 0370  
 TYP3 037F 038C



1130 DIAGNOSTIC MONITOR II

TYP4 0385 037F  
TYP5 0390 0366 0391  
UTLTY 0014 0544 0546  
WC 03F2 0341  
WCNT 0308 03A9 03B1 03B7 03B8 03BE 03C9  
XFVC 0008 021B  
XLFACT 0204 01E6 01F3 0232 0233 0236 02BB 051B  
XSC1 04AC 04B7  
XS1 0455 04AC  
X30 0456 044D  
X31 0467 045E  
X32 0478 046F  
X33 0489 0480  
X34 0499 0491  
X8000 0512  
ZEROS 0208 0270  
END OF ASSEMBLY

----- LAST PAGE -----

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. REQUIREMENTS . . . . .	1A
2.1 PROGRAM REQUIREMENTS	
2.2 EQUIPMENT REQUIREMENTS	
2.3 EQUIPMENT SETUP	
3. OPERATING PROCEDURE . . . . .	2
3.1 LOADING PROGRAM	
3.2 PROGRAM OPERATION	
3.2.1 DISK UNIT SELECTION - FUNCTION 2	
3.2.2 PROGRAM CONTROL - FUNCTION 0	
3.2.3 PROGRAM SELECTION - FUNCTION 1	
3.2.4 MISCELLANEOUS CONTROL - FUNCTION 3	
3.2.5 DECK SETUP FOR MULTI-FILE OVERLAP OPERATION	
3.3 PROGRAM HALTS	
3.3.1 NORMAL HALTS	
3.3.2 ERROR HALTS	
3.4 PROGRAM TERMINATIONS	
3.5 RESTART	
4. PRINTOUTS . . . . .	4
4.1 STATUS MESSAGES	
4.2 ERROR MESSAGES	
4.3 SPECIAL NOTES	
5. COMMENTS . . . . .	6A
5.1 DISK ADDRESSING SCHEME	
5.2 ROUTINES	
6. APPENDIX . . . . .	7A

NOTE -- DO NOT RUN THIS PROGRAM IN OVERLAP MODE.

1. PURPOSE

THE PURPOSE OF THIS PROGRAM IS TO PREPARE THE 2315 CE DISK PACK FOR USE BY THE DISK DIAGNOSTIC TEST PROGRAM. THIS PROGRAM IS RUN NORMALLY AT INSTALLATION TIME, AND WHEN THE PACK DATA HAS BEEN DESTROYED OR CHANGED. IN THIS PROGRAM ALL FILE ADDRESSES AND THE PROPER SECTOR PATTERNS ARE WRITTEN. THE EXCEPTIONS ARE CYLINDERS 90-110 INCLUSIVE.

THE PROGRAM PERFORMS THE FOLLOWING FUNCTIONS ON THE C.E. 2315 DISK PACK.

1. WRITES SECTOR IDENTIFICATION ADDRESS (SID) AND PROPER PATTERNS ON ALL CYLINDERS EXCEPT 90 THRU 110 INCLUSIVE. PROPER PATTERNS ARE -- 1313 ON SECTORS 0,2,5 AND 7 AND E5E5 ON SECTORS 1,3,4 AND 6.
2. READS AND VERIFIES EVERY SECTOR IDENTIFICATION ADDRESS WRITTEN. ANY MISSING OR ERRONEOUS SECTOR IDENTIFICATION ADDRESS CAUSES THE PROGRAM TO TERMINATE AND RESTART THE ENTIRE INITIALIZATION PROCESSES AGAIN.
3. WRITES THE SID, THE C.E. IDENTIFICATION 'CEDC', THE CYLINDER ERROR TABLE AND THE PROPER SECTOR PATTERN ON CYLINDER 199 SECTORS 3 AND SECTOR 7. THE CYLINDER ERROR TABLE CONTAINS ALL THE SECTOR ADDRESSES FOR EACH CYLINDER THAT HAS UNREADABLE AREAS. (EITHER ADDRESS OR DATA) WHEN A BAD SECTOR IS FOUND, ALL THE OTHER SECTOR ID'S IN THAT CYLINDER ARE ALSO RECORDED IN THE CYLINDER ERROR TABLE.

2. REQUIREMENTS

2.1\*\*\* PROGRAM REQUIREMENTS

1130 DIAGNOSTIC MONITOR II

2.2\*\*\* EQUIPMENT REQUIREMENTS

1. AN 1131 CPU WITH PROGRAM INPUT FROM CARD OR PAPER TAPE READER.
2. A DISK STORAGE DRIVE AND A 2315 CE DISK PACK.
3. AT LEAST 2350 AVAILABLE POSITIONS OF CORE STORAGE FOR THIS PROGRAM.

2.3\*\*\* EQUIPMENT SETUP

1. TURN POWER ON.
2. PLACE THE 2315 CE DISK PACK IN THE DISK DRIVE TO BE TESTED.
3. WAIT LONG ENOUGH FOR THE MACHINE TO BECOME READY. MACHINE MUST BE READY PRIOR TO EXECUTING PROGRAM.

3. OPERATING PROCEDURE

3.1\*\*\* PROGRAM LOADING

STANDARD MONITOR LOADING PROCEDURES APPLY

THESE PROCEDURES ARE SUMMARIZED HERE. SEE DM USE PROCEDURE FOR DETAILS.

1. SET FIRST TYPEWRITER TAB 20 CHARACTERS FROM LEFT MARGIN.
2. SET BIT SWITCH 15 OFF - LOAD AND GO  
ON - TO SPECIFY OPTIONS BEFORE RUNNING.

IF HALT AFTER LOADING, SELECT PROGRAM  
OPTIONS THEN TURN OFF HALT SWITCH OR  
FOLLOW NORMAL RESTART PROCEDURE  
(SECTION 3.5).

3. LOAD DIAGNOSTIC MONITOR AND THIS PROGRAM.
4. SELECT PROGRAM OPTIONS, IF DESIRED.

\*\*\*\*\*

3.2\*\*\* PROGRAM OPERATION.

NOTE--DO NOT RUN THIS PROGRAM IN OVERLAP MODE.

3.2.1 DISK UNIT SELECTION - FUNCTION 2

ANY SINGLE DISK DRIVE MAY BE SELECTED AND TESTED BY TURNING ON THE PROPER CONSOLE BIT.

1. TO SELECT DISK
  - A. SET SWITCHES 0-7 TO 01.
  - B. SET SWITCHES 12-15 FOR DESIRED DRIVE UNIT.

SW	DRIVE UNIT
ALL OFF	MAIN DRIVE
15	FIRST DRIVE
14	SECOND DRIVE
13	THIRD DRIVE
12	FOURTH DRIVE

2. PRESS INTERRUPT REQUEST KEY.

NOTE -- SELECTING A DISK THAT IS NOT ATTACHED TO THE SYSTEM WILL NOT CAUSE AN E0802 (NOT RDY) MESSAGE BUT WILL CAUSE AN E0801 (LOST INTERRUPT MESSAGE.)

3. INITIALIZING MORE THAN ONE DISK PACK PER PROGRAM LOAD WILL REQUIRE SELECTING THE NEW DISK UNIT (SW FNC 2) PRIOR TO RESTARTING THIS PROGRAM.

\*\*

3.2.2 PROGRAM CONTROL - FUNCTION 0

1. SET SWITCHES 0-7 TO 01.
2. SET SWITCHES 8-15 AS DESIRED.

SW	FUNCTION
8	RESTART
9	ROUTINE START MESSAGE
10	LOCK ON FUNCTION
11	LOOP PROGRAM
12	LOOP ON ERROR
13	BYPASS ERROR PRINTOUT
14	HALT ON ERROR
15	HALT

3. PRESS INT REQ KEY ON CONSOLE.

\*\*

3.2.3 ROUTINE SELECTION - FUNCTION 1

THE SELECTED ROUTINE WILL LOOP UNTIL A NEW ROUTINE IS SELECTED.

1. TO SET ROUTINE SELECTION

- A. SET SWITCHES 0-7 TO 41.
- B. SET ROUTINE NUMBER IN SWITCHES 12-15.

RTN	DESCRIPTION	
1	'SEEK-WRITE-READ' ADDRESSES AND TEST PATTERNS	• NORMAL ROUTINES-
2	'REVERSE READ' VERIFIES EACH VALID DISK ADDRESS	• THE PROGRAM STARTS WITH ROUTINE 1, RUNS EACH ROUTINE IN SEQUENCE THEN TERMINATES AFTER ROUTINE 3.
3	WRITE CE HISTOK DATA ON CE CYLINDER SECTOR 3 AND 7.	

NOTE

IF A COMPLETE PASS OF THE PROGRAM WITH A NORMAL TERMINATION IS NOT ALLOWED AND IS NOT THE FINAL RUN, THE DISK PACK WILL NOT OPERATE PROPERLY WITH THE DIAGNOSTIC TEST PROGRAM. IT IS RECOMMENDED THAT THE LOOP ROUTINE FUNCTION NOT BE USED WITH THIS PROGRAM.

- C. PRESS INT REQ KEY ON CONSOLE.

2. TO RESET ROUTINE SELECTION SET AS IF SELECTING ROUTINE ZERO.

3.2.4 MICELLANEOUS CONTROL - FUNCTION 3  
1. SET SWITCHES 0-7 TO C1.  
2. SET SWITCHES 8-15 AS DESIRED.

SW	FUNCTION
14	BYPASS RETURNING ARM TO HOME BETWEEN RTNS. SEMI-AUTO I/O AREA SCAN. DISPLAYS ONE WORD IN THE ACCUMULATOR EACH TIME THE START KEY IS PRESSED AFTER AN E0906 MESSAGE. THE COUNT IS DISPLAYED ON THE FIRST WAIT. (30DA) SCAN RT EXITS AFTER I/O WORD 321 OR FNC 3 BIT 15 OFF + CONSOLE INTERRUPT IS PRESSED.
15	

(NOTE ... IS A NON-OVERLAP FUNCTION)  
BIT 15

3. PRESS INT REQ KEY ON CONSOLE.

3.2.5 DECK SETUP FOR MULTI-FILE OVERLAP OPERATION  
1. REPRODUCE THE PROGRAM DECK FOR EACH FILE  
2. PUNCH TWO PATCH CARDS FOR EACH PROGRAM AS FOLLOWS— (START IN COL 1)

+05DC 03N9 (CARD #1 -- PROG ID PATCH )  
+05E1 000X (CARD #2 -- FILE SELECT PATCH)

WHERE + EQUALS A 12 PUNCH ONLY  
N EQUALS NEW PROG ID NUMBER (0 TO 4)  
X EQUALS THE FILE SELECT WORD (SEE 3.2.1-1)  
FILE 0 EQUALS 0000  
FILE 1 EQUALS 0001  
FILE 2 EQUALS 0002  
FILE 3 EQUALS 0004  
FILE 4 EQUALS 0008

3. PLACE THE TWO PATCH CARDS JUST IN FRONT OF THE LAST CARD OF THE PROGRAM THAT IS TO BE MODIFIED.

\*\*\*\*\*

3.3\*\*\* PROGRAM HALTS

3.3.1 NORMAL HALTS

HALT NO. (B RF6)	DESCRIPTION	RESTART ACTION
3001	PROGRAM STOP OR ADDRESS STOP	PRESS START
3002	HALT ON ERROR	DISPLAY MODE-PRESS START. RUN MODE - PRESS START
3033	WAIT FOR DISK UNIT SELECTION	SEL DISK (FNC 2)

\*\*\*\*\*

3.3.2 ERROR HALTS

HALT NO. (B REG)	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON FIRST CARD OF LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F3	CARD 2 OF LOADER DID NOT LOAD	RELOAD
30F4	CAN NOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F5	READER CHECK WHEN LOADING MONITOR OR TEST PROGRAM	NPRO THEN PLACE CARDS RUN OUT IN FRONT OF REMAINING DECK AND PRESS START.
30F6	MONITOR DID NOT LOAD	RELOAD
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START
30FA	CONSOLE PRINTER HANG UP - BUSY WILL NOT GO OFF	FIX THE CONSOLE PRINTER OR NOP THIS WAIT
30DA	DISPLAY I/O AREA WORD	PRESS START FOR NXT WD FNC 3 BIT 15 OFF + CNSL NTRPT

THIS PROGRAM HAS ONE HANGUP WAIT 7OFF. THIS OCCURS ON AN UNEXPECTED INTERRUPT OR IF THE INTERRUPT DOES NOT RESET.

\*\*\*\*\*

3.4\*\*\* PROGRAM TERMINATION

IF LOOP PROGRAM HAS NOT BEEN SPECIFIED THE PROGRAM WILL TERMINATE AN A0801 MESSAGE WILL INDICATE PROGRAM TERMINATION. AN A08ED WILL INDICATE A NONSCHEDULED TERMINATION. GENERALLY THE RESULT OF ILLEGAL OPERATION PROCEDURES. RECHECK CAREFULLY.

NOTE

IF THE PROGRAM IS NOT ALLOWED TO MAKE A NORMAL TERMINATION, THE DISK PACK WILL NOT BE ACCEPTED BY THE FUNCTION TEST.

\*\*\*\*\*

3.5\*\*\* RESTART

1. SET SWITCHES 0-7 TO 01.
2. TURN ON SWITCH 8.
3. SET DESIRED CONTROL IN SWITCHES 9-14.
4. PRESS INTERRUPT REQUEST KEY.

\*\*\*\*\*

2315 DISK INITIALIZATION

4. PRINTOUTS

ALL PRINTOUTS ARE IN THE STANDARD FORMAT.

APPNN OORR AAAA (MESSAGE)  
OR  
EPPNN OORR AAAA (MESSAGE)

WHERE A IDENTIFIES STATUS MESSAGES  
E IDENTIFIES ERROR MESSAGES  
PP IS THE PID OF THE PROGRAM CAUSING THE MESSAGE

THIS WILL BE EITHER 00 FOR MESSAGES  
ORIGINATED BY MONITOR OR  
08 FOR MESSAGES ORIGINATED BY  
THIS PROGRAM.

NN IS THE MESSAGE SEQUENCE NUMBER  
RR IS THE ROUTINE NUMBER  
AAAA IS THE ADDRESS OF THE ROUTINE  
MESSAGE IS ANY VARIABLE INFORMATION

4.1\*\*\* STATUS MESSAGES

A0000 NUM PID ADRS RELF LD  
XXXX XXXX XXXX XXXX

THIS MESSAGE IS PRINTED FOLLOWING THE LOADING OF ANY PROGRAM  
(EXCEPT MONITOR), THE MESSAGE GIVES THE LOAD SEQUENCE NUMBER,  
THE PROGRAM ID, THE ADDRESS INTO WHICH THE PROGRAM WAS LOADED,  
AND THE RELOCATION FACTOR.

A0001 SWS PID  
XXXX XXXX

THIS MESSAGE IS PRINTED EACH TIME A VALID SWITCH ENTRY IS READ  
BY THE MONITOR. THE MESSAGE CONTAINS THE SWITCH SETTING READ  
TOGETHER WITH THE PROGRAM ID OF THE PROGRAM INTO WHICH THE  
CONTENTS OF SWITCHES 8-15 WERE STORED. IF THE SWITCH ENTRY  
CALLED FOR HALT OF ANY PROGRAM THE WORD HALT WILL FOLLOW THE  
MESSAGE.

A0800 OORR AAAA

ROUTINE START MESSAGE - IF SWITCH 9, FUNCTION 0, IS TURNED ON,  
THIS MESSAGE WILL BE PRINTED BEFORE THE START OF EACH ROUTINE.  
R IS THE NUMBER OF THE NEXT ROUTINE AND AAAA IS THE STARTING  
ADDRESS.

A0801 OORR AAAA PASS= SEEKS RESKS  
WRITE SFTER HRDER  
READS SFTER HRDER

COMPLETE PASS OF PROGRAM AND STATISTICAL INFORMATION. THE  
STATISTICAL INFORMATION IS A TOTAL FOR ALL OF THE PASSES INDICATED  
BY THE PROGRAM PASS COUNT. ALL DATA IS GIVEN IN THE DECIMAL FORM.

A0803 OORR AAAA SEL FILE

SELECT DISK UNIT PROGRAM IS TO OPERATE WITH. (SEE 3.2.1-1)

DATE 02JAN66 15JUNE66 15NOV66 15JUN67  
EC NO. 415490 415490C 419643 420317

PROG ID 0308-  
PAGE 0004

2315 DISK INITIALIZATION

A0804 OORR AAAA THIS MESSAGE IS ALWAYS PRECEDED BY TWO EWOB MESSAGES. THE TWO  
EWOB MESSAGES INDICATE THAT THE WRONG ADDRESS WAS READ TWICE AFTER  
THE INITIAL SEEK. HOWEVER, REZEROING THE ARM AND RESEEKING THE  
PROPER ADDRESS WAS OBTAINED. THE PROGRAM COUNTS THIS AS ONE SEEK  
ERROR.

A08ED OORR AAAA RT =N/A ( OR ) ARM NOT HOME

END OF DISK DIAGNOSTIC. AN 'A08ED' FOLLOWING AN 'E0803'  
INDICATES THAT THE ACCESS ARM DID NOT GO TO HOME DURING  
INITIALIZATION OF PROGRAM. 'A08ED' ALONE MEANS A ROUTINE WAS  
SELECTED WHICH IS NOT AVAILABLE.

A08AD OORR AAAA ADDR RECORDED IN ERR TBL

THIS IS THE CYLINDER ERROR TABLE (CET) PRINTOUT. THE ADDRESSES HERE  
ARE IN ERROR. ONE BAD SID (SECTOR ID) WILL CAUSE ALL EIGHT ADDRESSES  
FOR SAID CYLINDER TO BE INCLUDED IN CET. FOUR LINES OF CET OUTPUT  
EQUAL A BAD PACK. (SEE E0866) ADRO TO AD7 ARE THE BAD ADDRESSES  
(SECTORS 0-7) OF SAID CYLINDER. SEE COMMENTS.  
\*\*\*\*\*

4.2\*\*\* ERROR MESSAGES

THE DSW IS CHECKED FOR ABSOLUTE CORRECTNESS AT ALL TIMES. IF AN  
ERROR IS DETECTED ONE OF THE MESSAGES BELOW WILL INDICATE THE  
PROBLEM. IT IS LEFT TO THE OPERATOR TO ANALYZE THE DSW FOR THE  
SPECIFIC PROBLEM AREA.

```

*****
* THE DISK DSW *
*-----*
* BIT DESCRIPTION *
* 0 ANY ERROR *
* 1 OP COMPLETE *
* 2 NOT READY *
* 3 BUSY *
* 4 CARRIAGE HOME *
* 5 NOT USED *
* 6 NOT USED *
* 7 NOT USED *
* 8 NOT USED *
* 9 NOT USED *
* 10 NOT USED *
* 11 NOT USED *
* 12 NOT USED *
* 13 NOT USED *
* 14 SECTOR HI COUNT *
* 15 SECTOR LO COUNT *
*****

```

E0001 SWS INVLD  
XXXX

THE SETTING OF SWITCHES 4-7 DID NOT EQUAL THE LOAD SEQUENCE  
NUMBER OF ANY PROGRAM IN CORE.

E0003 DVR CORE

THE PROGRAM WHICH THE LOADER WAS ATTEMPTING TO LOAD  
EXCEEDED AVAILABLE CORE. LOADING WAS TERMINATED.

DATE 02JAN66 15JUNE66 15NOV66 15JUN67  
EC NO. 415490 415490C 419643 420317

PROG ID 0308-  
PAGE 0004A

E0004 CKSUM

A CHECK SUM ERROR WAS DETECTED WHILE LOADING A TEST PROGRAM.  
THIS ERROR OCCURS UNDER ANY OF THE FOLLOWING CONDITIONS.

1. A CARD IS MISSING OR IS OUT OF SEQUENCE.
2. THERE IS AN EXTRA CARD IN THE DECK.
3. THE PUNCHED INFORMATION ON THE CARD IS NOT CORRECT.
4. DATA WAS LOST OR PICKED UP DUE TO A MACHINE MALFUNCTION.
5. DUE TO A CPU MALFUNCTION, THE CHECK SUM WAS NOT CORRECTLY CALCULATED.

WHEN THIS ERROR OCCURS ATTEMPT TO RELOAD THE PROGRAM.

E0005 000N XXXX

THIS ERROR WILL OCCUR IS AN INTERRUPT OCCURS, BUT THE ILSW  
WAS NOT CORRECT. N IS THE INTERRUPT LEVEL AND XXXX IS THE  
ILSW. THIS PRINTOUT WILL ONLY OCCUR IF THE INTERRUPT IS RESET  
BY A BOSI. NO ATTEMPT IS MADE BY THE ERROR ROUTINE TO RESET  
THE REQUEST BIT.

E0801 00RR AAAA INST DSW FROM TO READ

INTERRUPT WAS LOST. PROGRAM AUTOMATICALLY RETRIES TO EXECUTE ROUTINE.

NOTE  
SELECTING A FILE UNIT THAT IS NOT ATTACHED TO THE SYSTEM WILL NOT  
CAUSE AN 'E0802' (NOT READY), MESSAGE, BUT WILL CAUSE AN 'E0801'  
(LOST INTERRUPT) MESSAGE.

E0802 00RR AAAA INST DSW NOT RDY

FILE NOT READY, BUSY, OP COMPLT OR ANY ERROR IS ON.

E0803 00RR AAAA INST DSW

HOME BIT IN DSW FAILED, OR MECHANICAL RESTRICTION IN ACCESS ARM.  
SEEK-1 CYLINDER WAS GIVEN 203 TIMES AND ARM IS NOT AT HOME YET, OR  
DSW DOES NOT HAVE BIT 4 ON. DSW SHOULD BE 080X. X EQUALS 0 - 3.

E0804 00RR AAAA INST DSW ADDR RECORDED IN ERR TBL

DESIREd ADDRESS IS IN THE TABLE OF BAD ADDRESSES.  
NEXT CYLINDER IS TRIED.

E0805 00RR AAAA INST DSW DSW ERR

DSW HAS BIT/BITS ON THAT SHOULD NOT BE ON AT THIS TIME. BRANCH TO  
MONITOR END ROUTINE.

E0806 00RR AAAA INST DSW ADDR DATA

DSW ERROR BIT/BITS ON AFTER A READ OR WRITE. SOFT RD/WR ERROR.  
(NOTE ... SEE FNC 3 (3.2.4) TO SCAN I/O AREA)

E0807 00RR AAAA INST DSW ADDR DATA

HARD READ OR WRITE ERROR. INDICATES TEN SOFT READ/WRITE RETRIES WITH  
FAILURE STILL PRESENT. DSW INDICATES ERROR BITS. SEE E0806.

E0808 00RR AAAA INST DSW FROM TO READ

ACTUAL ADDRESS READ, AND THE DESIRED ADDRESS DO NOT AGREE. (READ,  
READ-CHECK FUNCTIONS ONLY). FIRST E0808 WILL CAUSE A RE-READ OF THE  
DESIRED ADDRESS. SECOND E0808 WILL CAUSE A RE-ZERO OF THE ARM AND  
A RE-SEEK AND A READ OF THE DESIRED ADDRESS.

E0809 00RR AAAA INST DSW FROM TO READ

THIS MESSAGE WILL ALWAYS BE PRECEDED BY TWO E0808 MESSAGES. THE  
ACTUAL AND THE DESIRED ADDRESSES STILL DO NOT AGREE. (SEE E0808  
MESSAGE.) THE ACTUAL ADDRESSES OF THE TWO E0808 MESSAGES ARE THE  
SAME. THEREFORE, A SEEK ERROR MOST LIKELY OCCURED. HOWEVER, THERE  
IS STILL A POSSIBILITY OF A READ, WRITE, OR DISK PACK RECORDED DATA  
ERROR OR ERRORS.

E0815 00RR AAA INST DSW FROM TO READ

THIS MESSAGE WILL ALWAYS BE PRECEDED BY TWO E0808 MESSAGES. THE  
ACTUAL AND THE DESIRED ADDRESSES STILL DO NOT AGREE. (SEE E0808  
MESSAGE.) THE ACTUAL ADDRESS OF THE TWO E0808 MESSAGES ARE NOT THE  
SAME. THEREFORE, A READ ERROR MOST LIKELY OCCURED. HOWEVER, THERE  
IS STILL A POSSIBILITY OF A SEEK, WRITE, OR DISK PACK RECORDED DATA  
ERROR OR ERRORS.

E0820 00RR AAAA INST DSW FROM TO READ

WRONG ADDRESS READ, DESIRED AND ACTUAL ADDRESS DO NOT AGREE. THIS  
IS A SECTOR ERROR.

E0821 00RR AAAA INST DSW FROM TO READ

READ SUBROUTINE ERROR RETURN. DSW, ADDRESS, OR DATA MAY BE IN ERROR.  
CHECK PRINTOUTS CAREFULLY.

E0860 00RR AAAA INST DSW FROM TO READ

WRITE ERROR RETURN. THIS OCCURS IN ROUTINE NO. 01, WHICH  
PLACES THE PROPER PATTERN ON THE DISK.

E0861 00RR AAAA INST DSW FROM TO READ

READ ERROR RETURN. THIS OCCURS IN ROUTINE NO. 01, WHICH  
VERIFIES THE PROPER PATTERN ON THE DISK.

E0866 00RR AAAA INITL BAD PACK

FOUR OR MORE CYLINDERS HAVE BAD SECTORS. THIS PACK IS THEREFORE BAD,  
ACCORDING TO THE DESIGN SPECIFICATIONS. DOES NOT CONTAIN 200 GOOD  
CYLINDERS.

E0875 00RR AAAA INST DSW FROM TO READ  
TRIED TWICE TO SEEK CE CYLINDER BY REZEROING ARM. FAILED TWICE.  
PROGRAM BRANCHED TO MONITOR END.

E0877 00RR AAAA CE CYL BAD  
THIS MESSAGE INDICATES THAT THE DESIGNATED  
'CE' CYLINDER (199) ADDRESS 0638 IS BAD. IT WILL BE NECESSARY TO  
PATCH THE PROGRAM. CHANGE ADDRESS (0638) TO SOME OTHER  
UNUSED ADDRESS. SUGGESTED ALTERNATE ADDRESS IS 0630 (CYLINDER 198).  
REFER TO LISTING FOR THE CORE ADDRESS OF THE DISK ADDRESS CONSTANT.  
PATCH THAT CORE ADDRESS WITH '0630'. CORE ADDRESS IS FOUND IN  
LISTING AT LABEL 'DCT' MINUS 40. 'DCT' IS FOUND IN CROSS REFERENCE  
LIST AT THE END OF THE LISTING.

E08CE 00RR AAAA INST DSW FROM TO READ  
ERROR IN WRITING CE DISK SECTORS 3 AND 7. THESE SECTORS CONTAIN  
SECTOR ID, 'CEDC' ID WORD, NUMBER OF ERROR SECTORS, CYLINDER ADDRESS  
ERROR TABLE, AND THE PROPER PATTERN.

4.3\*\*\* SPECIAL NOTES

- NOTE A. HARD WRITE (OR READ) ERRORS - A SOLID FAILURE TO  
WRITE ON THE DISK.  
IF THE TEST PROGRAM DETECTS THAT IT CANNOT WRITE  
ON THE DISK, IT WILL REPEAT THE OPERATION NINE  
TIMES. IF ALL TEN ATTEMPTS TO WRITE END IN A  
FAILURE, THE PROBLEM IS DEFINED AS A 'HARD WRITE(OR READ)  
ERROR.'
- NOTE B. SOFT WRITE (OR READ) ERROR - AN INTERMITTANT FAILURE  
TO WRITE ON THE DISK.  
IF THE TEST PROGRAM DETECTS THAT IT CANNOT WRITE  
ON THE DISK, IT WILL REPEAT THE OPERATION UP TO NINE TIMES.  
IF ONE OF THE TRIALS IS SUCCESSFUL, THE PROBLEM  
IS CALLED A 'SOFT WRITE (OR READ) ERROR.'
- NOTE C. DISK ADDRESSES ARE GIVEN IN DECIMAL FORM IN ALL MESSAGES. I.E. ADDR,  
FROM, TO, READ ALL APPEAR IN THE FORM 'CCCS' WHERE 'CCC' IS THE  
CYLINDER VALUE IN DECIMAL AND 'S' IS THE SECTOR VALUE IN DECIMAL.  
SECTOR 'S' IS ACTUALLY SECTORS AND HEAD.

5. COMMENTS

5.1\*\*\* DISK ADDRESSING SCHEME

THE FOLLOWING IS THE FORMAT FOR THE DISK ADDRESSING SCHEME --

HEX WD	N	N	N	N
BITS	0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15
CODE	X X X X	X C C C	C C C C	C H S S
CYL. POS	CNT	1 0 0	0 0 0 0	0
READ		2 6 3	1 0 0 0	0
DOWN		8 4 2	6 8 4 2	1

C = CYLINDER H = HEAD S = SECTOR X = NOT USED

THE LOWEST CYLINDER ADDRESS IN HEX = 0000  
THE HIGHEST CYLINDER ADDRESS IN HEX = 0657  
THE ADDRESSES ARE CYLINDER 0, HEAD 0, SECTOR 0 TO CYLINDER 202,  
HEAD 1, SECTOR 3.

THE ABOVE ADDRESSING FORMAT IS USED FOR ALL THE SECTOR IDENTIFICATION  
WORDS (CALLED SID). IT APPEARS ON THE DISK AND AS THE FIRST  
WORD OF DATA TO BE READ OR WRITTEN TO OR FROM CORE. IT IS THE  
SECOND WORD OF THE FIELD ADDRESSED BY THE IOCC. (THE FIRST WORD OF  
SAID FIELD IS THE WORD COUNT)

A. TO CONVERT HEX DISK ADDRESS WORD TO DECIMAL. PERFORM THE  
FOLLOWING USING SECTION 'B' BELOW

1. FIND CORRESPONDING C.V. FOR EACH N IN THE ADDRESS.
2. ADD THE C.V.'S TOGETHER.
3. C.V. TOTAL IS THE ACTUAL CYLINDER NUMBER IN DECIMAL.
4. FIND CORRESPONDING S.N. FOR UNITS N OF HEX ADDRESS.
5. S.N. IS THE ACTUAL DECIMAL HEAD - SECTOR NUMBER.

EXAMPLE --

CONVERT 03BD TO DECIMAL CYLINDER AND SECTOR NUMBERS.

SOLUTION -- FROM TABLE	C.V.	S.N.
0 3 B D		
.....		
.....	1	5
.....	22	
.....	96	
CYLINDER	119	5 SECTOR

B. HEX TO DEC ADDRESS CONVERSION

\*\*\*\*\*

	DISK ADDRESS	C.V. = CYLINDER VALUE
	WORD IN HEX	S.N. = SECTOR NUMBER
	. . . 0 N N N	NNN = HEX ADDRESS FROM
ZERO NOT USED . . . . .		000 TO 657

\*\*\*\*\*

N	C.V.	N	C.V.	N	C.V.	S.N.
0	= 00	0	= 0	0	= 0	+ 0
1	= 32	1	= 2	1	= 0	+ 1
2	= 64	2	= 4	2	= 0	+ 2
3	= 96	3	= 6	3	= 0	+ 3
4	= 128	4	= 8	4	= 0	+ 4
5	= 160	5	= 10	5	= 0	+ 5
6	= 192	6	= 12	6	= 0	+ 6
		7	= 14	7	= 0	+ 7
		8	= 16	8	= 1	+ 0
		9	= 18	9	= 1	+ 1
		A	= 20	A	= 1	+ 2
		B	= 22	B	= 1	+ 3
		C	= 24	C	= 1	+ 4
		D	= 26	D	= 1	+ 5
		E	= 28	E	= 1	+ 6
		F	= 30	F	= 1	+ 7

\*\*\*\*\*

5.2\*\*\* ROUTINES

IT IS THE INTENT OF THIS SECTION TO DESCRIBE THE FUNCTIONS OF EACH TEST ROUTINE AND THE DISK SUPERVISOR ROUTINES. THE FOLLOWING ARE THE IMPORTANT DISK SUPERVISOR ROUTINES-

PROGRAM LISTING LABEL	FUNCTION
DCARM	RETURN ARM TO HOME
DEXEQ	SETUP AND EXECUTE THE IOCC.
DCABP	BYPASS CYLINDERS 90 THRU 110.
DCRDY	FILE READY, NOT BUSY AND NO ERRORS.
DCDSW	SENSE DSW AND SAVE IT.
DCRTN	ROUTINE NUMBER AND PROGRAM CONTROL ROUTINE.
DCSK	SEEK SUBROUTINE.
DCWR	WRITE SUBROUTINE.
DCRD	READ SUBROUTINE.
CDTRT	COMMON DATA TRANSFER ROUTINE.
NTRPT	INTERRUPT ROUTINE.
START	MONITOR CONTROL RETURN.
END	MONITOR END ENTRY.

THE DISK SUPERVISOR ROUTINES ARE THE INTERFACE BETWEEN THE DIAGNOSTIC MONITOR AND THE TEST ROUTINES. THESE ROUTINES DO THE BASIC TESTING, CHECKING AND CONTROLLING FOR THE USING ROUTINES WHICH MAY INCLUDE OTHER SUPERVISOR ROUTINES AS WELL AS TEST ROUTINES. THEREFORE, THE ERROR MESSAGES OF SUPERVISOR ROUTINES POINT TO BASIC OR GENERAL PROBLEMS AND SHOULD NOT BE DISREGARDED OR NOTICED CASUALLY. IN SHORT, ALL ERROR MESSAGES SHOULD BE CAREFULLY ANALYZED TO SEE HOW THEY RELATE TO EACH OTHER.

NOTE

AN UNSCHEDULED INTERRUPT WILL CAUSE A PROGRAM HANG CONDITION. SEE THE INTERRUPT ROUTINE.

AGAIN IT MUST BE SAID, 'ALL ERROR MESSAGES MUST BE ANALYZED TO FIND THEIR ASSOCIATION WITH EACH OTHER.'

6. APPENDIX

NOTE

ROUTINES ARE NOT DESIGNED TO LOOP IN THE INITIALIZATION PROGRAM.

ROUTINE 01 WRITE SECTOR IDENTIFICATION ON CYLINDERS 000 (0000) THRU 089 (02C8) AND 111 (0378) THRU 202 (0650) WRITES ALTERNATE WORST CASE PATTERNS ON ALL CYLINDERS AND USES 2 SECTORS TO LOG ALL CYLINDERS THAT ARE BAD. THE CYLINDER ERROR TABLE (C.E.T.) IS LOCATED ON SECTOR ID 0638 AND 063F.

ROUTINE 02 VERIFIES CORRECT ADDRESSES ON ALL CYLINDERS (EXCEPT 90 - 110 INCLUSIVE). THIS IS A REVERSE READ. STARTS AT CYLINDER 202 AND READS TO HOME ADDRESS 0000 (HEX).

NOTE

ANY ERROR TYPEOUTS DURING ROUTINE 2 WILL CAUSE THE INITIALIZATION PROGRAM TO BE RESTARTED. THESE TYPEOUTS COULD INDICATE IMPROPER SEEK AND WRITING OF THE SECTOR ADDRESS, THEREFORE TO ENSURE PROPER INITIALIZATION THE PROGRAM IS AUTOMATICALLY RESTARTED. IF ERROR MESSAGES WITH ROUTINE 2 DESIGNATED KEEP REOCCURRING, THIS INDICATES IMPROPER SEEK INCREMENTING FROM CYLINDER 0 TO 202. INVESTIGATE SEEK ERRORS BEFORE TRYING TO INITIALIZE THE PACK.

ROUTINE 03 WRITE THE CE SECTORS WHICH CONTAIN THE CYLINDER ERROR TABLE DATA. THE CE SECTORS ARE IDENTIFIED BY THE WORD 'CEDC' FOLLOWING THE SECTOR ID. THE SECTOR PATTERNS ARE -- 1313 ON SECTORS 0,2,5, AND 7 AND E5E5 ON SECTORS 1,3,4, AND 6. THE C.E.T. IS PRINTED AT THE END OF THE PROGRAM IF THERE ARE ENTRIES IN IT.

NOTE

IF AN ERROR OCCURS DURING THIS ROUTINE IT INDICATES IMPROPER CE DATA SECTOR, AN ALTERNATE SHOULD BE SELECTED VIA PATCH CARDS. IF PATCH IS USED, PACK SHOULD INDICATE SAME. THE 2310 PROGRAM MUST BE PATCHED THE SAME AS THE 2315 PROGRAM IF THE 2310 PROGRAM IS TO BE RUN. SEE DM DOCUMENTATION FOR PATCH DETAILS.



2315 DISK INITIALIZATION

```

0000      ORG      *+1500      30800030
*****      30800040
*          30800050
* THIS ENGINEERING CHANGE REFLECTS MAJOR      30800060
* CHANGES TO THE DIAGNOSTIC MONITOR. PREVIOUS 30800070
* TESTS WILL NOT RUN WITH DIAGNOSTIC MONITCR II. 30800080
*          30800090
* THIS TEST WILL NOT RUN WITH PREVIOUS MONITORS. 30800100
*          30800110
* TESTS PRIOR TO EC 419643, DATED NOV 15, 1966 30800120
* WILL NOT OPERATE PROPERLY WITH DIAGNOSTIC     30800130
* MONITOR II.      30800140
*          30800150
*****      30800160
*          30800170
*          30800180
*          30800190
*          30800200
*          30800210
*          30800220
*          30800230
*          30800240
*          30800250
*          30800260
*          30800270
*          30800280
*          30800290
*          30800300
*          30800310
*          30800320
*          30800330
*          30800340
*          30800350
*          30800360
*          30800370
*          30800380
*          30800390
*          30800400
*          30800410
*          30800420
*          30800430
*          30800440
*          30800450
*          30800460
*          30800470
*          30800480
*          30800490
*          30800500
*          30800510
*          30800520
*          30800530
*          30800540
*          30800550
*          30800560
*          30800570
*          30800580
*          30800590
*          30800600
*          30800610
*          30800620
*          30800630
*          30800640
*          30800650
*          30800660
*          30800670
*          30800680
*          30800690
*          30800700

```

2315 DISK INITIALIZATION

```

05E9 00 44800160      *****      30800710
05EB 1 05DC           EXEQD BSI I BEGIN BR FOR INIT * SE 30800720
                        DC PID DISK ID ADDR * 30800730
*****      30800740
*          30800750
*          30800760
*          30800770
*          30800780
*          30800790
*          30800800
*          30800810
*          30800820
*          30800830
*          30800840
05EC 01 67000940     INITL LDX L3 DCT SET X3 CTRL ADDR SE 30800850
05EE 0 COF2          LD SW2 GET FILE CTRL WORD 30800860
05EF 0 FOF8          EOR TERM CHECK FOR ENTRY 30800870
05FO 01 4C200607     BSC L FILEX,Z BR TO PROCEED 30800880
*          30800890
*          30800900
05F2 01 66000ACO     LDX L2 SELDU GET ALPHA PHRASE 30800910
05F4 0 4328          BSI 3 FORMD-DCT BR TO MSAG FORM 0 MC 30800920
05F5 0 A003          DC /A003 -- MSAG = -- 30800930
05F6 0 3033          WAIT X /33 WAIT FOR SWITCHES 30800940
05F7 01 650005EC     LDX L1 INITL GET ENTRY ADDR 30800950
05F9 01 6D0C05E7     STX L1 MLN SET IT IN MLSCF 30800960
05FB 00 44800161     BSI I START RETURN TO MONITOR XM 30800970
*          30800980
*          30800990
*          30801000
*          30801010
05FD 01 67000940     RLPGM LDX L3 DCT SET X3 CTRL ADDR SE 30801020
05FF 0 C300          LD 3 0 RESET A REG 30801030
0600 0 D319          STO 3 25 RESET RT CTRL 30801040
0601 0 C0DF          LD SW2 GET SW2 DATA 30801050
0602 01 F4000655     EOR L HLDSK TEST WITH LAST DATA 30801060
0604 0 4820          BSC Z SKIP IF EUUAL 30801070
0605 0 70E6          MDX INITL BR TO INITIALIZATION SX 30801080
0606 0 706A          MDX DCTL2 BR TO LOOP CONTROL SX 30801090
*          30801100
*          30801110
*          30801120
*          30801130
*          30801140
0607 01 67000940     FILEX LDX L3 DCT SET X3 CTRL ADDR SE 30801150
0609 0 10A0          SLT 32 CLEAR A + Q REG 30801160
060A 01 C40005E1     LD L SW2 GET FILE SEL DATA 30801170
060C 0 E3F1          AND 3 -15 PASS ONLY GOOD BITS 30801180
060D 01 D4000655     STO L HLDSK SAVE IT IN SEL TBL 30801190
*          30801200
060F 0 4820          BSC Z TEST FOR MAIN FILE 30801210
0610 0 7001          MDX *+1 SKIP NOT MAIN FILE 30801220
0611 0 700C          MDX FILE0 BR TO MAIN FILE SUR 30801230
0612 0 4804          BSC E SKIP FOR NXT TEST 30801240
0613 0 7014          MDX FILE1 BR TO FILE SETUP RT 30801250
0614 0 1801          SRA 1 SHIFT FOR NXT TEST 30801260
0615 0 4804          BSC E SKIP FOR NXT TEST 30801270
0616 0 701B          MDX FILE2 BR TO FILE SETUP RT 30801280
0617 0 1801          SRA 1 SHIFT FOR NXT TEST 30801290
0618 0 4804          BSC E SKIP FOR NXT TEST 30801300
0619 0 7022          MDX FILE3 BR TO FILE SETUP RT 30901310
061A 0 1801          SRA 1 SHIFT FOR NXT TEST 30801320
061B 0 4804          BSC E SKIP FOR NXT TEST 30801330
061C 0 7029          MDX FILE4 BR TO FILE SETUP RT 30801340
061D 0 7000          MDX FILE0 BR MAIN FILE Q. CTRL 30801350
*          30801360
*          30801370
*          30801380

```

2315 DISK INITIALIZATION

```

061E 01 C4000650 FILE0 LD L MFST GET AREA CODE 30801390
0620 01 D40006DD STO L DVA SAVE IT 30801400
0622 0 D305 STO 3 5 SAVE IT 30801410
0623 01 660006DE LDX L2 NTRPT GET NTRPT ADDR 30801420
0625 00 6E00019A STX L2 IL2 SET ADDR IN MON TBL 30801430
0627 0 702E MDX SACTL BR TO SETUP 30801440
*
* FILE 1 SEL RT *
*
0628 01 C4000651 FILE1 LD L MFST+1 GET AREA CODE 30801450
062A 01 D40006DD STO L DVA SAVE IT 30801460
062C 0 D305 STO 3 5 SAVE IT 30801470
062D 01 660006DE LDX L2 NTRPT GET NTRPT ADDR 30801480
062F 00 6E000199 STX L2 IL2-1 SET ADDR IN MON TBL 30801490
0631 0 7024 MDX SACTL BR TO SETUP 30801500
*
* FILE 2 SEL RT *
*
0632 01 C4000652 FILE2 LD L MFST+2 GET AREA CODE 30801510
0634 01 D40006DD STO L DVA SAVE IT 30801520
0636 0 D305 STO 3 5 SAVE IT 30801530
0637 01 660006DE LDX L2 NTRPT GET NTRPT ADDR 30801540
0639 00 6E000198 STX L2 IL2-2 SET ADDR IN MON TBL 30801550
063B 0 701A MDX SACTL BR TO SETUP 30801560
*
* FILE 3 SEL RT *
*
063C 01 C4000653 FILE3 LD L MFST+3 GET AREA CODE 30801570
063E 01 D40006DD STO L DVA SAVE IT 30801580
0640 0 D305 STO 3 5 SAVE IT 30801590
0641 01 660006DE LDX L2 NTRPT GET NTRPT ADDR 30801600
0643 00 6E000197 STX L2 IL2-3 SET ADDR IN MON TBL 30801610
0645 0 7010 MDX SACTL BR TO SETUP 30801620
*
* FILE 4 SEL RT *
*
0646 01 C4000654 FILE4 LD L MFST+4 GET AREA CODE 30801630
0648 01 D40006DD STO L DVA SAVE IT 30801640
064A 0 D305 STO 3 5 SAVE IT 30801650
064B 01 660006DE LDX L2 NTRPT GET NTRPT ADDR 30801660
064D 00 6E000196 STX L2 IL2-4 SET ADDR IN MON TBL 30801670
064F 0 7006 MDX SACTL BR TO SETUP 30801680
*
*.....*
* SW2 FILE SELECT TBL *
*.....*
0650 0 2000 MFST DC /2000 PRY 1 CSL 0 ILSW 0 30801690
0651 0 8800 DC /8800 2 1 1 30801700
0652 0 9000 DC /9000 3 1 2 30801710
0653 0 9800 DC /9800 4 1 3 30801720
0654 0 A000 DC /A000 5 1 4 30801730
0655 0 0000 HLDSK DC 0 SW2 HOLD DATA WD 30801740
*
* SET IL AREA CODE CTRLS *
*
0656 0 C327 SACTL LD 3 39 GET SENSE RESET 30801750
0657 0 EB05 OR 3 5 OR IN A.C. 30801760
0658 0 D38B STO 3 -69 SAVE IT 30801770
0659 0 C326 LD 3 38 GET SENSE INST 30801780
065A 0 EB05 OR 3 5 OR IN A.C. 30801790
065B 0 D38D STO 3 -67 SAVE IT 30801800
*
065C 01 650005FD LDX L1 RLPGM GET RESTRT ADDR 30801810
065E 01 6D0005E4 STX L1 MLSCF-1 SET IN PGM CTL TBL 30801820
*
0660 0 700A MDX DCTRL BR TO START EXEQ PGM 30801830
*

```

2315 DISK INITIALIZATION

```

* **** ** PSEUDO REQ/REL DEVICE ***
*
0661 0 0000 RELDV DC 0 ENTRY / EXIT 30802070
0662 01 74020661 MDX L RELDV,2 ADV EXIT ADDR 30802080
0664 01 4C800661 BSC I RELDV BR TO USER 30802090
*
*
0666 0 0000 REQDV DC 0 ENTRY / EXIT 30802100
0667 01 74040666 MDX L REQDV,4 ADV EXIT ADDR 30802110
0669 01 4C800666 BSC I REQDV BR TO USER 30802120
*
*.....*
* MAINLINE PROGRAM CTRL *
*.....*
066B 0 6116 DCTRL LDX 1 22 SET CLEAR LOOP XTNT SE 30802130
066C 0 1010 SLA 16 CLR ACCUM 30802140
066D 01 D5000945 DCTL1 STO L1 DCT+5 RESET DCT FIELD 30802150
066F 0 71FF MDX 1 -1 DEC LOOP CTRL 30802160
0670 0 70FC MDX DCTL1 BR LOOP 30802170
*
0671 01 67000940 DCTL2 LDX L3 DCT SET X3 ADDR CTRL SE 30802180
*
0673 0 C300 LD 3 0 CLR A REG 30802190
0674 01 D4000842 STO L DCARM-1 CLEAR CTRL CTR 30802200
0676 0 D30A STO 3 10 SET PRESENT ADDR 30802210
0677 0 C3F8 LD 3 -8 GET SCI OF ONE 30802220
0678 0 D308 STO 3 11 SET DESIRED ADDR 30802230
*
0679 0 43AE BSI 3 -82 BR TO SEEK SUB RT SC 30802240
*
067A 0 4380 BSI 3 -128 DCARM RT SC 30802250
*
067B 01 C40005E0 DCTL3 LD L SW1 GET FNC SW 1 INFO 30802260
067D 01 4C200698 BSC L DCTL5,2 BR IF RT NUM PRESENT 30802270
*
067F 01 74010959 MDX L DCT+25,1 ADD ONE TO RT NUMBER 30802280
*
0681 01 66800959 DCTL4 LDX I2 DCT+25 GET RT NUMBER 30802290
0683 01 6E0005DD STX L2 RID SET RT NUM IN MLSCF 30802300
0685 01 C6000698 LD L2 DDSA-1 GET ROUTINE ADDR 30802310
0687 01 D40005DE STO L RAD SET ADDR IN MLSCF 30802320
0689 00 D4000165 STO L RTNSW SET RT NO. PRNT ADDR 30802330
068B 01 D4000693 STO L DCTL6-1 SET RT PIVOT ADDR 30802340
068D 01 66000694 LDX L2 DCTL6 GET ENTRY ADDR 30802350
068F 01 6E0005E5 STX L2 MLSCF SET ADDR IN CTRL FLD 30802360
0691 00 44800161 BSI I START BR TO MONITOR XM 30802370
*
0693 0 0000 DC 0 RT PIVOT ADDR HOLDER 30802380
0694 01 67000940 DCTL6 LDX L3 DCT SET X3 CTRL ADDR SE 30802390
0696 01 4C800693 BSC I DCTL6-1 BR TO RT ENTRY 30802400
*
0698 0 E3FD DCTL5 AND 3 -3 PASS MAX RT CNT 30802410
0699 0 D319 STO 3 25 SET RT NUMBER IN DCT 30802420
069A 0 70E6 MDX DCTL4 BR TO CONTINUE 30802430
*
*.....*
* DISK DIAG START ADDR *
*.....*
069B 1 06BD DC DCEOD NO RT-RETURN TO MON 30802440
069C 1 0C73 DC F01AA DISK TEST = 1 30802450
069D 1 0CC7 DC F02AA DISK TEST = 2 30802460
069E 1 0CF2 DC F03AA DISK TEST = 3 30802470
069F 1 06BD DC DCEOD NO RT-RETURN TO MON 30802480
06A0 1 06BD DC DCEOD NO RT-RETURN TO MON 30802490
*

```

2315 DISK INITIALIZATION

```

*.....*
* TEST RETURN CONTROL *
*.....*
06A1 01 C40005E2 DCRTN LD L SW3 GET SW FNC 3 DATA SE
06A3 0 1801 SRA 1 SHIFT TO TEST BIT 14
06A4 01 4C6406A7 BSC L DCNHH+E BR IF B14 IS A ONE
06A6 0 4380 BSI 3 -128 BR TO DCARM RT SC
06A7 01 C40005E0 DCNHH LD L SW1 GET SW FNC 1 DATA
06A9 01 4C200698 BSC L DCTL5,Z BR IF RT NUM PRESENT
30802750
30802760
30802770
30802780
30802790
30802800
30802810
30802820
30802830
30802840
30802850
30802860
30802870
30802880
30802890
30802900
30802910
30802920
30802930
30802940
30802950
30802960
30802970
30802980
30802990
30803000
30803010
30803020
30803030
30803040
30803050
30803060
30803070
30803080
30803090
30803100
30803110
30803120
30803130
30803140
30803150
30803160
30803170
30803180
30803190
30803200
30803210
30803220
30803230
30803240
30803250
30803260
30803270
30803280
30803290
30803300
30803310
30803320
30803330
30803340
30803350
30803360
30803370
30803380
30803390
30803400
30803410
30803420
*.....*
* LOST INTERRUPT RT *
*.....*
06C2 01 67000940 DLNRT LDX L3 DCT SET X3 CTRL ADDR SE
06C4 0 C318 LD 3 27 GET TIMER CNT
06C5 0 83FF A 3 -1 ADD +1 TO CNT
06C6 0 D318 STO 3 27 SAVE NEW CNT
06C7 0 F31C EOR 3 28 TEST FOR LIMIT
06C8 0 4820 BSC Z Q. EQ TO LIMIT
06C9 0 70CA MDX DLNRT1 NO, PROCEED
30803200
30803210
30803220
30803230
30803240
30803250
30803260
30803270
30803280
30803290
30803300
30803310
30803320
30803330
30803340
30803350
30803360
30803370
30803380
30803390
30803400
30803410
30803420
*.....*
* DISK INTERRUPT ROUTINE *
*.....*

```

2315 DISK INITIALIZATION

```

*.....*
* DVA DC /2000 DISK AREA CODE ID *
*.....*
06DD 0 2000 NTRPT DC 0
06DE 0 0000 LDX L3 DCT SET X3 CTRL ADDR SE
06DF 01 67000940 *
06E1 0 0BBA XIO 3 -70 SENSE RESET DSW
06E2 0 D308 STO 3 8 SAVE IT IN TBL
30803430
30803440
30803450
30803460
30803470
30803480
30803490
30803500
30803510
30803520
30803530
30803540
30803550
30803560
30803570
30803580
30803590
30803600
30803610
30803620
30803630
30803640
30803650
30803660
30803670
30803680
30803690
30803700
30803710
30803720
30803730
30803740
30803750
30803760
30803770
30803780
30803790
30803800
30803810
30803820
30803830
30803840
30803850
30803860
30803870
30803880
30803890
30803900
30803910
30803920
30803930
30803940
30803950
30803960
30803970
30803980
30803990
30804000
30804010
30804020
30804030
30804040
30804050
30804060
30804070
30804080
30804090
30804100
*.....*
* LOCK ON FNC CHECK RT *
*.....*
06F1 0 0000 LCKAF DC 0 ENTRY ADDR SE
06F2 01 668006F1 LDX I2 LCKAF GET LOOP ADDR
30803700
30803710
30803720
30803730
30803740
30803750
30803760
30803770
30803780
30803790
30803800
30803810
30803820
30803830
30803840
30803850
30803860
30803870
30803880
30803890
30803900
30803910
30803920
30803930
30803940
30803950
30803960
30803970
30803980
30803990
30804000
30804010
30804020
30804030
30804040
30804050
30804060
30804070
30804080
30804090
30804100
*.....*
* READ WRITE ADDR CK RT *
*.....*
06FF 0 0000 RWACK DC 0 ENTRY
0700 01 C4000D8D LD L CYLEX GET CYL ERR CNT SE
0702 01 4C18070D BSC L RWCKX,+ BR OUT IF ZERO
0704 01 65800D8D LDX I1 CYLEX GET LOOP COUNT
0706 01 C5000DA1 RWCKA LD L1 CYLET-1 GET LAST ADDR
0708 0 F308 EOR 3 11 TEST AGAINST DESIRED
0709 01 4C18070F BSC L RWCKT,+ BR IF EQ -- BAD ADDR
070B 0 71FF MDX 1 -1 DEC CTRL
070C 0 70F9 MDX RWCKA LOOP
30803930
30803940
30803950
30803960
30803970
30803980
30803990
30804000
30804010
30804020
30804030
30804040
30804050
30804060
30804070
30804080
30804090
30804100
*.....*
070D 01 4C8006FF RWCKX BSC I RWACK RETURN TO USER SX
30803940
30803950
30803960
30803970
30803980
30803990
30804000
30804010
30804020
30804030
30804040
30804050
30804060
30804070
30804080
30804090
30804100
*.....*
070F 01 66000A6B RWCKT LDX L2 ARIET GET ALPHA STQTEMENT
0711 01 6E0009EB STX L2 REMRK SET IT IN TWOPA
0713 01 74040974 MDX L FORM1+3,4 ADD TO PRINT CTRL
0715 0 C308 LD 3 11 GET DESIRED ADDR
0716 0 4399 BSI 3 -103 BR TO CONVERT IT SC
0717 01 D40009EE STO L MSG+2 SET ADDR IN TWOPA
0719 0 4331 BSI 3 FORM1-DCT BR TO MSAG FORM 1 MC
071A 0 E004 DC /E004 -- MSAG = --
071B 0 0000 DC 0 NO ERR LOOP ADDR
071C 01 74FC0974 MDX L FORM1+3,-4 SET PRNT CTL TO STD
30804060
30804070
30804080
30804090
30804100
*.....*
071E 0 C319 LD 3 25 GET RT EXEQ NUMBER
071F 0 F3FD EOR 3 -3 TEST FOR RT 3
0720 01 4C18072F BSC L RWRT3,+ BR TO SET RT 3 RETRN SX
30804080
30804090
30804100

```

2315 DISK INITIALIZATION

0722 0 C308 LD 3 11 GET CURRENT ADDR 30804110
0723 0 1803 SRA 3 SHIFT RIGHT HD-SECT 30804120
0724 0 1003 SLA 3 SHIFT LEFT HD-SECT 30804130
0725 01 4C1806A1 BSC L DCRTN,+- END RT BR IF ZERO 30804140
0727 0 93F8 S 3 -8 DEC DISK ADDR BY 1 30804150
0728 0 EBF9 OR 3 -7 SET HI SECT ADDR 30804160
0729 0 D30B STD 3 11 SET IN DESIRED ADDR 30804170
072A 01 65800CF1 LDX 11 F02XB GET CURRENT CTRL CNT 30804180
072C 0 71F9 MDX 1 -7 DEC FOR ONE TOT CYL 30804190
072D 01 64000CD7 LDX L F02AC+5 BR TO CONTINUE RT 2 SX 30804200
...
0758 0 1010 DCR10 SLA 16 CLR A REG 30804780

DATE 02JAN66 01JUL66 15NOV66 15JUN67
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3
PAGE 4

2315 DISK INITIALIZATION

0759 0 D3EB STD 3 -21 RESET RD-SK SW 1 30804790
075A 0 D3E0 STD 3 -32 RESET SFT ERR CNTR 30804800
075B 0 D3DE STD 3 -34 RESET HRD ERR CNTR 30804810
075C 0 C324 DCR12 LD 3 36 GET READ FUNCTION 30804820
075D 0 D306 STD 3 6 SET IT IN DCT 30804830
075E 0 D3DF STD 3 -33 MEM FNC HOLDER 30804840
075F 0 C31A LD 3 26 GET RD / RD-CK MOD 30804850
0760 01 4C500765 BSC L DCR16,Z- BR BY ADDR CLR IF + 30804860
0762 0 1010 SLA 16 CLR ACC 30804870
0763 01 D4000AC6 STD L DCDA+1 CLR READ I/O ADDR 30804880
...
0792 0 70C9 MDX DCR12 BR TO RE-READ ADDR 30805410
0793 0 C30C DCR23 LD 3 12 GET 2ND AAR OF E008 30805420
0794 0 D004 STD DCRE2 SAVE IT 30805430
0795 0 4386 BSI 3 -122 BR TO RE-ZERO ARM + 30805440
RE-SEEK DESIRED ADR 30805450

DATE 02JAN66 01JUL66 15NOV66 15JUN67
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3
PAGE 4A

2315 DISK INITIALIZATION

0796 0 C3BF LD 3 -65 GET 8000 HEX 30805470
0797 0 70F6 MDX DCR22+1 BR TO SET SW 1 30805480
\* DCRE1 DC 0 E008 AAR 1 30805490
DCRE2 DC 0 E008 AAR 2 30805500
\* DCRGA LD 3 -21 GET SW1 30805510
BSC L DCR0K,- BR IF RD ADDR OK 30805520
LD 3 -32 GET SFT RD ERR TEMP 30805530
S 3 -2 COUNT + CORRECT IT 30805540
STO 3 -32 SAVE CORRECTED CNT 30805550
DCR0K MDX L DRD,1 ADD TO EXIT RT OK 30805560
LD 3 12 GET ACTUAL DISK ADDR 30805570
STO 3 10 SET IT IN DCT PRESNT 30805580
STO 3 9 SET IT IN LAST GOOD 30805590
DCREL LD 3 23 GET TOT HRD RD ERRS 30805600
A 3 -34 ADD NEW HRD ERRS 30805610
STO 3 23 SAVE NEW TOTAL 30805620
LD 3 22 GET TOT SFT RD ERRS 30805630
A 3 -32 ADD NEW SFT ERRS 30805640
STO 3 22 SAVE NEW TOTAL 30805650
\* BSI L LCKAF BR TO LOOP ANY FNC SC 30805660
DC DCRD LOOP ADDR 30805670
DCRBB BSC I DRD EXIT TO CALL RT SX 30805680
\* .....\*
\* COMMON RD / WR ROUTINE \*
\* .....\*
CDTRT DC 0 ENTRY 30805690
SLA 16 CLR ACC SE 30805700
STO 3 14 CLR RETRY CTR 30805710
\* CDTBC BSI 3 -120 DCRDY RT SC 30805720
\* LD 3 13 GET WORD COUNT 30805730
STO L DCDA SET IT IN I/O FLD 30805740
LD 3 -33 GET MEM HLD FNC 30805750
STO 3 6 SET IT IN DCT FNC WD 30805760
LD 3 11 GET DESIRED ADDR 30805770
SLA 13 CLR CYL NUMBER 30805780
SRA 13 CLR CYL NUMBER 30805790
OR 3 26 RD/RD CK MODIFIER 30805800
STO 3 7 SET ADJ MOD IN DCT 30805810
LD 3 1 GET I/O ADDR 30805820
STO 3 4 SET IT IN DCT 30805830
\* CDTRC BSI 3 -110 DCDSW RT SC 30805840
\* AND 3 -59 TEST BITS 30805850
BSC L CDTSE,+ BR ZERO TO CONTINUE 30805860
\* LDX L2 DSWER GET ALPHA PHRASE 30805870
STX L2 REMRK SET IT IN THUPA 30805880
BSI 3 FORM1-DCT BR TO MSAG FORM 1 MC 30805890
DC /E005 -- MSAG = -- 30805900
DC CDTRC ERR LOOP ADDR 30805910
\* XIO 3 -70 SENSE RESET DSW 30805920
STO 3 8 SET DSW IN DCT 30805930
BSI 3 -106 BR TO MONITOR END RT SX 30805940
\* CDTSE LD 3 -73 GET RETURN ADDR 30805950
STO 3 29 SET IT IN DCT 30805960
\* BSI 3 -126 BR TO EXEQ RT SC 30805970
\* BSI 3 -100 MONITOR START RT XM 30805980
30805990
30806000
30806010
30806020
30806030
30806040
30806050
30806060
30806070
30806080
30806090
30806100
30806110
30806120
30806130
30806140

2315 DISK INITIALIZATION

07D2 01 67000940 CDTNR LDX L3 DCT SET X3 CTRL ADDR SE 30806150
07D4 0 C300 LD 3 0 GET ZERO XTNT 30806160
07D5 0 D307 STO 3 7 CLR MODIFIER 30806170
\* BSI 3 -114 RELEASE RT SC 30806180
\* BSC I CDTRT EXIT TO CALL RT SX 30806190
\* \*\*\*\*\* \*\* CHECK DSW RD/WR \*\* \* 30806200
\* CDTSN DC 0 ENTRY 30806210
LD 3 8 GET DSW IN DCT SE 30806220
AND 3 -59 PASS TEST BITS 30806230
EOR 3 -51 TEST ALL FOR OK 30806240
BSC L CDTGX,+ BR ZERO ALL OK 30806250
\* LD L DCDA+1 GET ACTUAL ADDR 30806260
STO 3 12 SET IT IN TBL 30806270
\* BSI 3 FORM4-DCT BR TO MSAG FORM 4 MC 30806280
DC /E006 -- MSAG = -- 30806290
DC 0 NO ERR LOOP ADDR 30806300
\* LD L SW3 GET SW FNC THREE 30806310
BSC E SKIP IF 15 NOT ON 30806320
MDX DSPLA BR TO DISPLAY IOA RT 30806330
\* CDTLK MDX L UCT-32,1 ADD TO SFT ERR HOLDER 30806340
\* NOP 0 SAFTY NOP 30806350
LD 3 14 GET ERR TRY CTR 30806360
A 3 -1 ADD TO TRY CTR 30806370
STO 3 14 SAVE CTR TOTAL 30806380
EOR 3 -10 TEST FOR TENTH TRY 30806390
BSC L CDTBC,Z LOOP TO RETRY AU'0 30806400
\* MDX L DCT-34,1 ADD TO HRD ERR HOLDER 30806410
NOP 0 SAFTY NOP 30806420
\* BSI 3 FORM4-DCT BR TO MSAG FORM 4 MC 30806430
DC /E007 -- MSAG = -- 30806440
DC CDTRT+1 ERR LOOP ADDR 30806450
\* MDX CDTGX+2 EXIT TO ERR ADDR SX 30806460
\* CDTGX MDX L CDTSN,1 ADD TO EXIT CTRL 30806470
BSC I CDTSN EXIT GOOD I/O OP SX 30806480
DSPLA STX 1 DSPX1+1 SAVE X1 CONTENTS 30806490
LDX L1 -321 SET X1 CONTROL COUNT 30806500
\* DSPLP LD L SW3 GET SW FNC 3 DATA 30806510
BSC E SKIP IF BIT 15 OFF 30806520
MDX \*+1 GO TO DISPLAY LOGIC 30806530
MDX DSPX1 BR TO RESET X1 30806540
\* LD L1 DCDA+321 GET I/O AREA WORD 30806550
DC /30DA WAIT TO LOOK AT IT 30806560
\* MDX 1 1 DEC X1 FOR NXT WORD 30806570
MDX DSPLP DISPLAY LOOP 30806580
\* DSPX1 LDX L1 0 RESTORE X1 30806590
MDX CDTLK BR OUT TO CONTINUE 30806600
\* .....\*
30806610
30806620
30806630
30806640
30806650
30806660
30806670
30806680
30806690
30806700
30806710
30806720
30806730
30806740
30806750
30806760
30806770
30806780
30806790
30806800
30806810
30806820

2315 DISK INITIALIZATION

```

*-----*
* DISK SEEK ROUTINE *
*-----*
0800 0 4388 DCSK BSI 3 -120 DCRDY RT SEC
080E 0 C300 LD 3 0 CLR ACC
080F 0 D302 STO 3 2 CLR SEEK INCREMENT
*
0810 0 C30A LD 3 10 GET PRESENT ADDR
0811 0 EBF9 OR 3 -7 SET HD-SECT TO MAX
0812 0 9308 S 3 11 SUB DESIRED ADDR
0813 0 1883 SRT 3 CLR HEAD-SECTOR BITS
0814 01 4C9808EE BSC I DSK,+ BR OUT IF DIFF ZERO SX
*
0816 01 4C100820 BSC L DSK10,- Q. DIRECTION NEG
0818 0 F3C2 EDR 3 -62 YES- EDR TO GEN PLUS
0819 0 83FF A 3 -1 ADD ONE TO CORRECT
081A 0 D304 STO 3 4 SAVE DIFF - INCR
081B 0 D302 STO 3 2 SAVE DIFF - INCR
081C 0 C321 LD 3 33 GET SK IN FUNCTION
081D 0 D306 STO 3 6 SET IT IN DCT
081E 0 D3DF STO 3 -33 SET MEM FNC HLDR
081F 0 7004 MDX DSK20 BR TO CONT
*
0820 0 D304 DSK10 STO 3 4 SAVE DIFF - INCR
0821 0 D302 STO 3 2 SAVE DIFF - INCR
0822 0 C322 LD 3 34 GET SK OUT FUNCTION
0823 0 D306 STO 3 6 SET IT IN DCT
*
0824 0 C388 DSK20 LD 3 -72 GET RETURN ADDR - SK
0825 0 D08A STO CDRTR SET IN CDT CMN RTRN
0826 0 7099 MDX CDRTR BR TO SET I/O EXEQ
*
0827 01 74010950 DSK30 MDX L DCT+16,1 ADD TO SEEK OPCMP CT
0829 0 1000 NOP 0 SAFTY NOP
*
082A 0 C30B LD 3 11 GET DESIRED ADDR
082B 0 D30A STO 3 10 SET IT IN PRESENT
*
082C 01 440006F1 BSI L LCKAF BR TO LOOP ANY FNC SC
082E 1 080D DC DCSK LOOP ADDR
082F 01 4C8008EE BSC I DSK EXIT TO CALL RT SX
*-----*
* CHANNEL BUSY ROUTINE *
*-----*
0831 01 6700089C CHNBZ LDX L3 CHNRQ GET CHNRQ ADDR SE
0833 01 6F0005E7 STX L3 MLN SET IT IN MLSCF
0835 00 44800161 BSI I START BR TO MONITOR XM
*-----*
* SEEK ADJ CYL 089 / 111 *
*-----*
0837 0 C30B SKADJ LD 3 11 GET DESIRED ADDR SE
0838 0 930C S 3 12 SUB PRESENT ADDR
0839 0 4808 BSC + SKIP IF POSITIVE
083A 0 7003 MDX SKOUT BR TO NEGATIVE RT
*
083B 0 C3E2 LD 3 -30 GET CYL 111
083C 0 D308 STO 3 11 SET ADDR
083D 0 7002 MDX SKOUT+2 BR TO BR BACK
*
083E 0 C3E3 SKOUT LD 3 -29 GET CYL 089
083F 0 D308 STO 3 11 SET ADDR
0840 01 4C000883 BSC L DCBPR BR TO USER RT SX
*
0842 0000 BSS E SET XFER X3 EVEN ADD

```

2315 DISK INITIALIZATION

```

*-----*
* ARM TO HOME ROUTINE *
*-----*
0842 0 0000 DC 0 DARM SEEK CNT
0843 0 C3FF DCARM LD 3 -1 GET CYL INCR SE
0844 0 D304 STO 3 4 PUT IN INCREMENT
*
0845 0 C322 DARM LD 3 34 GET SEEK --OUT--OP
0846 0 D306 STO 3 6 SET IT IN DCT
0847 0 C386 LD 3 -74 GET RETURN ADDR
0848 0 D31D STO 3 29 SET IT IN DCT
*
0849 0 4388 BSI 3 -120 DCRDY RT SC
*
084A 0 4392 BSI 3 -110 DCDSW RT SC
084B 0 1004 SLA 4 SHIFT FOR B4
084C 01 4C280860 BSC L DARTH,Z+ BR IF ARM AT HOME
*
084E 0 C0F3 LD DCARM-1 GET SEEK CNT
084F 0 83FF A 3 -1 ADD +1 TO CNT
0850 0 D0F1 STO DCARM-1 SAVE NEW CNT
0851 0 F3E7 EOR 3 -25 TEST WITH 204
0852 01 4C20085A BSC L DARMX,Z BR IF NOT EQ 204
*
0854 01 D40009EB STO L REMRK RESET REMRK WD THOPA
0856 0 4331 BSI 3 FORM1-DCT BR TO MSAG FORM 1 MC
0857 0 E003 DC /E003 -- MSAG = --
0858 0 0000 DC 0 NO ERR LOOP ADDR
*
0859 0 43A5 BSI 3 -91 BR TO END PROG SC
*
085A 0 4382 DARMX BSI 3 -126 DEXEQ RT SC
*
085B 0 439C BSI 3 -100 MONITOR START RT XM
*
085C 01 67000940 DARTN LDX L3 DCT SET X3 CTRL ADDR
085E 0 438E BSI 3 -114 RELEASE RT SC
085F 0 70E5 MDX DARM LOOP
*
0860 0 10A0 DARTH SLT 32 CLEAR A AND Q
0861 0 D309 STO 3 9 CLR LAST ADDR
0862 0 D80A STD 3 10 CLR DESIRED + PRESENT
0863 0 D30C STO 3 12 CLR ACTUAL ADDR
0864 0 D31D STO 3 29 CLR NTRPT RTRN ADDR
0865 0 D0DC STO DCARM-1 RESET SEEK CNTR
0866 00 650008B8 LDX L1 3000 SET DELAY XTNT
0868 0 4390 BSI 3 -112 DELAY RT - 20 MS SC
*
0869 01 4C8008C0 BSC I ARM EXIT TO CALL RT SX
*-----*
* COMMON EXECUTE I/O RT *
*-----*
086B 0 438A DEXEQ BSI 3 -118 REQUEST DEVICE SC
*
* ***** ** MAIN LINE MUST RELEASE **
*
086C 0 C389 LD 3 -71 GET NTRPT TIMR ADDR
086D 01 D40005E6 STO L LIV SET IN MLSCF TBL
*
086F 0 C305 LD 3 5 GET AREA CODE
0870 0 EB06 OR 3 6 SET IN FUNCTION
0871 0 EB07 OR 3 7 SET IN MODIFIER
0872 0 1890 SRT 16 PUT XIO IN Q REG

```

2315 DISK INITIALIZATION

```

0873 0 C304      *      LD    3 4      GET ADDR / INCREMENT
0874 0 D803      *      STD   DEXIO   SET IOCC WORD
0875 0 0802      *      XIO   DEXIO   DD I/O COMMAND
0876 01 4C8008C2 *      BSC  I EXQ    EXIT RETURN          SX
0878 0002      *      DEXIO BSS E 2    IOCC WORD          PH
*.....*
*      ADDR BYPASS CK RT
*.....*
087A 0 C308      *      DCABP LD  3 11   GET DESIRED ADDR      SE
087B 0 93EC      *      S      3 -20   SUB CYL 90
087C 01 4C280883 *      BSC  L DCBPR,Z+ BR IF LESS THAN
*.....*
087E 0 C308      *      LD    3 11   GET DESIRED ADDR
087F 0 93EA      *      S      3 -22   SUB CYL 110
0880 01 4C300883 *      BSC  L DCBPR,Z- BR IF GREATER THAN
*.....*
0882 0 7084      *      MDX   SKADJ   BR TO GET CYL +/-
0883 01 4C8008C4 *      DCBPR BSC  I ABP  EXIT TO CALL RT      SX
*.....*
*      RE-ZERO TO RE-SEEK RT
*.....*
0885 0 C308      *      DRESK LD  3 11   GET DESIRED ADDR      SE
0886 0 D3DC      *      STD   3 -36   SAVE IT
0887 01 74010953 *      MDX  L DCT+19,1 ADD TO SEEK ERR CNT
0889 0 1000      *      NOP                                SAFTY NOP
*.....*
088A 0 4380      *      BSI   3 -128  DCARM RT          SC
*.....*
088B 0 C3DC      *      LD    3 -36   GET SAVED ADDR
088C 0 D308      *      STD   3 11   SET IT IN DCT
*.....*
088D 0 43AE      *      BSI   3 -82   DCSK RT          SC
*.....*
088E 01 4C8008C6 *      BSC  I RSK    EXIT TO CALL RT      SX
*.....*
*      RDY-NOTBUSY-OK CK RT
*.....*
0890 0 420       *      DCRDY BSI  3 -110 DCDSW RT          SEC
0891 0 180C      *      SRA   12      SHIFT FOR B0-B3
0892 01 4C9808C8 *      BSC  I RDY,+  EXIT          SX
*.....*
0894 01 6E0009E8 *      LDX  L2 NTRDY  GET ALPHA PHRASE
0896 01 6E0009E8 *      STX  L2 REMRK  SET IT IN TWOPA
0898 0 4331      *      BSI   3 FORM1-DCT BR TO MSAG FORM 1  MC
0899 0 E002      *      DC    /E002   -- MSAG = --
089A 0 0000      *      DC    0       NO ERR LOOP ADDR
089B 0 70F4      *      MDX  DCRDY   BR TO PRINT AGAIN
*.....*
*      CHANNEL REQUEST RT
*.....*
089C 01 44000666 *      CHNRO BSI  L REQDV CHANNEL REQUEST RT  SEC
089E 1 0831      *      DC    CHNBZ   BUSY RT ADDR
089F 0 0000      *      CHNSA DC    0   DDEF ADDR

```

2315 DISK INITIALIZATION

```

08A0 1 06DD      *      DC    DVA     DEVICE REF ADDR      30808870
08A1 1 05E8      *      DC    TERM   ADDR OF/FFFF      30808880
08A2 01 4C8008CA *      BSC  I REQ   EXIT TO CALL RT      SX  30808890
*.....*
*      FILL I/O AREA RT
*.....*
08A4 00 6500FEC0 *      DFILL LDX L1 -320 SET PASS CTRL      SE  30808950
08A6 01 D5000C07 *      STO  L1 DCDA+322 RESET IDA WITH ACC  30808960
08A8 0 7101      *      MDX  1 1     DEC CTRL          30808970
08A9 0 70FC      *      MDX  DFILL+2 LOOP          30808980
08AA 01 4C8008CC *      BSC  I FLX   EXIT          SX  30808990
*.....*
*      CHANNEL RELEASE RT
*.....*
08AC 01 44000661 *      CHNRL BSI  L RELDV CHANNEL RELEASE RT  SEC  30809050
08AE 0 0000      *      CHNRA DC    0   DDEF ADDR          30809070
08AF 1 05E8      *      DC    TERM   ADDR OF /FFFF      30809080
08B0 01 4C8008CE *      BSC  I REL   EXIT TO CALL RT      SX  30809090
*.....*
*      DISK DELAY ROUTINE
*.....*
08B2 0 71FF      *      DCCLA MDX  1 -1  DEC NDX CNT          SE  30809150
08B3 0 70FE      *      MDX   *-2    LOOP IF NOT ZERO  30809160
08B4 0 C3C0      *      LD    3 -64   GET ADDR OF RETURN  30809170
08B5 01 D40005E7 *      STO  L MLN    SET ADDR IN MLSCF  30809180
08B7 0 439C      *      BSI   3 -100 BR TO MON START    XM  30809190
08B8 01 67000940 *      DLABB LDX L3 DCT SET X3 CTRL ADDR  30809200
08BA 01 4C8008D0 *      BSC  I DLA   EXIT TO CALL RT      SX  30809210
*.....*
*      READ DSW ROUTINE
*.....*
08BC 0 0B8C      *      DCDSW XIO  3 -68 READ DSW          SE  30809270
08BD 0 D308      *      STD   3 8    SAVE DSW          30809280
08BE 01 4C8008D2 *      BSC  I DSW   RETURN TO CALL RT  SX  30809290
*.....*
*      X3 - COMMON XFER TABLE
*.....*
08C0 0000      *      BSS  E      30809340
08C0 0 0000      *      ARM  DC    0  -128  30809350
08C1 0 7081      *      MDX   DCARM  SC  30809360
*.....*
08C2 0 0000      *      EXO  DC    0  -126  30809370
08C3 0 70A7      *      MDX   DEXEQ  SC  30809380
*.....*
08C4 0 0000      *      ABP  DC    0  -124  30809390
08C5 0 7084      *      MDX   DCABP  SC  30809400
*.....*
08C6 0 0000      *      RSK  DC    0  -122  30809410
08C7 0 70BD      *      MDX   DRESK  SC  30809420
*.....*
08C8 0 0000      *      RDY  DC    0  -120  30809430
08C9 0 70C6      *      MDX   DCRDY  SC  30809440
*.....*
08CA 0 0000      *      REQ  DC    0  -118  30809450
08CB 0 70D0      *      MDX   CHNRQ  SC  30809460
*.....*
08CC 0 0000      *      FLX  DC    0  -116  30809470

```

2315 DISK INITIALIZATION

Table with columns for address (e.g., 08CD 0 70D6), control code (MDX), data (DFILL), status (SC), and address (30809550). Includes a section for 'X3 - COMMON XTNT REF TBL' and a 'PATTERN ZERO' section.

2315 DISK INITIALIZATION

Table with columns for address (e.g., 0908 0 C800), control code (DC), data (/C800), status (-56), and address (30810230). Includes a section for 'X3 - DISK CTRL TABLE' and a 'ZERO CONSTANT' section.



2315 DISK INITIALIZATION

```

0947 0 0000      DC      0      MODIFIER      7      30810910
0948 0 0000      DC      0      LAST DSW READ  8      30810920
0949 0 0000      DC      0      LAST GOOD CYL READ 9 30810930
094A 0 0000      DC      0      PRESENT CYL    10     30810940
094B 0 0000      DC      0      DESIRED CYL HDDR 11 30810950
094C 0 0000      DC      0      ACTUAL ADDR READ 12 30810960
094D 0 0000      DC      0      CURRENT WORD CNT 13 30810970
094E 0 0000      DC      0      RD-WR ERR TRY CTR 14 30810980
094F 0 0000      DC      0      SEEK ERR TRY CTR 15 30810990
0950 0 0000      DC      0      TOTAL SEEKS    16     30811000
0951 0 0000      UC      0      TOTAL WRITES   17     30811010
0952 0 0000      DC      0      TOTAL READS    18     30811020
0953 0 0000      DC      0      TOTAL SEEK ERRORS 19 30811030
0954 0 0000      DC      0      TOT SFT WR ERRORS 20 30811040
0955 0 0000      DC      0      TOT HRD WR ERRORS 21 30811050
0956 0 0000      DC      0      TOT SFT RD ERRORS 22 30811060
0957 0 0000      DC      0      TOT HRD RD ERRORS 23 30811070
0958 0 0000      DC      0      TOT PRG PASSES 24     30811080
0959 0 0000      DC      0      ROUTINE EXEQ NUM 25     30811090
095A 0 0000      DC      0      RD-RDCK MODE CODE 26 30811100
095B 0 0000      DC      0      LOST TIME DLA CTR 27     30811110
095C 0 0FFF      DC      /OFFF    LOST TIME DLA XNT 28     30811120
095D 0 0000      DC      0      NTRUPT RTRN ADDR 29     30811130
095E 0 0000      DC      0      WD NUM OF PAT ERR 30     30811140
095F 0 0000      DC      0      DESIRED PATTERN 31     30811150
0960 0 0000      DC      0      ACTUAL PATTERN  32     30811160
0961 0 0400      DC      /0400    DISK SEEK --IN-- 33     30811170
0962 0 0404      DC      /0404    DISK SEEK --OUT-- 34 30811180
0963 0 0500      DC      /0500    DISK WRITE DATA 35     30811190
0964 0 0600      DC      /0600    DISK READ DATA  36     30811200
0965 0 0680      DC      /0680    DISK READ CHECK  37     30811210
0966 0 0700      DC      /0700    DISK SENSE NRSET 38     30811220
0967 0 0701      DC      /0701    DISK SENSE RESET 39     30811230
*
*.....*
* X3 - MSAG REF TBL *
*.....*
*
*.....*
* SETUP MSAG FORMAT RTNS *
*.....*
*
0968 0 0000      FORM0 DC      0      ENTRY ADDR      SE 30811340
0969 01 65800968  LDX  11 FORM0  SAVE ENTRY FOR RTRN 30811350
096B 0 6A7F      STX  2 REMRK  SET PHRASE IN TWOPA 30811360
096C 0 6200      LDX  2 0      GET PRINT CONTROL 30811370
096D 0 6A7A      STX  2 HDSW  CLEAR HEX-DEC CTRL 30811380
096E 0 6A7A      STX  2 DPCW  SET IT IN TWOPA 30811390
096F 0 6A7A      STX  2 ALPHA CLEAR ALPHA ADDR 1 30811400
0970 0 705C      MDX  MCALL  BR TO MSG CTRL RT 30811410
*
0971 0 0000      FORM1 DC      0      ENTRY ADDR      SE 30811420
0972 01 65800971  LDX  11 FORM1  SAVE ENTRY FRO RTRN 30811430
0974 0 6203      LDX  2 3      GET PRINT CONTROL 30811440
0975 0 6A73      STX  2 DPCW  SET IT IN TWOPA 30811450
0976 0 6200      LDX  2 0      SET X2 TO ZERO 30811460
0977 0 6A70      STX  2 HDSW  CLEAR HEX-DEC CTRL 30811470
0978 0 7013      MDX  TWFRM  BR TO FINISH SETUP 30811480
*
0979 0 0000      FORM2 DC      0      ENTRY ADDR      SE 30811490
097A 01 65800979  LDX  11 FORM2  SAVE ENTRY FOR RTRN 30811500
097C 0 6200      LDX  2 0      SET X2 TO ZERO 30811510
097D 0 6A6A      STX  2 HDSW  CLEAR HEX-DEC CTRL 30811520
097E 0 621F      LDX  2 31     GET PRINT CONTROL 30811530
097F 0 6A69      STX  2 DPCW  SET IT IN TWOPA 30811540
0980 0 C309      LD   3 9      GET LAST ADDR 30811550
0981 0 4399      BSI  3 -103  BR TO DSK CNVRT RT 30811560

```

2315 DISK INITIALIZATION

```

0982 0 D06B      STO  MSG+2     SET ADDR IN TWOPA 30811590
0983 0 C30B      LD   3 11     GET DFSIRED ADDR 30811600
0984 0 4399      BSI  3 -103  BR TO DSK CNVRT RT SC 30811610
0985 0 D069      STO  MSG+3     SET IN TWOPA 30811620
0986 0 C30C      LD   3 12     GET ACTUAL ADDR 30811630
0987 0 4399      BSI  3 -103  BR TO DSK CNVRT RT SC 30811640
0988 0 D067      STO  MSG+4     SET IN TWOPA 30811650
0989 01 66000A58  LDX  L2 FRTRD GET ALPHA ADDR 2 30811660
098B 0 6A5F      STX  2 REMRK  SET IN TWOPA 30811670
098C 01 66C00A4C  TWFRM LDX  L2 INDSW GET ALPHA ADDR 1 30811680
098E 0 6A5B      STX  2 ALPHA  SET IN TWOPA 30811690
098F 0 C308      LD   3 8      GET DSM 30811700
0990 0 D05C      STO  MSG+1     SET IT IN TWOPA 30811710
0991 01 C4000879  LD   L DEXIO+1 GET INST 30811720
0993 0 7038      MDX  MCALL-1  BR TO SET IN MSAG SX 30811730
*
0994 0 0000      * FORM3 DC      0      ENTRY ADDR      SE 30811740
0995 01 65800994  LDX  11 FORM3  SAVE ENTRY FOR RTRN 30811750
0997 0 623F      LDX  2 63     GET PRINT CONTROL 30811760
0998 0 6A50      STX  2 DPCW  SET IT IN TWOPA 30811770
0999 0 6208      LDX  2 8      GET HEX-DEC CONTROL 30811780
099A 0 6A4D      STX  2 HDSW  SET IT IN TWOPA 30811790
099B 0 C30C      LD   3 12     GET ACTUAL ADDR 30811800
099C 0 4399      BSI  3 -103  BR TO CNVRT RT SC 30811810
099D 0 D050      STO  MSG+2     SET IN TWOPA 30811820
099E 0 C31E      LD   3 30     GET WD = IN REC 30811830
099F 0 D04F      STG  MSG+3     SET IN TWOPA 30811840
09A0 0 C31F      LD   3 31     GET DESIRED PATTERN 30811850
09A1 0 D04E      STO  MSG+4     SET IT IN TWOPA 30811860
09A2 0 C320      LD   3 32     GET ACTUAL PATTERN 30811870
09A3 0 D049      STO  MSG+5     SET IT IN TWOPA 30811880
09A4 01 66000A60  LDX  L2 AWEAX GET ALPHA 2 ADDR 30811890
09A6 0 70E4      MDX  TWFRM-1  BR TO FINISH SETUP SX 30811900
*
09A7 0 0000      * FORM4 DC      0      ENTRY ADDR      SE 30811910
09A8 01 658009A7  LDX  11 FORM4  SAVE ENTRY FOR RTRN 30811920
09AA 0 6200      LDX  2 0      SET X2 TO ZERO 30811930
09AB 0 6A3C      STX  2 HDSW  CLEAR HEX-DEC CTRL 30811940
09AC 0 620F      LDX  2 15     GET PRINT CONTROL 30811950
09AD 0 6A3B      STX  2 DPCW  SET IT IN TWOPA 30811960
09AE 0 C30C      LD   3 12     GET ACTUAL ADDR READ 30811970
09AF 0 4399      BSI  3 -103  BR TO ADDR CNVRT RT SC 30811980
09B0 0 D03D      STO  MSG+2     SET IT IN TWOPA 30811990
09B1 01 C4000AC7  LD   L DCDA+2 GET FIRST DATA WD 30812000
09B3 0 D03B      STO  MSG+3     SET IT IN TWOPA 30812010
09B4 01 66000A52  LDX  L2 ADATA GET ALPHA 2 ADDR 30812020
09B6 0 70D4      MDX  TWFRM-1  BR TO FINISH SETUP SX 30812030
*
09B7 0 0000      * FORM5 DC      0      ENTRY ADDR      SE 30812040
09B8 01 658009B7  LDX  11 FORM5  SAVE ENTRY FOR RTRN 30812050
09BA 0 6200      LDX  2 0      SET X2 TO ZERO 30812060
09BB 0 6A2F      STX  2 REMRK  SET IT IN TWOPA 30812070
09BC 0 6207      LDX  2 7      GET CTRL XTNT 30812080
09BD 0 6A2A      STX  2 HDSW  SET HEX-DEC CTRL 30812090
09BE 0 6A2A      STX  2 DPCW  SET PRINT CTRL 30812100
09BF 0 C3CB      LD   3 -53    GET BR INST 30812110
09C0 0 D050      STO  CTLOG  SET IT IN STATUS CTL 30812120
09C1 01 66000A20  LDX  L2 TWMLP GET MULTILINE XFER 30812130
09C3 0 6A4E      STX  2 CTLOG+1 STATUS LOOP CTRL 30812140
09C4 01 66000AA2  LDX  L2 STAT1 GET ALPHA ADDR 30812150
09C6 0 6A23      STX  2 ALPHA  SET IT IN TWOPA 30812160
09C7 0 C313      LD   3 19     GET RESK COUNT 30812170
09C8 0 D025      STO  MSG+2     SET IT IN TWOPA 30812180
09C9 0 C310      LD   3 16     GET SEEK COUNT DATA 30812190
09CA 0 D022      STO  MSG+1     SET IT IN TWOPA 30812200
09CB 0 C318      LD   3 24     GET PASS COUNT 30812210
09CC 0 D01F      STO  MSG  SET IT IN TWOPA 30812220
09CD 0 702E      MCALL MDX  CALLM  BR TO MSAG CTRL RT SX 30812230

```

2315 DISK INITIALIZATION

```

*
09CE 01 660009DB FRM5A LDX L2 FRM5B LINE THREE SETUP SE 30812270
09D0 0 6A54 STX 2 TWMLB+1 GET NEW MLP RTRN 30812280
09D1 01 66000AAC LDX L2 STAT2 GET ALPHA ADDR 30812290
09D3 0 6A16 STX 2 ALPHA SET IT IN TWOPA 30812300
09D4 0 C311 LD 3 17 GET WRITE COUNT 30812310
09D5 0 D016 STO MSG SET IT IN TWOPA 30812320
09D6 0 C314 LD 3 20 GET WR SFT ERRS 30812330
09D7 0 D015 STO MSG+1 SET IT IN TWOPA 30812340
09D8 0 C315 LD 3 21 GET WR HRD ERRS 30812350
09D9 0 D014 STO MSG+2 SET IT IN TWOPA 30812360
09DA 0 7031 MDX TWLOG+1 BR TO LOG CALL SX 30812370
*
09DB 01 66000A26 FRM5B LDX L2 TWMLR GET MLP RESET ADDR SE 30812380
09DD 0 6A47 STX 2 TWMLB+1 SET RESET ADGR ML! 30812390
09DE 01 66000AB6 LDX L2 STAT3 GET ALPHA ADDR 30812400
09E0 0 6A09 STX 2 ALPHA SET IT IN TWOPA 30812410
09E1 0 C312 LD 3 18 GET READ COUNT 30812420
09E2 0 D009 STO MSG SET IT IN TWOPA 30812430
09E3 0 C316 LD 3 22 GET RD SFT ERRS 30812440
09E4 0 D008 STO MSG+1 SET IT IN TWOPA 30812450
09E5 0 C317 LD 3 23 GET RD HRD ERRS 30812460
09E6 0 70F2 MDX FRM5B-2 BR TO LOG CALL SX 30812470
*
*.....*
* TW MESSAGE OUTPUT AREA *
*.....*
09E7 0 0000 TWOPA DC MESSAGE NUMBER 30812500
09E8 0 0000 HDSW DC HEX / DEC SW 30812510
09E9 0 0000 DPCW DC DATA ID 30812520
09EA 0 0000 ALPHA DC ALPHA 1 30812530
09EB 0 0000 REMRK DC ALPHA 2 30812540
09EC 0 0000 MSG DC M 30812550
09ED 0 0000 DC E 30812560
09EE 0 0000 DC S 30812570
09EF 0 0000 DC S 30812580
09F0 0 0000 DC A 30812590
09F1 0 0000 DC G 30812600
09F2 0 0000 DC E 30812610
09F3 0 0000 DC 30812620
09F4 0 0000 DC D 30812630
09F5 0 0000 DC A 30812640
09F6 0 0000 DC T 30812650
09F7 0 0000 DC A 30812660
09F8 0 0000 DC 30812670
09F9 0 0000 DC W 30812680
09FA 0 0000 DC D 30812690
09FB 0 6100 CCRTN DC /8100 S 30812700
*
*.....*
* MESSAGE CALL CTRL RTS *
*.....*
09FC 0 C100 CALLM LD 1 0 GET MSAG NUM SE 30812710
09FD 0 D0E9 STO TWOPA SET IT IN TW OPA TBL 30812720
09FE 0 180C SRA 12 SHIFT FOR ERR RT CK 30812730
09FF 0 F3F2 EDR 3 -14 TEST FOR ERR IND 30812740
0A00 01 4C2G0A0B BSC L TWLOG,Z BR TO LOG CALL RT 30812750
0A02 0 7101 MDX 1 1 ADD FOR LOOP ADDR 30812760
0A03 0 C100 LD 1 0 GET ERROR ADDR 30812770
0A04 0 D004 STO TWERR+3 SET IT IN CALL SEQ 30812780
0A05 0 6927 STX 1 TWX1H SAVE INDEX 1 DATA 30812790
*
* ERROR RT CALL SEQ *
*
0A06 00 44800162 TWERR BSI I ERROR BR TO MON ERR RT SX 30812800
0A08 1 09E7 DC TWOPA GPA STRING 30812810

```

2315 DISK INITIALIZATION

```

0A09 0 0000 DC 0 LOOP ON ERR ADDR 30812950
0A0A 0 7008 MDX TWRN BR TO MSG RTRN CTRL SE 30812960
*
* LOG RT CALL SEQ *
*
0A0B 0 6921 TWLOG STX 1 TWX1H SAVE INDEX 1 DATA 30813000
0A0C 00 44800163 BSI I LOG BR TO MONITOR LOG 30813010
0A0E 1 09E7 DC TWOPA OPA STRING 30813020
0A0F 01 67000940 LDX L3 DCT SET XR3 CONTROL ADDR 30813030
*
0A11 0 1000 CTLOG NOP LOG - STATUS 30813040
0A12 0 100C NOP LOOP CONTROL 30813050
*
* MSAG RETRN CTRL RT *
*
0A13 01 67000940 TWRN LDX L3 DCT SET X3 CTRL ADDR SE 30813060
*
0A15 01 66000A1A LDX L2 TWEND GET ENTRY ADDR 30813100
0A17 01 6E0005E5 STX L2 MLSCF SET IT IN CTRL TBL 30813110
0A19 0 439C BSI 3 -100 BR TO MONITOR START SX 30813120
*
0A1A 01 67000940 TWEND LDX L3 DCT SET X3 CTRL ADDR SE 30813130
*
0A1C 01 65800A2D LDX I1 TWX1H GET X1 ENTRY ADDR 30813140
0A1E 00 4D000001 BSC L1 1 RETURN TO USER SX 30813150
*
0A20 0 C3D2 TWMLP LD 3 -46 GET CTRL WORD 30813160
0A21 0 1004 SLA 4 SHIFT FOR MLP CTRL 30813170
0A22 0 E8C6 OR DPCW OR IN CTRL COUNT 30813180
0A23 0 D0C5 STO DPCW SET IN FINAL CTRL 30813190
0A24 01 4C3009CE TWMLB BSC L FRM5A RETURN TO NEXT RT 30813200
*
0A26 01 660009CE TWMLR LDX L2 FRM5A GET BASIC ADDR 30813210
0A28 0 6AFC STX 2 TWMLB+1 SET IN MLB RETURN 30813220
0A29 0 C3D1 LD 3 -47 GET NOP INST 30813230
0A2A 0 D0E6 STO CTLOG CLEAR MLP CTRL XFER 30813240
0A2B 0 D0E6 STO CTLOG+1 CLEAR MLP CTRL XFER 30813250
0A2C 0 70E6 MDX TWRN RETURN TO NORMAL CTL 30813260
*
0A2D 0 0000 TWX1H DC 0 X1 ENTRY HOLD ADDR 30813270
*
*.....*
* ADDR CNVRT RT *
*.....*
0A2E 0 18D0 DSKAC RTE 16 SET Q DATA IN A 30813300
0A2F 0 1010 SLA 16 CLR Q DATA IN A 30813310
0A30 0 18D0 RTE 16 RESET Q TO ZERO 30813320
0A31 0 18D3 RTE 19 SHIFT A DATA 30813330
0A32 0 18D0 SRA 13 ADJ SECT-HEAD 30813340
0A33 0 D014 STO HCAP1 SAVE IT 30813350
0A34 0 1010 SLA 16 CLR A REG 30813360
0A35 0 ABF6 D 3 -10 DIVIDE FOR BASE 10 30813370
0A36 0 18D0 RTE 16 SET QUOTIENT INTO Q 30813380
0A37 0 D011 STO HCAP2 SAVE REMAINDER 30813390
0A38 0 1010 SLA 16 CLEAR A REG 30813400
0A39 0 ABF6 D 3 -10 DIVIDE FOR BASE 10 30813410
0A3A 0 18D0 RTE 16 SET QUOTIENT INTO Q 30813420
0A3B 0 D00E STO HCAP3 SAVE REMAINDER 30813430
0A3C 0 1010 SLA 16 CLEAR A REG 30813440
0A3D 0 ABF6 D 3 -10 DIVIDE FOR BASE 10 30813450
0A3E 0 18D0 RTE 16 SEG QUOTIENT INTO Q 30813460
0A3F 0 1004 SLA 4 SHIFT TO BUILD ADDR 30813470
0A40 0 E809 OR HCAP3 OR IN TENS CYL ADDR 30813480
0A41 0 1004 SLA 4 SHIFT AGAIN 30813490
0A42 0 E806 OR HCAP2 OR IN UNITS CYL ADDR 30813500
0A43 0 1004 SLA 4 SHIFT AGAIN 30813510
0A44 0 E803 OR HCAP1 OR IN HEAD-SECT 30813520

```

2315 DISK INITIALIZATION

OA45 0 D005	STD	HCAP4	SAVE FINAL ADDR
OA46 01 4C8008D9	BSC I	ARC	BR TO USER
OA48 0 0000	HCAP1 DC	0	HEAD-SECTOR
OA49 0 0000	HCAP2 DC	0	UNITS CYL ADDR
OA4A 0 0000	HCAP3 DC	0	TENS CYL ADDR
OA4B 0 0000	HCAP4 DC	0	TOTAL CYL ADDR
*.....*			
*.....ALPHA PHRASES TABLE.....*			
*.....*			
OA4C 0 2074	INDSM DC	/2074	INST DSW
OA4D 0 989C	DC	/989C	
OA4E 0 2121	DC	/2121	
OA4F 0 3098	DC	/3098	
OA50 0 9000	DC	/9000	
OA51 0 FFFF	DC	/FFFF	
OA52 0 3C30	ADATA DC	/3C30	ADDR DATA
OA53 0 3060	DC	/3060	
OA54 0 2130	DC	/2130	
OA55 0 3C9C	DC	/3C9C	
OA56 0 3C00	DC	/3C00	
OA57 0 FFFF	DC	/FFFF	
OA58 0 1060	FRTRD DC	/1060	FROM TO READ
OA59 0 5070	DC	/5070	
OA5A 0 2121	DC	/2121	
OA5B 0 9C50	DC	/9C50	
OA5C 0 2121	DC	/2121	
OA5D 0 6034	DC	/6034	
OA5E 0 3C30	DC	/3C30	
OA5F 0 FFFF	DC	/FFFF	
OA60 0 3C30	AWEAX DC	/3C30	ADDR WD = EXPT ACTL
OA61 0 3060	DC	/3060	
OA62 0 2190	DC	/2190	
OA63 0 3021	DC	/3021	
OA64 0 34C0	DC	/34C0	
OA65 0 2134	DC	/2134	
OA66 0 9454	DC	/9454	
OA67 0 9C21	DC	/9C21	
OA68 0 3C1C	DC	/3C1C	
OA69 0 9C5C	DC	/9C5C	
OA6A 0 FFFF	DC	/FFFF	
OA6B 0 3C30	ARIET DC	/3C30	ADDR RECORDED IN ERR
OA6C 0 3060	DC	/3060	TBL
OA6D 0 2160	DC	/2160	
OA6E 0 341C	DC	/341C	
OA6F 0 5060	DC	/5060	
OA70 0 3034	DC	/3034	
OA71 0 3021	DC	/3021	
OA72 0 2074	DC	/2074	
OA73 0 2134	DC	/2134	
OA74 0 6060	DC	/6060	
OA75 0 219C	DC	/219C	
OA76 0 185C	DC	/185C	
OA77 0 FFFF	DC	/FFFF	
OA78 0 609C	NORTN DC	/609C	RT = N/A
OA79 0 21C0	DC	/21C0	
OA7A 0 748C	DC	/748C	
OA7B 0 3C00	DC	/3C00	
OA7C 0 FFFF	DC	/FFFF	

30813630  
30813640  
30813650  
30813660  
30813670  
30813680  
30813690  
30813700  
30813710  
30813720  
30813730  
30813740  
30813750  
30813760  
30813770  
30813780  
30813790  
30813800  
30813810  
30813820  
30813830  
30813840  
30813850  
30813860  
30813870  
30813880  
30813890  
30813900  
30813910  
30813920  
30813930  
30813940  
30813950  
30813960  
30813970  
30813980  
30813990  
30814000  
30814010  
30814020  
30814030  
30814040  
30814050  
30814060  
30814070  
30814080  
30814090  
30814100  
30814110  
30814120  
30814130  
30814140  
30814150  
30814160  
30814170  
30814180  
30814190  
30814200  
30814210  
30814220  
30814230  
30814240  
30814250  
30814260  
30814270  
30814280  
30814290  
30814300

2315 DISK INITIALIZATION

OA7D 0 3C60	ARMNH DC	/3C60	ARM NOT HOME
OA7E 0 7021	DC	/7021	
OA7F 0 7450	DC	/7450	
OA80 0 9C21	DC	/9C21	
OA81 0 2450	DC	/2450	
OA82 0 7034	DC	/7034	
OA83 0 FFFF	DC	/FFFF	
OA84 0 2074	BDPAK DC	/2074	INITL BAD PACK
OA85 0 209C	DC	/209C	
OA86 0 5C21	DC	/5C21	
OA87 0 183C	DC	/183C	
OA88 0 3021	DC	/3021	
OA89 0 543C	DC	/543C	
OA8A 0 1C58	DC	/1C58	
OA8B 0 FFFF	DC	/FFFF	
OA8C 0 3098	DSWER DC	/3098	DSW
OA8D 0 9021	DC	/9021	
OA8E 0 3460	DC	/3460	ERR
OA8F 0 6021	DC	/6021	
OA90 0 FFFF	DC	/FFFF	
OA91 0 7450	NTRDY DC	/7450	NOT RDY
OA92 0 9C21	DC	/9C21	
OA93 0 6030	DC	/6030	
OA94 0 A442	DC	/A442	
OA95 0 FFFF	DC	/FFFF	
OA96 0 5C50	LNTRP DC	/5C50	LOST NTRPT
OA97 0 989C	DC	/989C	
OA98 0 2174	DC	/2174	
OA99 0 9C60	DC	/9C60	
OA9A 0 549C	DC	/549C	
OA9B 0 FFFF	DC	/FFFF	
OA9C 0 1C34	CECAB DC	/1C34	CE CYL BAD
OA9D 0 211C	DC	/211C	
OA9E 0 A45C	DC	/A45C	
OA9F 0 2118	DC	/2118	
AAA0 0 3C30	DC	/3C30	
AAA1 0 FFFF	DC	/FFFF	
AAA2 0 543C	STAT1 DC	/543C	PASS= SEEKS RESKS
AAA3 0 9898	DC	/9898	
AAA4 0 C021	DC	/C021	
AAA5 0 9834	DC	/9834	
AAA6 0 3458	DC	/3458	
AAA7 0 9821	DC	/9821	
AAA8 0 6034	DC	/6034	
AAA9 0 9858	DC	/9858	
AAAA 0 9800	DC	/9800	
AAAB 0 FFFF	DC	/FFFF	
AAAC 0 9060	STAT2 DC	/9060	WRITE SFTER HRDR
AAAD 0 209C	DC	/209C	
AAAE 0 3421	DC	/3421	
AAAF 0 9810	DC	/9810	
AAB0 0 9C34	DC	/9C34	
AAB1 0 6021	DC	/6021	
AAB2 0 2460	DC	/2460	
AAB3 0 3034	DC	/3034	
AAB4 0 6000	DC	/6000	
AAB5 0 FFFF	DC	/FFFF	
AAB6 0 6034	STAT3 DC	/6034	READS SFTER HRDR
AAB7 0 3C30	DC	/3C30	

30814310  
30814320  
30814330  
30814340  
30814350  
30814360  
30814370  
30814380  
30814390  
30814400  
30814410  
30814420  
30814430  
30814440  
30814450  
30814460  
30814470  
30814480  
30814490  
30814500  
30814510  
30814520  
30814530  
30814540  
30814550  
30814560  
30814570  
30814580  
30814590  
30814600  
30814610  
30814620  
30814630  
30814640  
30814650  
30814660  
30814670  
30814680  
30814690  
30814700  
30814710  
30814720  
30814730  
30814740  
30814750  
30814760  
30814770  
30814780  
30814790  
30814800  
30814810  
30814820  
30814830  
30814840  
30814850  
30814860  
30814870  
30814880  
30814890  
30814900  
30814910  
30814920  
30814930  
30814940  
30814950  
30814960  
30814970  
30814980

2315 DISK INITIALIZATION

```

OAB8 0 9821      DC      /9821      30814990
OAB9 0 9810      DC      /9810      30815000
OABA 0 9C34      DC      /9C34      30815010
OABB 0 6021      DC      /6021      30815020
OABC 0 2460      DC      /2460      30815030
OABD 0 3034      DC      /3034      30815040
OABE 0 6000      DC      /6000      30815050
OABF 0 FFFF      DC      /FFFF      30815060
*
OAC0 0 9834      SELDU DC    /9834      SEL FILE  30815070
OAC1 0 5C21      DC      /5C21      30815080
OAC2 0 1020      DC      /1020      30815090
OAC3 0 5C34      DC      /5C34      30815100
OAC4 0 FFFF      DC      /FFFF      30815110
*
*.....*
*          DISK IOCC AND CE HIST          *
*.....*
*
OAC5 0 0000      DCDA   DC      WORD COUNT  30815120
OAC6 0 0000      DC      DATA FIRST WORD 30815130
OAC7 0 0140      BSS    320   REMAINING AREA 30815140
OAC7 0 0050      BSS    80    SAFTY AREA FOR RD ERR 30815150
*
OC57 0 FFFF      DCE    DC      /FFFF   DISK HISTORY DATA 30815160
OC58 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815170
OC59 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815180
OC5A 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815190
OC5B 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815200
OC5C 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815210
OC5D 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815220
OC5E 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815230
OC5F 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815240
OC60 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815250
OC61 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815260
OC62 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815270
OC63 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815280
OC64 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815290
OC65 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815300
OC66 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815310
OC67 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815320
OC68 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815330
OC69 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815340
OC6A 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815350
OC6B 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815360
OC6C 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815370
OC6D 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815380
OC6E 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815390
OC6F 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815400
OC70 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815410
OC71 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815420
OC72 0 FFFF      DC      /FFFF   DISK HISTORY DATA 30815430
*
*****
*          ** START INITIALIZATION **          *
*****
*
*.....*
*          BEGIN RT F01 WR ADDR-PAT          *
*.....*
*
OC73 0 C3E5      F01AA LD  3 -27   GET WORD COUNT  SE 30815440
OC74 0 D30D      STO   3 13   SET WORD COUNT  30815450
OC75 0 C300      LD     3 0    CLR ACC          30815460

```

2315 DISK INITIALIZATION

```

OC76 01 D4000D8D  STO L CYLEX CLR CTR 30815670
OC78 01 D4000C99  STO L CYLEC CLR CTR 30815680
OC7A 0 C300      LD 3 0    DESIRED CYLINDER AND 30815690
OC7B 0 D30B      STO 3 11  SECTOR SET TO 000 30815700
*.....*
*          SELECTION OF NEXT SECTOR          *
*.....*
OC7C 0 4384      F01AB BSI 3 -124 BR TO CK BYPASS CYL SC 30815760
OC7D 0 43AE      F01SK BSI 3 -82  SEEK NEXT SECTOR SC 30815770
OC7E 01 4C000D8E  BSC L PATRT BR TO SETUP PATTERN/ 30815780
OC80 0 43AB      F01WR BSI 3 -85  WRITE SECTOR ID SC 30815790
OC81 0 700E      MDX F01AE ERROR RETURN ADDRESS 30815800
*.....*
*          IF ID IS OK, GEN NEW          *
*          SECTOR ADDRESS AND          *
*          CONTINUE TEST          *
*.....*
OC82 0 43A8      F01RD BSI 3 -88  BR TO RD RT SC 30815860
OC83 0 7010      MDX F01AF RD ERR RETURN 30815870
*
OC84 01 7401094B  F01AC MDX L DCT+11,1 INCREMENT SECTOR ADR 30815900
OC86 0 C3BE      LD 3 -66  GET DISK MAX CTRL 30815910
OC87 0 F30B      EOR 3 11  TEST CURRENT ADDR 30815920
OC88 0 4820      BSC Z BR OUT IF ZERO 30815930
OC89 0 70F2      MDX F01AB CONTINUE TEST 30815940
*
OC8A 0 C3D1      LD 3 -47  GET NOP INST 30815950
OC8B 01 D4000737  STO L DCWR SET BR/NOP SW TO NOP 30815960
OC8D 01 D4000756  STO L DCRD SET BR/NOP SW TO NOP 30815970
*
OC8F 0 439F      BSI 3 -97  BR TO RT RTRN CTRL SX 30816000
*.....*
*          ERR ANALYSS + LOGGING          *
*.....*
OC90 0 4339      F01AE BSI 3 FORM2-DCT BR TO MSAG FORM 2 MC 30816060
OC91 0 E060      DC /E060 -- MSAG = -- 30816070
OC92 1 0C80      DC F01WR ERROR LOOP RE-WRITE 30816080
OC93 0 70EE      MDX F01RD BR TO RD CK 30816090
*
OC94 0 4339      F01AF BSI 3 FORM2-DCT BR TO MSAG FORM 2 MC 30816110
OC95 0 E061      DC /E061 -- MSAG = -- 30816120
OC96 1 0C82      DC F01RD ERR LOOP ADDR 30816130
OC97 0 4002      BSI CETRT BR TO CYL ERR TBL RT SC 30816140
OC98 0 70EB      MDX F01AC BR TO CONTINUE 30816150
*.....*
*          ERROR CONTROL ROUTINE          *
*.....*
OC99 0 0000      CYLEC DC 0 CYL ERR CTR 30816200
*
OC9A 0 0000      CETRT DC 0 SAVE ENTRY 30816230
OC9B 0 C0FD      LD CYLEC TEST SE 30816240
OC9C 0 F3FD      EOR 3 -3 FOR 30816250
OC9D 01 4C180CE9  BSC L DSKNG,+ MAX ERR TEST 30816260
OC9F 01 74010C99  MDX L CYLEC,+1 CNT 30816270
*
OCA1 01 C4000D8D  CKCET LD L CYLEX GET ERR CNT 30816280
OCA3 0 4820      BSC Z SKIP IF ZERO 30816290
OCA4 0 7002      MDX ++2 BR TO SET ERR CTRL 30816300
OCA5 0 6100      LDX 1 0 SET CNT TO ZERO 30816310
OCA6 0 7002      MDX ++2 BR TO SETUP RT 30816320
OCA7 01 65800D8D  LDX 11 CYLEX SET IN ERR CTRL CNT 30816340

```

2315 DISK INITIALIZATION

```

OCA9 0 6208      *
OCAA 0 C30B      *
OCAB 0 1803      *
OCAC 0 1003      *
OCAD 01 05000DA2 CSADR STO L1 CYLET  PUT SECT ADDR IN CET
OCAE 0 7101      MDX 1 1      ADV CYL ERR TBL CTRL
OCB0 0 83FF      A 3 -1      ADV SECT ADDR
OCB1 0 72FF      MDX 2 -1      DEC ADDR C-S CTRL
OCB2 0 70FA      MDX CSADR  BR TO LOOP
OCB3 0 93FF      S 3 -1      RE-ADJ ADDR CTRLS
OCB4 0 D30B      STO 3 11      SET IN PROPER ADDR
OCB5 01 74080D8D MDX L CYLEX,8  INCREMENT ERR CNTR
OCB7 01 4C800C9A BSC I CETRT   RETURN TO CALL RT  SX
*
*.....*
*          SET BAD PACK ERR SWITCH *
*.....*
OCB9 0 C3D1      DSKNG LD 3 -47  GET NOP INST  SE
OCBA 01 D4000D35 STO L DNCSW  SET BR INST TO NOP
OCBC 0 70E4      MDX CKCET  BR TO SETUP  SX
*
*          CE DATA ERR ROUT
*
OCBD 0 0000      CETYP DC 0      SAVE ENTRY
OCBE 0 4339      BSI 3 FORM2-DCT BR TO MSAG FORM 2  MC
OCBF 0 E0CE      DC /EOCE  -- MSAG = --
OCC0 0 0000      DC 0      NO ERR LOOP ADDR
OCC1 01 4C800CBD BSC I CETYP  RETURN TO MAIN LINE
*
*          CESX3 BSI CETYP BR TO TYPE ERR MSAG
OCC3 0 40F9      MDX DNCSW  BR TO CONTINUE
OCC4 0 7070
*
*          CESX7 BSI CETYP BR TO TYPE ERR MSAG
OCC5 0 40F7      MDX CEXA7  BR TO CONTINUE
OCC6 0 7055
*
*          BEGIN ROUTINE 2
*
OCC7 0 C3FF      FOZAA LD 3 -1  SET WORD COUNT FOR  TE
OCC8 0 D30D      STO 3 13  READ TO 001
OCC9 0 C3CF      LD 3 -49  DESIRED CYLINDER ADR
OCCA 0 D30B      STO 3 11  SECTOR SET TO 000
OCCB 00 650005B0 LDX L1 1456  NUMBER OF SECTORS
*
*          SELECTION OF NEXT SECTOR*
*
OCCD 0 6923      FOZAB STX 1 FOZXB  SAVE X1 XTANT
OCE0 0 4384      BSI 3 -124  DCABP RT  SC
OCCF 0 43AE      FOZSK BSI 3 -82  DCSK RT  SC
OCD0 0 43A8      BSI 3 -88  DCRD RT  SC
OCD1 0 7008      MDX FOZAD  ERROR RETURN ADDRESS
*
*          IF ID IS OK, GEN NEW
*
OCD2 01 74FF094B FOZAC MDX L DCT+11,-1  DECREMENT SECTOR ADR
OCD4 0 1000      NOP 0      SAFTY NOP
OCD5 01 65800CF1 LDX 11 FOZXB  GET X1 XTANT
OCD7 0 71FF      MDX 1 -1  DECREMENT XR1 BY 1

```

```

30816350
30816360
30816370
30816380
30816390
30816400
30816410
30816420
30816430
30816440
30816450
30816460
30816470
30816480
30816490
30816500
30816510
30816520
30816530
30816540
30816550
30816560
30816570
30816580
30816590
30816600
30816610
30816620
30816630
30816640
30816650
30816660
30816670
30816680
30816690
30816700
30816710
30816720
30816730
30816740
30816750
30816760
30816770
30816780
30816790
30816800
30816810
30816820
30816830
30816840
30816850
30816860
30816870
30816880
30816890
30816900
30816910
30816920
30816930
30816940
30816950
30816960
30816970
30816980
30816990
30817000
30817010
30817020

```

2315 DISK INITIALIZATION

```

OCD8 0 70F4      MDX FOZAB CONTINUE TEST
OCD9 0 439F      BSI 3 -97  DCRTN RT  SX
*
*.....*
*          ERR ANALYSS + LOGGING *
*.....*
OCDA 0 C30B      FOZAD LD 3 11  GET DESIRED ADDR
OCDB 0 F30C      EOR 3 12  CMP WITH ACTUAL
OCDC 0 1803      SRA 3      CLR SECTOR/HEAD
OCDD 01 4C200CE5 BSC L FOZAE,Z BR IF CYL NOT EQ
*
OCDF 0 C30B      LD 3 11  GET DESIRED ADDR
OCEO 0 F30C      EOR 3 12  CMP WITH ACTUAL
OCE1 0 100C      SIA 12  CLR CYLINDER ADDR
OCE2 01 4C290CE9 BSC L FOZAF,Z BR IF SECT/HD NOT EQ
OCE4 0 70ED      MDX FOZAC  FALSE ERR CONTINUE
*
OCE5 0 4386      FOZAE BSI 3 -122 DRESK RT  SC
OCE6 0 43A8      BSI 3 -88  DCRD RT  SC
OCE7 0 7005      MDX FOZAG  ERR RD RETURN
OCE8 0 43A2      BSI 3 -94  RESTART EXIT  SX
*
OCE9 0 4339      FOZAF BSI 3 FORM2-DCT BR TO MSAG FORM 2  MC
OCEA 0 E020      DC /E020  -- MSAG = --
OCEB 0 0CCF      DC FOZSK  LOOP ON ERR
OCEC 0 43A2      BSI 3 -94  RESTART EXIT  SX
*
OCE9 0 4339      FOZAG BSI 3 FORM2-DCT BR TO MSAG FORM 2  MC
OCEA 0 E020      DC /E021  -- MSAG = --
OCEF 0 0000      DC 0      NO ERR LOOP ADDR
OCFO 0 43A2      BSI 3 -94  RESTART EXIT  SX
*
OCF1 0 0000      FOZXB DC 0      X1 XTANT HOLDER
*
*.....*
*          ROUTINE 3 WR CE SECTORS *
*.....*
OCF2 0 4388      FO3AA BSI 3 -120  RDY NBSY RT  SEC
OCF3 0 4380      FO3AB BSI 3 -128  DCARM RT  SC
*
OCF4 0 C3D8      FO3AC LD 3 -40  GET DISK ADDR
OCF5 0 D30B      STO 3 11  SET ADDR
OCF6 0 C3E5      LD 3 -27  GET WC
OCF7 0 D30D      STO 3 13  SET WC
OCF8 0 43AE      BSI 3 -82  SEEK CYL  SC
OCF9 0 43A8      BSI 3 -88  BR TO READ SUB RT  SC
OCFA 0 7001      MDX CESKE  BR TO SEEK ERR RT
OCFB 0 7008      MDX FOZAD  BR TO CONTINUE
*
OCFC 0 4386      CESKE BSI 3 -122  BR TO REZERO ARM RT  SC
*
OCFD 0 43A8      BSI 3 -88  BR TO READ SUB RT  SC
OCFE 0 7001      MDX SKZER  BR TO ERR MSAG RT
OCFF 0 7004      MDX FO3AD  BR TO CONTINUE
*
OD00 0 4339      SKZER BSI 3 FORM2-DCT BR TO MSAG FORM 2  MC
OD01 0 E075      DC /E075  -- MSAG = --
OD02 1 0CF2      DC FO3AA  BR ERROR LOOP ADDR
*
OD03 0 4396      BSI 3 -106  BR TO MONITOR END RT  SX
*
*          CE SECT 7 SETUP
*
OD04 0 C3D0      FO3AD LD 3 -48  GET 1313 PATTERN

```

2315 DISK INITIALIZATION

0D05 0 438C BSI 3 -116 DFILL RT SC 30817710
0D06 01 C4000D8D LD L CYLEX GET ERR AMT 30817720
0D08 01 4C180DOC BSC L F03XY,+ BR NO ERR CNT 30817730
0D0A 01 44000D7F BSI L FIOAX BR TO SET ERR IN IOA 30817740
...
0D3C 0 0000 DC 0 NO ERR LOOP ADDR 30818290
\*
\* ..... \*
\* CE ERR TBL TEST RT \*
\* ..... \*
0D3D 0 C04F ETEST LD CYLEX GET ERR CNT SE 30818340
0D3E 01 4C180D7E BSC L DIPND,+ BR IF ZERO ERR CNT 30818350
...
0D41 0 10Q4 WRCET LD 3 -15 GET BASIC PRNT CTRL 30818370
SLA 4 SHIFT FOR TOTAL CTRL 30818380

DATE 02JAN66 01JUL66 15NOV66 15JUN67
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3 PAGE 14

2315 DISK INITIALIZATION

0D42 0 EBF1 OR 3 -15 SET PARTIAL CTRL WD 30818390
0D43 01 D40009E9 STO L DPCW SET CTRL IN TWOPA 30818400
0D45 00 6500CBAD LDX L1 /7CBAD BAD CYL ERR MSAG NUM 30818410
0D47 01 6D0009E7 STX L1 TWOPA SET ERR MSAG NUM OPA 30818420
...
0D79 01 4C200D53 BSC L WRCPL-2,Z TEST IT - NOP IF 2RD 30818950
\*
\* ..... \*
\* PRINT CYL ERR TBL DATA \*
\* ..... \*
0D62 00 44800163 PRIBL BSI I LOG CALL MON LOG RT SC 30818740
0D64 1 09E7 DC TWOPA TYPE OUTPUT AREA 30818750
...
0D7E 0 439F DIPND BSI 3 -97 BR RETURN SX 30819000
\*
\* ..... \*
0D7F 0 0000 FIOAX DC 0 SET IN RETURN ADDR SE 30819060

DATE 02JAN66 01JUL66 15NOV66 15JUN67
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3 PAGE 14A

2315 DISK INITIALIZATION

```

OD80 01 65800D8D      LDX  I1 CYLEX      SET ERR CNT IN X1
OD82 01 C5000DA1     FIOAY LD  L1 CYLET-1 GET FIRSTADDR
OD84 01 D5000AC8     STO  L1 OODA+3     PUT IT IN IOA
OD86 0  71FF          MDX  1 -1          DEC INDEX CTRL
OD87 0  70FA          MDX  FIOAY        BR LOOP
OD88 01 4C800D7F     FIOAZ BSC I: FIOAY RETURN TO MAINLINE  SX
*
*
*
OD8A 0  0000          ETCNT DC  0          LINE CTRL COUNTER
OD8B 0  0000          X2SAV DC  0          INDEX 2 SAVE INFO
OD8C 0  0000          TWCNT DC  0          LINE COUNT HOLDER
OD8D 0  0000          CYLEX DC  0          SECT ERR CNT
*
*           END OF WR CE TEST DATA
*
*-----*
*           ALTERNATE PATTERN RT
*-----*
*
*
OD8E 0  C30B         PATRT LD  3 11         GET ADDR           SE
OD8F 0  100D         SLA  13          B13 TO B0
OD90 01 4C280D9D     BSC  L  TST10,Z+   BR IF HEAD 1
*
*
OD92 0  C30B         TST00 LD  3 11         GET ADDR
OD93 0  4804         BSC  E           SKIP IF ADDR EVEN
OD94 0  7004         MDX  PATE5        BR TO SET E5E5 PAT
*
*
OD95 0  C3D0         PAT13 LD  3 -48       GET PATTERN 1313
OD96 0  438C         BSI  3 -116       BR TO FILL IOA     SC
OD97 01 4C000C80     BSC  L  F01WR     BR TO WRITE RT     SX
*
*
OD99 0  C3C6         PATE5 LD  3 -58       GET PATTERN E5E5
OD9A 0  438C         BSI  3 -116       BR TO FILL IOA     SC
OD9B 01 4C000C80     BSC  L  F01WR     BR TO WRITE RT     SX
*
*
OD9D 0  C30B         TST10 LD  3 11         GET ADDR
OD9E 0  4804         BSC  E           SKIP IF ADDR EVEN
OD9F 0  70F5         MDX  PAT13        BR TO SETUP PAT E5
ODA0 0  70F8         MDX  PATE5        BR TO SETUP PAT 13
*
*
ODA2  0140          CYLET BSS E 320      CYLINDER ERR ACCUM
*
*-----*
*
*           END OF INITIALIZATION
*-----*
*
*
OEE2  0002          OMEGA BSS E 2
OEE4  05E9          END  EXEQD        BR TO BEGIN X'ER RT
    
```

```

30819070
30819080
30819090
30819100
30819110
30819120
30819130
30819140
30819150
30819160
30819170
30819180
30819190
30819200
30819210
30819220
30819230
30819240
30819250
30819260
30819270
30819280
30819290
30819300
30819310
30819320
30819330
30819340
30819350
30819360
30819370
30819380
30819390
30819400
30819410
30819420
30819430
30819440
30819450
30819460
30819470
30819480
30819490
30819500
30819510
30819520
30819530
30819540
30819550
30819560
30819570
30819580
30819590
30819600
30819610
30819620
30819630
30819640
    
```

2315 DISK INITIALIZATION

CROSS REFERENCE LISTING

SYMBOL	VALUE	REFERENCES
ABP	08C4	0883
ADATA	0A52	09B4
ALPHA	09EA	06BA,0731,096F,098E,09C6,09D3,09E0,0D38,0D4E
ARC	08D9	0A46
AFIET	0A68	070F,0D4C
ARM	08C0	0869
ARMNH	0A7D	0688
AWEAX	0A60	09A4
BDPAK	0A84	0D36
BEGIN	0160	05E9
CALLM	09FC	09CD
CCRTN	09FB	
CDTBC	07B3	07F0
CDTGX	07F9	07DD,07F8
CDTLK	07E9	080C
CDTNR	07D2	08F7
CDTRC	07C0	07CA,0826
CDTRT	07B0	0742,0765,07D7,07F7,0825
CDTSE	07CE	07C2
CDTSN	07D9	0746,0769,07F9,07FB
CECAB	0A9C	072F
CERDE	0D33	0D1B,0D31
CESKE	0CFC	0CFA
CESX3	0CC3	0D2F
CESX7	0CC5	0D19
CETRT	0C9A	0C97,0C87
CETYP	0C8D	0CC1,0CC3,0CC5
CEAA7	0D1C	0CC6
CHNBZ	0831	089E
CHNRA	08AE	
CHNRL	08AC	08CF
CHNRQ	089C	0831,08CB
CHNSA	089F	
CKCET	0CA1	0C8C
CSADR	0CAD	0CB2
CTLOG	0A11	09C0,09C3,0A2A,0A2B
CYLEC	0C99	0C78,0C9B,0C9F
CYLET	0DA2	0706,0CAD,0D56,0D82
CYLEX	0D8D	0700,0704,0C76,0CA1,0CA7,0CB5,0D06,0E12,0D1E,0D2A,0D3D,0D50,0D80
DARM	0845	085F
DARMX	085A	0852
DARTH	0860	084C
DARTN	085C	08F6
DCABP	087A	08C5
DCANH	06B8	08E6
DCARM	0843	0674,084E,0850,0865,08C1
DCBPR	0883	0840,087C,0880
DCDA	0AC5	0740,0763,076B,0785,07DF,0805,08A6,0941,0981,0D10,0D14,0D28,0D2C,0D84
DCDLA	08B2	08D1
DCDSW	08BC	08D3
DCE	0C57	
DCEOD	06BD	069B,069F,06A0,068C
DCNHM	06A7	06A4
DCRAB	07AE	
DCRD	0756	077A,0783,07AD,08E9,0C8D
DCRDY	0890	089B,08C9
DCREL	07A5	076A,0780
DCRE1	0798	0774,078C
DCRE2	0799	0775,0794
DCRGA	079A	076F
DCRND	06B6	06C1
DCROK	07A0	079B
DCRTN	06A1	0725,08E0

2315 DISK INITIALIZATION

DCR10	0758	0756
DCR12	075C	0792
DCR16	0765	0760
DCR19	077D	0784
DCR20	0781	0776
DCR21	0785	0772
DCR22	078D	0797
DCR23	0793	0789
DCKS	080D	082E,08EF
DCT	0940	05EC,05F4,05FD,0607,066D,0671,067F,0681,0694,06AF, 06B4,06BF,06C2,06CC,06DF,0719,0733,0743,0766,0778, 077D,0781,0785,07C8,07D2,07E2,07E9,07F2,07F5,0827, 0856,085C,0887,0898,08B8,0A0F,0A13,0A1A,0C84,0C90, 0C94,0CBE,0CD2,0CE9,0CED,0D00,0D3A,0D6B
DCTL1	066D	0670
DCTL2	0671	05E3,0606,08F5
DCTL3	067B	06AD
DCTL4	0681	069A,06D3
DCTL5	0698	067D,06A9
DCTL6	0694	068B,068D,0696
DCTRL	066B	0660,08E3,08F4
DCWBB	0754	
DCWEL	074B	0748
DCWR	0737	0753,08EC,0C8B
DCW10	073C	0737
DDSA	069C	0685
DEND	08E5	
DEXE0	086B	08C3
DEX10	0878	0874,0875,0991
DFILL	08A4	08A9,08CD
DIPND	0D7E	0D3E
DLA	08D0	088A
DLABB	08B8	0900
DLAND	06D8	06D5
DLNRT	06C2	06D7,08F9
DLNPI	06D4	06C9
DNGSW	0D35	0CBA,0CC4,0D32
DPCW	09E9	096E,0975,097F,0998,09AD,09BE,0A22,0A23,0D43,0D6F, 0D74,0D7C
DRD	08E8	07A0,07AE
DRESK	0885	08C7
DSK	08EE	0814,082F
DSKAC	0A2E	08DA
DSKNG	0C89	0C9D
DSK10	0820	0816
DSK20	0824	081F
DSK30	0827	08F8
DSPLA	07FD	07E8
DSPLP	080C	0809
DSPX1	080A	07FD,0804
DSW	08D2	08BE
DSWER	0A8C	07C4
DVA	06DD	0620,062A,0634,063E,0648,06A0
DVR	08E8	0749,0754
EMF	08F1	08F2
END	0164	0686,08D7
ERLCK	0166	06F6
ERROR	0162	0A06
ETCNT	0D8A	0D51,0D76,0D78
ETEST	0D3D	0D35
EXE0D	05E9	0EE4
EXO	08C2	0876
FILEX	0607	05F0
FILE0	061E	0611,061D
FILE1	0628	0613
FILE2	0632	0616
FILE3	063C	0619
FILE4	0646	061C

DATE 02JAN66 01JUL66 15NOV66 15JUN67  
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3  
PAGE 16

2315 DISK INITIALIZATION

FIOAX	0D7F	0D0A,0D22,0D88
FIOAY	0D82	0D87
FIOAZ	0D88	
FLX	08CC	08AA
FORM0	0968	05F4,06BF,0733,0969,0D3A
FORM1	0971	0713,0719,071C,07C8,0856,0898,0972
FORM2	0979	06CC,0778,0781,0785,097A,0C90,0C94,CCBE,0CE9,0CED, 0D00
FORM3	0994	0995
FORM4	09A7	07E2,07F5,09AB
FORM5	09B7	06B4,09B8
FRM5A	09CE	0A24,0A26
FRM5B	09DB	09CE,09E6
FRTRD	0A58	0989
F01AA	0C73	069C
F01AB	0C7C	0C89
F01AC	0C84	0C98
F01AE	0C90	0C81
F01AF	0C94	0C83
F01RD	0C82	0C93,0C96
F01SK	0C7D	
F01WR	0C80	0C92,0D97,0D9B
F02AA	0CC7	069D
F02AB	0CCD	0CD8
F02AC	0CD2	072D,0CE4
F02AD	0CDA	0CD1
F02AE	0CE5	0CDD
F02AF	0CE9	0CE2
F02AG	0CED	0CE7
F02SK	0CCF	0CEB
F02XB	0CF1	072A,0CCD,0CD5
F03AA	0CF2	069E,0D02
F03AB	0CF3	
F03AC	0CF4	
F03AD	0D04	0CFB,0CFF
F03XY	0D0C	0D08
F03XZ	0D24	0D20
HCAP1	0A48	0A33,0A44
HCAP2	0A49	0A37,0A42
HCAP3	0A4A	0A3B,0A40
HCAP4	0A4B	0A45
HDSW	09E8	096D,0977,097D,099A,09AB,09BD
HLDSK	0655	0602,060D
HNG	08D4	
ILO	017A	
IL1	018A	
IL2	019A	0625,062F,0639,0643,064D
IL3	01AA	
IL4	01BA	
INDSW	0A4C	098C
INITL	05EC	05E4,05F7,0605
LCKAF	06F1	06F2,06FB,06FD,0751,07AB,082C
LHOLD	0D6B	0D65
LIV	05E6	06D9,06EC,086D
LNTRP	0A96	
LOG	0163	0A0C,0D62
LOGBY	0167	
LSTRT	0D65	
MCALL	09CD	0970,0993
MEND	08D6	
MFST	0650	061E,0628,0632,063C,0646
MLN	05E7	05F9,06E4,0833,08B5
MLSCF	05E5	065E,068F,0A17,0D67
MSG	09EC	0717,0982,0985,0988,0990,099D,099F,09A1,09A3,09B0, 09B3,09C8,09CA,09CC,09D5,09D7,09D9,09E2,09E4,0D59
NORTN	0A78	068D
NTRDY	0A91	0894
NTRER	06F0	06E6,06F0

DATE 02JAN66 01JUL66 15NOV66 15JUN67  
EC NO. 415490 415490C 419643 420317

PROG ID 0308-3  
PAGE 16A



2315 DISK INITIALIZATION

NTRPT	06DE	0623,062D,0637,0641,064B,06EE
NTRST	06E6	
NTRXT	06EE	
OMEGA	0EE2	
PATES	0D99	0D94,0DA0
PATR:	0D8E	0C7E
PAT13	0D95	0D9F
PID	05DC	05EB
PRTBL	0D62	
RAD	05DE	0687
RCY	08C8	0892
REL	08CE	0880
RELDV	0661	0662,0664,08AC
REMRK	09EB	0711,07C6,0854,0896,096B,098B,098B,0D4A
REQ	08CA	08A2
REQDV	0666	0667,0669,089C
REST	08E2	
RID	05DD	0683
RLPGM	05FD	065C
RQKB	01DA	
ROTY	01CA	
RSK	08C6	088E
RTN	08DF	
RTNSW	0165	0689
RWACK	C6FF	070D,073B,0757
RWCKA	0706	070C
RWCKT	070F	0709
RWCKX	070D	0702
RWRT3	072F	0720,0D33
SACTL	0656	0627,0631,063B,0645,064F
SELDU	0AC0	05F2
SETUP	0CA9	
SKADJ	0837	0882
SKDUT	083E	083A,083D
SK2ER	0D00	0CFE
START	0161	05FB,0691,06DB,0835,08DD,0D69
STAT1	0AA2	09C4
STAT2	0AAC	09D1
STAT3	0AB6	09DE
STRT	08DC	
SVKB	01EA	
SWO	05DF	06F4
SW1	05E0	067B,06A7
SW2	05E1	05EE,0601,060A
SW3	05E2	06A1,07E5,0800
TERM	05E8	05EF,08A1,08AF
TST00	0D92	
TST10	0D9D	0D90
TWCNT	0D8C	0D55,0D5D
TWEND	0A1A	0A15
TWERR	0A06	0A04
TWFRM	098C	097B,09A6,09B6
TWLDG	0A0B	09DA,0A00
MLB	0A24	09D0,09DD,0A28
TWMLP	0A20	09C1
TWMLR	0A26	09DB
TWOPA	09E7	09FD,0A08,0A0E,0D47,0D64
TWRTN	0A13	0A0A,0A2C
TWX1H	0A2D	0A05,0A0B,0A1C
WRCET	0D40	
WRCPL	0D55	0D60,0D79
X2SAV	0D8B	0D61,0D6D

TABLE OF CONTENTS

1. PURPOSE . . . . .1A

2. PREREQUISITES . . . . .1A

3. USE PROCEDURE . . . . .1A

    3.1 PROGRAM LOADING

    3.2 PROGRAM OPERATION

    3.3 PROGRAM HALTS

    3.4 PROGRAM TERMINATION

    3.5 PROGRAM RESTART

4. PRINTOUTS . . . . .3A

    4.1 STATUS MESSAGES

    4.2 COMMAND MESSAGES

    4.3 DATA MESSAGES

    4.4 ERROR MESSAGES

5. COMMENTS . . . . .6A

    5.1 DESCRIPTION OF TEST ROUTINES

    5.2 DESCRIPTION OF SUB-ROUTINES

6. APPENDIX . . . . .NONF

1. PURPOSE

THE 2310B FUNCTION TEST IS DESIGNED TO TEST EACH FUNCTION OF THE DISK FOR COMPLIANCE WITH THE PRODUCT SPECIFICATIONS.

THIS TEST IS WRITTEN TO ACCOMMODATE SYSTEMS WITH ONE OR MORE '135D' DISK DRIVES. ANY ONE OF THE DRIVES MAY BE SELECTED TO BE TESTED.

2. PREREQUISITES

THIS PROGRAM MUST RUN UNDER CONTROL OF THE 1130 DIAGNOSTIC MONITOR. THE DIAGNOSTIC MONITOR USES 1,500 STORAGE WORDS AND THIS PROGRAM USES 2,596 STORAGE WORDS, SO THAT A 4K MACHINE IS SUPPORTED.

THIS PROGRAM REQUIRES THAT A DISK WHICH IS INITIALIZED BY THE 2315 DISK INITIALIZATION PROGRAM (0308) BE INSTALLED ON THE DISK DRIVE TO BE TESTED PRIOR TO THE EXECUTION OF THE TEST. ANY INITIALIZED PACK WILL BE ACCEPTABLE, INCLUDING AN 1130 DIMAL PACK.

3. USE PROCEDURE

3.1 PROGRAM LOADING

3.1.1 PROGRAM DECK OR PAPER TAPE

TO LOAD THE PROGRAM DECK OR PAPER TAPE, USE THE STANDARD LOADING PROCEDURE AS DESCRIBED IN THE DIAGNOSTIC MONITOR USE PROCEDURE.

3.1.2 USING 1130 DIMAL

1. USING 1130 DIMAL, PUT ONLY ONE COPY OF THE 2310 DISK FUNCTION TEST ON THE DIMAL PACK. SEE DIMAL OPERATING PROCEDURES FOR INSTRUCTIONS.
2. TO EXECUTE ONE OR MORE 2310 FUNCTION TESTS IN OVERLAP, IPL THE DIMAL COLD START LOADER, SELECT PROGRAMS TO BE EXECUTED (09 FOR DISK TEST). LEAVE BIT 15 ON WHEN THE DISK PROGRAM IS BEING LOADED. THIS IS THE MONITOR HALT BIT. SELECT THE DISK TO BE TESTED USING SWITCH FNC 2 BEFORE EXECUTING THE DISK TEST.
3. TO EXECUTE MORE THAN ONE DISK TEST, SELECT PID 09 FOR EACH DISK TO BE TESTED. A COPY OF THE DISK TEST WILL BE LOADED INTO CORE FOR EVERY DISK TO BE TESTED. SELECT A DIFFERENT DISK TO BE TESTED FOR EVERY PROGRAM LOADED, USING SWITCH FNC 2, BEFORE EXECUTING EACH TEST.

3.2 PROGRAM OPERATION

3.2.1 PROGRAM EXECUTION

A. LOAD AND GO MODE

IF BIT SWITCH 15 IS OFF WHILE THE PROGRAM IS LOADED, THE PROGRAM WILL BE EXECUTED WHEN LOADING IS COMPLETED. ALL TESTS WHICH ARE NORMALLY RUN WITH ZEROES IN THE PROGRAM SWITCH OPTIONS WILL BE EXECUTED.

NOTE--LOAD AND GO MODE MUST NOT BE USED WHEN RUNNING MORE THAN ONE DISK TEST IN OVERLAP, OR WHEN THE DRIVE TO BE TESTED IS NOT THE CPU DRIVE.

B. PAUSE BEFORE EXECUTION

IF BIT SWITCH 15 IS ON WHILE THE PROGRAM IS LOADED, THE PROGRAM WILL NOT BE EXECUTED WHEN LOADING IS COMPLETE. THIS MODE MUST ALWAYS BE USED WHEN EXECUTING MORE THAN ONE DISK TEST IN OVERLAP, OR WHEN THE DRIVE TO BE TESTED IS NOT THE CPU DRIVE. THE DRIVE TO BE TESTED MUST BE SELECTED USING SW FNC 2. THE FOLLOWING PROCEDURE MUST BE USED TO EXECUTE THE PROGRAM(S).

1. SET UP THE DESIRED OPTIONS AS SPECIFIED IN SECTION 3.2.2. IF NO OPTIONS ARE DESIRED, PROCEED TO THE NEXT STEP.
2. START PROGRAM EXECUTION BY SETTING HEXADECIMAL 0X80 IN THE DATA ENTRY SWITCHES, WHERE X IS THE PROGRAM SEQUENCE NUMBER.
3. DEPRESS CONSOLE INTERRUPT. AFTER A PRINTOUT OCCURS, THE PROGRAM WILL BE EXECUTED.

3.2.2 PROGRAM SWITCH OPTIONS

THE OPERATOR MAY MODIFY THE EXECUTION OF THE PROGRAM ANY TIME BEFORE OR AFTER IT HAS STARTED EXECUTION BY ENTERING PROGRAM CONTROL OPTIONS OR ROUTINE SELECTION OPTIONS.

THE OPTIONS ARE SELECTED AS FOLLOWS.

- A. SET BIT SWITCHES ACCORDING TO TABLE 1.
- B. DEPRESS CONSOLE INTERRUPT AND WAIT FOR PRINTOUT.
- C. IF MORE OPTIONS ARE REQUIRED, REPEAT STEPS A AND B.

TABLE 1. 1130 DISK TEST OPTIONS

DATA ENTRY SWITCH POSITION (X INDICATES NOT USED AND D INDICATES DATA)		DESCRIPTION OF OPTIONS
FUNCTION	PROGRAM SEQUENCE NUMBER	MODIFIERS
0 1	2 3 4 5 6 7	8 9 10 11 12 13 14 15
0 0	0 0 P P P P	1 X
		X 1
		X 1
		X 1
		X 1
		X 1
		X 1
0 1	0 0 P P P P	0 0 0 0 0 0 0 0
		0 0 0 0 D D D D
		0 0 0 0 1 1 0 1
1 0	0 0 P P P P	0 0 0 0 0 0 0 0
		0 0 0 0 0 0 0 1
		0 0 0 0 0 0 1 0
		0 0 0 0 0 1 0 0
		0 0 0 0 1 0 0 0

TABLE 2. DSW BITS

DSW BITS																MEANING
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
.	.	.	.	.	X	X	X	X	X	X	X	X	.	.	.	.....SECTOR COUNT (TWO BITS)
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	..CARRIAGE HOME
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	..DISK BUSY (READ/WRITE/CONTROL)
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	..DISK NOT READY
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	..OPERATION COMPLETE (W/INTERRUPT)
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	..DATA ERROR

3.2.3 PATCH OPTIONS

THERE ARE FIVE PATCH OPTIONS AVAILABLE WITH THIS PROGRAM. TO USE ONE OR ALL OF THESE OPTIONS, MAKE UP PATCH CARD(S) AS SHOWN.

```

&XXXX DDDD
.....
.....
.....
..... DATA TO BE ENTERED IN SPECIFIED CORE LOC*N
..... 'CORE ADDRESS' TAKEN FROM PROGRAM LISTING
.....
..... 12-PUNCH IN COLUMN ONE (1)
    
```

NOTE - THESE PATCH OPTIONS MAY BE INSERTED, AFTER PROGRAM LOAD, THROUGH THE CONSOLE SWITCHES BUT GREAT CARE MUST BE TAKEN TO COMPUTE CORRECTED ADDRESSES USING NECESSARY RELOCATION FACTORS.

1. SELECTION OF FIRST RANDOM NUMBER TO BE USED BY THE RANDOM SEEK ROUTINE (3). THIS NUMBER WILL BE USED AS THE FIRST RANDOM NUMBER ON EVERY ROUTINE PASS. TO SELECT THIS OPTION, DETERMINE THE ADDRESS OF THE VARIABLE 'RNSK' FROM THE PROGRAM LISTING. THIS ADDRESS IS THE 'CORE ADDRESS' TO BE PUNCHED IN THE PATCH CARD. THE RANDOM NUMBER TO BE INSERTED SHOULD BE TAKEN FROM A PREVIOUS SUMMARY PRINTOUT, LINE 5.

2. SELECTION OF FIRST RANDOM NUMBER TO BE USED BY THE RANDOM PATTERN WRITE-READ ROUTINES, 9 AND 10. THIS NUMBER WILL BE USED AS THE FIRST RANDOM NUMBER OF THE FIRST PATTERN GENERATED, ON EVERY ROUTINE PASS. TO SELECT THIS OPTION, DETERMINE THE ADDRESS OF THE VARIABLE 'RNDWR' FROM THE PROGRAM LISTING. THIS ADDRESS IS THE 'CORE ADDRESS' TO BE PUNCHED IN THE PATCH CARD. THE RANDOM NUMBER TO BE INSERTED SHOULD BE TAKEN FROM A PREVIOUS SUMMARY PRINT, EITHER LINE 6 OR LINE 7, DEPENDING UPON THAT WAS FAILING.

OPTIONS 3, 4, AND 5 BELOW ALLOW CONTINUATION OF RANDOM GENERATION FROM A PREVIOUS LOAD OF THE THE PROGRAM. THESE PATCHES ALLOW THE CE TO SPECIFY THE STARTING POINT OF THE RANDOM PATTERNS. VALUES TO BE ENTERED CAN BE FOUND IN THE LAST SUMMARY TABLE PRINTED.

3. SELECTION OF FIRST RANDOM NUMBER OF THE FIRST PROGRAM PASS ONLY, TO BE USED BY THE RANDOM SEEK ROUTINE. THIS OPTION WILL CAUSE THE RANDOM PATTERN GENERATION TO CONTINUE FROM THE LAST NUMBER USED ON PREVIOUS LOADS OF LOADS OF THE PROGRAM. THE ADDRESS OF THE VARIABLE 'PRNSK' IS THE 'CORE ADDRESS' TO BE PUNCHED ON THE PATCH CARD.
4. SELECTION OF FIRST RANDOM NUMBER TO BE USED BY ROUTINE NINE (9). THIS PATCH WILL CAUSE THE RANDOM PATTERN GENERATION TO CONTINUE FROM A PREVIOUS LOAD OF THE PROGRAM. THE ADDRESS OF THE VARIABLE 'LRN1' IS THE 'CORE ADDRESS' TO BE PUNCHED IN THE PATCH CARD.
5. SELECTION OF FIRST RANDOM NUMBER TO BE USED BY THE RANDOM WRITE-READ ROUTINE (10). THIS OPTION WILL CAUSE THE RANDOM PATTERN GENERATION TO CONTINUE FROM A PREVIOUS LOAD OF THE PROGRAM. THE ADDRESS OF THE VARIABLE 'LRN2' IS THE CORE ADDRESS TO BE PUNCHED IN THE PATCH CARD.

3.3 PROGRAM HALTS

THIS PROGRAM HAS ONE ERROR TRAP (7OFF), IN THE SEEK TIMING ROUTINE (D). THIS TRAP WILL OCCUR IF THERE IS NO INTERRUPT FROM THE DISK AFTER A SEEK XIO IS EXECUTED. THE FUNCTION TEST WILL HAVE TO BE RESTARTED USING THE SWITCH OPTIONS TO CONTINUE.

NOTE - THE SEEK TIMING TEST IS A SPECIAL ROUTINE AND MUST BE SELECTED IN ORDER TO BE RUN.

3.4 PROGRAM TERMINATION

THE PROGRAM WILL NORMALLY TERMINATE AFTER ONE COMPLETE PASS, UNLESS THE DIAGNOSTIC MONITOR OPTION OF LOOP ALL PROGRAMS IS SELECTED. SEE DM USE PROCEDURE FOR THIS OPTION.

THE PROGRAM CAN BE MANUALLY TERMINATED IN ONE OF TWO WAYS.

1. BY THE MONITOR HALT OPTION (BIT 15 ONLY).
2. BY USE OF THE HALT OPTION (SEE SW OPTIONS).

THIS PROGRAM WILL ALSO BE TERMINATED WHEN CERTAIN ERRORS OCCUR. ALL SUCH TERMINATIONS WILL BE ACCOMPANIED BY AT LEAST TWO PRINTOUTS, ONE STATING THE ERROR, AND THE SECOND STATING THAT THE TEST WAS TERMINATED. THE DFT WILL HAVE TO BE RESTARTED TO CONTINUE.

3.5 PROGRAM RESTART

THE PROGRAM CAN BE RESTARTED FOLLOWING ANY TERMINATION BY SETTING THE BIT SWITCHES TO '0X80', WHERE X IS THE PROGRAM LOAD NUMBER, AND THEN PRESSING CONSOLE INTERRUPT.

4. PRINTOUTS

ALL PRINTOUTS FROM THIS FUNCTION TEST WILL INCLUDE THE LAST DISK DSW THAT WAS SENSED, AND THE NUMBER OF THE FILE BEING TESTED. ALL MESSAGE MODIFIERS MOST EASILY UNDERSTOOD AS DECIMAL NUMBERS, SUCH AS ERROR COUNTS, CYLINDER NUMBERS, AND FILE NUMBER, WILL BE PRINTED AS DECIMAL NUMBERS. ALL NUMBERS WHICH SHOULD BE ANALYZED FOR BITS BEING ON OR OFF, ARE PRINTED IN HEX.

4.1 STATUS MESSAGES

A0900 FILE DSW CPU CYCLE TIME \_ NN 2310 DFT START  
000X XXXX

THE PROGRAM DETERMINED THAT THE CPU CYCLE TIME IS NN, WHERE NN IS EITHER '22' OR '36', REPRESENTING 2.2 AND 3.6 USEC MEMORY SPEED. THE DETERMINATION IS MADE BY EXECUTING A 'SHIFT TO RED' IOCC TO THE CONSOLE TYPEWRITER. THIS MESSAGE WILL OCCUR EVERY TIME THE FUNCTION TEST BEGINS A LOOP.

A0901 RTN# RTN ADRS FILE DSW DSK NOT INIT  
000X XXXX

THE HEXIDECIMAL WORD 'CEDC' WAS NOT FOUND ON CYLINDER 199, SECTOR 6. THE DISK PROBABLY IS NOT INITIALIZED WITH THE 2315 DIAGNOSTIC DISK INITIALIZER (0308).

A0902 RTN# RTN ADRS FILE DSW INVLD RTN#  
000X XXXX

AN INVALID ROUTINE NUMBER WAS SELECTED USING SWITCH FUNCTION ONE. THE DFT MUST BE RESTARTED TO CONTINUE.

A0903 RTN# RTN ADRS FILE DSW RECOVERED RD ERR  
000X XXXX

AFTER ONE OR MORE DSW ERRORS ON A READ, A SUCCESSFUL READ WAS EXECUTED WITHOUT DSW ERRORS. THIS PRINTOUT WILL BE PRECEDEE BY ONE OR MORE ERROR MESSAGES.

A0904 RTN# RTN ADRS FILE DSW HRD CMP ERR  
000X XXXX

AFTER EIGHT (8) RETRIES, DATA READ FROM THE DISK DID NOT COMPARE WITH THAT EXPECTED. THIS MESSAGE INDICATES THAT THE DATA WAS PROBABLY WRITTEN WRONG BY A WRITE ROUTINE, OR THAT THE DISK IS NOT INITIALIZED. LOOK AT THE ROUTINE NUMBER TO DETERMINE IF IT IS A WRITE ROUTINE.

A0906 RTN# RTN ADRS FILE DSW RECOVERED SK ERR  
000X XXXX

AFTER ONE OR MORE SEEK ERRORS, THE DESIRED CYLINDER WAS REACHED. THIS PRINTOUT WILL BE PRECEDEE BY ONE OR MORE SEEK ERROR PRINTOUTS.

A0907 RTN# RTN ADRS FILE DSW DFT TERMINATED  
000X XXXX

THE DISK FUNCTION TEST HAS BEEN TERMINATED. THE PREVIOUS PRINTOUTS WILL EXPLAIN THE REASON FOR THE TERMINATION. THE DFT WILL HAVE TO BE RESTARTED TO CONTINUE THE TEST.

A0908 RTN# RTN ADRS FILE DSW SEEK TIMING IN PROGRESS  
000X XXXX

FILE DSW MAX MIN AVG INCR SK TIME (MS)  
000X XXXX NNNN NNNN NNNN NNNN

SEEK TIMING PRINTOUT (ROUTINE D). THIS PRINTOUT OCCURS WHEN THE SEEK TIMING ROUTINE IS EXECUTED. WHEN THE TIMING ROUTINE STARTS, THE CPU IS DEDICATED TO THE DISK AND ALL OTHER TESTS WHICH MAY BE RUNNING IN OVERLAP, ARE SUSPENDED UNTIL THE CONCLUSION OF THIS TEST.

SEEK INCREMENT IS THE NUMBER OF CYLINDERS SEEKED.

MAX \_ MAXIMUM SEEK TIME OUT OF 20 SEEKS FOR THE GIVEN INCR.  
MIN \_ MINIMUM SEEK TIME OUT OF 20 SEEKS FOR THE GIVEN INCR.  
AVG \_ AVERAGE OF THE 20 SEEKS FOR THE GIVEN INCREMENT  
INCR \_ SEEK INCREMENT

NOTE -- AVERAGE SEEK TIME \_ 15MS X (NUMBER OF 20 MIL STEPS +  
NUMBER OF 10 MIL STEPS)

A0909 RTN# RTN ADRS FILE DSW PASS SKS SFT HRD 7-LINE SMRY  
000X XXXX 00NN NNNN NNNN NNNN

FILE DSW PASS RDS SFT HRD  
000X XXXX 00NN NNNN NNNN NNNN

FILE DSW PASS WRTS SFT HRD  
000X XXXX 00NN NNNN NNNN NNNN

FILE DSW PASS #WDS IN SECTOR  
000X XXXX 00NN NNNN

FILE DSW PASS SK-RNDM WDS  
000X XXXX 00NN FRST LAST

FILE DSW PASS WRT-RNDM WDS/RTN9  
000X XXXX 00NN FRST LAST

FILE DSW PASS WRT-RNDM WDS/RTN10  
000X XXXX 00NN FRST LAST

THIS PRINTOUT OCCURS EVERY TIME THE DFT COMPLETES A PASS.  
IT ALSO OCCURS WHENEVER THE DFT IS TERMINATED WITH AN ERROR.

PASS \_ 00NN THE TOTAL NUMBER OF COMPLETE PASSES  
SKS \_ NNNN THE TOTAL NUMBER OF SEEKS EXECUTED  
SFT \_ NNNN THE NUMBER OF SOFT SEEK ERRORS  
HRD \_ NNNN THE NUMBER OF HARD SEEK ERRORS  
RDS \_ NNNN THE TOTAL NUMBER OF READS EXECUTED  
SFT \_ NNNN THE NUMBER OF SOFT READ ERRORS  
HRD \_ NNNN THE NUMBER OF HARD READ ERRORS  
WRTS \_ NNNN THE TOTAL NUMBER OF WRITES EXECUTED  
SFT \_ NNNN THE NUMBER OF SOFT WRITE ERRORS  
HRD \_ NNNN THE NUMBER OF HARD WRITE ERRORS  
#WDS IN SECTOR \_ AVG # WORDS WRITTEN WITH WRITE 400  
SK RNDM WDS WORDS USED TO GENERATE RANDOM SEEKS.  
HIGH ORDER EIGHT BITS ONLY ARE USED

WRT RNDM WDS RTN9 WORDS USED TO GENERATE RANDOM  
PATTERNS FOR RTN 9

WRT RNDM WDS RTN10 WORDS USED TO GENERATE RANDOM  
PATTERNS FOR RTN 10

FRST \_ FIRST RANDOM NUMBER USED ON THIS PASS  
LAST \_ LAST RANDOM NUMBER USED ON THIS PASS - ALSO  
IS THE FIRST NUMBER USED ON THE NEXT PASS

4.2 ERROR MESSAGES

E0901 RTN# RTN ADRS FILE DSW LAST IOCC LOST INT  
000X XXXX XXXX XXXX

AFTER THE EXECUTION OF AN I/O INSTRUCTION, THE DFT LOOPS THROUGH THE MONITOR WAITING FOR THE EXPECTED INTERRUPT. IF A TIMEOUT OCCURS, THIS MESSAGE IS PRINTED AND THE DFT IS TERMINATED.

E0902 RTN# RTN ADRS FILE DSW LAST IOCC EXTRA INT  
000X XXXX XXXX XXXX

AN UNEXPECTED INTERRUPT OCCURRED. IF THIS PRINTOUT IS PRECEDED BY THE 'LOST INTERRUPT' MESSAGE, THE INTERRUPT OCCURRED BUT WAS LATE.

E0903 RTN# RTN ADRS FILE DSW DSK NOT RDY  
000X XXXX

THE DISK DRIVE 'FILE#' IS NOT READY AND/OR BUSY WHEN IT SHOULD BE BOTH READY AND NOT BUSY. THIS CHECK IS MADE PRIOR TO THE EXECUTION OF ANY SFEK, READ, OR WRITE. THE FUNCTION TEST WILL LOOP WAITING UNTIL THE DISK COMES READY, PRINTING THIS MESSAGE APPROXIMATELY EVERY TWELVE SECONDS.

E0904 RTN# RTN ADRS FILE DSW LAST IOCC DSW WRONG  
000X XXXX XXXX XXXX

THE DISK IS CHECKED THAT IT GOES BUSY AND NOT READY IMMEDIATELY AFTER AN XIO FOR A READ, SEEK, OR WRITE. THIS MESSAGE IS PRINTED IF THE DISK DID NOT GO BUSY AND/OR NOT READY AFTER SUCH AN XIO. THE PROGRAM THEN ENTERS A LOOP THROUGH THE MONITOR CHECKING FOR A LOST INTERRUPT.

IF THIS ERROR OCCURS DURING OVERLAP SEE NOTE ON I/O TEST INDEX PAGE, P/N 2191291.

E0905 RTN# RTN ADRS FILE DSW LAST IOCC DSW SK ERR  
000X XXXX XXXX XXXX

A DSW ERROR WAS INDICATED AFTER THE DP COMPLETE INTERRUPT OCCURRED FOR THE SEEK IOCC SPECIFIED. THE SEEK WILL THEN BE VERIFIED BY READING SECTOR IDS BEFORE RE-SEEKING.

E0906 RTN# RTN ADRS FILE DSW LAST IOCC DSW RD ERR  
000X XXXX XXXX XXXX

THE DSW INDICATED AN ERROR OCCURRED WHILE READING. THE DATA WILL BE COMPARED WITH THAT EXPECTED BEFORE ANY RETRIES ARE MADE.

E0907 RTN# RTN ADRS FILE DSW LAST IOCC HRD DSW RD ERR  
000X XXXX XXXX XXXX

AFTER EIGHT (8) RETRIES, A SUCCESSFUL READ USING THE GIVEN IOCC COULD NOT BE EXECUTED. THIS PRINTOUT WILL BE PRECEDED BY EIGHT 'E0906' MESSAGES.

E0908 RTN# RTN ADRS FILE DSW LAST IOCC DSW WRT ERR  
000X XXXX XXXX XXXX

THE DSW INDICATED AN ERROR OCCURRED AFTER THE EXECUTION OF A WRITE IOCC. THE DATA WRITTEN WILL BE CHECKED FOR VALIDITY AND RE-WRITTEN IF ANY COMPARE ERRORS OCCUR.

E0909 RTN# RTN ADRS FILE DSW FROM S/B WAS SEEK ERR  
000X XXXX 00CC 00CC 00CC

WHEN A SEEK WAS ATTEMPTED AND VERIFIED, IT WAS FOUND THAT THE WRONG CYLINDER WAS OBTAINED. SEEKS ARE ALL VERIFIED BY READING ALL EIGHT SECTOR ID'S. THE CYLINDER NUMBER OF ALL ID'S MUST BE THE SAME, AND THE EIGHT SECTOR COUNTS (BITS 13-15) MUST BE SEQUENTIAL (0-7) BEFORE THE SEEK IS ACCEPTED AS VERIFIED.

E0910 RTN# RTN ADRS FILE DSW CYL# OOH# #WDS CMP ERR  
000X XXXX 00CC 00HS NNNN

FILE DSW WD# S/B WAS  
000X XXXX NNNN DDDD DDDD

FILE DSW WD# S/B WAS  
000X XXXX NNNN DDDD DDDD

FILE DSW WD# S/B WAS  
000X XXXX NNNN DDDD DDDD

FILE DSW #ERS  
000X XXXX NNNN

COMPARE ERROR. THIS PRINTOUT WILL OCCUR IF THE DATA READ FROM THE DISK DOES NOT COMPARE WITH THE DATA THAT WAS EXPECTED FROM THE DISK. THERE ARE THREE THINGS THAT THIS PRINTOUT CAN MEAN.

1. DATA WAS MISREAD FROM THE DISK
2. DATA WAS WRITTEN INCORRECTLY ON A PRECEDING WRITE
3. THE DISK IS NOT INITIALIZED

IN ORDER TO DETERMINE WHICH WAS THE CAUSE OF THIS MESSAGE, FIRST LOOK FOR A PREVIOUS PRINTOUT STATING THAT A WRITE ERROR OCCURRED. IF THERE IS NOT SUCH A PRINTOUT, THEN THIS PRINTOUT WAS NOT CAUSED BY FAULTY WRITE CIRCUITRY.

IF THIS PRINTOUT OCCURS REPEATEDLY FOR THE SAME CYL. AND SECTOR AND NO WRITE ERROR PRINTOUT HAS OCCURRED, THEN THE DISK IS PROBABLY NOT INITIALIZED.

IF THIS PRINTOUT OCCURS INTERMITTENTLY, THEN THE ERRORS ARE PROBABLY DUE TO FAULTY READ CIRCUITRY.

USE CAREFUL JUDGMENT IN DETERMINING WHICH CONDITION EXISTS BEFORE LOOKING FOR ANY HARDWARE FAULTS.

CYL# - CYLINDER WHICH WAS READ (DECIMAL)  
OOHS - TRACK ON THAT CYLINDER  
H - HEAD USED (0 OR 1)  
S - SECTOR READ (0, 1, 2, OR 3)  
#WDS - READ WORD COUNT USED IN THE READ XIO  
WD# - POSITION IN THE I/O TABLE OF THE WORD  
S/B - DATA EXPECTED  
WAS - DATA FOUND  
#ERS - NUMBER OF BAD DATA WORDS FOUND

NOTE - FOR EVERY BAD WORD THAT IS FOUND, THE WORD PRECEDING IT, AND THE WORD FOLLOWING IT, AS WELL AS THE BAD WORD, ARE PRINTED TO GIVE AN OVERALL PICTURE OF THE I/O AREA.

E0911 RTN# RTN ADRS FILE DSW CYL# OOH# #WDS OVERREAD  
000X XXXX OCCC OOH# NNNN  
OVERREAD. BEFORE ANY READS ARE EXECUTED, THE I/O AREA IS SET TO HEXADECIMAL 'FFFF'. THE TWO WORDS FOLLOWING THE EXPECTED READ AREA ARE ALSO SET TO 'FFFF'. WHEN DATA IS READ AND COMPARED, THESE TWO WORDS ARE CHECKED TO SEE THAT THEY ARE STILL SET TO 'FFFF'. IF NOT, THEN THIS MESSAGE IS PRINTED.

RTN# RTN ADRS FILE DSW CYL# OOH# #WDS  
CYL# - CYLINDER NUMBER ON WHICH THE ERROR OCCURRED (DECIMAL)  
OOH# - TRACK ON WHICH ERROR OCCURRED  
#WDS - WORD COUNT USED IN READ IOCC

E0912 RTN# RTN ADRS FILE DSW WRITE ERROR  
000X XXXX  
THE DATA JUST WRITTEN HAS BEEN READ AND COMPARED WITH THE DATA EXPECTED, SUPPRESSING ALL PRINTOUTS WHICH MIGHT HAVE OCCURRED. A HARD COMPARE ERROR WAS INDICATED (EIGHT RETRIES). AFTER THIS MESSAGE IS PRINTED, THE DATA WILL BE COMPARED AGAIN, THIS TIME NOT SUPPRESSING PRINTOUTS. ALL COMPARE ERRORS MAY THEN BE INTERPRETED AS WRITE ERRORS.

E0913 RTN# RTN ADRS FILE DSW CYL# HOME BIT WRONG  
000X XXXX OCCC  
HOME BIT IN DSW (BIT 4) IS WRONG. IF THE CYLINDER NUMBER GIVEN IS ZERO, THEN THE HOME BIT WAS FOUND OFF WHEN THE ARM WAS AT HOME. IF NON-ZERO, THEN THE HOME BIT WAS ON WHEN THE ARM WAS AT THE GIVEN POSITION (DECIMAL).

E0914 RTN# RTN ADRS FILE DSW LAST IOCC NO DSW ERR/WRT 400  
000X XXXX XXXX XXXX  
A WRITE WITH A WORD COUNT OF 400 WAS EXECUTED. THIS WRITE OPERATION SHOULD HAVE BEEN TERMINATED BY THE NEXT SECTOR GAP AND AN ERROR INDICATED IN THE DSW. NO SUCH ERROR WAS INDICATED.

E0915 RTN# RTN ADRS FILE DSW LAST IOCC NO DSW ERR/RD 370  
000X XXXX XXXX XXXX  
A READ WITH A WORD COUNT OF 370 WAS EXECUTED. THIS READ OPERATION SHOULD HAVE BEEN TERMINATED BY THE NEXT SECTOR GAP AND AN ERROR INDICATED IN THE DSW. NO SUCH INDICATION WAS GIVEN.

E0916 RTN# RTN ADRS FILE DSW WDCNT SECT WD CNT WRONG  
000X XXXX NNNN  
A READ WITH A WORD COUNT OF 370 WAS EXECUTED. THE I/O AREA WAS PRESET TO HEXADECIMAL 'FFFF' SO THAT THE NUMBER OF WORDS TRANSFERRED COULD BE DETERMINED. IF THE NUMBER OF WORDS TRANSFERRED IS LESS THAN 331 OR GREATER THAN 358, THIS ERROR MESSAGE WILL BE PRINTED.

E0917 RTN# RTN ADRS FILE DSW WRT 400 WROTE NEXT SECT  
000X XXXX  
A WRITE WITH A WORD COUNT OF 400 WAS EXECUTED. THE NEXT SECTOR WAS THEN READ TO DETERMINE IF THE WRITE OPERATION TERMINATED WHEN THE SECTOR GAP CAME UP. IF NOT, THIS ERROR MESSAGE IS PRINTED.

E0918 RTN# RTN ADRS FILE DSW LAST IOCC #TMS RDCK XFPD DATA  
000X XXXX XXXX XXXX NNNN  
A READ USING THE READ-CHECK MODIFIER BIT WAS EXECUTED. THE I/O AREA WAS PRESET TO HEX 'FFFF' BEFORE THE RD CK. AFTERWARDS, THE I/O WAS NOT 'FFFF', INDICATING DATA WAS TRANSFERRED. #TMS - NUMBER OF READ-CHECKS WHICH WERE DATA DVD NORD

E0919 RTN# RTN ADRS FILE DSW SECT CNTR BAD  
000X XXXX XXXX XXXX NNNN  
THE SECTOR COUNTS IN BITS 14-15 OF THE DSW INDICATE THE NEXT THEY CHANGE FROM 0-1-2-3 AS THE DISK ROTATES. IF NOT, THIS MESSAGE IS PRINTED AND THE DSW IS TERMINATED SINCE ALL READ AND WRITE OPERATIONS HINGE ON GOOD SECTOR COUNT.

E0920 RTN# RTN ADRS FILE DSW 8 CYLINDER IDS  
000X XXXX XXXX XXXX XXXX XXXX XXXX  
BEFORE LOOKING FOR SECTOR IDS IN SECTOR 1, THE SECTOR IDS IN SECTORS 2-8 ARE CHECKED FOR INDICATING THE SAME CYLINDER AND FOR HAVING SEQUENTIAL SECTOR COUNTS. IF NOT, THIS MESSAGE IS PRINTED UP TO SEVEN RETRIES ARE MADE TO MEET THE ABOVE CONDITIONS. IF THIS MESSAGE IS PRINTED EIGHT TIMES FOR THE SAME CYLINDER, THEN THE DISK IS PROBABLY NOT INITIALIZED.

E0924 RTN# RTN ADRS FILE DSW SK/HOME BIT ERR  
000X XXXX XXXX XXXX  
THE SEEK TIMING ROUTINE SEEKS FORWARD TEN (10) TIMES, THEN BACKWARDS TEN (10) TIMES FOR EVERY SEEK INCREMENT. THE ROUTINE STARTS WITH THE ARM AT HOME AND THE ARM SHOULD BE AT HOME WHEN THE TEST IS FINISHED. THE ONLY CHECK FOR THE ARM BEING AT HOME IS THE HOME BIT (BIT 4) IN THE DSW. IF THIS BIT IS NOT SET, A SEEK ERROR IS PROBABLY INDICATED. THE TIMING TEST RESULTS ARE PROBABLY INVALID.

5. COMMENTS

5.1 TEST ROUTINES

HEXADECIMAL ROUTINE NUMBER	ROUTINE DESCRIPTION
0	<p>---- PRECN ----</p> <p>THIS ROUTINE MAY NEVER BE EXPLICITLY EXECUTED AS A ROUTINE. IT IS RUN ONCE EACH TIME THE DISK FUNCTION TEST IS LOOPED REGARDLESS OF ANY OPTIONS. THE ROUTINE WILL -</p> <p>A. ISSUE AN XIO SEEK TOWARD HOME OF 203 CYLINDERS TO TEST THE HOME BIT IN THE DSW (BIT 4).</p> <p>B. CHECK THE SECTOR COUNT IN THE DSW FOR PROPER STEPPING - 0123 . IF THE SECTORS COUNTS FOUND ARE NOT SEQUENTIAL THE DFT IS TERMINATED AND MUST BE RESTARTED USING THE SWITCH OPTIONS (SEE PROGRAM TERMINATION ABOVE).</p> <p>C. SEEK CYLINDER 199. READ THE HISTORY TABLE ON SECTOR THREE (3) TO SET UP THE BAD CYLINDER TABLE IN THE PROGRAM. IF THE HEXADECIMAL WORD 'CEDC' IS NOT THE SECOND WORD ON THE SAME TRACK THE DFT IS TERMINATED AND MUST BE RESTARTED USING THE SWITCH OPTIONS (SEE PROGRAM TERMINATION ABOVE).</p>
1	<p>---- RNT1 ----</p> <p>THIS ROUTINE FIRST SEEKS AND VERIFIES HOME. ALL EIGHT SECTORS ON THE HOME CYLINDER ARE THEN READ, IN ORDER 0 THROUGH 7. THE READ WORD COUNT IS SET TO 1 AT THE START, AND IS DOUBLED EVERY TIME ALL EIGHT SECTORS ARE READ, UNTIL THE READ WORD COUNT REACHES 256. THE READ COUNT IS THEN SET TO 320. THE DATA THAT IS READ IS NOT VERIFIED AS BEING CORRECT UNLESS THE DSW INDICATES AN ERROR OCCURRED, EXCEPT THAT THE DATA IN SECTOR 7 IS ALWAYS VERIFIED THIS ROUTINE IS LOOPED 20 TIMES</p>
2	<p>---- RTN2 ----</p> <p>INCREMENTAL SEEK ROUTINE. THIS ROUTINE SEEKS 2 CYLINDERS IN AND 1 OUT FROM CYLINDER 0 TO 202. THEN THEN ROUTINE SEEKS 2 OUT AND 1 IN FROM CYLINDER 202 TO 0 (HOME). EACH SEEK IS VERIFIED BY READING ALL EIGHT SECTOR IDS ON EACH CYLINDER. CYLINDERS 90-110 WILL NEVER BE ATTEMPTED.</p>
3	<p>---- RTN3 ----</p> <p>RANDOM SEEK ROUTINE. THIS ROUTINE WILL EXECUTE 100 RANDOM SEEKS. EACH SEEK IS VERIFIED BY READING ALL EIGHT SECTOR IDS. CYLINDERS 90-110 ARE NEVER ATTEMPTED.</p> <p>- NOTE - THE SAME RANDOM SEEK PATTERN WILL ALWAYS BE EXECUTED IF THE VARIABLE 'RNSK' IS PATCHED (SEE PATCH OPTIONS).</p>

4	<p>---- RTN4 ----</p> <p>READ AND VERIFY 320 WORDS OF HEXADECIMAL '1313' FROM CYLINDER 1 SECTOR 0. THIS ROUTINE IS LOOPED 50 TIMES.</p>
5	<p>---- RTN5 ----</p> <p>READ AND VERIFY 320 WORDS OF HEXADECIMAL 'E5E5' FROM CYLINDER 201 SECTOR 6. THIS ROUTINE IS LOOPED 50 TIMES.</p>
6	<p>---- RTN6 ----</p> <p>WRITE 320 WORDS OF HEXADECIMAL '1313' ON CYLINDER 2 SECTOR 0. READ BACK THE RECORD AND VERIFY THAT IT WAS WRITTEN CORRECTLY. LOOP THIS ROUTINE 50 TIMES.</p>
7	<p>---- RTN7 ----</p> <p>WRITE 320 WORDS OF HEXADECIMAL 'E5E5' ON CYL. 202, SECTOR 6. READ BACK THE RECORD AND VERIFY THAT IT WAS WRITTEN CORRECTLY. LOOP THIS ROUTINE 50 TIMES.</p>
8	<p>---- RTN8 ----</p> <p>READ-CHECK CYLINDER 1, SECTOR 0 AND VERIFY THAT NO DATA WAS TRANSFERRED TO CORE. IF DATA IS TRANSFERRED, PRINT AN ERROR MESSAGE IMMEDIATELY INDICATING THE NUMBER OF TIMES THAT DATA WAS TRANSFERRED TO CORE. LOOP THIS ROUTINE 50 TIMES.</p>
9	<p>---- RTN9 ----</p> <p>GENERATE 320 WORDS OF RANDOM DATA AND WRITE THESE WORDS ON CYLINDER 2, SECTOR 0. READ AND VERIFY THE DATA WAS WRITTEN CORRECTLY. LOOP THE ROUTINE 50 TIMES WITH DIFFERENT PATTERNS.</p> <p>- NOTE - THE SAME RANDOM PATTERN WILL ALWAYS BE WRITTEN IF THE VARIABLE 'LRN1' IS PATCHED (SEE PATCH OPTIONS).</p>
A	<p>---- RTN10 ----</p> <p>GENERATE 320 WORDS OF RANDOM DATA AND WRITE THESE WORDS ON CYLINDER 202 SECTOR 6. READ AND VERIFY THAT THE DATA WAS WRITTEN CORRECTLY. LOOP THIS ROUTINE 50 TIMES WITH DIFFERENT PATTERNS.</p> <p>- NOTE - THE SAME RANDOM PATTERN WILL ALWAYS BE WRITTEN IF THE VARIABLE 'LRN2' IS PATCHED (SEE PATCH OPTIONS).</p>
B	<p>---- RTN11 ----</p> <p>WRITE 320 WORDS OF HEXADECIMAL '1313' ON CYLINDER 2, SECTOR 0. REWRITE THE SAME SECTOR WITH A WORD COUNT OF ONE (1). READ 321 WORDS FROM THE SECTOR AND CHECK THAT ALL 320 WORDS FOLLOWING THE ONE WORD THAT WAS WRITTEN ARE ZERO. RESTORE THE ORIGINAL DATA (HEX '1313') ON SECTOR.</p>



C ---- RTN12 ----

ISSUE A WRITE COMMAND WITH A WORD COUNT OF 400 ON CYLINDER 2, SECTOR 0. THE HARDWARE SHOULD TERMINATE THE WRITE AT NEXT SECTOR GAP. READ THE DATA JUST WRITTEN TO DETERMINE THE NUMBER OF WORDS THAT WERE WRITTEN. CALCULATE THE AVERAGE WORD COUNT FOR FIFTY (50) PASSES AND SAVE FOR SUMMARY TABLE PRINTOUT. IF WORD COUNT IS LESS THAN 331 OR GREATER THAN 358, AN ERROR MESSAGE IS PRINTED IMMEDIATELY.

D ---- RTN13 --- \*\*\*SPECIAL ROUTINE\*\*\*

\*\*\*SEEK TIMING ROUTINE\*\*\*

THIS ROUTINE WILL BE RUN ONLY IF IT IS SPECIFICALLY SELECTED. IT WILL TAKE CONTROL OF THE CPU FOR THE DURATION OF THE TIMING TEST. SEEK INCREMENTS OF 1, 2, 4, 8, 16, 32, 64, 128, AND 200 CYLINDERS ARE TIMED. THE MAXIMUM TIME FOUND, THE MINIMUM TIME FOUND AND THE AVERAGE TIME ARE ALL PRINTED AT THE END OF THE ROUTINE FOR EVERY SEEK INCREMENT.

SEEK TIME \_ 15MS X (NUMBER OF 20 MIL STEPS +  
NUMBER OF 10 MIL STEPS)

5.2 DESCRIPTION OF SUB-ROUTINES

INDEX REGISTER 2 IS SET TO THE ADDRESS 'TB' AT THE START OF THE PROGRAM AND EVERY ROUTINE EXPECTS THIS REGISTER TO CONTAIN THAT ADDRESS. 'TB' IS THE POINTER TO A COMMON TABLE OF CONSTANTS AND ROUTINE CALLS. ALL RETURNS FROM CALLS ON THE MONITOR GO THROUGH A ROUTINE WHICH SET UP THIS POINTER ALSO. REGISTER 2 IS NEVER SET TO ANY OTHER VALUE.

ALL THE SUBROUTINES DESCRIBED PROVIDE ANY NECESSARY ERROR AND STATUS PRINTOUTS. (REFERENCE SECTION 4.)

ALL SUBROUTINES WHICH REQUIRE THE OPTION ARE PROVIDED WITH THE 'LOCK ON ERROR' OPTION. THIS OPTION WILL LOOP THE SUBROUTINE IN THE SMALLEST POSSIBLE LOOP WHICH CAUSED THE ORIGINAL ERROR. ONCE AN ERROR OCCURS, AND THE OPTION IS SELECTED, THE ROUTINE WILL REMAIN IN THE LOOP AS LONG AS THE 'LOCK ON ERROR' SWITCH IS ON, WHETHER THE ERROR RECURS OR NOT. IF NO ERROR OCCURS THEN THE OPTION HAS NO EFFECT.

\*\*\*CALL\*\*\*

...BSI 2 STMLS-TB

THIS SUBROUTINE IS USED TO SET AN ENTRY IN THE MAIN LINE SEQUENCE CONTROL FIELD (MLSCF) WHEN EXITING TO THE MONITOR. THE ROUTINE SAVES ALL INDEX REGISTERS, SETS A RETURN ADDRESS IN 'MLSCF+1', AND GOES TO MONITOR 'START'. UPON RETURN FROM MONITOR, THE SUBROUTINE RESTORES THE INDEX REGISTERS, AND RETURNS TO THE CALLING ROUTINE.

...BSI 2 XEQ-TB  
DC RETURN1

THIS SUBROUTINE BUILDS THE IOCC WITH THE DEVICE AREA CODE, SETS THE INTERRUPT SWITCH NON-ZERO, AND SETS UP A LOOP COUNT TO TEST THE SELECTED DEVICE FOR READY. IF THE DEVICE IS READY, THIS ROUTINE TESTS THE VALUE OF 'COMA'. IF ZERO, EXIT IS TAKEN IMMEDIATELY. IF NON-ZERO, THE XIO SET UP IN LOCATION 'ZXIO' IS EXECUTED. A LOOP IS THEN SET UP THROUGH THE MONITOR WAITING FOR A LOST INTERRUPT. WHEN THE INTERRUPT OCCURS, EXIT IS MADE TO CALL+1.

...BSI 2 CKLK-TB  
...DC RETURN1

THIS SUBROUTINE IS USED TO CHECK FOR THE LOCK ON ERROR OPTION SELECTED. IF THE SWITCH IS ON THE ROUTINE RETURNS INDIRECTLY VIA THE ADDRESS RETURN1 IN CALL+1. IF THE SWITCH IS OFF THE ROUTINE RETURNS DIRECTLY TO CALL+2.

...BSI 2 SETV-TB  
A-REG CONTAINS DATA TO SET

THIS SUBROUTINE SETS THE I/O AREA TO THE CONTENTS OF THE A REGISTER. THE NUMBER OF WORDS TO BE SET MUST BE STORED IN LOCATION 'CCMA' PRIOR TO THE CALL.

...BSI 2 RNDOM-TB  
A-REG CONTAINS LAST RANDOM NUMBER

THIS ROUTINE USES THE NUMBER IN THE A REGISTER TO GENERATE A RANDOM NUMBER, AND RETURNS TO CALL+1 WITH THE NEW NUMBER IN THE A REGISTER. THE NUMBER IN THE A REGISTER AT THE TIME OF THE CALL IS NORMALLY THE LAST RANDOM NUMBER USED.

...BSI 2 STMSG-TB  
DC FMM  
F \_ FORM NUMBER  
MMM\_ MESSAGE ID.

ALTERNATE CALL

...BSI L STMSG+/8000  
DC FMM

THIS MESSAGE WILL BE PRINTED AS AN ADDITIONAL LINE MESSAGE, AND MID-RID-RAD WILL NOT BE PRINTED.

THIS ROUTINE WILL SET UP THE MESSAGE SPECIFIED BY THE FORM NUMBER. THE MESSAGE ID IS THEN CHECKED TO SEE IF HEX CHARACTER ONE IS AN E. IF IT IS AN E THE DIAGNOSTIC MONITOR ERROR ROUTINE IS CALLED. OTHERWISE THE LOG ROUTINE IS CALLED, UNLESS THE BYPASS LOG MESSAGES OPTION IS SELECTED. IF THE CALL WAS TO THE ERROR ROUTINE THE DIAGNOSTIC MONITOR OPTION OF LOOP ON ERROR IS CHECKED AND IF SELECTED THE TEST ROUTINE CAUSING THE ERROR WILL BE LOOPED. THIS ROUTINE NORMALLY EXITS TO CALL+2.

...BSI 2 VERIFY-TB  
DC CYL. NO. DESIRED  
DC ERROR ADDRESS RETURN

THIS ROUTINE WILL SEEK THE DESIRED CYLINDER, USING THE CYL-ADDRESS IN 'PCYL#' AS THE PRESENT DISK POSITION. ALL EIGHT SECTOR IDS ARE THEN READ. IF ALL EIGHT SECTOR ADDRESSES INDICATE THE SAME CYLINDER NUMBER, AND ALL EIGHT SECTOR IDS (BITS 13-15) ARE SEQUENTIAL (0-7), THE ROUTINE EXITS TO CALL +3.

IF NOT, AN ERROR MESSAGE IS PRINTED IMMEDIATELY. AN ERROR RETRY PROCEDURE IS THEN INITIATED. ALL EIGHT SECTOR IDS ARE REREAD UP TO SEVEN TIMES AND EVALUATED USING THE ABOVE CRITERION. IF THIS CRITERION IS NOT THEN MET, THE DFT IS TERMINATED.

...BSI 2 READ-TB  
DC WORD COUNT  
DC NUMBER (USED IN CALL TO CMP ROUTINE)  
DC ERROR RETURN  
(A REG. CONTAINS SECTOR DESIRED.)

THIS ROUTINE WILL BUILD THE READ IOCC, PRESET THE I/O AREA TO HEX 'FFFF', SET THE READ WORD COUNT IN THE I/O AREA, AND EXECUTE THE READ THROUGH THE XEQ SUBROUTINE.

UPON RETURN FROM XEQ, THE DSW IS CHECKED TO SEE IF ANY ERROR IS INDICATED. IF A DSW ERROR IS INDICATED AFTER ANY READ, AN ERROR MESSAGE IS PRINTED.

WITH OR WITHOUT DSW ERRORS, A CALL IS MADE ON THE 'CMP' SUBROUTINE TO CHECK THE DATA TO SEE THAT IT WAS AS EXPECTED. IF NOT, UP TO EIGHT RETRIEES ARE MADE FOR A GOOD READ. IF ALL RETRIES FAIL, THE ROUTINE EXITS TO THE ERROR RETURN ADDRESS.

...BSI CMP  
MDX ERROR RETURN

THIS ROUTINE WILL MAKE A WORD BY WORD COMPARISON OF THE DATA READ AGAINST THE DATA EXPECTED. THE ROUTINE WILL ALSO MAKE A CHECK OF THE WORD COUNTER BY COMPARING TO SEE IF MORE WORDS WERE TRANSFERRED THAN EXPECTED. IF NO ERRORS ARE FOUND THE ROUTINE EXITS DIRECTLY TO

CALL+2. OTHERWISE THE EXIT IS TO CALL+1.

...BSI 2 WRITE-TB  
DC NUMBER OF WORDS (IF BIT 0 \_ 1, DO NOT PRESET I/O AREA)  
DC DATA TO BE PRESET IN I/O AREA  
DC ERROR RETURN  
(A REG. CONTAINS THE SECTOR NUMBER)

THIS ROUTINE WILL PRESET THE I/O AREA 'COMA' WITH THE DATA TO BE WRITTEN. THE ROUTINE WILL PLACE THE CURRENT VERIFIED SECTOR ID AT THE I/O AREA PLUS 1 AND THE WRITE WORD COUNT AT I/O AREA. THE ROUTINE WILL BUILD THE WRITE IOCC, AND ISSUE THE XIO THROUGH THE 'XEQ' ROUTINE. UPON RETURN FROM 'XEQ' THE DSW IS CHECKED FOR ERRORS.

IF NO DSW ERROR IS INDICATED, EXIT IS MADE TO CALL+4 WITHOUT ANY CHECKING.

IF A DSW ERROR IS INDICATED, AN ERROR MESSAGE IS PRINTED IMMEDIATELY AND EXIT IS MADE TO THE ERROR RETURN ADDRESS. NO RETRIES ARE MADE TO WRITE A GOOD RECORD WITHIN THIS SUBROUTINE.

----- LAST PAGE -----



2310 B DISK FUNCTION TEST

```

**          H _ HEAD (0-1)          30901380
**          S _ SECTOR (0-3)        30901390
*          30901400
* SETV          30901410
*   A REG._NO.TO BE SET IN COMA AREA 30901420
*   COMA _ WORD COUNT                30901430
*   BSI  2 SETV-TB   PRESET COMA+1 TO COMA+N+2 30901440
**          TO VALUE IN A REG.      30901450
*          30901460
* STMLS        30901470
*   BSI  2 STMLS-TB  SAVE IX 1, IX 2 AND EXIT 30901480
**          * MONITOR. RETURN IS TO      30901490
**          * CALL+1.                   30901500
*          30901510
* STMSG        30901520
*   BSI  2 STMSG-TB  PRINT MESSAGE          30901530
*   DC   MESSAGE ID  30901540
*          30901550
****ALTERNATE CALL USED 30901560
*   BSI  L STMSG+/8000 30901570
*   DC   MESSAGE ID  30901580
*          30901590
* VERIFY      30901600
*   BSI  2 VERIFY-TB  SEEK A CYLINDER AND VER- 30901610
**          * IFY CYL. BY READING AT      30901620
**          * LEAST TWO SECTOR ID'S.     30901630
*   DC   CYL # DESIRED 30901640
*   DC   RETURN ADRS IF CYL # INVALID OR     30901650
**          BAD FROM 2315 DISK INITIALIZER. 30901660
*          30901670
* WRITE       30901680
*   A REG._SECTOR COUNT 30901690
*   BSI  2 WRITE-TB   WRITE IN PRESENT CYLINDER 30901700
*   DC   NO.OF WORDS TO BE WRITTEN          30901710
**          IF BIT 0_1, DONT PRESET      30901720
**          THE I/O AREA                 30901730
*   DC   DATA TO BE WRITTEN                30901740
*   DC   ADRS OF DSW ERROR RETURN          30901750
*          30901760
****ALTERNATE CALL USED 30901770
*   BSI  L WRITE+/8000 30901780
*   DC   NO OF WORDS  30901790
*   DC   DATA TO BE WRITTEN                30901800
*          30901810
* XEQ         30901820
*   BSI  2 XEQ-TB    EXECUTE I/O AND WAIT     30901830
**          * FOR INTERRUPT              30901840
*          30901850
*          30901860
*          30901870
*          30901880
*          30901890
*          30901900
*          30901910
*          30901920
*          30901930
*          30901940
*          30901950
*          30901960
*          30901970
*          30901980
*          30901990
*          30902000
*          30902010
*          30902020
*          30902030
*          30902040
*          30902050

```

0160 0  
0161 0

2310 B DISK FUNCTION TEST

```

0162 0
0163 0
0164 0
0165 0
0166 0
0167 0
0168 0
017A 0
018A 0
019A 0
01AA 0
01BA 0
01BB 0
01BC 0
01BD 0
05DC
05DC 0 0309
05DD 0 0000
05DE 0 0000
05DF 0 0000
05E0 0 0000
05E1 0 0000
05E2 0 0000
05E3 1 0855
05E4 1 0814
05E5 0 0000
05E6 0 0000
05E7 0 FFFF
05E8 1 0FFA
05E9 0 2000
05EA 0 8800
05EB 0 9000
05EC 0 9800
05ED 0 A000
05EE 0 0000
05EF 0 0000
05F0 0 FFFF
05F1 0 FFFF
05F2 0 FFFF
05F3 0 4480 0160
05F5 1 05DC

```

```

ERROR EQU      START+1  ERROR LOG ROUTINE  30902060
LOG EQU        ERROR+1  STATUS LOG ROUTINE  30902070
END EQU        LOG+1   END ROUTINE          30902080
*              30902090
*-----*
* MONITOR CONTROL WORD ADDRESSES 30902100
*-----*
RTNSW EQU      END+1   ROUTINE START SWITCH  30902130
ERLCK EQU      END+2   LOCK ON ERROR CONTROL 30902140
LOGBY EQU      END+3   I/O BUSY SWITCH ADDRESS 30902150
ADDRS EQU      END+4   ACTIVE PGRM MLSCF ADRS 30902160
*              30902170
*-----*
* INTERRUPT TRANSFER VECTOR ADDRESSES 30902180
*-----*
ILO EQU        /017A   INTERRUPT LEVEL ZERO  30902210
IL1 EQU        IL0+16  INTERRUPT LEVEL ONE   30902220
IL2 EQU        IL1+16  INTERRUPT LEVEL TWO   30902230
IL3 EQU        IL2+16  INTERRUPT LEVEL THREE  30902240
IL4 EQU        IL3+16  INTERRUPT LEVEL FOUR  30902250
RQTY EQU       IL4+1   CONSOLE PRINTER REQUEST 30902260
RQKB EQU       RQTY+1  USE KEYBOARD REQUEST  30902270
SVKB EQU       RQKB+1  KB SERVICE REQUEST    30902280
*              30902290
*****
* ORG *+1500 30902300
*****
* PROGRAM CONTROL TABLE 30902330
*****
*
* MONITOR INTERFACE TABLES 30902370
*
PID DC /0309 2310-1130 FUNCTION TEST 30902390
RID DC 0 ROUTINE ID 30902400
RAD DC 0 ROUTINE ADRS 30902410
SWO DC 0 SWITCH FUNCTION 00 30902420
SW1 DC 0 SWITCH FUNCTION 01 30902430
SW2 DC 0 SWITCH FUNCTION 10 30902440
SW3 DC 0 SWITCH FUNCTION 11 30902450
LPA DC ZLPA LOOP PROGRAM ADDRESS 30902460
IPA DC ZIPA RESTART ADDRESS 30902470
* 30902480
MLSCF DC 0 MAIN LINE SEQUENCE 30902490
DC 0 * CONTROL FIELD 30902500
TERM DC /FFFF TERMINATOR 30902510
DC PEND LAST PROGRAM ADDRESS 30902520
*-----*
DVAS DC /2000 AREA CODE-CPU DRIVE 30902540
DC /8800 AREA CODE-UNIT 1 30902550
DC /9000 AREA CODE-UNIT 2 30902560
DC /9800 AREA CODE-UNIT 3 30902570
DC /A000 AREA CODE-UNIT 4 30902580
RNSDK DC 0 PATCH OPTION-RANDOM SEEK 30902590
RNDWR DC 0 PATCH OPTION RANDOM WRT 30902600
BADCY DC -1 30902610
DC -1 30902620
DC -1 30902630
* 30902640
* BEGIN ROUTINE 30902650
* 30902660
BEG BSI I BEGIN EXIT TO MONITOR 30902670
DC PID ADRS OF PID 30902680
*-----*
* INTERRUPT ROUTINE 30902700
* 30902710
* 30902720
*-----*
* 30902730

```

2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

05F6 0 0000	INTSW DC	0	INTERRUPT SWITCH	30902740
05F7 0 0000	DVA DC	0	AREA CODE STORAGE	30902750
05F8 0 0000	INTR DC	0	INTRPT ENTRY	30902760
05F9 0 C0FC	LD	INTSW	GET INTRPT SWITCH	30902770
05FA 1 6700 0604	LDX L3	INTRB	GET RETURN ADDRESS	30902780
05FC 0 4818	SKP	+-	SKIP IF INT EXPECTED	30902790
05FD 0 6BE7	STX	3 MLSCF	SET IN TABLE FOR MONITOR	30902800
05FE 0 1010	INTRC SLA	16	CLEAR INTRPT SW	30902810
05FF 0 D0F6	STD	INTSW	*	30902820
* 30902830				
0600 0 0815	XID	ZSNS	SENSE DSW	30902840
0601 0 D014	STU	ZSNS	STORE DSW IN TEMP LOC*N	30902850
0602 1 4C80 05F8	BSC	I INTR	EXIT ROUTINE	30902860
* 30902870				
* 30902880				
* 30902890				
* 30902900				
0604 0 C011	INTRB LD	ZSNS	GET INTRPT DSW	30902910
0605 0 D021	STO	TBDSW	STORE FOR PRINT	30902920
0606 1 6600 065C	LDX L2	TB	SET UP POINTER	30902930
0608 0 4232	BSI	2 STMMSG-TB	PRINT EXTRA INT MSG	30902940
0609 0 5E02	DC	/5E02	MESSAGE ID	30902950
060A 0 4480 0161	BSI	I START	EXIT TO MONITOR	30902960
* 30902970				
* 30902980				
* 30902990				
* 30903000				
* 30903010				
* 30903020				
* 30903030				
065C 0	TB EQU	P10+128	CONSTANT TABLE POINTER	30903040
0014 0	LP13 EQU	20	NUMBER OF LOOPS RTN 13	30903050
* 30903060				
060C 0 0404	XSKBK DC	/0404	FNC/MOD-SEEK OUT	30903070
060D 0 3000	H3000 DC	/3000	FORM NUMBER FOR PRSUM	30903080
060E 0 1313	H1313 DC	/1313	CONSTANT HEX 1313	30903090
060F 0 0123	MASK DC	/0123	ORDER OF SECTOR COUNTS	30903100
0610 0 0500	WRMOD DC	/0500	WRITE FUNCTION	30903110
0611 0 0600	DSKMD DC	/0600	READ FNC/MOD	30903120
0612 0 0680	RDCHK DC	/0680	READ-CHECK IOCC	30903130
0613 0 0700	SNRES DC	/0700	SENSE/RESET CONSTANT	30903140
0614 0 0900	TYRED DC	/0900		30903150
0615 0 8000	H8000 DC	/8000	CONSTANT HEX 8000	30903160
0616 0 0000	BSS E	0		30903170
0616 0 0000	ZSNS DC	*-*	USED AS TEMP LOC*N	EVEN 30903180
0617 0 0000	DC	*-*	SENSE-RESET IOCC	ODD 30903190
0618 0 0000	ZXID DC	*-*	COMMON IOCC STORAGE	EVEN 30903200
0619 0 0000	DC	*-*	*	ODD 30903210
061A 0 0000	SNXID DC	*-*	IOCC-STORAGE-SENSE	EVEN 30903220
061B 0 0000	DC	*-*		ODD 30903230
061C 1 0614	TYRED DC	/0614	SHIFT TO RED IOCC	30903240
061D 0 0900	DC	/0900		30903250
061E 0 0000	SNSTY DC	*-*	SENSE DSW W/RESET	30903260
061F 0 0F01	DC	/0F01		30903270
0620 0 E5E5	HE5E5 DC	/E5E5	CONSTANT HEX E5E5	30903280
0621 0 0000	MSG0 DC	*-*	MESSAGE STORAGE	ODD 30903290
0622 0 0000	HXDC DC	*-*	HEX/DEC FLAG	EVEN 30903300
0623 0 0000	MSGID DC	*-*	MESSAGE ID	ODD 30903310
0624 1 0E1F	ADDR1 DC	ALPHO		30903320
0625 0 0000	ADDR2 DC	0	ALPHA ADDRESS 2	30903330
0626 0 0000	FILE# DC	*-*	FILE NUMBER	ODD 30903340
0627 0 0000	TBDSW DC	*-*	DSW	EVEN 30903350
0628 0 0000	MOD3 DC	*-*	MODIFIERS	EVEN 30903360
0629 0 0000	MOD4 DC	*-*	*	ODD 30903370
062A 0 0000	MOD5 DC	*-*	*	EVEN 30903380
062B 0 0000	MOD6 DC	*-*	*	ODD 30903390
062C 0 0000	MOD7 DC	*-*	*	EVEN 30903400
062D 0 0000	BNTMP DC	*-*	TEMP STORAGE	ODD 30903410

062E 0 0000	PCYL# DC	*-*	PRESENT CYLINDER	EVEN 30903420
062F 0 0000	NCYL# DC	*-*	NEXT DESIRED CYLINDER	ODD 30903430
0630 0 0000	ERSK1 DC	*-*	SEEK RTN ERROR SW	EVEN 30903440
0631 0 0000	CNTB DC	*-*	SEEK RTN-RETRY CTR B	ODD 30903450
0632 0 0000	RTRYA DC	*-*	RETRY CTRS	EVEN 30903460
0633 0 0000	RTRYB DC	*-*	*	ODD 30903470
0634 0 0000	RDDSW DC	*-*	ERROR CTRS	EVEN 30903480
0635 0 0000	RDCMP DC	*-*	*	ODD 30903490
0636 0 0001	K1 DC	1	CONSTANT 1	30903500
0637 0 0002	K2 DC	2	CONSTANT 2	30903510
0638 0 0003	K3 DC	3	CONSTANT 3	30903520
0639 0 0004	K4 DC	4	CONSTANT 4	30903530
063A 0 0007	K7 DC	7	CONSTANT 7	30903540
063B 0 0008	K8 DC	8	CONSTANT 8	ODD 30903550
063C 0 000A	TEN DC	10	TEN	30903560
063D 0 FFF2	KM14 DC	-14	USED TO CALCULATE CPU TIME	30903570
063E 0 0024	K36 DC	36	USED TO CALCULATE CPU TIME	30903580
063F 0 0064	HUNDR DC	100	ONE HUNDRED	30903590
0640 0 00CA	K202 DC	202	CONSTANT 202	30903600
0641 0 00CB	K203 DC	203	CONSTANT 203	30903610
0642 0 00FD	K253 DC	253	DECIMAL CONSTANT	30903620
0643 0 014B	K331 DC	331	CONSTANT	30903630
0644 0 03E8	THOUS DC	1000	ONE THOUSAND	30903640
0645 0 2710	K10TH DC	10000	CONSTANT 10000 DEC	30903650
0646 0	SUMRY EQU	*	SUMMARY TABLE ORIGIN	30903660
0646 0 0000	PSRCT DC	*-*	PASS COUNT	30903670
0647 0 0000	SKCNT DC	*-*	NUMBER OF SEEKS	30903680
0648 0 0000	SFTSK DC	*-*	NUMBER SOFT SK ERRORS	30903690
0649 0 0000	HRDSK DC	*-*	NUMBER HARD SK ERRORS	30903700
064A 0 0000	RDCTN DC	*-*	NUMBER OF READS	30903710
064B 0 0000	SFTRD DC	*-*	NUMBER SOFT RD ERRORS	30903720
064C 0 0000	HRDRD DC	*-*	NUMBER HARD RD ERRORS	30903730
064D 0 0000	WRCNT DC	*-*	NUMBER OF WRITES	30903740
064E 0 0000	SFTWR DC	*-*	NUMBER SOFT WR ERRORS	30903750
064F 0 0000	HRDWR DC	*-*	NUMBER HARD WR ERRORS	30903760
0650 0 0000	WRLNG DC	*-*	AVG SECT LNGTH/WRT 400	30903770
0651 0 0000	PRSW DC	*-*	IF NON ZERO-BYPASS PRT	30903780
000C 0	SMLNG EQU	*-SUMRY	LENGTH OF SUMMARY TABLE	30903790
0652 0 0000	FRNSK DC	*-*	FIRST RANDOM SK ISSUED	30903800
0653 0 0000	PRNSK DC	*-*	LAST RANDOM SEEK ISSUED	30903810
0654 0 0000	FRN1 DC	*-*	FIRST RANDOM # RTN 9	30903820
0655 0 0000	LRN1 DC	*-*	LAST RANDOM # RTN 9	30903830
0656 0 0000	FRN2 DC	*-*	FIRST RANDOM # RTN 10	30903840
0657 0 0000	LRN2 DC	*-*	LAST RANDOM # RTN 10	30903850
0658 0 0000	CNTA DC	*-*	SEEK RTN-RETRY CTR	30903860
0659 0 0000	RNDCK DC	*-*	TEMP STORAGE	30903870
065A 0 0000	NOCK DC	*-*	BYPASS RD CKS IF NON 0	30903880
065B 0 0000	LNGTH DC	*-*	RECORD LENGTH STORAGE	30903890
065C 0 0000	INDEX DC	*-*	INDEX POINTER	30903900
065D 0 0000	ERCT DC	*-*	CMP RTN ERROR CTR	30903910
065E 0 0000	DC	*-*	TEMP STORAGE CMP RTN	EVEN 30903920
065F 0 0000	S#B DC	*-*	*	30903930
0660 0 0000	DC	*-*	*	30903940
0661 0 0000	IDS#B DC	*-*	PRESENT SECT/CYL ID	ODD 30903950
0662 0 0000	LPRNT DC	*-*	LAST WORD PRINTED	30903960
0663 0 0000	ZCNT DC	*-*	DELAY COUNT	30903970
0664 0 0000	RTCNT DC	*-*	RETRY COUNTER	30903980
0665 0 0000	WRTY DC	*-*	RETRY COUNTER	30903990
0666 0 0000	WRERR DC	*-*	ERROR SWITCH	30904000
0667 0 0000	RTNER DC	*-*	RTN ERROR COUNTER	30904010
0668 0 0000	CMPTM DC	*-*	TEMP STORAGE	30904020
0669 0 007B	K123 DC	123	DECIMAL CONSTANT	30904030
066A 0001	CPUT2 TYPE	.22.		30904040
066B 0001	CPUT3 TYPE	.36.		30904050
066C 0 1510	TLP13 DC	5392	RTN 13 LOOP TIME (NSEC)	30904060
066D 0 0014	KLP13 DC	LP13	AVERAGING CONSTANT RTN 13	30904070
066E 0 0000 0001	TIMCN DC	1	LOOP ADDEND RTN 13	EVEN 30904080
0670 0 0000	TIMSL DC	*-*	RTN 13 LOOP (NSEC)	30904090

2310 B DISK FUNCTION TEST

```

*
0671 0 019A ADIL2 DC IL2 INTERRUPT ADDRESS 30904100
0672 1 05F8 ADINT DC INTR ADDRESS OF INTERRUPT RTN 30904110
0673 1 0AE0 ADIN2 DC INT2 INTRPT ADRS TIME RTN 30904120
0674 1 069E ADCMA DC COMA ADRS OF COMA 30904130
0675 1 0869 ADPCN DC PRECN ADRS PRE CNTL RTN 30904140
0676 0 0000 CKLK DC ** ADRS PRE CNTL RTN 30904150
0677 1 4C00 0D07 BSC L CKLKE ENTRY TO CK LOCK OPTION 30904160
* RTN 30904170
30904180
0679 0 0000 CKPRT DC ** CHECK PRINT-ALL-ERRORS OPT 30904190
067A 1 4C00 0D15 BSC L CKPRE ENTRY POINT 30904200
* 30904210
067C 0 0000 CNTLE DC ** BRANCH TO CONTROL 30904220
067D 1 4C00 08B7 BSC L CNTL * ROUTINE 30904230
* 30904240
067F 0 0000 COUNT DC ** ENTRY TO INCR SUMMARY 30904250
0680 1 4C00 0D5E BSC L COUNE * COUNT RTN 30904260
* 30904270
0682 0 0000 READ DC ** ENTRY TO DISK READ 30904280
0683 1 4C00 08EB BSC L R DEN * RTN 30904290
* 30904300
0685 0 0000 RNDOM DC ** ENTRY TO GENERATE 30904310
0686 1 4C00 0D01 BSC L RNDME * RANDOM NUMBER RTN 30904320
* 30904330
0688 0 0000 SETV DC ** ENTRY TO SET I/O AREA 30904340
0689 1 4C00 0D1F BSC L SETVE * RTN 30904350
* 30904360
068B 0 0000 STMLS DC ** ENTRY TO SAVE INDEXING 30904370
068C 1 4C00 0830 BSC L STMLE * AND EXIT TO MONITOR 30904380
* 30904390
068E 0 0000 STMSG DC ** ENTRY TO SET UP AND 30904400
068F 1 4C00 0D67 BSC L STMSE * PRINT A MESSAGE 30904410
* 30904420
0691 0 0000 TEXT DC ** ENTRY OF CALL 30904430
0692 1 4C00 0D28 BSC L DFTXT TO TERMINATE DFT 30904440
* 30904450
0694 0 0000 VERIFY DC ** ENTRY TO SEEK A CYL AND 30904460
0695 1 4C00 083F BSC L VRFYE * VERIFY THE SEEK 30904470
* 30904480
0697 0 0000 WRITE DC ** ENTRY TO WRITE DATA 30904490
0698 1 4C00 0CD5 BSC L WRTEN * ON THE DISK 30904500
* 30904510
069A 0 0000 XEQ DC ** ENTRY TO EXECUTE AN 30904520
069B 1 4C00 0AF9 BSC L XEQE * XIO INSTRUCTION 30904530
* 30904540
* 30904550
* 30904560
* 30904570
* 30904580
* 30904590
069E 0176 COMA BSS E 374 THIS SETS *BDCYL* EVEN 30904600
080C 0 BDCYL EQU COMA+366 USED IN VERIFY ONLY 30904610
* 30904620
* 30904630
* 30904640
* 30904650
* 30904660
* 30904670
* 30904680
* 30904690
* 30904700
* 30904710
* 30904720
0814 0 630C ZIPA LDX 3 SHLNG LENGTH OF SUMMARY TABLE 30904730
0815 0 1010 SLA 16 CLEAR A REGISTER 30904740
0816 1 D700 0645 ZIPL STO L3 SUMRY-1 CLEAR SUMMARY TABLE 30904750
0818 0 73FF MDX 3 -1 DECR POINTER 30904760
0819 0 70FC B ZIPL LOOP 30904770
*
    
```

2310 B DISK FUNCTION TEST

```

081A 1 6600 065C LDX L2 TB SET UP TABLE POINTER 30904780
081C 0 C285 LD 2 SW2-TB GET SW FNC TWO 30904790
081D 0 100C SLA 12 CLEAR BITS 0-11 30904800
081E 0 180C SRA 12 ** 30904810
081F 0 6310 LDX 3 16 SET INITIAL COUNT 30904820
0820 0 1340 SLCA 3 0 CREATE DRIVE NUMBER 30904830
0821 1 C700 05E9 LD L3 DVAS GET CORRECT AREA CODE 30904840
0822 0 D29B STO 2 DVA-TB SET IN DEV. AREA CODE 30904850
0824 1 6F00 0626 STX L3 FILE# FOR PRINTOUTS 30904860
0826 0 C215 LD 2 ADIL2-TB GET ADDRESS OF IL2 30904870
0827 0 92CA S 2 FILE#-TB CREATE INTERRUPT ENTRY 30904880
0828 1 D400 0AFB STO L XEQ+1 SET FOR XIO RTN 30904890
* 30904900
082A 0 C480 0167 ZIPB LD I LOGBY GET TYPEWRITER BUSY SWITCH 30904910
082C 1 4C18 0830 BZ ZIPC BRANCH IF NOT BUSY 30904920
082E 0 422F BSI 2 STMLS-TB ELSE GO TO MONITOR 30904930
082F 0 70FA MDX ZIPC RETURN HERE..GO TEST AGAIN 30904940
* 30904950
0830 0 6500 0F00 ZIPC LDX L1 /OF00 SENSE DSW FOR TYPEWRITER 30904960
0832 1 6D00 061F STX L1 SNSTY+1 STORE FOR TESTING TY 30904970
0834 0 0AC2 XIO 2 SNSTY-TB SENSE TYPEWRITER DSW 30904980
0835 0 1004 SLA 4 GET BUSY BIT 30904990
0836 1 4C28 082A BN ZIPC LOOP IF BUSY 30905000
0838 1 7401 061E MDX L SNSTY,1 SET RESET BIT 30905010
083A 0 C216 LD 2 ADINT-TB GET INTERRUPT VECTOR 30905020
083B 0 D400 018B STO L RQTY SET FOR TYPEWRITER INTRPT 30905030
083D 0 D29A STO 2 INTSW-TB SET INTRPT SWITCH FOR INTR 30905040
083E 0 6500 07D0 LDX L1 2000 SET TIME CONSTANT FOR CNT 30905050
0840 0 C20F LD 2 CPUT3-TB GET TIME FOR MESSAGE 30905060
0841 1 D400 0E2D STO L CPUTO SET TIME IN MESSAGE 30905070
0843 0 C2C3 LD 2 SNSTY+1-TB GET TYPEWRITER SNS DSW 30905080
0844 0 D2BB STO 2 ZSNS+1-TB SET FOR INTR ROUTINE 30905090
0845 0 0AC0 XIO 2 TYPE-TB SHIFT TO RED 30905100
* 30905110
0846 0 C29A ZIPD LD 2 INTSW-TB GET INTRPT SWITCH 30905120
0847 1 4C18 084F BZ ZIPE BRANCH IF INTRPT OCCURRED 30905130
0849 0 71FF MDX 1 -1 ELSE BUMP TIMER 30905140
084A 0 70FB MDX ZIPD LOOP IF NOT THROUGH 30905150
084B 0 C20E LD 2 CPUT2-TB GET TIME FOR MESSAGE 30905160
084C 1 D400 0E2D STO L CPUTO SET FOR PRINT 30905170
084E 0 C2E1 LD 2 KM14-TB FALL THRU TO HERE FOR 2.2 30905180
084F 0 82E2 ZIPE A 2 K36-TB BR HERE IF 3.6 MEMORY 30905190
0850 0 D2C2 STO 2 SNSTY-TB SAVE IN UPPER WORD.SNSTY 30905200
0851 0 0AC2 ZIPE XIO 2 SNSTY-TB SENSE TYPEWRITER DSW 30905210
0852 0 1004 SLA 4 GET BUSY BIT 30905220
0853 1 4C28 0851 BN ZIPE LOOP IF BUSY 30905230
* 30905240
* 30905250
* 30905260
* 30905270
* 30905280
* 30905290
* 30905300
* 30905310
0855 1 6600 065C ZLPA LDX L2 TB POINTER TO TABLE 30905320
0857 0 C283 LD 2 SWO-TB GET FNC SW 00 30905330
0858 1 4C04 0866 BOD ZLPB BRANCH # BIT 15 SET 30905340
085A 0 C219 LD 2 ADPCN-TB GET ADDRESS OF PRE-CONTROL 30905350
085B 0 D28A STO 2 MLSCF+1-TB SET IN MLSCF 30905360
085C 0 D282 STO 2 RAD-TB SET FOR MESSAGES AND ERROR 30905370
085D 0 D242 STO 2 COMA-TB MAKE COMA NON-ZERO FOR XEQ 30905380
* 30905390
* 30905400
* 30905410
* 30905420
* 30905430
* 30905440
* 30905450
0861 0 C29B LD 2 DVA-TB GET AREA CODE OF SEL. DEV. 30905450
    
```

2310 B DISK FUNCTION TEST

0862 0 EAB7 OR 2 SNRES-TB COBINE WITH SNS IOCC 30905460
0863 0 D2BF STO 2 SNXIO+1-TB SET FOR SENSING 30905470
0864 0 EADA OR 2 K1-TB SET RESET MODIFIER BIT 30905480
0865 0 D2BB STO 2 ZSNS+1-TB SET FOR INTRPT SNS 30905490
0866 0 4480 0161 ZLPB BSI 1 START GO TO MONITOR 30905500
\* PRE-CONTROL ROUTINE
\* THIS 'ROUTINE' TESTS FOR SEQUENTIAL SECTOR COUNTS IN THE DSW, ISSUES A SEEK TOWARD HOME OF 202 CYLINDERS TO ASSURE THAT THE ARM IS HOME AFTER WHICH THE HOME BIT (BIT 4) IS TESTED FOR ON.
\* CALLS ON THE 'VERIFY' ROUTINE TO MAKE SURE THE ARM IS AT HOME.
\* READS THE CE HISTORY TRACK ON CYLINDER 199 SECTOR 6 AND SETS UP THE BAD CYLINDER TABLE.
0868 0 CEDC HCEDC DC /CEDC CE DISK PACK 30905700
0869 1 6600 065C PRECN LDX L2 TB IX2 \_ TABLE POINTER 30905740
086B 0 4232 BSI 2 STMSG-TB PRINT '2310 DFT START' 30905750
086C 0 5A00 DC /5A00 FORM/MESSAGE ID 30905760
086D 0 C2B0 LD 2 XSKBK-TB GET SEEK-TOWARD-HOME IOCC 30905770
086E 0 1890 SRT 16 PUT IN Q REGISTER 30905780
086F 0 D2D2 STO 2 PCYL#-TB SET CYL POINTER 30905790
0870 0 C2E5 LD 2 K203-TB SEEK OUT 203 CYLINDERS 30905800
0871 0 DABC STD 2 ZXIO-TB SET FOR XEQ RTN 30905810
0872 0 423E BSI 2 XEQ-TB EXECUTE IOCC IN ZXIO 30905820
0873 0 6132 LDX 1 50 LOOP COUNTER 30905830
0874 0 1010 PRECA SLA 16 CLEAR A REG. 30905840
0875 0 D242 STO 2 COMA-TB SET FLAG FOR NOT EXECUTE 30905850
0876 0 423E BSI 2 XEQ-TB GO TO CHECK FOR READY 30905860
0877 0 63FC LDX 3 -4 SET COUNTER AND SECTORS 30905870
0878 0 402E PRECB BSI PRECJ GO WAIT FOR SECTOR 30905880
0879 0 C02C LD PRECT GET SECTOR JUST TESTED 30905890
087A 0 E2DC AND 2 K3-TB \* 30905900
087B 0 100C SLA 12 SAVE HEX DIGIT IN Q REG. 30905910
087C 0 18DC RTE 28 \* 30905920
087D 0 7301 MDX 3 1 COUNT NUMBER SECTORS 30905930
087E 0 70F9 B PRECB LOOP 30905940
\* 30905950
087F 0 1090 SLT 16 GET FOUR SECTORS 30905960
0880 0 D2CC STO 2 MOD3-TB SAVE IN CASE PRINT 30905970
0881 0 F2B3 EDR 2 MASK-TB TEST FOR /0123 30905980
0882 1 4C18 0887 BZ PRECD 30905990
0884 0 4232 BSI 2 STMSG-TB ELSE PRINT ERROR MESSAGE 30906000
0885 0 9E13 DC /9E13 SECTOR COUNTER BAD 30906010
0886 0 4235 BSI 2 TEXTIT-TB TERMINATE DFT 30906020
0887 0 71FF PRECD MDX 1 -1 LOOP COUNT DECREMENTED 30906030
0888 0 70EB B PRECA LOOP 30906040
\* 30906050
\* LOOP IS FINISHED 30906060
\* 30906070
0889 0 4238 PRECE BSI 2 VERIFY-TB GO VERIFY ARM AT HOME 30906080
088A 0 0000 DC 0 CYLINDER ZERO 30906090
088B 1 0D28 DC DFTXT ERROR RETURN ADDRESS 30906100
088C 0 4238 BSI 2 VERIFY-TB GO TO CYLINDER 199 30906110
088D 0 00C7 DC 199 \* CE HISTORY TRACK 30906120
088E 1 0D28 DC DFTXT ERROR RETURN ADDRESS 30906130

2310 B DISK FUNCTION TEST

088F 0 C2DC \* LD 2 K3-TB GET 3 30906140
0890 1 4400 8682 BSI L READ+/8000 READ SECTOR 3 30906150
0892 0 001B DC 27 READ 27 WORDS 30906160
\* 30906170
0893 0 C244 \* LD 2 COMA+2-TB GET WORD 2 30906180
0894 0 F0D3 EOR HCEDC CMP WITH S/B 30906190
0895 1 4C18 089A BZ PRECG BRANCH IF OK 30906200
0897 0 4232 BSI 2 STMSG-TB ELSE PRINT MESSAGE 30906210
0898 0 5A01 DC /5A01 NOT INITIALIZED DISK 30906220
0899 0 4235 BSI 2 TEXTIT-TB TERMINATE DFT 30906230
089A 1 6780 06A1 PRECG LDX I3 COMA&3 NO BAD SECTORS-NOT CYLS 30906240
089C 1 6500 05F0 LDX L1 BADCY ADRS OF BAD CYL TABLE 30906250
\* 30906260
089E 1 C700 06A0 PRECF LD L3 COMA&2 GET BAD SECTOR NO. 30906270
08A0 0 1883 SRT 3 RIGHT JUSTIFY 30906280
08A1 0 D100 STO 1 0 STORE IN BAD CYL TABLE 30906290
08A2 0 7101 MDX 1 1 INCR POINTER 30906300
08A3 0 73F8 MDX 3 -8 DECR SECTOR COUNTER 30906310
08A4 0 70F9 B PRECF LOOP 30906320
08A5 0 7011 MDX CNTL GO TO CONTROL ROUTINE 30906330
\* 30906340
\* 30906350
\* 30906360
\* 30906370
\* 30906380
\* 30906390
\* 30906400
\* 30906410
\* 30906420
08A6 0 0000 PRECT DC 0 TEMPORARY STORAGE 30906430
\* 30906440
08A7 0 C000 PRECJ DC \*-\* SUBROUTINE ENTRY POINT 30906450
08A8 0 6BFD STX 3 PRECT SAVE SECTOR COUNT WORD 30906460
08A9 0 C2E9 LD 2 K10TH-TB SET TIMEOUT COUNTER 30906470
08AA 0 D207 STO 2 ZCNT-TB SAVE TIMEOUT COUNT 30906480
08AB 0 0ABE PRECL XIO 2 SNXIO-TB SENSE DSW 30906490
08AC 0 D2CB STO 2 TBDSW-TB SET IN CASE PRINT 30906500
08AD 0 F0F8 EOR PRECT LOOK FOR SAME SECTOR 30906510
08AE 0 E2DC AND 2 K3-TB \* COUNTS AS IN IX3 30906520
08AF 1 4C98 08A7 PRECM BZ I PRECJ 30906530
08B1 1 74FF 0663 MDX L ZCNT,-1 ELSE COUNT TIME 30906540
08B3 0 70F7 MDX PRECL LOOP IF NOT TIMEOUT 30906550
08B4 0 1010 SLA 16 CLEAR A REGISTER 30906560
08B5 0 D0F0 STO PRECT SET SECTOR TO ZERO 30906570
08B6 0 70F8 B PRECM RETURN 30906580
\* 30906590
\* 30906600
\* 30906610
\* 30906620
\* 30906630
\* 30906640
\* 30906650
\* 30906660
\* 30906670
\* 30906680
\* 30906690
0032 0 LPCNT EQU 50 ROUTINE LOOP COUNT 30906700
08B7 1 6600 065C CNTL LDX L2 TB IX2 \_ TABLE ADDRESS 30906710
08B9 0 C283 LD 2 SW0-TB GET SW FNC 0 30906720
08BA 1 4C04 0D28 BUD DFTXT BRANCH # TERM DFT SW ON 30906730
08BC 0 C284 LD 2 SW1-TB GET SW FNC 1 30906740
08BD 1 4C20 08CF BNZ CNTLD BRANCH IF NO RTN SELECTED 30906750
\* 30906760
08BF 0 C281 LD 2 RID-TB GET RTN ID 30906770
08C0 0 82DA A 2 K1-TB ADD 1 30906780
08C1 0 D281 STO 2 RID-TB SAVE 30906790
08C2 0 9013 S RTTBL CK FOR VALID 30906800
08C3 1 4C10 0D2D BNN PRSUM BRANCH IF COMPLETE 30906810
\* 30906820

2310 B DISK FUNCTION TEST

08C5 1 6580 05DD	CNTLB LDX 11 RID	IX EQUAL RTN ID	30906820
08C7 1 C500 08D6	LD L1 RTTBL	GET RTN ADRS	30906830
	*		30906840
08C9 0 D282	STO 2 RAD-TB	SAVE RTN ADRS	30906850
08CA 0 6132	LDX 1 LPCNT	SET LOOP COUNT	30906860
08CB 0 1010	SLA 16	CLEAR ROUTINE ERROR SW	30906870
08CC 0 D20B	STO 2 RTNER-TB	*	30906880
08CD 1 4C80 05DE	B I RAD	BRANCH TO TEST RTN	30906890
	*		30906900
08CF 0 D281	CNTLD STO 2 RID-TB	SET IN RTN ID	30906910
08D0 0 9013	S SPRTN		30906920
08D1 1 4C28 08C5	BN CNTLB	BRANCH IF SPECIAL RTN	30906930
08D3 0 4232	BSI 2 STMSG-TB	ELSE PRINT MESSAGE	30906940
08D4 0 5A02	DC /5A02	INVALID RTN SELECTED	30906950
08D5 0 4235	BSI 2 TEXIT-TB	TERMINATE DFT	30906960
	*		30906970
	*	ROUTINE ADDRESS TABLE	30906980
	*		30906990
08D6 0 000D	RTTBL DC LRTN-RTTBL	LENGTH OF RTN ADRS TABLE	30907000
08D7 1 08E5	DC RTN1	ADDRESS OF TEST ROUTINE 1	30907010
08D8 1 0919	DC RTN2	2	30907020
08D9 1 093C	DC RTN3	3	30907030
08DA 1 094D	DC RTN4	4	30907040
08DB 1 0954	DC RTN5	5	30907050
08DC 1 095A	DC RTN6	6	30907060
08DD 1 095C	DC RTN7	7	30907070
08DE 1 098B	DC RTN8	8	30907080
08DF 1 0983	DC RTN9	9	30907090
08E0 1 0989	DC RTN10	10	30907100
08E1 1 09FC	DC RTN11	11	30907110
08E2 1 0A12	DC RTN12	12	30907120
08E3 0	LRTN EQU *	END OF RTN TABLE	30907130
	*	SPECIAL RTN TABLE	30907140
08E3 1 0A6C	DC RTN13	13	30907150
08E4 0 000E	SPRTN DC SPRTN-RTTBL		30907160
	*		30907170
	*		30907180
	*		30907190
	*		30907200
	*	TEST ROUTINE ONE	30907210
	*		30907220
	*		30907230
	*		30907240
08E5 0 6114	RTN1 LDX 1 20	LOOP COUNTER	30907250
08E6 0 4238	BSI 2 VERIFY-TB	SEEK CYLINDER ZERO	30907260
08E7 0 0000	DC 0	CYLINDER NUMBER	30907270
08E8 1 08E9	DC RTN1A	ERROR RETURN ADDRESS	30907280
	*		30907290
08E9 0 6922	RTN1A STX 1 RTN1E+1	SAVE ROUTINE LOOP COUNTER	30907300
08EA 0 C2DA	LD 2 K1-TB	INITIAL WORD COUNT	30907310
08EB 0 D006	STO WDCNT	SET FOR CALL ON READ RTN	30907320
	*		30907330
08EC 0 61F8	RTN1B LDX 1 -8	READ EIGHT SECTORS	30907340
	*		30907350
08ED 0 6922	RTN1C STX 1 SCTOR	SET FOR MASK	30907360
08EE 0 C021	LD SCTOR	GET PRESENT SECTOR	30907370
08EF 0 E2DE	AND 2 K7-TB	SAVE BITS 13-15	30907380
08F0 1 4400 8682	BSI L READ+/8000	READ WITHOUT CHECKING DATA	30907390
08F2 0 0001	WDCNT DC 1	WORD COUNT	30907400
08F3 0 C2BA	LD 2 ZSNS-TB	GET INTERRUPT DSW	30907410
08F4 1 4C28 08F8	BN RTN1D	BRANCH IF ERROR INDICATED	30907420
08F6 0 7101	MDX 1 1	ELSE READ NEXT SECTOR	30907430
08F7 0 70F5	B RTN1C	LOOP	30907440
	*		30907450
08F8 1 C500 0918	RTN1D LD L1 STORG+7	GET DATA LAST READ	30907460
08FA 0 D20C	STO 2 CMPTM-TB	SET FOR COMPARE	30907470
08FB 1 4400 0C48	BSI L CMP	GO COMPARE	30907480
08FD 0 1000	NOP 0	ERROR RETURN	30907490

2310 B DISK FUNCTION TEST

08FE 0 C0F3	LD WDCNT	GET LAST WORD COUNT	30907500
08FF 0 1001	SLA 1	DOUBLE WORD COUNT	30907510
0900 0 D0F1	STO WDCNT	SET FOR NEXT READ	30907520
0901 0 1006	SLA 6	TEST FOR GREATER THAN 256	30907530
0902 1 4C30 08EC	BP RTN1B	BRANCH IF NOT	30907540
0904 0 1001	SLA 1	SHIFT LEFT	30907550
0905 1 4C30 090B	BP RTN1E	BRANCH IF WORD CNT _ 321	30907560
0907 0 6700 0141	LDX L3 321	ELSE SET MAX WORD COUNT	30907570
0909 0 6BE8	STX 3 WDCNT	SET FOR CALL	30907580
090A 0 70E1	B RTN1B	GO DO IT	30907590
	*		30907600
090B 0 6500 0000	RTN1E LDX L1 *-*	GET RTN LOOP COUNTER	30907610
090D 0 71FF	MDX 1 -1	DECREMENT COUNT	30907620
090E 0 70DA	B RTN1A	LOOP	30907630
090F 0 4220	BSI 2 CNTLE-TB	TERMINATE RTN	30907640
	*		30907650
0910 0 FFF8	SCTOR DC -8	SECTOR POINTER	30907660
	*		30907670
0911 0 1313	STORG DC /1313	SEQUENTIAL SECTOR DATA	30907680
0912 0 E5E5	DC /E5E5	*	30907690
0913 0 1313	DC /1313	*	30907700
0914 0 E5E5	DC /E5E5	*	30907710
0915 0 E5E5	DC /E5E5	*	30907720
0916 0 1313	DC /1313	*	30907730
0917 0 E5E5	DC /E5E5	*	30907740
0918 0 1313	DC /1313	*	30907750
	*		30907760
	*		30907770
	*		30907780
	*	TEST ROUTINE TWO	30907790
	*		30907800
	*	THIS ROUTINE WILL SEEK 2 IN AND 1 OUT FROM	30907810
	*	CYLINDER ZERO TO CYLINDER 202 EXCEPT	30907820
	*	CYLINDERS 90-110. THE ROUTINE WILL THEN	30907830
	*	SEEK 2 OUT AND 1 IN FROM CYLINDER 202 TO	30907840
	*	CYLINDER ZERO. EACH SEEK IS VERFIED FOR	30907850
	*	PROPER CYLINDER REACHED.	30907860
	*		30907870
	*		30907880
0919 0 63FF	RTN2 LDX 3 -1	DECR SEEKS BY ONE	30907890
091A 0 6B1F	STX 3 INCR1	*	30907900
091B 0 6302	LDX 3 2	INCR SEEKS BY TWO	30907910
091C 0 6B1E	STX 3 INCR2	*	30907920
091D 0 C2DA	LD 2 K1-TB	FIRST CYL. WILL BE HOME	30907930
091E 0 4007	BSI RTN2P	EXECUTE SUCCESSIVE SEEKS	30907940
	*		30907950
091F 0 63FE	LDX 3 -2	DECR SEEKS BY TWO	30907960
0920 0 6B19	STX 3 INCR1	*	30907970
0921 0 6301	LDX 3 1	INCR SEEKS BY ONE	30907980
0922 0 6B18	STX 3 INCR2	*	30907990
0923 0 C2E5	LD 2 K203-TB	INI. CYL. + 1	30908000
0924 0 4001	BSI RTN2P	EXECUTE SUCCESSIVE SEEKS	30908010
0925 0 4220	BSI 2 CNTLE-TB	EXIT ROUTINE	30908020
	*		30908030
0926 0 0000	RTN2P DC *-*	ENTRY POINT	30908040
0927 0 D00C	STO RTN2T	STOR FOR CALL	30908050
0928 0 6500 00CA	LDX L1 202	NO. IF SEEKS	30908060
	*		30908070
092A 0 C009	RTN2Q LD RTN2T	GET CYL. NO.	30908080
092B 0 800E	A INCR1	INCR OR DECR	30908090
092C 0 D001	STO RTN2R	STORE FOR SEEK	30908100
092D 0 4238	BSI 2 VERIFY-TB	SEEK CYL. AND VERIFY	30908110
092E 0 C000	RTN2R DC *-*	CYLINDER NO.	30908120
092F 1 0930	DC RTN2S	ADRS ERROR RETURN	30908130
0930 0 C0FD	RTN2S LD RTN2R	GET LAST CYL. NO.	30908140
0931 0 8C09	A INCR2	INCR OR DECR	30908150
0932 0 D001	STO RTN2T	STORE FOR SEEK	30908160
0933 0 4238	BSI 2 VERIFY-TB	CALL VERIFY ROUTINE	30908170



2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

```

0934 0 0000 RTN2T DC *-* CYLINDER DESIRED 30908180
0935 1 0936 DC RTN2U INVALID ADRS RETURN 30908190
0936 0 71FF RTN2U MDX 1 -1 DECR COUNTER 30908200
0937 0 70F2 B RTN2Q NOT FINISHED 30908210
0938 1 4C80 0926 B 1 RTN2P RETURN 30908220
* 30908230
INCR1 DC 0 30908240
INCR2 DC 0 30908250
*----- 30908260
*----- 30908270
*----- 30908280
*----- 30908290
* TEST ROUTINE THREE 30908300
* 30908310
* THIS ROUTINE WILL ISSUE AND VERIFY 100 RANDOM 30908320
* SEEKS. CYLINDERS 90-110 WILL NEVER BE 30908330
* ATTEMPTED. 30908340
* 30908350
*----- 30908360
RTN3 MDX 1 100-LPCNT 100 LOOPS 30908370
LD 2 RND5K-TB GET RNDM SEEK SWITCH 30908380
SKP +- USE AS FIRST IF NOT ZERO 30908390
LD 2 PRNSK-TB ELSE USE LAST GENERATED 30908400
STO 2 FRNSK-TB FIRST RANDOM NUMBER 30908410
B RTN3B * 30908420
* 30908430
RTN3A LD 2 PRNSK-TB GET LAST RNDM NUMBER 30908440
BSI 2 RNDOM-TB GENERATE NEXT 30908450
* 30908460
RTN3B STO 2 PRNSK-TB SAVE 30908470
SRA 8 * FROM 0-202 30908480
STO RTN3C STORE FOR SEEK 30908490
* 30908500
BSI 2 VERIFY-TB SEEK AND VERIFY CYL. 30908510
RTN3C DC *-* CYL. DESIRED 30908520
DC RTN3A IF INVALID, DON'T COUNT 30908530
MDX 1 -1 ELSE COUNT 30908540
B RTN3A LOOP 30908550
BSI 2 CNTLE-TB END OF RTN 30908560
* 30908570
*----- 30908580
*----- 30908590
*----- 30908600
*----- 30908610
* TEST ROUTINE FOUR 30908620
* READ AND CHECK 320 WORDS OF 30908630
* HEX 1313 FROM CYL 1-SECTOR 0 30908640
* 30908650
*----- 30908660
RTN4 LDX 3 1 IX EQUAL CYL 30908670
LDX 1 0 IX EQUAL WRITE SW 30908680
CMN2 LD 2 H1313-TB GET DATA EXPECTED 30908690
LDX 2 0 IX EQUAL SECTOR 30908700
CMN1 STX 1 RTN3C SET RTN VALUES 30908710
LDX 1 LPCNT IX EQUAL LOOP CTR 30908720
B CMRT2 BRANCH TO COMMON RTN 30908730
*----- 30908740
* 30908750
* TEST ROUTINE FIVE 30908760
* READ AND CHECK 320 WORDS OF 30908770
* HEX E5E5 FROM CYL 201-SECTOR 6 30908780
* 30908790
*----- 30908800
RTN5 LDX L3 201 IX EQUAL CYL 30908810
LDX 1 0 IX EQUAL WRT SW 30908820
CMN3 LD 2 HE5E5-TB GET DATA EXPECTED 30908830
LDX 2 6 IX EQUAL SECTOR 30908840
MDX CMN1 BRANCH TO COMPLETE SETUP 30908850

```

```

095A 0 6302
095B 0 70F3
095C 0 6700 00CA
095E 0 70F8
095F 0 6B07
0960 0 6A75
0961 0 D00F
0962 0 D015
0963 0 D01B
0964 1 6600 065C
0966 0 4238
0967 0 C000
0968 1 08B7
0969 0 C288
096A 0 D243
096B 0 C0DC
096C 1 4C18 097A
096E 0 C067
096F 0 423B
0970 0 0141
0971 0 0000
0972 1 0973
0973 0 C2DA
0974 0 D2F5
0975 0 C060
0976 0 4226
0977 0 0141
0978 0 0000
0979 1 0984
097A 0 1010
097B 0 D2F5
097C 0 C059
097D 0 4226
097E 0 0141
097F 0 0000
0980 1 0981
0981 0 71FF
0982 0 70E6
0983 0 4220
0984 0 1010
0985 0 D2F5

```

```

*----- 30908860
* 30908870
* TEST ROUTINE SIX 30908880
* WRITE AND READ 320 WORDS 30908890
* OF HEX 1313 ON CYL 2-SECT 30908900
* 0-CHECK DATA READ 30908910
* 30908920
*----- 30908930
RTN6 LDX 3 2 IX EQUAL CYL 30908940
B CMN2 BRANCH TO COMPLETE SETUP 30908950
*----- 30908960
* 30908970
* TEST ROUTINE SEVEN 30908980
* WRITE AND READ 320 WORDS 30908990
* OF HEX E5E5 ON CYL 202- 30909000
* SECTOR 6-CHECK DATA READ 30909010
* 30909020
*----- 30909030
RTN7 LDX L3 202 IX EQUAL CYL 30909040
B CMN3 BRANCH TO COMPLETE SETUP 30909050
*----- 30909060
* 30909070
* THIS ROUTINE IS COMMON TO 30909080
* TEST ROUTINES 3,4,5,6,AND 7 30909090
* 30909100
*----- 30909110
* 30909120
*----- 30909130
* 30909140
CMRT2 STX 3 CYL SET CYLINDER TO USE 30909150
STX 2 SECTD * 30909160
STO DATA1 SET DATA EXPECTED 30909170
STO DATA2 SET DATA EXPECTED 30909180
STO DATA3 SET DATA EXPECTED 30909190
LDX L2 TB INDEX EQUAL TBL ADRS 30909200
BSI 2 VERIFY-TB SEEK DESIRED CYL AND 30909210
CYL DC *-* * VERIFY THE SEEK 30909220
DC CNTL EXIT ON ERROR 30909230
CMRTL LD 2 TERM-TB SET DATA EXPECTED 30909240
STO 2 COMA+1-TB SET FOR COMPARE RTN 30909250
LD RTN3C IS ROUTINE TO WRITE 30909260
BZ CMRTB BRANCH TO READ ONLY 30909270
LD SECTD GET SECTOR DESIRED 30909280
BSI 2 WRITE-TB GO WRITE DATA 30909290
DC 321 WORD COUNT 30909300
DATA1 DC *-* DATA EXPECTED 30909310
DC CMRTA ERROR EXIT ADDRESS 30909320
CMRTA LD 2 K1-TB SET SW FOR NO PRINT 30909330
STO 2 PRSW-TB * 30909340
LD SECTD GET DESIRED SECTOR 30909350
BSI 2 READ-TB GO READ WITHOUT ERR PRINTS 30909360
DC 321 WORD COUNT 30909370
DATA2 DC *-* EXPECTED DATA 30909380
DC CMRTG ERROR ADDRESS 30909390
* 30909400
CMRTB SLA 16 CLEAR NO-PRINT SWITCH 30909410
STO 2 PRSW-TB * 30909420
LD SECTD GET DESIRED SECTOR 30909430
BSI 2 READ-TB GO READ DATA 30909440
DC 321 WORD COUNT 30909450
DATA3 DC *-* EXPECTED DATA 30909460
DC CMRTD CMP ERROR ADDRESS 30909470
* 30909480
CMRTD MDX 1 -1 DECREMENT LOOP COUNT 30909490
MDX CMRTL LOOP ROUTINE 30909500
BSI 2 CNTLE-TB EXIT ROUTINE 30909510
* 30909520
CMRTG SLA 16 CLEAR NO-PRINT SWITCH 30909530
STO 2 PRSW-TB * 30909540

```



```

09EB 0 73F7 MDX 3 -9 SKP IF RTN 9 30910900
09EC 0 7301 MDX 3 1 INDEX - 2 30910910
09ED 1 0700 0654 STO L3 FRN1 SAVE FIRST NUMBER USED 30910920
09EF 0 C00A LD LSTNO GET LAST NUMBER USED 30910930
09F0 1 0700 0655 STO L3 LRN1 SAVE 30910940
09F2 0 4220 BSI 2 CNTLE-TB EXIT ROUTINE 30910950
* 30910960
* READ OR COMPARE ERROR 30910970
* 30910980
RTN9G SLA 16 CLEAR NO-PRINT SWITCH 30910990
STO 2 PRSW-TB * 30911000
LDX 3 HRDWR-SUMRY COUNT WRITE ERROR 30911010
BSI 2 COUNT-TB COUNT ERROR 30911020
BSI 2 STMSG-TB PRINT ERROR MSG 30911030
DC /5EOC MESSAGE ID 30911040
MDX RTN9D GO PRINT ERRORS 30911050
LSTNO DC 123 LAST RANDOM NUMBER 30911060
FSTNO DC *-* FIRST NUMBER USED 30911070
* 30911080
* 30911090
* TEST ROUTINE ELEVEN- 30911100
* 1. WRITE SECTOR ID PLUS 30911110
* 320 WORDS OF /1313 ON 30911120
* CYL 2-SECT 0. 30911130
* 2. REWRITE CYL 2-SECT 0 30911140
* WITH 1 WORD OF SECTOR 30911150
* ID ONLY. 30911160
* 3. READ 321 WORDS FROM 30911170
* CYL 2-SECT 0. 30911180
* 4. CK DATA READ FOR 30911190
* CORRECT SECT ID AND 30911200
* DATA OF ALL ZEROS. 30911210
* 5. RESTORE ORIGINAL DATA 30911220
* ON CYL 2-SECT 0. 30911230
* 30911240
* 30911250
RTN11 BSI 2 VERFY-TB SEEK CYL 2-VERIFY 30911260
DC 2 * THE SEEK 30911270
DC CNTL EXIT ON ERROR 30911280
B RT11B WRITE 321 WORDS FIRST 30911290
* 30911300
RT11L SLA 16 WRITE CYL 2-SECT 0 30911310
BSI 2 WRITE-TB * WITH SECTOR ID 30911320
DC 1+/8000 * ONLY 30911330
AVG DC 0 AVERAGE OF 50 PASSES 30911340
DC RT11A EXIT IF ERROR 30911350
* 30911360
RT11A SLA 16 READ CYL 2-SECT 0 30911370
BSI 2 READ-TB * 30911380
DC 321 * 30911390
DC 0 DATA EXPECTED 30911400
DC RT11B EXIT IF ERROR 30911410
* 30911420
RT11B SLA 16 WRITE CYL 2-SECT 0 30911430
BSI 2 WRITE-TB * WITH /1313 30911440
DC 321 321 WORDS 30911450
DC /1313 DATA 30911460
DC RT11C EXIT IF ERROR 30911470
* 30911480
RT11C MDX 1 -1 DECREMENT LOOP COUNT 30911490
MDX RT11L LOOP ROUTINE 30911500
BSI 2 CNTLE-TB EXIT ROUTINE 30911510
* 30911520
* 30911530
* TEST ROUTINE 12-WRITE 400 30911540
* WORDS ON CYL 2-SECT 0. 30911550
* READ AND CK NUMBER OF 30911560
* WORDS WRITTEN-SAVE IN 30911570

```

```

* SUMMARY. 30911580
* RESTORE SECTOR 1 IF 30911590
* DESTROYED. 30911600
* NOTE-SUMMARY VALUE IS THE 30911610
* AVERAGE OF 50 PASSES. 30911620
* 30911630
* 30911640
RTN12 BSI 2 VERFY-TB SEEK CYL 2 AND VERIFY 30911650
DC 2 * THE SEEK 30911660
DC CNTL EXIT IF ERROR 30911670
RT12Z SLA 16 CLEAR AVERAGE 30911680
STO AVG * 30911690
* 30911700
RT12L LD K370 SET WORD COUNT 30911710
STO 2 COMA-TB IN I/O AREA 30911720
LD 2 H1313-TB GET DATA 30911730
BSI 2 SETV-TB PRESET I/O AREA 30911740
* 30911750
* SLA 16 WRITE CYL 2-SECT 0 30911760
BSI L WRITE&/8000 * WITH NO DSW CK 30911770
DC 400&/8000 NO PRESET OF I/O AREA 30911780
K27 DC 27 CONSTANT 27 30911790
BSI R12CK CK DSW 30911800
* 30911810
BSI 2 STMSG-TB PRINT IF ERROR 30911820
DC /OE0E DSW ERROR BIT NOT SET 30911830
* 30911840
SLA 16 READ SECTOR 0 30911850
BSI L READ&/8000 WITH NO CKING 30911860
DC 370 WORD COUNT 30911870
BSI R12CK CHECK DSW 30911880
BSI 2 STMSG-TB PRINT IF ERROR 30911890
DC /OE0F DSW ERROR BIT NOT SET 30911900
* 30911910
OA2A 0 6700 0172 LDX L3 370 CK NUMBER OF WORDS 30911920
OA2C 1 C700 069E RT12A LD L3 COMA * 30911930
OA2E 0 F2B2 EOR 2 H1313-TB * 30911940
OA2F 1 4C18 OA33 BZ RT12B BRANCH IF FOUND 30911950
OA31 0 73FF MDX 3 -1 DECR WD COUNT 30911960
OA32 0 70F9 MDX RT12A LOOP 30911970
* 30911980
OA33 0 68C7 RT12B STX 3 FSTNO SAVE NUMBER OF WORDS 30911990
OA34 1 7780 OA03 MDX I3 AVG ADD TO TOTAL 30912000
OA36 0 68CC STX 3 AVG SAVE TOTAL 30912010
* 30912020
OA37 0 C2DA LD 2 K1-TB READ SECTOR 1 30912030
OA38 1 4400 8682 BSI L READ&/8000 WITH NO CKING 30912040
OA3A 0 0001 DC 1 WORD COUNT 30912050
OA3B 0 C243 LD 2 COMA&1-TB GET ID READ 30912060
OA3C 0 F205 EOR 2 IDS#B-TB CHECK FOR CORRECT ID 30912070
OA3D 1 4C18 OA49 BZ RT12D BRANCH IF CORRECT 30912080
OA3F 0 C20B LD 2 RTNER-TB GET ERROR SW 30912090
OA40 0 D2BA STO 2 ZSNS-TB DUMMY ENTRY 30912100
OA41 0 401B BSI R12CK CK PRINT SW ONLY 30912110
* 30912120
OA42 0 4232 BSI 2 STMSG-TB PRINT ERROR 30912130
OA43 0 5E11 DC /5E11 WRITE 400 DESTROYED NXT SC 30912140
OA44 0 C2DA LD 2 K1-TB RESTORE SECTOR 1 30912150
OA45 0 423B BSI 2 WRITE-TB * 30912160
OA46 0 0141 DC 321 * 30912170
OA47 0 E5E5 DC /E5E5 * 30912180
OA48 1 OA49 DC RT12D ERROR RETURN 30912190
OA49 0 C0B1 RT12D LD FSTNO GET CURRENT WD CT 30912200
OA4A 0 D2CC STO 2 MOD3-TB SET IN CASE PRINT 30912210
OA4B 0 92E7 S 2 K331-TB SUB 331 30912220
OA4C 1 4C28 OA51 BN RT12G BRANCH IF LESS 30912230
OA4E 0 90D0 S K27 SUB 27 30912240
OA4F 1 4C08 OA55 BNP RT12F BRANCH IF LESS THAN 358 30912250

```

2310 B DISK FUNCTION TEST

0A51 0 C20B RT12G LD 2 RTNER-TB GET ERROR SW 30912260
0A52 0 400A BSI R12CK CHECK PRT SW ONLY 30912270
0A53 0 4232 BSI 2 STMSG-TB PRINT ERROR 30912280
0A54 0 4E10 DC /4E10 SECTOR WORD COUNT NOT WITH 30912290
\* IN LIMITS 331-358 30912300
\* RT12F MDX 1 -1 DECREMENT LOOP COUNT 30912310
RT12L LOOP 30912320
MDX RT12L LD AVG GET TOTAL WORDS 30912330
LD AVX EXCHANGE REGISTERS 30912340
RTE 16 CLEAR A REGISTER 30912350
SLA 16 DIVIDE BY 50 30912360
D NLOOP DIVIDE BY 50 30912370
STO 2 WRLNG-TB SET IN SUMMARY 30912380
BSI 2 CNTLE-TB EXIT ROUTINE 30912390
\* R12CK DC \*-\* ENTRY 30912400
LD 2 ZSNS-TB GET INTERRUPT DSW 30912410
BN RCKX BRANCH IF DN 30912420
LD 2 RTNER-TB GET ERROR SW 30912430
BSI 2 CKPRT-TB CHECK PRINT ALL SW 30912440
MDX RCKX RETURN IF OFF 30912450
LD K27 SET ERROR SW 30912460
STO 2 RTNER-TB \* 30912470
RCKX1 BSC I R12CK EXIT ROUTINE 30912480
RCKX MDX L R12CK,2 INCR RETURN TO NO PRINT 30912490
MDX RCKX1 GO EXIT 30912500
NLOOP DC LPCNT LOOP COUNT 30912510
\* 30912520
\* 30912530
\* 30912540
\* 30912550
\* 30912560
\* 30912570
\* 30912580
\* 30912590
\* 30912600
\* 30912610
\* 30912620
\* 30912630
\* 30912640
\* 30912650
\* 30912660
\* 30912670
\* 30912680
\* 30912690
\* 30912700
\* 30912710
\* 30912720
\* 30912730
\* 30912740
\* 30912750
\* 30912760
\* 30912770
\* 30912780
\* 30912790
\* 30912800
\* 30912810
\* 30912820
\* 30912830
\* 30912840
\* 30912850
\* 30912860
\* 30912870
\* 30912880
\* 30912890
\* 30912900
\* 30912910
\* 30912920
\* 30912930

2310 B DISK FUNCTION TEST

0A8B 0 7101 MDX 1 1 BUMP SAVE POINTER 30912940
0A8C 0 1001 SLA 1 DOUBLE SEEK INCREMENT 30912950
0A8D 0 D2BC STO 2 ZXIO-TB SET IN IOCC 30912960
0A8E 0 1007 SLA 7 TEST FOR 128 30912970
0A8F 1 4C10 0A7F BNN RT13A BRANCH IF LESS THAN 128 30912980
0A91 0 F2B9 EOR 2 H8000-TB ELSE CLEAR BIT 0 30912990
0A92 1 4C30 0A98 BP RT13K BRANCH IF INCR 200 30913000
0A94 0 C2E4 LD 2 K202-TB SET SEEK INCREMENT 30913010
0A95 0 92DB S 2 K2-TB \*\* TO 200 30913020
0A96 0 D2BC STO 2 ZXIO-TB SET FOR EXECUTION 30913030
0A97 0 70E7 MDX RT13A GO SEEK 30913040
\* 30913050
RT13K LD 2 TBDSW-TB GET INTERRUPT DSW 30913060
SLA 4 CHECK HOME BIT 30913070
BN RT13C BRANCH IF OK 30913080
BSI 2 STMSG-TB ELSE PRINT ERROR MESSAGE 30913090
DC /5E18 FORM/MESSAGE ID 30913100
\* 30913110
RT13C LDX L1 COMA SET FROM POINTER 30913120
LDX 3 9 SET LOOP COUNTER 30913130
\* 30913140
RT13D LD 2 KLP13-TB GET LOOP COUNT 30913150
STO 2 CNTB-TB SET FOR LOOPING 30913160
LD 2 TERM-TB SET A-REG TO GREATEST 30913170
SRA 1 \* POSITIVE NUMBER 30913180
SRT 16 MOVE TO Q/ZERO A-REG 30913190
STD 2 MOD3-TB SET INITIAL COUNTS 30913200
SRT 16 CLEAR Q-REG NOW 30913210
STD 2 MOD5-TB SET INITIAL TOTAL 30913220
\* 30913230
RT13E SLT 16 CLEAR Q-REG 30913240
LD X1 0 GET TIME COUNT 30913250
RTE 16 PUT IN POSITION TO ADD 30913260
AD 2 MOD5-TB ADD IN EXISTING TOTAL 30913270
STD 2 MOD5-TB SAVE IN MOD5 30913280
LD 2 MOD3-TB GET MAXIMUM COUNT SINK 30913290
S X1 0 TEST FOR GREATER 30913300
BP RT13F BRANCH IF NOT 30913310
LD X1 0 ELSE SET IN NEW MAXIMUM 30913320
STO 2 MOD3-TB \*\* 30913330
\* 30913340
RT13F LD 2 MOD4-TB GET MINIMUM TIME SINK 30913350
S X1 0 TEST FOR LESS THAN MINIMUM 30913360
BN RT13G BRANCH IF GREATER 30913370
LD X1 0 ELSE SET IN NEW MINIMUM 30913380
STO 2 MOD4-TB \*\* 30913390
\* 30913400
RT13G MDX 1 1 INCREMENT SOURCE POINTER 30913410
MDX L CNTB,-1 DECREMENT LOOP COUNTER 30913420
MDX RT13E LOOP IF NOT THROUGH 30913430
\* 30913440
LDD 2 MOD5-TB GET TOTAL COUNT 30913450
D 2 KLP13-TB CALCULATE AVERAGE SEEK TIM 30913460
STO 2 MOD5-TB SET FOR PRINT 30913470
LD X1 0 GET SEEK INCREMENT 30913480
MDX 1 1 INCREMENT POINTER 30913490
STO 2 MOD6-TB SET FR PRINT 30913500
BSI L STMSG+/8000 PRINT W/O PID-MID.. 30913510
DC /1E19 FORM/MSG ID 30913520
MDX 3 -1 DECREMENT LOOP COUNT 30913530
MDX RT13D LOOP IF NOT THROUGH 30913540
\* 30913550
RT13H BSI I END GO TO MONITOR END 30913560
DC PID ADDRESS OF PRG ID 30913570
\* 30913580
RT13J SLA 16 SET CYL ZERO IN MSG WORD 30913590
STO 2 MOD3-TB \*\* 30913600
BSI 2 STMSG-TB CALL MSG ROUTINE 30913610

2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

OACF 0 4E17 DC /4E17 FORM/MSG ID 30913620
OADO 0 70F8 MDX RT13H CONTINUE PROCESSING ANYWAY 30913630
\* TIMING ROUTINE 30913640
\* 30913650
\* 30913660
OAD1 0 0000 TIMIT DC \*-\* ENTRY POINT 30913670
OAD2 0 C217 LD 2 ADIN2-TB SET INT ADRS IN MONITOR 30913680
OAD3 1 D480 OAFB STD I XEQA+1 \* 30913690
OAD5 0 OABC XIO 2 ZXIO-TB EXECUTE SEEK 30913700
OAD6 0 IOAO SLT 32 CLEAR AQ 30913710
\* 30913720
OAD7 1 8400 0636 TIM2 A L K1 BUMP TIMER 30913730
OAD9 1 4C20 OAD7 BNZ TIM2 BRANCH IF NOT OV AND CARRY 30913740
OADB 1 8C00 066E AD L TIMCN ADD DOUBLE WORD COUNT 30913750
OADD 1 4C18 OAD7 BZ TIM2 IF Q OVERFLOWS/A NONZERO 30913760
\* 30913770
\* 10 MINUTE TIMEOUT 30913780
\* 30913790
OADF 0 70FF TIMOT MDX TIMOT TRAP STOP/NO INTRPT 30913800
\* 30913810
\* INTERRUPT ROUTINE FOR TIMER 30913820
\* 30913830
OAE0 0 0000 INT2 DC \*-\* ENTRY POINT 30913840
OAE1 0 OABA XIO 2 ZSNS-TB SENSE WITH RESET 30913850
OAE2 1 C480 OAE0 LD I INT2 GET INSTRUCTION AT CALL+1 30913860
OAE4 0 1008 SLA 8 GET DISPLACEMENT 30913870
OAE5 0 1808 SRA 8 \* 30913880
OAE6 0 82DA A 2 K1-TB IX2 NOT DISTURBED 30913890
OAE7 0 80F8 A INT2 ADD CALL ADDRESS 30913900
OAE8 0 D001 STD INT2A+1 SET ABS ADDRESS 30913910
OAE9 0 CC00 0000 INT2A LDD L \*-\* GET SAVED AQ REGISTERS 30913920
OAEB 0 18D0 RTE 16 A TO Q/Q TO A 30913930
OAE C 0 AAE8 D 2 THOUS-TB DIVIDE BY 1000 30913940
OAE D 0 DAC6 STD 2 MOD3-TB SAVE AQ 30913950
OAE E 0 A214 M 2 TIMSL-TB MULT BY LOOP TIME 30913960
OAE F 0 DAC6 STD 2 MOD5-TB SAVE AQ 30913970
OAF 0 0 C2CD LD 2 MOD4-TB GET PREVIOUS REMAINDER 30913980
OAF 1 0 A214 M 2 TIMSL-TB MULT BY SINGLE LP TIME 30913990
OAF 2 0 AAE8 D 2 THOUS-TB DIVIDE BY 1000 30914000
OAF 3 0 1890 SRT 16 A TO Q 30914010
OAF 4 0 8ACE AD 2 MOD5-TB ADD DOUBLE SAVED AQ 30914020
OAF 5 0 AAE8 D 2 THOUS-TB DIVIDE BY 1000 30914030
OAF 6 0 D100 STD X1 0 SAVE FOR LATER 30914040
OAF 7 1 4CC0 OAD1 BOSC I TIMIT EXIT TO MY RTN 30914050
\* 30914060
\*-----\* 30914070
\* 30914080
\* SUB-ROUTINE XEQ 30914090
\* 30914100
\*-----\* 30914110
\* 30914120
\* 30914130
\* THIS ROUTINE WILL BUILD AND ISSUE 30914140
\* AN XIO INSTRUCTION. 30914150
\* 30914160
\* IT WILL THEN WAIT FOR AN INTERRUPT, 30914170
\* LOOPING THROUGH THE MONITOR. THE 30914180
\* LOCATION (XCNT) WILL KEEP 30914190
\* THE CURRENT DELAY COUNT FOR A LOST 30914200
\* INTERRUPT. 30914210
\* 30914220
\* IF AN INTERRUPT IS LOST AN ERROR 30914230
\* MESSAGE IS PRINTED AND THE ROUTINE 30914240
\* TERMINATES THE DFT 30914250
\* 30914260
\* \*\*\*\* ROUTINE CALL 30914270
\* BSI 2 XEQ-TB 30914280
\* IOCC MUST BE PRESET AT LOCATION (ZXIO). 30914290

\* 30914300
\*-----\* 30914310
XEQE LD 2 ADINT-TB GET INTRPT RTN ADDRESS 30914320
XEQA STO L \*-\* DEVICE REQUEST ADDRESS 30914330
OAF9 0 C216 LD 2 ZXIO+1-TB GET IOCC 30914340
OAF A 0 C2BD OR 2 DVA-TB COMBINE WITH AREA CODE 30914350
OAF D 0 EA98 STO 2 ZXIO&1-TB SAVE 30914360
OAF E 0 D2BD STO 2 INTSW-TB SET INTRPT SWITCH 30914370
OAF F 0 D29A AND 2 DSKMD-TB TEST FOR READ IOCC 30914380
OB0 0 E2B5 EOR 2 DSKMD-TB \*\* 30914390
OB1 0 F2B5 BZ XEQB NO RDY CK IF READ OPERATIO 30914400
\* 30914410
\* CKRD1 XIO 2 SNXIO-TB SENSE DSW 30914420
\* 30914430
\* 30914440
\* 30914450
\* 30914460
\* 30914470
\* DISK IS BUSY AND/OR NOT READY 30914480
\* 30914490
\* 30914500
\* 30914510
\* 30914520
\* 30914530
\* 30914540
\* 30914550
\* 30914560
\* 30914570
\* 30914580
\* 30914590
\* 30914600
\* 30914610
\* 30914620
\* 30914630
\* 30914640
\* 30914650
\* 30914660
\* 30914670
\* 30914680
\* 30914690
\* 30914700
\* 30914710
\* 30914720
\* 30914730
\* 30914740
\* 30914750
\* 30914760
\* 30914770
\* 30914780
\* 30914790
\* 30914800
\* 30914810
\* 30914820
\* 30914830
\* 30914840
\* 30914850
\* 30914860
\* 30914870
\* 30914880
\* 30914890
\* 30914900
\* 30914910
\* 30914920
\* 30914930
\* 30914940
\* 30914950
\* 30914960
\* 30914970
\* 30914980
\* 30914990
\* 30915000



2310 B DISK FUNCTION TEST

OB55 0 73A7	MDX	3	-89	TEST FOR CYLINDERS 0-89	30915660
OB56 0 7001	B		TSTCC	90-...	30915670
OB57 0 7006	B		SEEK	0-89	30915680
	*				30915690
OB58 0 73EB	TSTCC	MDX	3 -21	TEST FOR 90-110	30915700
OB59 0 7002	B		TSTCD	111-...	30915710
OB5A 0 4C00 0000	TSTCF	B	L *-*	90-110 TAKE ADRS ERR EXIT	30915720
	*				30915730
OB5C 0 73A4	TSTCD	MDX	3 -92	TEST FOR CYLS 111-202	30915740
OB5D 0 70FC	B		TSTCF	GREATER THAN 202	30915750
	*		FALL THRU TO SEEK	DESIRED CYLINDER	30915760
	*				30915770
	*				30915780
	*				30915790
	*				30915800
	*				30915810
OB5E 0 C2B0	SEEK	LD	2 XSKBK-TB	GET SEEK TOWARD HOME IOCC	30915820
OB5F 0 1890	SRT		16	PUT IN Q REGISTER	30915830
OB60 0 D242	STO	2	COMA-TB	CLEAR COMA..DON'T XEQ IO	30915840
OB61 0 C2D3	LD	2	NCYL#-TB	GET DESIRED CYLINDER	30915850
OB62 0 92D2	S	2	PCYL#-TB	CREATE DISPLACEMENT	30915860
OB63 1 4C18 OB86	BZ		VERFB	DON'T SEEK IF ZERO	30915870
OB65 1 4C28 OB6A	BN		SEEKB	IF NEGATIVE-BRANCH	30915880
OB67 0 1883	SRT		3	ELSE CLEAR SEEK OUT BIT	30915890
OB68 0 1083	SLT		3	* IN IOCC	30915900
OB69 0 7002	B		SEEKD	GO EXECUTE SEEK	30915910
	*				30915920
OB6A 0 F28B	SEEKB	EOR	2 TERM-TB	FORM TWO'S COMPLEMENT OF	30915930
OB6B 0 82DA	A		2 K1-TB	* SEEK DISPLACEMENT	30915940
	*				30915950
OB6C 0 DABC	SEEKD	STD	2 ZXIO-TB	STORE FOR XEQ ROUTINE	30915960
OB6D 0 423E	BSI	2	XEQ-TB	GO TO TEST FOR READY ONLY	30915970
OB6E 0	SEEKE	EQU	*	RETURN FROM XEQ	30915980
OB6E 0 C2DF	LD	2	K8-TB	SET READ RETRY COUNTER	30915990
OB6F 0 D208	STO	2	RTCNT-TB	* FOR SEEK VERIFIES	30916000
OB70 0 6301	LDX	3	SKCNT-SUMRY	COUNT SEEK	30916010
OB71 0 4223	BSI	2	COUNT-TB	COUNT	30916020
OB72 0 0ABC	XIO	2	ZXIO-TB	EXECUTE SEEK IOCC	30916030
OB73 0 C2E9	LD	2	K10TH-TB	SET COUNT IN ACC	30916040
OB74 0 D207	STO	2	ZCNT-TB	SET PRIMARY COUNTER	30916050
OB75 0 422F	SEEKF	BSI	2 STMLS-TB	GO TO MONITOR	30916060
OB76 0 0ABE	XIO	2	SNXIO-TB	SENSE DSW	30916070
OB77 0 1003	SLA		3	SHIFT DISK BUSY BIT	30916080
OB78 1 4C10 OB80	BNN		SEEKG	BRANCH IF NOT BUSY	30916090
OB7A 1 74FF 0663	MDX	L	ZCNT,-1	DECREMENT COUNT	30916100
OB7C 0 70F8	MDX		SEEKF	LOOP	30916110
OB7D 1 7412 069A	MDX	L	XEQ,SEEKG-S	SEEKE SET RETURN TO SEEKG	30916120
OB7F 0 709E	MDX		XEQC	GO WAIT FOR INTERRUPT	30916130
	*				30916140
OB80 0 0ABE	SEEKG	XIO	2 SNXIO-TB	SENSE DISK DSW	30916150
OB81 0 D2CB	STO	2	TBDSW-TB	SET IN CASE PRINT	30916160
OB82 1 4C10 OB86	BNN		VERFB	BRANCH IF NO ERROR	30916170
OB84 0 4232	BSI	2	STMSG-TB	ELSE PRINT E005 MESSAGE	30916180
OB85 0 0E05	DC		/0E05		30916190
	*				30916200
OB86 0 0ABE	VERFB	XIO	2 SNXIO-TB	SENSE DISK DSW	30916210
OB87 0 61F8	LDX		1 -8	SET LOOP COUNTER	30916220
	*				30916230
OB88 0 E2DE	VERFC	AND	2 K7-TB	SAVE BITS 13-15 FOR SECTOR	30916240
OB89 0 D0D1	STO		TSTCF+1	* WANTED	30916250
OB8A 1 4400 8682	BSI	L	READ+/8000	READ DESIRED SECTOR	30916260
OB8C 0 0001	DC		1	ONE WORD	30916270
OB8D 0 C243	LD	2	COMA+1-TB	GET SECTOR ID READ	30916280
OB8E 1 D500 0814	STO	L1	BDCYL+8	SAVE IN TABLE	30916290
OB90 0 COCA	LD		TSTCF+1	GET SECTOR POINTER	30916300
OB91 0 82DA	A		2 K1-TB	INCREMENT BY ONE	30916310
OB92 0 7101	MDX		1 1	BUMP LOOP COUNTER	30916320
OB93 0 70F4	MDX		VERFC	LOOP IF NOT THROUGH	30916330

2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

OB94 0 DOC6 STD TSTCF+1 ELSE SAVE FIRST SECTOR 30916340
OB95 0 61F8 LDX 1 -8 SET LOOP COUNTER 30916350
\* VERFD LD TSTCF+1 GET SECTOR POINTER 30916360
OB96 0 COC4 VERFD LD TSTCF+1 GET SECTOR POINTER 30916370
OB97 1 F500 0814 EOR L1 BDCYL+8 TEST CORRECT SECTOR 30916380
OB99 0 E2DE AND 2 K7-TB SAVE BITS 13-15 30916390
OB9A 1 4C20 08BD BNZ VERFF ERROR MESSAGE IF ERROR 30916400
OB9C 1 7401 085B MDX L TSTCF+1,1 BUMP SECTOR POINTER 30916410
OB9E 0 7101 MDX 1 1 BUMP LOOP COUNTER 30916420
OB9F 0 70F6 MDX VERFD LOOP IF NOT THRU 30916430
\* LDX 1 -7 SET LOOP COUNTER 30916440
OBA0 0 61F9 VERFE LD L BDCYL GET FIRST SECTOR ID READ 30916450
OBA1 1 C400 080C EOR L1 BDCYL+8 COMPARE WITH NEXT ONE 30916460
OBA3 1 F500 0814 SRA 3 \* FOR SAME CYLINDER NUM. 30916480
OBA5 0 1803 BNZ VERFF BRANCH CH IF NOT SAME 30916490
OBA6 1 4C20 08BD MDX 1 1 ELSE BUMP LOOP COUNTER 30916500
OBA8 0 7101 MDX VERFE LOOP IF NOT THROUGH 30916510
OBA9 0 70F7 \* 30916520
\* 30916530
\* 30916540
OBA A 0 CAD2 LDD 2 PCYL#-TB SET PRINT MESSAGE 30916550
OBA B 0 DACC STD 2 MOD3-TB \* 30916560
OBA C 0 C243 LD 2 COMA&1-TB CK FOR EXPECTED CYLINDER 30916570
OBA D 0 1883 SRT 3 \* 30916580
OBA E 0 D2CE STD 2 MOD5-TB SAVE FOR PRINT 30916590
OBA F 0 D2D2 STO 2 PCYL#-TB SET AS PRESENT CYLINDER 30916600
OB B 0 F2D3 EOR 2 NCYL#-TB IS IT EXPECTED 30916610
OB B 1 4C18 0BD2 BZ VERFI BRANCH IF YES 30916620
\* 30916630
\* 30916640
\* 30916650
OB B 3 0 4232 BSI 2 STMSG-TB PRINT MESSAGE 30916660
OB B 4 0 2E09 DC /2E09 SEEK ERROR 30916670
OB B 5 0 6302 LDX 3 SFTSK-SUMRY COUNT SOFT SEEK ERROR 30916680
OB B 6 0 4223 BSI 2 COUNT-TB \* 30916690
OB B 7 1 74FF 0665 MDX L WRRTY,-1 DECR RETRY COUNTER 30916700
OB B 9 0 70A4 B SEEK RETRY THE SEEK 30916710
OB B A 0 6303 LDX 3 HRDSK-SUMRY COUNT HARD SEEK ERROR 30916720
OB B B 0 4223 BSI 2 COUNT-TB \* 30916730
OB B C 0 4235 BSI 2 TEXTIT-TB TERMINATE DFT 30916740
\* 30916750
OB B D 0 61FB VERFF LDX 1 -5 SET LOOP COUNTER 30916760
OB B E 1 CD00 0811 VERFG LDD L1 BDCYL&5 SET HARD ERROR MSG 30916770
OB C 0 0 DACC STD 2 MOD3-TB \* 30916780
OB C 1 1 CD00 0813 LDD L1 BDCYL&7 \* 30916790
OB C 3 0 DACE STD 2 MOD5-TB \* 30916800
OB C 4 0 7104 MDX 1 4 DECR CTR 30916810
OB C 5 0 7009 MDX VERFH BRANCH 30916820
OB C 6 1 4400 868E BSI L STMSG&/8000 PRINT SECOND LINE 30916830
OB C 8 0 8E00 DC /8E00 MSG ID NOT PRINTED 30916840
OB C 9 1 74FF 0664 MDX L RTCNT,-1 BUMP ERROR RETRY COUNTER 30916850
OB C B 0 70BA MDX VERFB LOOP IF NOT MAXIMUM 30916860
OB C C 0 4232 BSI 2 STMSG-TB PRINT MESSAGE 30916870
OB C D 0 5A01 DC /5A01 DSK NOT INIT 30916880
OB C E 0 4235 BSI 2 TEXTIT-TB TERMINATE DFT 30916890
OB C F 0 4232 VERFH BSI 2 STMSG-TB PRINT LINE 1 30916900
OB D 0 0 8E14 DC /8E14 HARD READ ERROR 30916910
OB D 1 0 70EC B VERFG BRANCH 30916920
\* 30916930
\* 30916940
\* 30916950
\* 30916960
OB D 2 0 CA08 VERFI LDD 2 RTCNT-TB CK PREVIOUS ERRORS 30916970
OB D 3 0 9ADF SD 2 K8-TB \* 30916980
OB D 4 1 4C20 0BD9 BNZ VERFJ BRANCH IF ERROR OCCURRED 30916990
OB D 6 0 1090 SLT 16 GET Q REG 30917000
OB D 7 1 4C18 0BDD BZ VERFX GO EXIT RTN 30917010

OB D 9 0 421A VERFJ BSI 2 CKLK-TB CHECK LOCK ON ERROR 30917020
OB D A 1 085E DC SEEK RETURN IF ON 30917030
OB D B 0 4232 BSI 2 STMSG-TB PRINT RECOVERED ERROR 30917040
OB D C 0 5A06 DC /5A06 MESSAGE ID 30917050
OB D D 0 6500 0000 VERFX LDX L1 \*- \* RESTORE INDEX 1 30917060
\* 30917070
\* 30917080
\* 30917090
\* 30917100
OB D F 0 C2D2 LD 2 PCYL#-TB GET PRESENT CYLINDER 30917100
OB E 0 0 D2CC STO 2 MOD3-TB \* 30917110
OB E 1 0 4820 SKP 2 SKIP IF CYL ZERO 30917120
OB E 2 0 C28B LD 2 TERM-TB MAKE HOME BIT ZERO 30917130
OB E 3 0 F2BA EOR 2 ZSNS-TB \* 30917140
OB E 4 0 1004 SLA 4 CHECK HOME BIT 30917150
OB E 5 1 4CA8 0694 BN I VERFY EXIT ROUTINE 30917160
OB E 7 0 4232 BSI 2 STMSG-TB PRINT HOME BIT INCORRECT 30917170
OB E 8 0 4E0D DC /4E0D FORM/MESSAGE ID 30917180
OB E 9 1 4C80 0694 B I VERFY EXIT ROUTINE 30917190
\* 30917200
\* 30917210
\* 30917220
\* 30917230
\* 30917240
\* 30917250
\* 30917260
\* 30917270
\* 30917280
\* 30917290
\* 30917300
\* 30917310
\* 30917320
\* 30917330
\* 30917340
\* 30917350
\* 30917360
\* 30917370
\* 30917380
\* 30917390
\* 30917400
\* 30917410
\* 30917420
\* 30917430
\* 30917440
\* 30917450
\* 30917460
\* 30917470
\* 30917480
\* 30917490
\* 30917500
\* 30917510
\* 30917520
\* 30917530
\* 30917540
\* 30917550
\* 30917560
\* 30917570
\* 30917580
\* 30917590
\* 30917600
\* 30917610
\* 30917620
\* 30917630
OB E B 0 EAB5 RDEN OR 2 DSKMD-TB COMBINE WITH READ 30917640
OB E C 0 1890 SRT 16 A TO Q 30917650
OB E D 0 D2D8 STO 2 RDDSW-TB CLEAR DSW ERROR COUNTER 30917660
OB E E 0 D2D9 STO 2 RDCMP-TB CLEAR CMP ERROR COUNTER 30917670
OB E F 0 C218 LD 2 ADCMA-TB 30917680
OB F 0 0 DABC STD 2 ZXIO-TB SET FOR XEQ CALL 30917690



2310 B DISK FUNCTION TEST

OBF1 0 108D	SLT	13	SECTOR TO 0-2 Q REG.	30917700
OBF2 0 C2D3	LD	2 NCYL#-TB	GET CYL. # WHERE DISK S/B	30917710
	*		* AFTER LAST SEEK	30917720
OBF3 0 1083	SLT	3	COMBINE FOR COMPARE	30917730
OBF4 0 D205	STO	2 IDS#B-TB	PRESENT SECTOR ID	30917740
	*			30917750
	*			30917760
OBF5 1 6780 0682	LDX	13 READ	GET CALLING ADRS+1	30917770
OBF7 0 C3FF	LD	3 -1	SET NOCK SW	30917780
OBF8 0 D2FE	STO	2 NOCK-TB	*	30917790
OBF9 0 C300	LD	3 0	GET WORD COUNT	30917800
OBFA 0 D2FD	STO	2 RNDCK-TB	IF NEG. USE RANDOM NUMBERS	30917810
OBFB 0 1001	SLA	1	CLEAR BIT 0	30917820
OBFC 0 1801	SRA	1	*	30917830
OBFD 0 D242	STO	2 COMA-TB	WORD COUNT	30917840
OBFE 0 D2FF	STO	2 LNGTH-TB	* FOR COMPARE RTN	30917850
OBFF 0 C301	LD	3 1	GET NUMBER	30917860
OC00 0 D20C	STO	2 CMPTM-TB	* FOR COMPARE RTN	30917870
OC01 0 C302	LD	3 2	GET CMP ERROR ADRS	30917880
OC02 0 D029	STO	RDCPX+1	SET FOR CMP ERROR EXIT	30917890
OC03 1 7401 0682	MDX	L READ,1	INCR FOR NO-CHECK RETURN	30917900
OC05 0 6304	LDX	3 RDCNT-SUMRY	POINTER FOR SUMMARY	30917910
OC06 0 4223	BSI	2 COUNT-TB	INCR READ COUNT	30917920
	*			30917930
OC07 0 CADF	READA	LD 2 K8-TB	SET RETRY COUNTER	30917940
OC08 0 DAD6	STD	2 RTRYA-TB	* TO 8	30917950
	*			30917960
	*		PRESET I/O AREA TO /FFFF	30917970
	*			30917980
OC09 0 C28B	READB	LD 2 TERM-TB	/FFFF	30917990
OC0A 0 422C	BSI	2 SETV-TB	CALL PRESET ROUTINE	30918000
	*			30918010
	*		ISSUE A READ OPERATION	30918020
	*		THRU THE XEQ SUB-ROUTINE	30918030
	*			30918040
OC0B 0 423E	BSI	2 XEQ-TB	ISSUE READ COMMAND	30918050
	*			30918060
	*		IF NOCK NEG DONT CHECK FOR	30918070
	*		DSW ERRORS, BUT EXIT AS IF NORMAL READ.	30918080
	*			30918090
OC0C 0 C2FE	LD	2 NOCK-TB	GET NO CHECK SWITCH	30918100
OC0D 1 4C28 OC20	BN	RDNCK	EXIT IF NEGATIVE	30918110
	*			30918120
	*		CHECK DSW FOR ERRORS	30918130
	*			30918140
OC0F 0 C2BA	LD	2 ZSNS-TB	INTRPT DSW	30918150
OC10 1 4C28 OC2D	BN	RDER2	BRANCH ON ERROR	30918160
OC12 0 4038	BSI	CMP	CALL COMPARE RTN	30918170
OC13 0 700E	B	RDER1	COMPARE ERROR RETURN	30918180
OC14 0 C2D8	LD	2 RDDSW-TB	GET ERROR CONT	30918190
OC15 0 82D9	A	2 RDCMP-TB	ADD COMPARE ERRORS	30918200
OC16 1 4C18 OC1E	BZ	READX	BRANCH IF NO ERRORS	30918210
OC18 0 421A	BSI	2 CKLK-TB	CHECK LOCK ON ERROR SW	30918220
OC19 1 0C07	DC	READA	IF SET	30918230
OC1A 0 6305	LDX	3 SFTRD-SUMRY	POINTER	30918240
OC1B 0 4223	BSI	2 COUNT-TB	SOFT READ ERROR	30918250
OC1C 0 4232	BSI	2 STMSG-TB	PRINT MESSAGE	30918260
OC1D 0 5A03	DC	/5A03	SOFT READ ERROR	30918270
	*			30918280
OC1E 1 7402 0682	READX	MDX L READ,2	INCR FOR NORMAL RETURN	30918290
	*			30918300
OC20 1 4C80 0682	RDNCK	B I READ	RETURN	30918310
	*			30918320
OC22 1 7401 0635	RDER1	MDX L RDCMP,1	COUNT COMPARE ERROR	30918330
OC24 1 74FF 0632	MDX	L RTRYA,-1	DECR RETRY COUNTER	30918340
OC26 0 70E2	B	READB	LOOP	30918350
OC27 0 4232	BSI	2 STMSG-TB	PRINT MESSAGE	30918360
OC28 0 5A04	DC	/5A04	HARD CMP ERROR	30918370

2310 B DISK FUNCTION TEST

OC29 0 6306	LDX	3	HRDRD-SUMRY	POINTER TO HRD RD ERR CNT	30918380	
OC2A 0 4223	BSI	2	COUNT-TB	COUNT ERROR	30918390	
OC2B 0 4C00 0000	RDCPX	B	L	*--*	COMPARE ERROR EXIT	30918400
	*					30918410
OC2D 0 C2D8	RDER2	LD	2	RDDSW-TB	GET DSW ERROR COUNT	30918420
OC2E 1 7401 0634	MDX	L	RDDSW,1	BUMP COUNT	30918430	
OC30 0 421D	BSI	2	CKPRT-TB	CK PRINT-ALL-ERRORS OPTION	30918440	
OC31 0 7002	B		RDR2B	BR AROUND PRINT	30918450	
OC32 0 4232	RDR2A	BSI	2	STMSG-TB	PRINT MESSAGE	30918460
OC33 0 0E06	DC		/0E06	HARD DSW ERROR	30918470	
OC34 0 4016	RDR2B	BSI		CMP	CALL COMPARE RTN	30918480
OC35 0 700F	B		RDER4	COMPARE ERROR RETURN	30918490	
OC36 1 74FF 0632	MDX	L	RTRYA,-1	DECR RETRY COUNTER	30918500	
OC38 0 70D0	B		READB	LOOP	30918510	
	*					30918520
OC39 0 CABC	RDER3	LDD	2	ZXIO-TB	GET LAST IOCC ISSUED	30918530
OC3A 0 DACC	STD	2	MOD3-TB	STORE IN CASE PRINT	30918540	
OC3B 0 421A	BSI	2	CKLK-TB	CHECK LOCK-ON-ERROR	30918550	
OC3C 1 0C07	DC		READA	IF SET	30918560	
OC3D 0 4232	BSI	2	STMSG-TB	PRINT MESSAGE	30918570	
OC3E 0 0E07	DC		/0E07	DSW ERROR	30918580	
OC3F 0 6306	LDX	3	HRDRD-SUMRY	POINTER	30918590	
OC40 0 4223	BSI	2	COUNT-TB	HARD READ ERROR	30918600	
OC41 0 C2D6	LD	2	RTRYA-TB	GET DSW RETRY COUNTER	30918610	
OC42 1 4C18 0C1E	BZ		READX	NORMAL EXIT/NO CMP ERRORS	30918620	
	*			* ON LAST READ	30918630	
OC44 0 70E6	B		RDCPX	TAKE CMP ERROR EXIT	30918640	
	*					30918650
OC45 1 7401 0635	RDER4	MDX	L	RDCMP,1	INCR CMP ERROR COUNT	30918660
OC47 1 74FF 0633	MDX	L	RTRYB,-1	DECR CMP ERROR COUNT	30918670	
OC49 0 70BF	B		READB	LOOP	30918680	
OC4A 0 70EE	B		RDER3	GO EXIT	30918690	
	*					30918700
	*					30918710
	*					30918720
	*					30918730
	*					30918740
	*					30918750
	*					30918760
	*					30918770
	*					30918780
	*					30918790
	*					30918800
	*					30918810
	*					30918820
	*					30918830
	*					30918840
	*					30918850
	*					30918860
	*					30918870
	*					30918880
	*					30918890
	*					30918900
	*					30918910
	*					30918920
	*					30918930
	*					30918940
	*					30918950
	*					30918960
	*					30918970
	*					30918980
	*					30918990
	*					30919000
	*					30919010
	*					30919020
	*					30919030
	*					30919040
	*					30919050
	*					30919060
	*					30919070
	*					30919080
	*					30919090
	*					30919100
	*					30919110
	*					30919120
	*					30919130
	*					30919140
	*					30919150
	*					30919160
	*					30919170
	*					30919180
	*					30919190
	*					30919200
	*					30919210
	*					30919220
	*					30919230
	*					30919240
	*					30919250
	*					30919260
	*					30919270
	*					30919280
	*					30919290
	*					30919300
	*					30919310
	*					30919320
	*					30919330
	*					30919340
	*					30919350
	*					30919360
	*					30919370
	*					30919380
	*					30919390
	*					30919400
	*					30919410
	*					30919420
	*					30919430
	*					30919440
	*					30919450
	*					30919460
	*					30919470
	*					30919480
	*					30919490
	*					30919500
	*					30919510
	*					30919520
	*					30919530
	*					30919540
	*					30919550
	*					30919560
	*					30919570
	*					30919580
	*					30919590
	*					30919600
	*					30919610
	*					30919620
	*					30919630
	*					30919640
	*					30919650
	*					30919660
	*					30919670
	*					30919680
	*					30919690
	*					30919700
	*					30919710
	*					30919720
	*					30919730
	*					30919740
	*					30919750
	*					30919760
	*					30919770
	*					30919780
	*					30919790
	*					30919800
	*					30919810
	*					30919820
	*					30919830
	*					30919840
	*					30919850
	*					30919860
	*					30919870
	*					30919880
	*					30919890
	*					30919900
	*					30919910
	*					30919920
	*					30919930
	*					30919940
	*					30919950
	*					30919960
	*					30919970
	*					30919980
	*					30919990
	*					30920000

2310 B DISK FUNCTION TEST

OC53 0 CA05 LDD 2 IDS#B-TB GET ID EXPECTED 30919060
OC54 0 DA02 STD 2 S#B-1-TB SET 30919070
OC55 0 402D BSI CMPB GO MAKE COMPARISION 30919080
OC56 0 C20C LD 2 CMPM-TB SET NUMBER 30919090
OC57 0 D203 STO 2 S#B-TB SET 30919100
OC58 0 7004 B CMP2 CHECK FOR FINISHED 30919110
OC59 0 C2FD CMP1 LD 2 RNDCK-TB GET RANDOM INDICATOR 30919120
OC5A 1 4C28 OC85 BN CMRND BRANCH IF ON 30919130
OC5C 0 705B B CMRNI GO TEST FOR BAD DATA 30919140
OC5D 1 7401 065C CMP2 MDX L INDEX,1 INCR INDEX 30919150
OC5F 0 C200 LD 2 INDEX-TB GET CONTENTS INDEX 30919160
OC60 0 92FF S 2 LNGTH-TB CK FOR COMPLETE 30919170
OC61 1 4C08 OC59 BNP CMP1 BRANCH IF NOT FINISHED 30919180
OC63 0 C201 LD 2 ERCT-TB GET ERROR CNT 30919190
OC64 0 D2CC STO 2 MOD3-TB STORE FOR PRINT 30919200
OC65 1 4C18 OC6D BZ CMP3 BRANCH IF ZERO 30919210
OC67 0 C2D9 LD 2 RDCMP-TB GET CMP ERROR COUNT 30919220
OC68 0 421D BSI 2 CKPRT-TB CK PRINT-ALL-ERRORS 30919230
OC69 0 7003 B CMP3 BRANCH IF NOT SET 30919240
OC6A 1 4400 868E BSI L STMSG&/8000 PRINT NO. ERRORS 30919250
OC6C 0 4E15 DC /4E15 # OF COMPARE ERRORS 30919260
OC6D 1 6780 065C \* CMP3 LDX I3 INDEX IX EQUAL WD CT 30919270
OC6F 1 C700 069E LD L3 COMA GET REC LNGTH &1 30919280
OC71 1 8700 069F A L3 COMA&1 ADD REC LNGTH &2 30919290
OC73 0 82DB A 2 K2-TB ADD 2 30919310
OC74 1 4C18 OC7C BZ CMPX BRANCH IF AS EXPECTED 30919320
OC76 0 C2D9 LD 2 RDCMP-TB GET CMP ERROR COUNT 30919330
OC77 0 421D BSI 2 CKPRT-TB CK PRINT-ALL-ERRORS OPTION 30919340
OC78 0 7002 B CMP4 BRANCH IF NOT SET 30919350
OC79 0 4232 BSI 2 STMSG-TB PRINT OVERREAD 30919360
OC7A 0 7E0B DC /7E0B 30919370
OC7B 0 C2DA CMP4 LD 2 K1-TB MAKE SURE A REG. NON-ZERO 30919380
OC7C 0 8201 CMPX A 2 ERCT-TB CK IF ANY ERRORS 30919390
OC7D 1 4C20 OC81 BNZ CMPEX BRANCH IF YES 30919400
OC7F 1 7401 OC4B MDX L CMP,1 INCR RETURN 30919410
OC81 1 4C80 OC4B CMPEX BSC I CMP EXIT ROUTINE 30919420
OC83 0 0000 \* CMPB DC \*-\* ENTRY 30919430
OC84 1 6780 065C LDX I3 INDEX GET INDEX 30919440
OC86 1 F700 069E EOR L3 COMA COMPARE WITH S/B 30919450
OC88 1 4C98 OC83 BZ I CMPB EXIT IF EQUAL 30919470
OC8A 1 7700 069E MDX L3 COMA INCR IX BY ADRS OF COMA 30919480
OC8C 0 6B37 STX 3 PNTC+1 SET FOR POINTER 30919490
OC8D 0 C201 LD 2 ERCT-TB GET ERROR CNT MINUS 1 30919500
OC8E 1 7401 065D MDX L ERCT,1 INCREMENT ERROR COUNT 30919510
OC90 1 4C18 OC9F BZ CMPB2 BRANCH IF FIRST CMP ERROR 30919520
\* CHECK PRINT ALL ERRORS OPTION 30919530
\* BSI 2 CKPRT-TB CK PRINT-ALL-ERRORS OPTION 30919550
OC92 0 421D B CMPB6 BRANCH IF NOT SET 30919560
OC93 0 701F \* PRINT ADDITIONAL ERRORS 30919570
\* 30919580
\* 30919590
\* 30919600
OC94 0 C206 CMPB1 LD 2 LPRNT-TB TEST TO SEE IF BAD WORDS 30919610
OC95 0 82DA A 2 K1-TB \* AND PREVIOUS WORD HAVE 30919620
OC96 0 9200 S 2 INDEX-TB \* BEEN PRINTED YET 30919630
OC97 1 4C18 OCAB BZ CMPB4 BRANCH TO PRINT BAD WORD 30919640
OC99 1 4C28 OCA9 BN CMPB3 BRANCH/PRINT PREVIOUS WORD 30919650
OC9B 0 92DA S 2 K1-TB TEST FOR NEXT WORD PRINTED 30919660
OC9C 1 4CA0 OC83 BNZ I CMPB EXIT IF LAST WORD 30919670
OC9E 0 700E B CMPB5 BAD WORD+1 NOT PRINTED YET 30919680
\* 30919690
OC9F 0 C2D9 CMPB2 LD 2 RDCMP-TB GET CMP ERROR COUNT 30919700
OCA0 0 421D BSI 2 CKPRT-TB CK PRINT-ALL-ERRORS OPTION 30919710
OCA1 0 7002 B CMP2A BRANCH IF NOT SET 30919720
OCA2 0 4232 BSI 2 STMSG-TB PRINT MESSAGE 30919730

2310 B DISK FUNCTION TEST

OCA3 0 7E0A DC /7E0A MESSAGE ID 30919740
OCA4 0 C200 CMP2A LD 2 INDEX-TB GET POINTER 30919750
OCA5 0 92DA S 2 K1-TB \* TO ONE (1) 30919760
OCA6 1 4C20 OC94 BNZ CMPB1 BRANCH IF NOT EQUAL 30919770
OCA8 0 7002 B CMPB4 PRINT BAD WORD 30919780
\* 30919790
\* PRINT WORD PREVIOUS TO BAD WORD 30919800
\* 30919810
CMPB3 LDX 3 -1 INDEX-1 30919820
BSI PNTB PRINT 30919830
\* 30919840
\* PRINT BAD WORD 30919850
\* 30919860
CMPB4 LDX 3 0 SET POINTER 30919870
BSI PNTB PRINT 30919880
\* 30919890
\* TEST IF LAST WORD 30919900
\* 30919910
CMPB5 LD 2 LNGTH-TB GET RECORD LNGTH 30919920
S 2 INDEX-TB TEST FOR END 30919930
BZ I CMPB EXIT IF LAST WORD WAS JUST 30919940
\* PRINTED 30919950
\* 30919960
\* PRINT WORD FOLLOWING BAD WORD 30919970
\* 30919980
LDX 3 1 SET POINTER 30919990
BSI PNTB PRINT FORM 2 30920000
CMPB6 B I CMPB RETURN 30920010
\* 30920020
CMRND LD 2 S#B-TB GET CURRENT NUMBER 30920030
BSI 2 RNDOM-TB GET NEXT NUMBER 30920040
STO 2 S#B&1-TB SET 30920050
CMRNI LD 2 S#B-TB CK CURRENT NUMBER 30920060
BSI CMPB \* 30920070
LD 2 S#B-TB SET PREVIOUS NUMBER 30920080
STO 2 S#B-1-TB \* 30920090
LD 2 S#B&1-TB SET CURRENT NUMBER 30920100
STO 2 S#B-TB \* 30920110
B CMP2 CK FOR COMPLETE 30920120
\* 30920130
PNTB DC \*-\* ENTRY 30920140
LD 2 RDCMP-TB GET CMP ERROR COUNT 30920150
BSI 2 CKPRT-TB CK PRINT-ALL-ERRORS OPTION 30920160
B PNTBX DON'T PRINT IF NOT SET 30920170
PNTC LD L3 \*-\* GET WORD READ 30920180
STO 2 MOD5-TB SET FOR PRINT 30920190
LD L3 S#B GET WORD EXPECTED 30920200
STO 2 MOD4-TB SET FOR PRINT 30920210
MOX I3 INDEX ADJUST INDEX 30920220
NOP IN CASE SKIP OCCURS 30920230
STX L3 MOD3 SET FOR PRINT 30920240
STX L3 LPRNT SET AS LAST WORD PRINTED 30920250
UCD0 1 4400 868E BSI L STMSG+/8000 PRINT W/O MID-PID--- 30920260
UCD2 0 6E16 DC /6E16 S/B WAS 30920270
UCD3 1 4C80 OCBF PNTBX B I PNTB EXIT 30920280
\* 30920290
\* 30920300
\* 30920310
\* 30920320
\* WRITE ROUTINE 30920330
\* 30920340
\* 30920350
\* 30920360
\* 30920370
\* THE ROUTINE WILL INSERT 30920380
\* THE WORD COUNT AND THE DISK ADRS 30920390
\* AS THE FIRST TWO WORDS OF THE 30920400
\* I/O AREA AND ISSUE THE WRITE 30920410

```

*        THRU THE XEQ ROUTINE.                30920420
*        30920430
*        AFTER INTRPT THE DSW IS CHECKED      30920440
*        FOR ANY ERRORS.                      30920450
*        30920460
*        IF ERRORS ARE FOUND THE ROUTINE      30920470
*        WILL RETRY UP TO EIGHT TIMES.        30920480
*        30920490
*        LOCK ON ERROR OPTION IS              30920500
*        PROVIDED AND WILL LOCK THE           30920510
*        ROUTINE IN WRITE ON EITHER AN        30920520
*        INTERMITTENT OR SOLID ERROR.         30920530
*        30920540
*        THE LOCK WILL REMAIN IN EFFECT        30920550
*        UNTIL THE SWITCH IS CLEARED.         30920560
*        30920570
**** ROUTINE CALL
*
*   A REG. _ SECTOR COUNT                    30920580
*   BSI  2 WRITE-TB                          30920590
*   DC   WORD COUNT                          30920600
*   BIT 0 EQUAL 1 MEANS                      30920610
*   DON'T PRESET I/O AREA.                   30920620
*   DC   NUMBER USED AS PRESET                30920630
*   DC   ADRS TO RETURN TO                   30920640
*   IF 8 RETRYS FAIL                         30920650
*                                               30920660
*                                               30920670
****ALTERNATE CALL USED
*
*   A REG _ SECTOR COUNT                    30920680
*   BSI  L WRITE+/8000                       30920690
*   DC   WORD COUNT                          30920700
*   DC   NUMBER USED AS PRESET                30920710
*                                               30920720
*                                               30920730
*                                               30920740
*   THE ALTERNATE CALL WILL CAUSE THE ROUTINE TO
*   PRESET THE I/O AREA, WRITE AND RETURN WITH-
*   OUT CHECKING THE DSW.                   30920750
*                                               30920760
*                                               30920770
*                                               30920780
*                                               30920790
*                                               30920800
-----
* WRTEN OR  2 WRMOD-TB  COMBINE WITH WRT FNC  30920810
*          SRT   16     A TO Q                30920820
*          LD   2 ADCMA-TB  ADRS OF COMA       30920830
*          STO  2 ZXIO-TB  SET FOR XEQ        30920840
*          SLT   13     POSITION SECTOR COUNT   30920850
*          LD   2 PCYL#-TB  GET PRESENT CYLINDER
*          SLT   3     CREATE SECTOR ID        30920860
*          STO  2 IDS#B-TB  UPDATE SECTOR ID   30920870
*          STO  2 COMA+1-TB SET FOR WRITE OP    30920880
*          LDX  3 WRCNT-SUMRY POINTER           30920890
*          BSI  2 COUNT-TB  INCR WRITE COUNT    30920900
*          LDX  13 WRITE   GET ADRS OF CALL+1   30920910
*          MDX  L WRITE,2   CREATE NORMAL RETURN ADRS
*          LD   3 -1     30920920
*          STO  2 NOCK-TB  SET FOR NO CHECK IF NEG.
*          LD   3 2     DSW ERR ADRS           30920930
*          STO  WRITE+1   SET FOR ERROR RETURN  30920940
*          LD   3 0     GET WORD COUNT         30920950
*          STO  2 COMA-TB  SET FOR XIO         30920960
*          BN   WRTA     BIT 0 SET MEANS DON'T  30920970
*          * PRESET THE I/O AREA              30920980
*          LD   3 1     NUMBER                 30920990
*          BSI  2 SETV-TB  PRESET I/O AREA      30921000
*                                               30921010
*                                               30921020
*                                               30921030
*                                               30921040
*                                               30921050
*          ISSUE WRITE THRU XEQ RTN          30921060
*                                               30921070
*          WRTA BSI  2 XEQ-TB  ISSUE WRITE COMMAND
*          LD   2 TERM-TB  GET ALL F'S         30921080
*                                               30921090
    
```

```

OCD5 0 EAB4
OCD6 0 1890
OCD7 0 C218
OCD8 0 DABC
OCD9 0 108D
OCDA 0 C2D2
OCDB 0 1083
OCDC 0 D205
OCDD 0 D243
OCDE 0 6307
OCDF 0 4223
OCE0 1 6780 0697
OCE2 1 7402 0697
OCE4 0 C3FF
OCE5 0 D2FE
OCE6 0 C302
OCE7 0 D018
OCE8 0 C300
OCE9 0 D242
OCEA 1 4C28 OCEE
OCEC 0 C301
OCED 0 422C
OCEE 0 423E
OCEF 0 C288
    
```

```

OCFO 0 D243
OCF1 0 C2FE
OCF2 1 4CA8 0697
OCF4 1 7401 0697
OCF6 0 C2BA
OCF7 1 4C90 0697
OCF9 0 CABC
OCFA 0 DACC
OCFB 0 4232
OCFC 0 0E08
OCFD 0 6308
OCFE 0 4223
OCFF 0 4C00 0000
OD01 0 A2E6
OD02 0 1090
OD03 0 4818
OD04 0 820D
OD05 1 4C80 0685
OD07 1 4C18 0D11
OD09 1 C480 0676
OD0B 0 D004
OD0C 0 18D0
    
```

```

*        STO  2 COMA+1-TB  PRESETS IO AREA    30921100
*        30921110
*        IF NOCK NEGATIVE, DON'T CHECK FOR    30921120
*        DSW ERRORS BUT EXIT AS IF NORMAL WRITE.
*        30921130
*        30921140
*        LD   2 NOCK-TB   GET NO CHECK SWITCH  30921150
*        BN   I WRITE   IF NEGATIVE, EXIT     30921160
*        MDX  L WRITE,1  INCR RETURN           30921170
*        30921180
*        CHECK FOR DSW ERRORS                  30921190
*        30921200
*        LD   2 ZSNS-TB   INTRPT DSW          30921210
*        BNN  I WRITE   EXIT IF NO ERRORS     30921220
*        30921230
*        PRINT UNRECOVERABLE WRT              30921240
*        30921250
*        LDD  2 ZXIO-TB   GET LAST IOCC ISSUED 30921260
*        STD  2 MOD3-TB   SET FOR PRINTING     30921270
*        BSI  2 STMSG-TB  DSW WRITE ERROR      30921280
*        DC   /OE08                                     30921290
*        30921300
*        LDX  3 SFTWR-SUMRY POINTER            30921310
*        BSI  2 COUNT-TB  INCR HARD WRITE ERROR
*        WRTA B   L *-*   DSW ERROR EXIT       30921320
*        30921330
*        -----
*        30921340
*        30921350
*        30921360
*        30921370
*        SUB-ROUTINE RNDOM                     30921380
*        30921390
*        30921400
*        RANDOM NUMBER GENERATOR               30921410
*        30921420
*        ROUTINE IS ENTERED WITH PREVIOUS NUMBER
*        THAT WAS GENERATED, IN THE A REG.     30921430
*        30921440
*        30921450
*        ROUTINE EXITS WITH A NEW 16-BIT
*        NUMBER IN THE A REGISTER.             30921460
*        30921470
*        30921480
*        30921490
**** ROUTINE CALL
*
*   A REG _ NUMBER                            30921500
*   BSI  2 RNDOM-TB                           30921510
*                                               30921520
*                                               30921530
*                                               30921540
*                                               30921550
*   RNDME M  2 K253-TB  GENERATE RANDOM NUMBER 30921560
*   SLT   16     Q TO A                        30921570
*   SKP   +-     SKIP IF NON-ZERO             30921580
*   A     2 K123-TB  ELSE ADD 123 BASE 10     30921590
*   B     I RNDOM   EXIT ROUTINE              30921600
*                                               30921610
*                                               30921620
*                                               30921630
*        -----
*        30921640
*        ROUTINE TO CHECK FOR LOCK             30921650
*        ON ERROR OPTION                       30921660
*        30921670
**** ROUTINE CALL
*
*   BSI  2 CKLK-TB                            30921680
*   DC   ADRS TO RETURN TO IF SW IS SET.      30921690
*                                               30921700
*                                               30921710
*                                               30921720
*        -----
*        30921730
*        CKLKE BZ  CKLK2  DON'T CK IF A REG. ZERO
*        LD   I CKLK  GET LOCK ADRS           30921740
*        STO  CKLK1+1 SET AS RETURN           30921750
*        RTE  16     SET IN Q                 30921760
*                                               30921770
    
```

2310 B DISK FUNCTION TEST

```

OD0D 0 C283          LD      2 SWO-TB   GET SW FNC 0      30921780
OD0E 0 108C          SLT      12          CK FOR LOCK ON ERROR 30921790
OD0F 0 4C28 0000    CKLK1 BN  L *-*          BRANCH IF ON      30921800
OD11 1 7401 0676    CKLK2 MDX L CKLK,1      INCR RETURN       30921810
OD13 1 4C80 0676    BSC      I CKLK        EXIT SUB-ROUTINE  30921820
*-----30921830
*-----30921840
OD15 1 4C18 OD18    CKPRE BZ      CKPRA      DON'T TEST IF ZERO 30921850
OD17 0 C283          LD      2 SWO-TB   GET FNC SW 00     30921860
OD18 0 100A          SLA      10          BIT 10            30921870
OD19 1 4C90 0679    BNN      I CKPRT      EXIT IF NOT SET   30921880
OD1B 1 7401 0679    CKPRA MDX L CKPRT,1     BUMP RETURN       30921890
OD1D 1 4C80 0679    B        I CKPRT      EXIT                30921900
*-----30921910
*-----30921920
*          SUB-ROUTINE SETV          30921930
*-----30921940
*-----30921950
*-----30921960
**** ROUTINE CALL          30921970
*-----30921980
*          (A)_WORD TO BE PRESET IN I/O AREA 30921990
*          BSI      2 SETV-TB        30922000
*-----30922010
*-----30922020
OD1F 1 6780 069E    SETVE LDX  I3 COMA      GET NO. OF WORDS TO BE SET 30922030
OD21 0 7301          MDX      3 1          INCR BY ONE       30922040
OD22 1 D700 069F    SETVA STO  L3 COMA&1    STORE WORD IN I/O AREA 30922050
OD24 0 73FF          MDX      3 -1         DECR COUNT        30922060
OD25 0 70FC          B        SETVA         LOOP                30922070
OD26 1 4C80 0688    B        I SETV        EXIT                30922080
*-----30922090
*-----30922100
*-----30922110
*-----30922120
*          PRINT SUMMARY TABLE      30922130
*-----30922140
**** ROUTINE CALL          30922150
*          BSI      2 TEXIT-TB        30922160
*-----30922170
****ALTERNATE CALL        30922180
*          BSC      L DFTXT          30922190
*-----30922200
*          THIS ROUTINE WILL PRINT THE SUMMARY TABLE 30922210
*          AND SET SW 15 IN FUNCTION ZERO. 30922220
*-----30922230
*-----30922240
*-----30922250
*          TERMINATE DFT              30922260
*-----30922270
*-----30922280
OD28 0 C283          DFTXT LD      2 SWO-TB   FUNCTION 00        30922280
OD29 0 EADA          OR        2 K1-TB     SET BIT 15 TO TERMINATE 30922290
OD2A 0 D283          STO      2 SWO-TB   * DFT PROGRAM      30922300
OD2B 0 4232          BSI      2 STMSG-TB  TERMINATE DFT      30922310
OD2C 0 5A07          DC        /5A07      TERMINATE DFT      30922320
OD2D 1 7401 0646    PRSUM MDX  L PSSCT,1   INCR PASS COUNT    30922330
*-----30922340
*-----30922350
OD2F 1 6700 0647    LDX      L3 SUMRY+1   SET POINTER        30922350
OD31 0 6500 1A09    LDX      L1 /1A09     FORM/MESSAGE ID    30922360
OD33 0 6922          STX      1 ID        SET FOR STMSG CALL 30922370
OD34 0 C035          LD        STMD+1     GET ADDRESS OF STMSG RTN 30922380
OD35 0 D01F          STO      FRM+1       SET FOR CALL       30922390
OD36 0 6103          LDX      1 3         SET LOOP COUNT     30922400
OD37 0 4013          PRLN1 BSI  PRSML      GO PRINT A LINE    30922410
OD38 0 7303          MDX      3 3         INCREMENT POINTER  30922420
OD39 0 71FF          MDX      1 -1        DECREMENT LOOP COUNT 30922430
OD3A 0 70FC          MDX      PRLN1      LOOP IF NOT THROUGH 30922440
OD3B 0 6500 3A05    LDX      L1 /3A05     MESSAGE ID         30922450

```

2310 B DISK FUNCTION TEST

```

OD3D 0 6918          STX      1 ID        SET FOR CALL       30922460
OD3E 0 400C          BSI      PRSML      PRINT A LINE       30922470
OD3F 0 7302          MDX      3 2         BUMP POINTER      30922480
OD40 0 6500 6A0C    LDX      L1 /6A0C    MESSAGE ID         30922490
OD42 0 6913          STX      1 ID        SET FOR CALL       30922500
OD43 0 6103          LDX      1 3         SET LOOP COUNTER  30922510
OD44 0 4C06          PRLN2 BSI  PRSML      PRINT A LINE       30922520
OD45 0 7302          MDX      3 2         INCREMENT POINTER 30922530
OD46 0 71FF          MDX      1 -1        DECREMENT LOOP COUNTER 30922540
OD47 0 70FC          MDX      PRLN2      LOOP IF NOT THROUGH 30922550
OD48 0 4480 0164    BSI      I END       ELSE GO TO MONITOR END RTN 30922560
OD4A 1 05DC          DC        PID        ADDRESS OF PROGRAM ID 30922570
*-----30922580
*-----30922590
PRSM DC      *-*      ENTRY          30922590
LD      2 PSSCT-TB   GET PASS COUNT    30922600
STO     2 MOD3-TB   SET IN MSG        30922610
LD      3 0         GET MODIFIER WORD 30922620
STO     2 MOD4-TB   SET IN MSG        30922630
LD      3 1         GET MODIFIER       30922640
STO     2 MOD5-TB   SET IN MSG        30922650
LD      3 2         GET MODIFIER       30922660
STO     2 MOD6-TB   SET IN MSG        30922670
FRM BSI  L STMSG    GO PRINT LINE     30922680
ID   DC      *-*      FORM/MSG ID      30922690
LD      FRM&1      SET NO PID-MID----PRINT 30922700
OR      2 H8000-TB  30922710
STO     FRM&1      30922720
MDX     L ID,1     INCREMENT MSG ID 30922730
BSC     I PRSML    EXIT                30922740
*-----30922750
*-----30922760
*-----30922770
*          INCREMENT AN ENTRY IN THE SUMMARY. 30922780
*-----30922790
*-----30922800
*          IX 1 IS THE POSITION IN THE SUMMARY 30922810
*          TABLE TO BE INCREMENTED. IF THE 30922820
*          COUNT EXCEEDS 9999, THE COUNTER IS 30922830
*          RESET TO ZERO SO THAT A MODULO 10,000 30922840
*          IS SIMULATED. THIS IS TO RELIEVE 30922850
*          PRINTOUT PROBLEMS. 30922860
*-----30922870
*-----30922880
OD5E 1 C700 0646    COUNE LD      L3 SUMRY   GET PROPER COUNTER 30922880
OD60 0 82DA          A        2 K1-TB     ADD ONE            30922890
*-----30922900
*-----30922910
OD61 1 D700 0646    COUNA STO     L3 SUMRY   STORE              30922910
OD63 0 92E9          S        2 K10TH-TB  SUBTRACT 10,000. THIS WIL 30922920
*          * BE MEANINGLESS ON LOOP. 30922930
*          BNZ      I COUNT   EXIT IF NOT 10,000 30922940
*          B        COUNA    ELSE LOOP        30922950
*-----30922960
*-----30922970
*-----30922980
*-----30922990
*          SUB-ROUTINE STMSG          30923000
*-----30923010
*-----30923020
*-----30923030
*          COMMON MESSAGE SETUP RTN  30923040
*-----30923050
*          THIS ROUTINE WILL BUILD THE 30923060
*          MESSAGE DESIRED AND CALL RTN 30923070
*          PRINT TO PRINT THE MESSAGE. 30923080
*-----30923090
**** ROUTINE CALL          30923100
*-----30923110
*          BSI      2 STMSG-TB        30923120
*          DC        MSG ID          30923130

```

```

* MSG ID _ FMM (WHERE) 30923140
* F _ FORM NUMBER 30923150
* MMM _ MESSAGE ID 30923160
* 30923170
****ALTERNATE CALL USED 30923180
* BSI L STMSG+/8000 30923190
* DC MSG ID 30923200
* 30923210
* THE ALTERNATE CALL WILL PRINT THE MESSAGE 30923220
* AS AN ADDITIONAL LINE WHERE PID-MID-RID-RAD 30923230
* ARE NOT PRINTED. 30923240
* 30923250
*-----*
* FORM MOD MOD MOD MOD MOD MOD MOD * 30923260
* NO. 1 2 3 4 5 6 7 * 30923270
*-----*
* 0 FILE# DSW HEX HEX * 30923280
*-----*
* 1 FILE# DSW DEC DEC DEC DEC * 30923290
*-----*
* 2 FILE# DSW DEC DEC DEC * 30923300
*-----*
* 3 FILE# DSW DEC DEC * 30923310
*-----*
* 4 FILE# DSW DEC * 30923320
*-----*
* 5 FILE# DSW * 30923330
*-----*
* 6 FILE# DSW DEC HEX HEX * 30923340
*-----*
* 7 FILE# DSW CYL OOHs #WDS * 30923350
*-----*
* 8 FILE# DSW HEX HEX HEX HEX * 30923360
*-----*
* 9 FILE# DSW HEX * 30923370
*-----*
* 30923380
* 30923390
* 30923400
* 30923410
* 30923420
* 30923430
* 30923440
* 30923450
* 30923460
* 30923470
* 30923480
* 30923490
* 30923500
* 30923510
* 30923520
* 30923530
* 30923540
* 30923550
* 30923560
* 30923570
* 30923580
* 30923590
* 30923600
* 30923610
* 30923620
* 30923630
* 30923640
* 30923650
* 30923660
* 30923670
* 30923680
* 30923690
* 30923700
* 30923710
* 30923720
* 30923730
* 30923740
* 30923750
* 30923760
* 30923770
* 30923780
* 30923790
* 30923800
* 30923810
* 30923820
* 30923830
* 30923840
* 30923850
* 30923860
* 30923870
* 30923880
* 30923890
* 30923900
* 30923910
* 30923920
* 30923930
* 30923940
* 30923950
* 30923960
* 30923970
* 30923980
* 30923990
* 30924000
* 30924010
* 30924020
* 30924030
* 30924040
* 30924050
* 30924060
* 30924070
* 30924080
* 30924090
* 30924100
* 30924110
* 30924120
* 30924130
* 30924140
* 30924150
* 30924160
* 30924170
* 30924180
* 30924190
* 30924200
* 30924210
* 30924220
* 30924230
* 30924240
* 30924250
* 30924260
* 30924270
* 30924280
* 30924290
* 30924300
* 30924310
* 30924320
* 30924330
* 30924340
* 30924350
* 30924360
* 30924370
* 30924380
* 30924390
* 30924400
* 30924410
* 30924420
* 30924430
* 30924440
* 30924450
* 30924460
* 30924470
* 30924480
* 30924490

```

```

0D67 0 691C
0D68 0 6B1F
0D69 1 6580 068E
0D6B 0 C1FF
0D6C 0 E2B9
0D6D 0 D2C7
0D6E 0 10A0
0D6F 0 C100
0D70 0 18D8
0D71 0 1808
0D72 0 D2C5
0D73 0 1010
0D74 0 108C
0D75 0 D00A
0D76 0 1083
0D77 0 D2C9
0D78 1 7401 068E
*
0D7A 0 C2F5
0D7B 1 4C20 0D83
*
0D7D 0 6301
0D7E 0 C2DB
0D7F 0 6500 0000
0D81 1 4D80 0D88
*
0D83 0 6500 0000
0D85 1 6600 065C
0D87 0 6700 0000
0D89 1 4C80 068E

```

```

STMSE STX 1 STMSX+1 SAVE IX 1
STX 3 STMSX+5 SAVE IX3
STMAD LDX 11 STMSG GET CALL ADRS
LD 1 -1 *
AND 2 H8000-TB SAVE BIT 0
STO 2 MSGID-TB STORE FOR LOG CALL
SLT 32 CLEAR AQ
LD 1 0 GET FORM NUJBER
RTE 24 *
SRA 8 GET MSG ID
STO 2 MSGO-TB SAVE FOR LATER
SLA 16 CLEAR A-REG
SLT 12 GET FORM NUMBER TO A-REG
STO STMSA+1 SAVE FOR INDEXING
SLT 3 GET TYPE MSG INDICATOR
STO 2 ADDR2-TB SAVE FOR LATER TEST
MDX L STMSG,1 INCR RETURN
*
LD 2 PRSW-TB GET PRINTER SWITCH
BNZ STMSX EXIT IF SW IS NON-ZERO
*
LDX 3 1 SET CONV CTR
LD 2 K2-TB GET MOD WD CT
STMSA LDX L1 *- *
B 11 FRMTB
*
STMSX LDX L1 *- * RESTORE
LDX L2 TB SET UP TABLE POINTER
LDX L3 *- * RESTORE IX
B I STMSG EXIT

```

```

0D8B 1 0D95
0D8C 1 0D97
0D8D 1 0D98
0D8E 1 0D99
0D8F 1 0D9A
0D90 1 0D9C
0D91 1 0DA5
0D92 1 0DA8
0D93 1 0DB6
0D94 1 0DB7
*
0D95 0 82DB
0D96 0 7005
*
0D97 0 82DA
*
0D98 0 82DA
*
0D99 0 82DA
*
0D9A 0 82DA
0D9B 0 6306
*
0D9C 0 D2D1
0D9D 1 6580 062D
0D9F 0 CA88
0DA0 0 1010
0DA1 0 11C0
0DA2 0 EAC7
0DA3 0 D2C7
0DA4 0 7019
*
0DA5 0 82DC
0DA6 0 6302
0DA7 0 70F4
*
0DA8 0 C205
0DA9 0 1803
0DAA 0 4037
0DAB 0 D2CC
0DAC 0 C205
0DAD 0 E2DE
0DAE 0 1882
0DAF 0 1002
0DAB 0 1082
0DB1 0 D2CD

```

```

*-----*
* FORM TABLE
*
FRMTB DC FORM0 ADRS OF RTN FORM 0
DC FORM1 1
DC FORM2 2
DC FORM3 3
DC FORM4 4
DC FORM5 5
DC FORM6 6
DC FORM7 7
DC FORM8 8
DC FORM9 9
*
* FORM IS 0
*
FORM0 A 2 K2-TB INCR MOD CNT
B FORM5 COMMON ROUTINE
*
* FORM IS 1
*
FORM1 A 2 K1-TB INCR MOD CNT
*
* FORM IS 2
*
FORM2 A 2 K1-TB INCR MOD CNT
*
* FORM IS 3
*
FORM3 A 2 K1-TB INCR MOD CNT
*
* FORM IS 4
*
FORM4 A 2 K1-TB INCR MOD CNT
LDX 3 6 SET CONVERSION CTR
*
* FORM IS 5
*
COMMON ROUTINE
*
FORM5 STO 2 BNTMP-TB SAVE IN TEMP STORAGE
LDX 11 BNTMP SET COUNT IN IX1
LDD 2 TERM-TB SET AQ TO FOXES
SLA 16 CLEAR A REGISTER
SLC 1 0 SET BITS FOR MODIFIERS
OR 2 MSGID-TB GET BIT 0
STO 2 MSGID-TB SET IN TABLE
B PRINT PRINT
*
* FORM IS 6
*
FORM6 A 2 K3-TB INCR MOD CNTR
LDX 3 2 SET CONVERSION CNTR
B FORM5
*
* FORM IS 7
*
FORM7 LD 2 IDS#B-TB GET PRESENT SECTOR
SRA 3 SAVE CYLINDER NUMBER ONLY
BSI BNDEC CONVERT TO DECIMAL
STO 2 MOD3-TB SET FOR PRINTING
LD 2 IDS#B-TB GET PRESENT SECTOR ID
AND 2 K7-TB SAVE SECTOR ONLY
SRT 2 SET SECTOR COUNT IN Q
SLA 2 CLEAR TWO BITS
SLT 2 MOVE BACK SECTOR CNT
STO 2 MOD4-TB SET FOR PRINTING

```

2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

```

ODB2 0 C2FF      LD 2 LNGTH-TB  GET READ LENGTH      30924500
ODB3 0 402E      BSI BNDEC    CONVERT TO DECIMAL      30924510
ODB4 0 D2CE      STO 2 MOD5-TB  SET IN PRINT MOD      30924520
ODB5 0 C2DA      LD 2 K1-TB    SET WORD CNT          30924530
*              FALL THRU TO GET WRD CNT_5      30924540
*              *                              30924550
*              FORM IS 8                      30924560
*              *                              30924570
ODB6 0 82DC      FORM8 A 2 K3-TB    BUMP MSG WD COUNT  30924580
*              *                              30924590
*              FORM IS 9                      30924600
*              *                              30924610
ODB7 0 82DA      FORM9 A 2 K1-TB    INCR MOD CNT      30924620
ODB8 0 70E3      B          FORM5    COMMON ROUTINE      30924630
*              *                              30924640
*              *                              30924650
ODB9 1 C700 0627 MSGC1 LD L3 MOD3-1  GET MODIFIER      30924660
ODB8B 0 4026      BSI BNDEC    CONVERT TO DECIMAL      30924670
ODB8C 1 D700 0627 STO L3 MOD3-1  SET IN MODIFIER      30924680
ODB8E 0 73FF      PRINT MDX 3 -1    DECR CONVERSION CTR      30924690
ODB8F 0 70F9      MDX          MSGC1    LOOP              30924700
ODC0 1 6780 0621 PRINB LDX I3 MSGO    GET MSG ID NUMBER  30924710
ODC2 0 C2C5      LD 2 MSGO-TB  GET SAME TO A-REG      30924720
ODC3 0 401E      BSI BNDEC    CONVERT TO DECIMAL      30924730
ODC4 0 D2C5      STO 2 MSGO-TB  SET FOR MSG ID      30924740
ODC5 0 C2C9      LD 2 ADDR2-TB GET MSG TYPE INDICATOR  30924750
ODC6 0 E2DE      AND 2 K7-TB   SAVE BITS 13-15      30924760
ODC7 0 F2DE      EOR 2 K7-TB   (NO MESSAGES WITH F)  30924770
ODC8 1 4C18 0DD5 BZ          PRINA   BRANCH FOR ERROR MESSAGE  30924780
ODCA 1 C700 0DF6 LD L3 AALPH  GET ALPHA ADDRESS  30924790
ODCC 0 D2C9      STO 2 ADDR2-TB SET ALPHA ADDR FOR LOG  30924800
ODCD 0 C283      LD 2 SWO-TB  GET SW FNC 0          30924810
ODCE 0 100D      SLA 13       BIT 13-BYPASS ALL LOGS  30924820
ODCF 1 4C28 0D83 BN          STMSX   EXIT IF SET      30924830
*              *                              30924840
ODD1 0 4480 0163 BSI I LOG      MONITOR LOG CALL      30924850
ODD3 1 0621      DC          MSGO    ADRS MSG          30924860
ODD4 0 70AE      B          STMSX   EXIT PRINT ROUTINE  30924870
*              *                              30924880
ODD5 1 C700 0E05 PRINA LD L3 EALPH  GET ALPHA ADDRESS  30924890
ODD7 0 D2C9      STO 2 ADDR2-TB SET ALPHA ADDR FOR ERROR  30924900
ODD8 0 4480 0162 BSI I ERROR    MONITOR ERROR CALL      30924910
ODDA 1 0621      DC          MSGO    ADRS MSG          30924920
ODDB 1 0DDD      DC          PRLP   LOOP RETURN        30924930
ODDC 0 70A6      B          STMSX   EXIT              30924940
*              *                              30924950
*              LOOP ON ERROR RETURN          30924960
*              *                              30924970
ODDD 0 C281      PRLP LD 2 RID-TB  GET PRESENT ROUTINE NUMBER  30924980
ODDE 1 4C18 0869 BZ          PRECN  IF ZERO, GO TO PRE-CONTROL  30924990
ODE0 1 4C00 08CF B          L CNTLD ELSE LOOP SAME ROUTINE  30925000
*              *                              30925010
*              *                              30925020
*              CONVERT A POSITIVE NUMBER IN THE  30925030
*              RANGE FROM 0-9999 TO IT'S DECI-  30925040
*              MAL EQUIVALENT.              30925050
*              *                              30925060
*              ENTER ROUTINE WITH THE NUMBER    30925070
*              IN THE A REGISTER. EXIT WITH    30925080
*              THE CONVERTED NUMBER IN THE A REG.  30925090
*              *                              30925100
ODE2 0 0000      BNDEC DC *-*     ENTRY POINT          30925110
ODE3 1 4CAB 0DE2 BN          I BNDEC  EXIT IF NEGATIVE  30925120
ODE5 0 189C      SRT          16    A TO Q          30925130
ODE6 0 AAE8      D 2 THOUS-TB  MOST SIGNIF. DIGIT  30925140
ODE7 0 100C      SLA 12       POSITION DIGIT      30925150
ODE8 0 D2BE      STO 2 SNXIO-TB SAVE              30925160
ODE9 0 1004      SLA 4          CLEAR A REG          30925170

```

```

ODEA 0 AAE3      D 2 HUNDR-TB  NEXT SIGNIF. DIGIT      30925180
ODEB 0 1008      SLA 8          POSITION              30925190
ODEC 0 EABE      OR 2 SNXIO-TB  COMBINE IN HEX WORD      30925200
ODED 0 D2BE      STO 2 SNXIO-TB  SAVE              30925210
ODEE 0 1008      SLA 8          CLEAR A              30925220
ODEF 0 AAE0      D 2 TEN-TB    NEXT SIGNIF. DIGIT      30925230
ODFO 0 108C      SLT 12       COMBINE LAST TWO DIGITS  30925240
ODF1 0 180C      SRA 12       *                      30925250
ODF2 0 1084      SLT 4          *                      30925260
ODF3 0 EABE      OR 2 SNXIO-TB  COMBINE IN HEX WORD      30925270
ODF4 1 4C80 0DE2 BSC I BNDEC  EXIT WITH WORD IN A REG.  30925280
*              *                              30925290
*              *                              30925300
*              *                              30925310
*              *                              30925320
*              *                              30925330
*              *                              30925340
*              *                              30925350
*              *                              30925360
*              *                              30925370
*              *                              30925380
*              *                              30925390
*              *                              30925400
*              *                              30925410
*              *                              30925420
*              *                              30925430
*              *                              30925440
*              *                              30925450
*              *                              30925460
*              *                              30925470
*              *                              30925480
*              *                              30925490
*              *                              30925500
*              *                              30925510
*              *                              30925520
*              *                              30925530
*              *                              30925540
*              *                              30925550
*              *                              30925560
*              *                              30925570
*              *                              30925580
*              *                              30925590
*              *                              30925600
*              *                              30925610
*              *                              30925620
*              *                              30925630
*              *                              30925640
*              *                              30925650
*              *                              30925660
*              *                              30925670
*              *                              30925680
*              *                              30925690
*              *                              30925700
*              *                              30925710
*              *                              30925720
*              *                              30925730
*              *                              30925740
*              *                              30925750
*              *                              30925760
*              *                              30925770
*              *                              30925780
*              *                              30925790
*              *                              30925800
*              *                              30925810
*              *                              30925820
*              *                              30925830
*              *                              30925840
*              *                              30925850

```

2310 B DISK FUNCTION TEST

0E48 0 FFFF	DC	/FFFF	30925860
0E49 0	ALPA3 EQU	*	30925870
0E49 0010	TYPE	. RECOVERED RD ERR.	30925880
0E53 0 FFFF	DC	/FFFF	30925890
0E54 0	ALPA4 EQU	*	30925900
0E54 0007	TYPE	. HRD CMP ERR.	30925910
0E5B 0 FFFF	DC	/FFFF	30925920
0E5C 0	ALPA5 EQU	*	30925930
0E5C 0012	TYPE	. PASS #WDS WRTN IN SECT.	30925940
0E68 0 FFFF	DC	/FFFF	30925950
0E69 0	ALPA6 EQU	*	30925960
0E69 0010	TYPE	. RECOVERED SK ERR.	30925970
0E73 0 FFFF	DC	/FFFF	30925980
0E74 0	ALPA7 EQU	*	30925990
0E74 0009	TYPE	. DFT TERMINATED.	30926000
0E7D 0 FFFF	DC	/FFFF	30926010
0E7E 0	ALPA8 EQU	*	30926020
0E7E 0012	TYPE	. SEEK TIMING IN PROGRESS.	30926030
0E8A 0 FFFF	DC	/FFFF	30926040
0E8B 0	ALPA9 EQU	*	30926050
0E8B 0016	TYPE	. PASS SKS SFT HRD 7-LINE SMRY.	30926060
0E9B 0 FFFF	DC	/FFFF	30926070
0E9C 0	ALPAA EQU	*	30926080
0E9C 0010	TYPE	. PASS RDS SFT HRD.	30926090
0EA6 0 FFFF	DC	/FFFF	30926100
0EA7 0	ALPAB EQU	*	30926110
0EA7 0010	TYPE	. PASS WRTS SFT HRD.	30926120
0EB1 0 FFFF	DC	/FFFF	30926130
0EB2 0	ALPAC EQU	*	30926140
0EB2 0009	TYPE	. PASS SK-RNDM WDS.	30926150
0EBB 0 FFFF	DC	/FFFF	30926160
0EBC 0	ALPAD EQU	*	30926170
0EBC 0012	TYPE	. PASS WRT-RNDM WDS/RTN9.	30926180
0EC8 0 FFFF	DC	/FFFF	30926190
0EC9 0	ALPAE EQU	*	30926200
0EC9 0013	TYPE	. PASS WRT-RNDM WDS/RTN10.	30926210
0ED6 0 FFFF	DC	/FFFF	30926220
0ED7 0	ALPE1 EQU	*	30926230
0ED7 0011	TYPE	. LAST IOCC LOST INT.	30926240
0EE2 0 FFFF	DC	/FFFF	30926250
0EE3 0	ALPE2 EQU	*	30926260
0EE3 0006	TYPE	. EXTRA INT.	30926270
0EE9 0 FFFF	DC	/FFFF	30926280
0EEA 0	ALPE3 EQU	*	30926290
0EEA 0007	TYPE	. DSK NOT RDY.	30926300
0EF1 0 FFFF	DC	/FFFF	30926310
0EF2 0	ALPE4 EQU	*	30926320
0EF2 0011	TYPE	. LAST IOCC DSW WRONG.	30926330
0EFD 0 FFFF	DC	/FFFF	30926340
0EFE 0	ALPE5 EQU	*	30926350
0EFE 0012	TYPE	. LAST IOCC DSW SK ERR.	30926360
0FOA 0 FFFF	DC	/FFFF	30926370
0F0B 0	ALPE6 EQU	*	30926380
0F0B 0012	TYPE	. LAST IOCC DSW RD ERR.	30926390
0F17 0 FFFF	DC	/FFFF	30926400
0F18 0	ALPE7 EQU	*	30926410
0F18 0014	TYPE	. LAST IOCC HRD DSW RD ERR.	30926420
0F26 0 FFFF	DC	/FFFF	30926430
0F27 0	ALPE8 EQU	*	30926440
0F27 0012	TYPE	. LAST IOCC DSW WRT ERR.	30926450
0F33 0 FFFF	DC	/FFFF	30926460
0F34 0	ALPE9 EQU	*	30926470
0F34 0013	TYPE	. FROM S/B WAS SEEK ERR.	30926480
0F41 0 FFFF	DC	/FFFF	30926490
0F42 0	ALPEA EQU	*	30926500
0F42 0013	TYPE	. CYL# 00HS #WDS CMP ERR.	30926510
0F4F 0 FFFF	DC	/FFFF	30926520
0F50 0	ALPEB EQU	*	30926530

2310 B DISK FUNCTION TEST

0F50 0013	TYPE	. CYL# 00HS #WDS OVERREAD.	30926540
0F50 0 FFFF	DC	/FFFF	30926550
0F5E 0	ALPEC EQU	*	30926560
0F5E 0007	TYPE	. WRITE ERROR.	30926570
0F65 0 FFFF	DC	/FFFF	30926580
0F66 0	ALPED EQU	*	30926590
0F66 0012	TYPE	. CYL# HOME BIT WRONG.	30926600
0F72 0 FFFF	DC	/FFFF	30926610
0F73 0	ALPEE EQU	*	30926620
0F73 0016	TYPE	. LAST IOCC NO DSW ERR/WRT 400.	30926630
0F83 0 FFFF	DC	/FFFF	30926640
0F84 0	ALPEF EQU	*	30926650
0F84 0015	TYPE	. LAST IOCC NO DSW ERR/RD 370.	30926660
0F93 0 FFFF	DC	/FFFF	30926670
0F94 0	ALPFO EQU	*	30926680
0F94 0013	TYPE	. WDCNT SECT WD CNT WRONG.	30926690
0FA1 0 FFFF	DC	/FFFF	30926700
0FA2 0	ALPF1 EQU	*	30926710
0FA2 0012	TYPE	. WRT 400 WROTE NEXT SECT.	30926720
0FAE 0 FFFF	DC	/FFFF	30926730
0FAF 0	ALPF2 EQU	*	30926740
0FAF 0016	TYPE	. LAST IOCC #TMS RDCK XFRD DATA.	30926750
0FBF 0 FFFF	DC	/FFFF	30926760
0FC0 0	ALPF3 EQU	*	30926770
0FC0 0008	TYPE	. SECT CNTR BAD.	30926780
0FC8 0 FFFF	DC	/FFFF	30926790
0FC9 0	ALPF4 EQU	*	30926800
0FC9 0008	TYPE	. 8 CYLINDER IDS.	30926810
0FD1 0 FFFF	DC	/FFFF	30926820
0FD2 0	ALPF5 EQU	*	30926830
0FD2 0003	TYPE	. #ERS.	30926840
0FD5 0 FFFF	DC	/FFFF	30926850
0FD6 0	ALPF6 EQU	*	30926860
0FD6 0007	TYPE	. WD# S/B WAS.	30926870
0FDD 0 FFFF	DC	/FFFF	30926880
0F66 0	ALPF7 EQU	ALPED	30926890
0FDE 0	ALPF8 EQU	*	30926900
0FDE 0008	TYPE	. SK/HOME BIT ERR.	30926910
0FE6 0 FFFF	DC	/FFFF	30926920
0FE7 0	ALPF9 EQU	*	30926930
0FE7 0018	TYPE	. MAX MIN AVG INCR SK TIME (MS).	30926940
0FF9 0 FFFF	DC	/FFFF	30926950
*			30926960
*			30926970
OFFA 0	PEND EQU	* LAST PROGRAM ADRS	30926980
	*****	*****	30926990
OFFA 05F3	END BEG	INITIAL XFER ADDRESS	30927000
	NO STATEMENTS FLAGGED IN THE ABOVE ASSEMBLY		



2310 B DISK FUNCTION TEST

2310 B DISK FUNCTION TEST

AALPH 0DF6 0DCA  
 ADCMA 0674 0BEF 0CD7  
 ADDRS 0168  
 ADDR1 0624  
 ADDR2 0625 0D77 0DC5 0DCC 0DD7  
 ADIL2 0671 0826  
 ADINT 0672 083A 0AF9  
 ADIN2 0673 0AD2  
 ADPCN 0675 085A  
 ALPAA 0E9C 0E00  
 ALPAB 0EA7 0E01  
 ALPAC 0EB2 0E02  
 ALPAD 0EBC 0E03  
 ALPAE 0E69 0E04  
 ALPA0 0E24 0DF6  
 ALPA1 0E38 0DF7  
 ALPA2 0E41 0DF8  
 ALPA3 0E49 0DF9  
 ALPA4 0E54 0DFA  
 ALPA5 0E5C 0DFB  
 ALPA6 0E69 0DFC  
 ALPA7 0E74 0DFD  
 ALPA8 0E7E 0DFE  
 ALPA9 0E8B 0DFF  
 ALPEA 0F42 0E0F  
 ALPEB 0F50 0E10  
 ALPEC 0F5E 0E11  
 ALPED 0F66 0E12  
 ALPEE 0F73 0E13  
 ALPEF 0F84 0E14  
 ALPE1 0ED7 0E06  
 ALPE2 0EE3 0E07  
 ALPE3 0EEA 0E08  
 ALPE4 0EF2 0E09  
 ALPE5 0EFE 0E0A  
 ALPE6 0F0B 0E0B  
 ALPE7 0F18 0E0C  
 ALPE8 0F27 0E0D  
 ALPE9 0F34 0E0E  
 ALPF0 0F94 0E15  
 ALPF1 0FA2 0E16  
 ALPF2 0FAF 0E17  
 ALPF3 0FC0 0E18  
 ALPF4 0FC9 0E19  
 ALPF5 0FD2 0E1A  
 ALPF6 0FD6 0E1B  
 ALPF7 0F66 0E1C  
 ALPF8 0FDE 0E1D  
 ALPF9 0FE7 0E1E  
 ALPH0 0E1F 0624  
 AVG 0A03 0A16 0A34 0A36 0A57  
 BADCY 05F0 089C 0B4D  
 BDCYL 080C 088E 0B97 0BA1 0BA3 0BBE 0BC1  
 BEG 05F3 0FFB  
 BEGIN 0160 05F3  
 BNDEC 00E2 00AA 0DB3 0DBB 0DC3 0DE3 0DF4  
 BNTMP 062D 0D9C 0D9D  
 CKLK 0676 0BD9 0C18 0C3B 0D09 0D11 0D13  
 CKLKE 0D07 0677  
 CKLK1 0D0F 0D0B  
 CKLK2 0D11 0D07  
 CKPRA 0D1B 0D15  
 CKPRE 0D15 067A  
 CKPRT 0679 099D 0A62 0C30 0C68 0C77 0C92 0CA0 0CC1 0D19 0D1B 0D1D  
 CKRD1 0804 0811  
 CKRD3 080D 0810  
 CMN1 0951 0959  
 CMN2 094F 0958

CMN3 0957 095E  
 CMP 0C4B 08FB 0C12 0C34 0C7F 0C81  
 CMPB 0C83 0C55 0C8B 0C9C 0CAF 0CB3 0CB9  
 CMPB1 0C94 0CA6  
 CMPB2 0C9F 0C90  
 CMPB3 0CA9 0C99  
 CMPB4 0CAB 0C97 0CAB  
 CMPB5 0CAD 0C9E  
 CMPB6 0CB3 0C93  
 CMPEX 0C81 0C7D  
 CMPTM 0668 08FA 0C00 0C51 0C56  
 CMPX 0C7C 0C74  
 CMP1 0C59 0C61  
 CMP2 0C5D 0C58 0CBE  
 CMP2A 0CA4 0CA1  
 CMP3 0C6D 0C65 0C69  
 CMP4 0C7B 0C78  
 CMRND 0CB5 0C5A  
 CMRN1 0CB8 0C5C  
 CMRTA 0973 0972  
 CMRTB 097A 096C 098A  
 CMRTD 0981 0980  
 CMRTG 0984 0979  
 CMRTL 0969 0982  
 CMRT2 095F 0953  
 CNTA 0658  
 CNTB 0631 09A8 09B0 0A80 0A86 0AA2 0ABB  
 CNTL 08B7 067D 08A5 0968 098D 09AB 09B5 09BB 09FE 0A14  
 CNTLB 08C5 08D1  
 CNTLD 08CF 08BD 0DE0  
 CNTLE 067C 090F 0925 094C 0983 09AF 09F2 0A11 0A5C  
 CGMA 069E 0674 085D 0875 0893 089A 089E 096A 0992 09CD 0A18 0A2C 0A3B 0A7D  
 0A9E 0B12 0B60 0B8D 0BAC 0BFD 0C6F 0C71 0C86 0C8A 0CDD 0CE9 0CFO  
 0D1F 0D22  
 COUNA 0D61 0D66  
 COUNE 0D5E 0680  
 COUNT 067F 0987 09F6 0B71 0B86 0BBB 0C06 0C1B 0C2A 0C40 0CDF 0CFE 0D64  
 CPUT0 0E2D 0841 084C  
 CPUT2 066A 084B  
 CPUT3 066B 0840  
 CYL 0967 095F  
 DATA1 0971 0961  
 DATA2 0978 0962  
 DATA3 097F 0963  
 DFXT 0D28 0692 088B 088E 08BA  
 DSKMD 0611 0800 0801 08EB  
 DVA 05F7 0823 0861 0AFD  
 DVAS 05E9 0821  
 EALPH 0E05 0DD5  
 END 0164 0AC9 0D48  
 ERCT 065D 0C4D 0C63 0C7C 0C8D 0C8E  
 ERLCK 0166  
 ERROR 0162 0DD8  
 ESK1 0630 098F 099B 09AA  
 FILE# 0626 0824 0827  
 FORM0 0D95 0D8B  
 FORM1 0D97 0D8C  
 FORM2 0D98 0D8D  
 FORM3 0D99 0D8E  
 FORM4 0D9A 0D8F  
 FORM5 0D9C 0D90 0D96 0DA7 0DB8  
 FORM6 0DA5 0D91  
 FORM7 0DAB 0D92  
 FORM8 0DB6 0D93  
 FORM9 0DB7 0D94  
 FRM 0D54 0D35 0D57 0D59  
 FRMTB 0DB8 0DB1  
 FRNSK 0652 0940

FRN1 0654 09ED  
FRN2 0656  
FSINO 09FB 09C4 09E8 0A33 0A49  
HCEDC 0868 0894  
HE5E5 0620 0957  
HRDRD 064C 0C29 0C3F  
HRDSK 0649 0B8A  
HRDWR 064F 0986 09F5  
HUNDR 063F 0DEA  
HXDC 0622  
H1313 060E 094F 0A19 0A2E  
H3000 060D 0B06 0B18 0B19  
H8000 0615 0A91 0D58 0D6C  
ID 0D56 0D33 0D3D 0D42 0D5A  
IDS#B 0661 0A3C 0BF4 0C53 0CDC 0DAB 0DAC  
ILO 017A  
IL1 018A  
IL2 019A 0671  
IL3 01AA  
IL4 01BA  
INCR1 093A 091A 0920 092B  
INCR2 093B 091C 0922 0931  
INDEX 065C 0C50 0C5D 0C5F 0C6D 0C84 0C96 0CA4 0CAE OCC9  
INTR 05F8 0602 0672  
INTRB 0604 05FA  
INTRC 05FE  
INTSW 05F6 05F9 05FF 083D 0846 0AFF 0B21  
INT2 0AE0 0673 0AE2 0AE7  
INT2A 0AE9 0AE8  
IPA 05E4  
KLP13 066D 0A7F 0AA1 0ABF  
KM14 063D 084E  
K1 0636 0864 08C0 08EA 091D 0973 09A0 09D8 0A37 0A44 0A7B 0AD7 0AE6 0B6B  
0891 0C4F 0C7B 0C95 0C9B 0CA5 0D29 0D60 0D97 0D98 0D99 0D9A 0DB5  
0DB7  
K10TH 0645 08A9 0B0B 0B1E 0B73 0D63  
K123 0669 0D04  
K2 0637 0A95 0C73 0D7E 0D95  
K202 0640 0A94  
K203 0641 0870 0923 0A6E  
K253 0642 0D01  
K27 0A1F 0A4E 0A64  
K3 0638 087A 088F 08AE 0DA5 0DB6  
K331 0643 0A4B  
K36 063E 084F  
K370 0A26 0A17  
K4 0639 0A82  
K7 063A 08EF 0B88 0B99 0DAD 0DC6 0DC7  
K8 063B 0B3F 0B6E 0BD3 0C07  
LNGTH 065B 0BFE 0C60 0CAD 0DB2  
LOG 0163 0DD1  
LOGBY 0167 082A  
LPA 05E3  
LPCNT 0032 08CA 093C 0952 0A6B  
LPRNT 0662 0C4E 0C94 0CCE  
LP13 0014 066D  
LRN1 0655 09B7 09F0  
LKN2 0657 098C  
LRTN 08E3 08D6  
LSTNO 09FA 09C0 09C3 09C7 09CA 09D2 09EF  
MASK 060F 0881  
MLSCF 05E5 05FD 085B 0B33  
MOD3 0628 0880 09A7 0A4A 0AA6 0AAE 0AB3 0ACD 0AED 0B2D 0BAB 0BC0 0BE0 0C3A  
0C64 0CCC 0CFA 0D4D 0DAB 0DB9 0DBC  
MOD4 0629 0AB4 0AB9 0AFO 0CC8 0D4F 0DB1  
MOD5 062A 09A9 0AA8 0AAC 0AAD 0ABE 0ACO 0AEF 0AF4 0BAE 0BC3 0CC5 0D51 0DB4  
MOD6 062B 0AC3 0D53  
MOD7 062C

MSGC1 0DB9 0DBF  
MSGID 0623 0D6D 0DA2 0DA3  
MSGO 0621 0D72 0DC0 0DC2 0DC4 0DD3 0DDA  
NCYL# 062F 0990 0B4A 0B4C 0B53 0B61 0B80 0BF2  
NLOOP 0A68 0A5A  
NOCK 065A 0BF8 0C0C 0CE5 0CF1  
PCYL# 062E 086F 0B62 0BAA 0BAF 0BDF 0CDA  
PEND 0FFA 05E8  
PID 05DC 05F5 0ACB 0D4A  
PNTB 0CBF 0CAA 0CAC 0CB2 0CD3  
PNTBX 0CD3 0CC2  
PNTC 0CC3 0C8C  
PRECA 0874 0888  
PRECB 0878 087E  
PRECD 0887 0882  
PRECE 0889  
PRECF 089E 08A4  
PRECG 089A 0895  
PRECJ 08A7 0878 08AF  
PRECL 08AB 08B3  
PRECM 08AF 08B6  
PRECN 0869 0675 0DDE  
PRECT 08A6 0879 08A8 08AD 08B5  
PRINA 0DD5 0DC8  
PRINB 0DC0  
PRINT 0DBE 0DA4  
PRLN1 0D37 0D3A  
PRLN2 0D44 0D47  
PRLP 0DDD 0DD8  
PKNSK 0653 093F 0942 0944  
PRSM 0D4B 0D37 0D3E 0D44 0D5C  
PRSUM 0D2D 08C3  
PKSW 0651 0860 0974 097B 0985 09A1 09A5 09D9 09E0 09F4 0D7A  
PSSCT 0646 0D2D 0D4C  
RAD 05DE 085C 08C9 08CD  
RCKX 0A68 0A5F 0A63  
RCKX1 0A66 0A6A  
RDCHK 0612 0993  
RDCMP 0635 0BEE 0C15 0C22 0C45 0C67 0C76 0C9F 0CC0  
RDCNT 064A 0C05  
RDCPX 0C2B 0C02 0C44  
RDDSW 0634 0BED 0C14 0C2D 0C2E  
RDEN 08EB 0683  
RDER1 0C22 0C13  
RDER2 0C2D 0C10  
RDER3 0C39 0C4A  
RDER4 0C45 0C35  
RDNCK 0C20 0C0D  
RDR2A 0C32  
RDR2B 0C34 0C31  
READ 0682 0890 08F0 0976 097D 0994 09DB 09E2 0A06 0A24 0A38 0B8A 0BF5 0C03  
0C1E 0C20  
READA 0C07 0C19 0C3C  
READB 0C09 0C26 0C38 0C49  
READX 0C1E 0C16 0C42  
RID 05DD 085F 08BF 08C1 08C5 08CF 09E9 0DDD  
RNDCK 0659 0BFA 0C59  
KNDME 0D01 0686  
RNDOM 0685 0943 09CF 0CB6 0D05  
RNDSK 05EE 093D  
RNDWR 05EF 09C1 09C5  
RQKB 01BC  
RQTY 01BB 083B  
RTCNT 0664 0B40 0B6F 0B89 0BD2  
RTNER 0667 08CC 0A3F 0A51 0A61 0A65  
RTNSW 0165  
RTN1 08E5 08D7  
RTN1A 08E9 08E8 090E

2310 B DISK FUNCTION TEST

RTN1B 08EC 0902 090A  
 RTN1C 08ED 08F7  
 RTN1D 08F8 08F4  
 RTN1E 0908 08E9 0905  
 RTN10 09B9 08E0  
 RTN11 09FC 08E1  
 RTN12 0A12 08E2  
 RTN13 0A6C 08E3  
 RTN2 0919 08D8  
 RTN2P 0926 091E 0924 0938  
 RTN2Q 092A 0937  
 RTN2R 092E 092C 0930  
 RTN2S 0930 092F  
 RTN2T 0934 0927 092A 0932  
 RTN2U 0936 0935  
 RTN3 093C 08D9  
 RTN3A 0942 0949 094B  
 RTN3B 0944 0941  
 RTN3C 0948 0946 0951 096B  
 RTN4 094D 08DA  
 RTN5 0954 08DB  
 RTN6 095A 08DC  
 RTN7 095C 08DD  
 RTN8 098B 08DE  
 RTN8A 09A2 0999 099F  
 RTN8J 09B0 0997  
 RTN8L 0990 09A3  
 RTN8M 099D 09B2  
 RTN8N 09A0 099E  
 RTN9 09B3 08DF  
 RTN9A 09CD 09D1  
 RTN9B 09D8 09D7  
 RTN9C 09DD 09CB  
 RTN9D 09DF 09F9  
 RTN9E 09E4 09CC  
 RTN9F 09E6 09E5  
 RTN9G 09F3 09DE  
 RTN9L 09C5 09E7  
 RTRYA 0632 0C08 0C24 0C36 0C41  
 RTRYB 0633 0C47  
 RTTBL 08D6 08C2 08C7 08D6 08E4  
 RT11A 0A05 0A04  
 RT11B 0A0A 09FF 0A09  
 RT11C 0A0F 0A0E  
 RT11L 0A00 0A10  
 RT12A 0A2C 0A32  
 RT12B 0A33 0A2F  
 RT12D 0A49 0A3D 0A48  
 RT12F 0A55 0A4F  
 RT12G 0A51 0A4C  
 RT12L 0A17 0A56  
 RT12Z 0A15  
 RT13A 0A7F 0A8F 0A97  
 RT13B 0A81 0A88  
 RT13C 0A9E 0A9A  
 RT13D 0AA1 0AC8  
 RT13E 0AA9 0ABD  
 RT13F 0AB4 0AB0  
 RT13G 0ABA 0AB6  
 RT13H 0AC9 0ADO  
 RT13J 0ACC 0A73  
 RT13K 0A98 0A92  
 RT910 09BE 09B8  
 R12CK 0A5D 0A20 0A27 0A41 0A52 0A66 0A68  
 S#B 065F 0C52 0C54 0C57 0CB5 0CB7 0CB8 0CBA 0CBB 0CBC 0CBD OCC6  
 SCTOR 0910 08ED 08EE  
 SECTD 09D6 0960 096E 0975 097C 09BE 09D3 09DA 09E1  
 SEEK 085E 0857 08B9 08DA

2310 B DISK FUNCTION TEST

SEEKB 0B6A 0B65  
 SEEKD 0B6C 0B69  
 SEEKE 0B6E 0B70  
 SEEKF 0B75 0B7C  
 SEEKG 0B80 0B78 0B7D  
 SETV 0688 0A1A 0C0A 0CED 0D26  
 SETVA 0D22 0D25  
 SETVE 0D1F 0689  
 SFTRD 0648 0C1A  
 SFTSK 0648 0BB5  
 SFTWR 064E 0CFD  
 SKCNT 0647 0B70  
 SMLNG 000C 0814  
 SNRES 0613 0862  
 SNSTY 061E 0832 0834 0838 0843 0850 0851 0A78  
 SNXIO 061A 0863 08AB 0B04 0B16 0B76 0B80 0886 0DE8 0DEC 0DED 0DF3  
 SPRTN 08E4 08D0 08E4  
 START 0161 060A 0866 0B34  
 STMAD 0D69 0D34  
 STML 0B3E 0B32  
 STMLE 0B30 068C  
 STMLR 0B36 0B30 0B31 0B3E  
 STMLS 0688 082E 0B0D 0B20 0B3C 0B75  
 STMSA 0D7F 0D75  
 STMSG 0D67 068F  
 STMSG 068E 0608 086B 0884 0897 08D3 0988 09AD 09F7 0A21 0A28 0A42 0A53 0A75  
 0A9C 0AC4 0ACE 0B09 0B1C 0B27 0884 0BB3 0BC6 0BCC 0BCF 0BDB 0BE7  
 0C1C 0C27 0C32 0C3D 0C6A 0C79 0CA2 0CDO 0CFB 0D2B 0D54 0D69 0D78  
 0D89  
 STMSX 0D83 0D67 0D68 0D7B 0DCF 0DD4 0DDC  
 STDRG 0911 08F8  
 SUMRY 0646 0816 0986 09F5 0B70 0BB5 0BBA 0C05 0C1A 0C29 0C3F 0CDE 0CFD 0D2F  
 0D5E 0D61  
 SVKB 01BD  
 SW0 05DF 0857 08B9 0D0D 0D17 0D28 0D2A 0DCD  
 SW1 05E0 08BC  
 SW2 05E1 081C  
 SW3 05E2  
 TB 065C 0606 0608 081A 081C 0823 0826 0827 082E 0834 083A 083D 0840 0843  
 0844 0845 0846 084B 084E 084F 0850 0851 0855 0857 085A 085B 085C  
 085D 085F 0860 0861 0862 0863 0864 0865 0869 086B 086D 086F 0870  
 0871 0872 0875 0876 087A 0880 0881 0884 0886 0889 088C 088F 0893  
 0897 0899 08A9 08AA 08AB 08AC 08AE 08B7 08B9 08BC 08BF 08C0 08C1  
 08C9 08CC 08CF 08D3 08D5 08E6 08EA 08EF 08F3 08FA 090F 091D 0923  
 0925 092D 0933 093D 093F 0940 0942 0943 0944 0947 094C 094F 0957  
 0964 0966 0969 096A 096F 0973 0974 0976 097B 097D 0983 0985 0987  
 0988 098B 098F 0990 0992 0993 0994 0998 099D 09A0 09A1 09A5 09A6  
 09A7 09A8 09A9 09AA 09AD 09AF 09B3 09B7 09B9 09BC 09C1 09C5 09CF  
 09D4 09D8 09D9 09DB 09E0 09E2 09F2 09F4 09F6 09F7 09FC 0A01 0A06  
 0A0B 0A11 0A12 0A18 0A19 0A1A 0A21 0A28 0A2E 0A37 0A3B 0A3C 0A3F  
 0A40 0A42 0A44 0A45 0A4A 0A4B 0A51 0A53 0A5B 0A5C 0A5E 0A61 0A62  
 0A65 0A6C 0A6E 0A6F 0A70 0A71 0A75 0A77 0A78 0A79 0A7A 0A7B 0A7C  
 0A7F 0A80 0A81 0A82 0A83 0A89 0A8D 0A91 0A94 0A95 0A96 0A98 0A9C  
 0AA1 0AA2 0AA3 0AA6 0AA8 0AAC 0AAD 0AAE 0AB3 0AB4 0AB9 0ABE 0ABF  
 0ACO 0AC3 0ACD 0ACE 0AD2 0AD5 0AE1 0AE6 0AEC 0AED 0AEE 0AEF 0AF0  
 0AF1 0AF2 0AF4 0AF5 0AF9 0AFD 0AFE 0AFE 0AFF 0B00 0B01 0B04 0B05  
 0B06 0B09 0B0B 0B0C 0B0D 0B12 0B15 0B16 0B17 0B18 0B19 0B1C 0B1E  
 0B1F 0B20 0B21 0B27 0B29 0B2A 0B2B 0B2C 0B2D 0B33 0B36 0B3F 0B40  
 0B4A 0B4C 0B5E 0B60 0B61 0B62 0B6A 0B6B 0B6C 0B6D 0B6E 0B6F 0B71  
 0B72 0B73 0B74 0B75 0B76 0B80 0B81 0B84 0B86 0B88 0B8D 0B91 0B99  
 0BAA 0BAB 0BAC 0BAE 0BAF 0BB0 0BB3 0BB6 0BBB 0BBC 0BC0 0BC3 0BCC  
 0BCE 0BCF 0BD2 0BD3 0BD9 0BD8 0BDF 0BE0 0BE2 0BE3 0BE7 0BE8 0BED  
 0BEE 0BEF 0BF0 0BF2 0BF4 0BF8 0BFA 0BFD 0BEF 0C00 0C06 0C07 0C08  
 0C09 0C0A 0C0B 0C0C 0C0F 0C14 0C15 0C18 0C1B 0C1C 0C27 0C2A 0C2D  
 0C30 0C32 0C39 0C3A 0C3B 0C3D 0C40 0C41 0C4D 0C4E 0C4F 0C50 0C51  
 0C52 0C53 0C54 0C56 0C57 0C59 0C5F 0C60 0C63 0C64 0C67 0C68 0C73  
 0C76 0C77 0C79 0C7B 0C7C 0C8D 0C8E 0C92 0C94 0C95 0C96 0C9B 0C9C  
 0CA2 0CA4 0CA5 0CAD 0CAE 0CB5 0CB6 0CB7 0CB8 0CBA 0CBB 0CBC 0CRD

2310 B DISK FUNCTION TEST

OCC0 OCC1 OCC5 OCC8 OCD5 OCD7 OCD8 OCDA OCDC OCDD OCDF OCE5 OCE9  
 OCED OCEE OCEF OCF0 OCF1 OCF6 OCF9 OCFA OCFB OCFE OD01 OD04 OD0D  
 OD17 OD28 OD29 OD2A OD2B OD4C OD4D OD4F OD51 OD53 OD58 OD60 OD63  
 OD6C OD6D OD72 OD77 OD7A OD7E OD85 OD95 OD97 OD98 OD99 OD9A OD9C  
 OD9F ODA2 ODA3 ODA5 ODA8 ODAB ODAC ODAD ODB1 ODB2 ODB4 ODB5 ODB6  
 ODB7 ODC2 ODC4 ODC5 ODC6 ODC7 ODCC ODCE ODD7 ODDD ODE6 ODE8 ODEA  
 ODEC ODED ODEF ODF3

TBDSW 0627 0605 08AC OA71 OA98 0B05 0B17 0B28 0B81  
 TEN 063C OA79 ODEF  
 TERM 05E7 0969 0AA3 0B6A 0BE2 0C09 OCEF 0D9F  
 TEXT 0691 0886 0899 08D5 0B29 0B8C 0BCE  
 THOUS 0644 0AEC 0AF2 0AF5 0DE6  
 TIMCN 066E 0ADB  
 TIMIT 0AD1 0A84 0AF7  
 TIMOT 0ADF 0ADF  
 TIMSL 0670 0A7A 0AEE 0AF1  
 TIM2 0AD7 0AD9 0ADD  
 TLP13 066C 0A77  
 TSTCC 0B58 0B56  
 TSTCD 0B5C 0B59  
 TSTCF 0B5A 0B46 0B4F 0B5D 0B89 0B90 0B94 0B96 0B9C  
 TYPE 061C 0845  
 TYRED 0614 061C  
 VERFA 0B4C 0B52  
 VERFB 0B86 0B63 0B82 0BCB  
 VERFC 0B88 0B93  
 VERFD 0B96 0B9F  
 VERFE 0BA1 0BA9  
 VERFF 0BBD 0B9A 0BA6  
 VERFG 0BBE 0BD1  
 VERFH 0BCF 0BC5  
 VERFI 0BD2 0BB1  
 VERFJ 0BD9 0BD4  
 VERFX 0BDD 0B41 0BD7  
 VERFY 0694 0889 088C 08E6 092D 0933 0947 0966 0988 09B3 09B9 09FC 0A12 0B43  
 0B47 0BE5 0BE9  
 VRFYE 0B3F 0695  
 WDCNT 08F2 08EB 08FE 0900 0909  
 WRCNT 064D 0CDE  
 WRERR 0666  
 WRITE 0697 096F 09D4 0A01 0A0B 0A1C 0A45 0CE0 0CE2 0CF2 0CF4 0CF7  
 WRLNG 0650 0A5B  
 WRMOD 0610 0CD5  
 WRRTY 0665 0BB7  
 WRTA 0CEE 0CEA  
 WRTE 0CFE 0CE7  
 WRTEN 0CD5 0698  
 XEQ 069A 0872 0876 0A70 0B13 0B2E 0B6D 0B7D 0C0B 0CEE  
 XEQA 0AFA 0828 0AD3  
 XEQB 0B12 0B02 0B07  
 XEQC 0B1E 0B1A 0B7F  
 XEQE 0AF9 069B  
 XEQLP 0B20 0B26  
 XEQX 0B2A 0B22  
 XSKBK 060C 086D 0A6C 0B5E  
 ZCNT 0663 08AA 08B1 0B0C 0B0E 0B1F 0B24 0B74 0B7A  
 ZIPA 0814 05E4  
 ZIPB 082A 082F 0836  
 ZIPC 0830 082C  
 ZIPD 0846 084A  
 ZIPE 084F 0847  
 ZIPF 0851 0853  
 ZIPL 0816 0819  
 ZLPA 0855 05E3  
 ZLPB 0866 0858  
 ZSNS 0616 0600 0601 0604 0844 0865 08F3 0998 0A40 0A5E 0AE1 0B2A 0BE3 0C0F  
 0CF6  
 ZXIO 0618 0871 09A6 0A6F 0A7C 0A81 0A83 0A89 0A8D 0A96 0AD5 0AFC 0AFE 0B15

2310 B DISK FUNCTION TEST

OB2C OB6C OB72 OBF0 OC39 OCD8 OCF9  
 END OF ASSEMBLY

----- LAST PAGE -----

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. REQUIREMENTS . . . . .	1
2.1 PROGRAM REQUIREMENTS	
2.2 EQUIPMENT REQUIREMENTS	
3. USE PROCEDURE . . . . .	1A
3.1 LOADING AND OPERATING	
3.2 WAITS	
4. PRINTOUTS (NONE)	
5. COMMENTS . . . . .	2
6. APPENDIX (NONE)	

1. PURPOSE

THE 1130 RELOCATING DIAGNOSTIC LOADER IS USED TO LOAD THE DIAGNOSTIC MONITOR AND PROGRAMS WHICH RUN UNDER CONTROL OF THE DIAGNOSTIC MONITOR. THE LOADER ALSO LOADS NON-MONITOR PROGRAMS WHOSE OBJECT DECKS ARE IN THE SAME FORM AS THE OUTPUT OF THE 1130 ASSEMBLER (12-4 FORMAT). (THE 1130 RELOCATING DIAGNOSTIC LOADER WILL NOT LOAD PROGRAMS WHOSE OBJECT DECKS ARE COMPATIBLE WITH THE 1130 BASIC DIAGNOSTIC LOADER)

2. PREREQUISITES

2.1 PROGRAM PREREQUISITES.

THIS LOADER USES THE FIRST 340 WORDS OF STORAGE.

2.2 EQUIPMENT PREREQUISITES

1. 1130 CPU
2. PROGRAM LOAD FROM CARD READER

3. OPERATING PROCEDURE

3.1 TYPES OF PROGRAMS THAT MAY BE LOADED.

THE RELOCATING DIAGNOSTIC LOADER IS DESIGNED PRIMARILY FOR LOADING THE 1130 DIAGNOSTIC MONITOR II AND RELOCATABLE DIAGNOSTIC PROGRAMS WHICH RUN UNDER CONTROL OF MONITOR II. THE FOLLOWING TYPES OF PROGRAMS CAN ALSO BE LOADED USING THIS LOADER.

1. ANY ABSOLUTE PROGRAM ASSEMBLED IN THE STANDARD 1130 ASSEMBLY FORMAT SO LONG AS THAT PROGRAM DOES NOT LOAD INTO ANY CORE LOCATIONS LOWER THAN 7160.
2. ANY RELOCATABLE PROGRAM.
3. A HEXIDECIMAL OBJECT DECK OR HEXIDECIMAL PATCH CARDS PUNCHED IN THE FORMAT DESCRIBED IN SECTION 5.2. SUCH PROGRAMS CAN LOAD INTO CORE LOCATIONS 0-39 IF THE PROGRAM IS LOADED WITH INTERRUPT DELAY ON. THIS ALLOWS INTERRUPT TRANSFER VECTORS TO BE SET DURING LOADING. ALSO INDEX REGISTERS ARE NOT USED BY THE HEX LOADER, THEREFORE, THESE MAY BE SET DURING LOADING. THIS FACILITY ALLOWS MANY PRESENTLY WRITTEN KEY IN ROUTINES TO BE PUNCHED INTO CARDS AND LOADED WITH VERY LITTLE MODIFICATION TO THESE ROUTINES.

3.2 TO LOAD A PROGRAM-

1. AT CARD READER

- A. DEPRESS NPRO PUSHBUTTON TO EJECT ANY CARDS LEFT IN MACHINE.
- B. PLACE RELOCATING LOADER IN FRONT OF DECK(S) TO BE LOADED AND PLACE CARDS IN HOPPER.
- C. DEPRESS START PUSHBUTTON. READY INDICATOR SHOULD LIGHT.

2. AT 1131 CONSOLE

- A. SET DATA ENTRY SWITCHES AS REQUIRED BY PROGRAM TO BE LOADED.
- B. DEPRESS RESET PUSHBUTTON.
- C. DEPRESS PROGRAM LOAD PUSHBUTTON. (PROGRAM SHOULD LOAD)

3. TRANSFER TO PROGRAM

- A. IF THE PROGRAM DECK IS TERMINATED BY A BLANK CARD, THE LOADER WILL AUTOMATICALLY TRANSFER TO THE ADDRESS ON THE END OF PROGRAM CARD.
- B. IF THERE IS NO BLANK CARD ON THE END OF THE DECK, THE CARD READER WILL GO NOT READY AND THE LOADER WILL STOP AT WAIT 30F8. TO CONTINUE, PRESS READER START THEN 1131 START.

3.3\*\*\* PROGRAM HALTS

HALT NO. (B REG.)	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON FIRST CARD OF LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F3	CARD 2 OF LOADER DID NOT LOAD	RELOAD
30F4	CAN NOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F5	READER CHECK WHEN LOADING MONITOR OR TEST PROGRAM	NPRO THEN PLACE CARDS RUN OUT IN FRONT OF REMAINING DECK AND PRESS START.
30F6	PROGRAM DID NOT LOAD	RELOAD
30F7	CHECK SUM WHEN LOADING PROGRAM	RELOAD
30F8	READER NOT READY	MAKE READER READY

\*\*\*\*\*

4. PRINTCUTS (NONE)

5. COMMENTS

5.1\*\*\* THE FOLLOWING ARE THE MAJOR ELEMENTS OF THE 1130 RELOCATING DIAGNOSTIC LOADER-

- 5.1.1 BOOTSTRAP ROUTINE -- IS A SET OF INSTRUCTIONS ENTERED INTO THE 1130 CPU CONTROLLED BY THE IPL (INITIAL PROGRAM LOADER) MODE WHOSE FUNCTION IS TO READ IN THE REMAINDER OF THE LOADER.
- 5.1.2 READ ROUTINE -- CHECKS READER FOR PROPER STATUS, READS A CARD INTO LOCATION /0028 THROUGH /0078, CHECKS FOR SATISFACTORY COMPLETION OF THE READ OPERATION, AND DETERMINES WHETHER THE CARD READ IS A BINARY CARD OR A HEXADECIMAL CARD.
- 5.1.3 BINARY PACK ROUTINE -- COMPUTES CHECKSUM OF A BINARY CARD, WAITS IF CHECKSUM IS IN ERROR.
- 5.1.5 MOVE ROUTINE -- MOVES DATA FROM /0028 THROUGH /0050 TO PROPER CORE LOCATION. CHECKS FOR EXCEEDING CORE SIZE. ADDS IN RELOCATION FACTOR WHEN REQUIRED.
- 5.1.6 RELOCATABLE HEADER ROUTINE -- ENTERED WHEN A RELOCATABLE HEADER CARD IS FOUND. COMPUTES A RELOCATION FACTOR FOR PROGRAM THAT FOLLOWS.
- 5.1.7 ABSOLUTE HEADER ROUTINE -- ENTERED WHEN AN ABSOLUTE HEADER CARD IS FOUND. SETS RELOCATION FACTOR TO ZERO.
- 5.1.8 TRANSFER ROUTINE -- ENTERED WHEN A TRANSFER CARD IS FOUND. COMPUTES THE NEXT LOCATION AVAILABLE FOR LOADING IF ANOTHER PROGRAM FOLLOWS. TRANSFERS CONTROL TO THE LOCATION SPECIFIED ON THE TRANSFER CARD.

5.1.9 HEX TO BINARY CONVERSION ROUTINE -- CONVERTS A HEXADECIMAL CARD TO BINARY. ADDS IN RELOCATION FACTOR IF REQUIRED.

5.2\*\*\* CARD FORMAT

THE FOLLOWING DESCRIBES THE FORMAT OF CARDS WHICH CAN BE LOADED BY THE 1130 RELOCATING DIAGNOSTIC LOADER.

- 5.2.1 ABSOLUTE HEADER CARDS HAVE A 1 PUNCH IN COLUMN 4.
- 5.2.2 RELOCATABLE HEADER CARDS HAVE A 0 (ZERO) PUNCH IN COLUMN 4.
- 5.2.3 NORMAL DATA CARDS HAVE NO PUNCHES IN ROW 12 IN COLUMN 1. AN ADDRESS IN ROWS 11 THROUGH 9 IN COLUMN 1 AND ROWS 2 AND ROWS 12 THROUGH 5 OF COLUMN 3. A 12, 0 PUNCH IN COLUMN 4, A WORD COUNT IN ROWS 4 THROUGH 9 OF COLUMN 4. A RELOCATION FIELD (WHICH MAY BE BLANK) IN COLUMN 5 THROUGH 12. DATA IN COLUMNS 13 THROUGH 72. A SEQUENCE NUMBER IN COLUMNS 73-80.
- 5.2.4 BINARY TRANSFER CARDS HAVE 12,11,0, 1 PUNCHES IN COLUMN 4 AND A WORD COUNT OF ZERO (NO PUNCHES IN ROWS 4 THROUGH 9 IN COLUMN 4).
- 5.2.5 HEXADECIMAL TRANSFER CARDS HAVE A 12 PUNCH IN COLUMN 1, A TRANSFER ADDRESS IN COLUMN 2 THROUGH 5 AND NO PUNCHES IN COLUMNS 6 AND 7.
- 5.2.6 HEXADECIMAL DATA CARDS HAVE A 12 PUNCH IN COLUMN 1. THE ADDRESS AT WHICH THE DATA IS TO BE LOADED IN COLUMN 2 THROUGH 5. DATA IN COLUMNS 6 THROUGH 75. DATA IS GROUPED 5 COLUMNS TO ONE CORE WORD. THE FIRST COLUMN OF EACH GROUP SPECIFIES WHETHER OR NOT THE GROUP REQUIRES A RELOCATION FACTOR. IF THE FIRST COLUMN OF A GROUP IS BLANK, A RELOCATION FACTOR WILL NOT BE ADDED. IF THE FIRST COLUMN OF A GROUP CONTAINS AN R (11,9 PUNCH) A RELOCATION FACTOR WILL BE ADDED TO THE FIELD. LOADING OF THE CARD IS TERMINATED BY TWO SEQUENTIAL BLANK COLUMNS. COLUMNS 76 AND 77 MUST BE BLANK. A SEQUENCE NUMBER MAY BE PUNCHED IN COLUMNS 78-80. ALSO IF ONLY A PORTION OF THE CARD IS USED FOR DATA AFTER TWO BLANK COLUMNS ANY COMMENT MAY BE ADDED TO THE CARD.

RELOCATING LOADER - 1442

028C	ABS	3AA00000
	ORG /0000	3AA00010
	----- 1130 LOADER CARD 1 -----	3AA00020
	* LOAD WITH PROGRAM LOAD BUTTON	3AA00030
		3AA00040
0000 0 C02C	START LD RDIN+1	3AA00050
0001 0 1802	SRA 2	3AA00060
0002 0 D02A	STO RDIN+1	3AA00070
0003 0 C023	LD STRE	3AA00080
0004 0 1801	SRA 1	3AA00090
0005 0 D021	STO STRD	3AA00100
0006 0 F038	EOR STORE	3AA00110
0007 0 D037	STO STORE	3AA00120
0008 0 C022	LD SENSE	3AA00130
0009 0 1803	SRA 3	3AA00140
000A 0 D020	STO SENSE	3AA00150
000B 0 F01D	EOR RESET	3AA00160
000C 0 D01C	STO RESET	3AA00170
000D 0 1805	SRA 5	3AA00180
000E 0 F031	EOR STORE+1	3AA00190
000F 0 D030	STO STORE+1	3AA00200
0010 0 C017	LD INTAD	3AA00210
0011 0 D0F6	STO /0008	3AA00220
0012 0 D0F9	STO /000C	3AA00230
		3AA00240
0013 0 C016	STRT LD CHKSM	3AA00250
0014 0 8000	A *	3AA00260
0015 0 D014	STO CHKSM	3AA00270
0016 0 C0FD	LD STRT+1	3AA00280
0017 0 800E	A K0001	3AA00290
0018 0 D0FB	STO STRT+1	3AA00300
0019 0 F00B	EOR CON.	3AA00310
001A 0 4820	BSC Z	3AA00320
001B 0 70F7	MDX STRT	3AA00330
001C 0 C000	LD CHKSM	3AA00340
001D 0 4820	BSC Z	3AA00350
001E 0 30F1	WAIT -15	3AA00360
001F 0 7010	ENDCK MDX	3AA00370
		3AA00380
0020 0 8823	INT DC /B823	3AA00390
0021 0 0806	XIO RESET-1	3AA00400
0022 0 48F8	DC /48F8	3AA00410
0023 0 0803	K8003 DC /0803	3AA00420
0024 0 700E	MDX PACK	3AA00430
		3AA00440
0025 0 8039	CON1 A X /0039	3AA00450
0026 0 0001	K0001 DC /0001	3AA00460
0027 0 2808	STRD DC /2808	3AA00470
0028 0 0020	INTAD DC INT	3AA00480
0029 0 0003	RESET DC /C003	3AA00490
002A 0 3829	CHKSM DC /3829	3AA00500
002B 0 8800	SENSE DC /B800	3AA00510
002C 0 0000	RDIN DC /0000	3AA00520
002D 0 4800	DC /4800	3AA00530
		3AA00540
002E 0 F017	EPROR EOR K8000	3AA00550
002F 0 30F2	WAIT -14	3AA00560
		3AA00570
0030 0 08F5	SRTRD XIO STRD-1	3AA00580
0031 0 08F6	XIO RESET-1	3AA00590
0032 0 08F7	XIO SENSE-1	3AA00600
0033 0 F011	PACK EOR K8003	3AA00610
0034 0 4820	BSC Z	3AA00620
0035 0 7011	MDX CONT1	3AA00630
0036 0 08F5	XIO RDIN	3AA00640
0037 0 C0F4	LD RDIN	3AA00650
0038 0 F0ED	EOR K0001	3AA00660
0039 0 D0F2	STO RDIN	3AA00670

RELOCATING LOADER - 1442

003A 0 4820	BSC Z	SKIP BOTH HALVES IN	3AA00680
003B 0 70F5	MDX SRTRD+1	GET 2ND HALF WORD	3AA00690
003C 0 C0C4	LD START+1	GET LAST 8 BITS	3AA00700
003D 0 1808	SRA B	SHIFT IT	3AA00710
003E 0 F0C1	EOR START	GET FIRST 8 BITS	3AA00720
003F 0 C004	STORE DC /C004	FIRST WORD OF STO L	3AA00730
0040 0 00F7	DC /00F7	2ND WORD OF STORE LONG	3AA00740
	* STORE + STORE +1 CHANGED BY PROG9 TO STO L /U04F		3AA00750
0041 0 C0FE	LD STORE+1		3AA00760
0042 0 80E3	A K0G01	MODIFY STORE ADDRESS	3AA00770
0043 0 D0FC	STO STORE+1		3AA00780
0044 0 70EC	MDX SRTRD+1		3AA00790
			3AA00800
0045 0 8003	K8003 DC /8003		3AA00810
0046 0 0800	K8000 DC /0800		3AA00820
			3AA00830
0047 0 F003	CONT1 EOR K8000	CHECK FOR BITS 14+15 ONLY	3AA00840
0048 0 4820	BSC Z	SKIP BUSY AND NOT READY	3AA00850
0049 0 7002	MDX CONT2		3AA00860
004A 0 70E7	MDX SRTRD+2		3AA00870
004B 0 8000	K8000 DC /8000		3AA00880
			3AA00890
004C 0 F0D6	CONT2 EOR K8003	CHECK FOR BIT 4 ONLY	3AA00900
004D 0 4820	BSC Z	SKIP END OF CARD	3AA00910
004E 0 70DF	MDX ERROR		3AA00920
0C4F 0 30F3	WAIT -13	**ERR IF PRGM STOPS AT WAIT	3AA00930
			3AA00940
			3AA00950
			3AA00960
			3AA00970
			3AA00980
			3AA00990
0028	INPUT EQU /28	INPUT AREA	3AA01000
			3AA01010
			3AA01020
			3AA01030
			3AA01040
			3AA01050
			3AA01060
0050	DRG /004F		3AA01070
004F 00 74000074	MDX L FIRST		3AA01080
0051 0 7003	MDX CNT		3AA01090
			3AA01100
0052 0 6379	LDX 3 /79		3AA01110
0053 0 68EC	STX 3 /40		3AA01120
0054 0 681F	STX FIRST		3AA01130
			3AA01140
0055 00 74FF0040	CNT MDX L /40,-1		3AA01150
0057 00 74FF0075	MDX L COUNT,-1		3AA01160
0059 0 70D6	MDX /0030	GO START READ	3AA01170
005A 0 C04B	LD INTE	SET INTERRUPT XFER	3AA01180
005B 0 D0AC	STO /B		3AA01190
005C 0 D0AF	STO /C		3AA01200
			3AA01210
005D 0 68A2	CORE STX 0		3AA01220
005E 00 6C007FFF	STX L /7FFF		3AA01230
0060 00 6700015C	LDX L3 END-4		3AA01240
0062 0 7302	CORE2 MDX 3 2		3AA01250
0063 0 10A0	SLT 32		3AA01260
0064 0 D802	STD 3 2		3AA01270
0065 00 74000000	MDX L /0000,0	SKIP IF ZERO CLEARED	3AA01280
0067 0 70FA	MDX CORE2		3AA01290
0068 00 C4007FFF	LD L /7FFF	CK IF LAST ADR CLEARED	3AA01300
006A 00 4C180071	BSC L CORE3,+	BR IF CLEARED	3AA01310
006C 00 6F00015F	STX L3 ULIM	FETCH THE LAST ADR	3AA01320
006E 00 2400015F	LD L ULIM	A = BIT THAT FAILED	3AA01330
0070 0 30F4	W30F4 WAIT -12	CAN NOT CLEAR CORE	3AA01340
		ADDRESSING BIT FAILED	3AA01350
0071 00 6F00015F	CORE3 STX L3 ULIM	SET THE CORE SIZE.	3AA01360

RELOCATING LOADER - 1442

```

0073 0 700A      MOX   RDCD   GO LOAD PROGRAM      3AA01360
*
0074 0 0000      FIRST DC  *--
0075 0 0007      COUNT DC  7
*
0076             ORG    /78
*****
*          1442 LOADER
*****
* THIS LOADER READS DATA FROM A CARD AND STORES
* THAT DATA IN LOCATIONS /40 TO /77
-----
0078 0 082D      LOAD  XIO   SENSN-1
0079 00 4C040156 BSC  L  MLCDE
0078 0 C02A      LD    INTE   SET INTERRUPT XFER
007C 0 D300      STO   3 0     XR3=MONITOR ILO
007D 0 D33E      STO   3 /3E
*
007E 0 C029      RDCD  LD    DSW    CK LAST CARD SW
007F 0 1003      SLA   3
0080 00 4C280155 BSC  L  FEED,+Z  GO TO MONITOR IF ON
0082 0 C022      RDCD1 LD   ADRS   SET INPUT ADRS
0083 0 D028      STO   READ
0084 0 D02A      STO   DATA
*
0085 0 0822      XIO   SENSR-1  SENSE DSW FOR READY
0086 0 4844      BOSC  E          SKIP IF READY
0087 0 30F8      WAIT  -8        WAIT (30F8) IF NOT READY
*
0088 0 0821      XIO   STRD1-1  START READ
*
0089 0 081E      RDCD2 XIO  SENSR-1  RESET DSW
008A 0 4878      BOSC  +-Z
*
008B 0 0000      INTO  DC  *--    SERVICE COLUMN INTERRUPT
008C 0 0819      RDCD3 XIO  SENSN-1  SENSE DSW FOR READY
008D 0 F01C      EOR   K0003
008E 00 4C18008C BSC  L  RDCD3,+  LOOP UNTIL CHANGE IN DSW
*
0090 0 18CD      RTE   13        LOOK AT ERR IND
0091 00 4C040153 BSC  L  W30F5,E  BR IF ON
0093 0 1082      SLT   2
0094 00 4C04009D BSC  L  OPCOM,E  BR IF OP COMPLETE
0096 0 100B      SLA   11
0097 00 4C10008C BSC  L  RDCD3,-  BR IF NO INTRPT RESPONSE
0099 0 0812      XIO   READ      READ COLUMN
009A 00 740100AC MDX  L  READ,+1  UPDATE READ IN AREA
009C 0 70EC      MDX   RDCD2
*
009D 0 080A      OPCOM XIO  SENSR-1  RESET DSW
009E 0 D009      STO   DSW      SAVE DSW
009F 0 C088      LD    INPUT
00A0 0 1001      SLA   1
00A1 00 4C5A0114 BOSC  L  HEX,C+- BR IF HEX PATCH CARD
00A3 00 4C4000B3 BOSC  L  SB05
*
00A5 0 0028      ADRS  DC  INPUT
00A6 0 0000      BSS   E
00A6 0 0088      INTE  DC  INTO   INTERRUPT TRANSFER ADRS
00A7 0 1700      SENSN DC  /1700  SENSE - NO RESET
00A8 0 0000      DSW   DC  *--    OP COMPLETE DSW
00A9 0 1703      SENSR DC  /1703  SENSE RESET
00AA 0 0003      K0003 DC /0003
00AB 0 1404      STRD1 DC /1404  START READ COMMAND
00AC 0 0000      READ  DC  *--    READ ONE COL COMMAND
00AD 0 1204      DC    /1204

```

DATE 01DEC66  
EC NO. 419643

PRG ID 03AA-2  
PAGE 2

RELOCATING LOADER - 1442

```

COAE 0 0001      ONE  DC  1
00AF 0 0000      DATA DC  *--
00E0 0 1084      SL   SLT  4
00B1 0 1088      SL   SLT  8
00B2 0 108C      SL   SLT 12
*
*****
*          CONVERT
*****
* THIS RT PACKS BINARY DATA AND LEAVES IT IN
* LOCATIONS 0028 - 0078.
-----
00B3 0 6188      SB05 LDX  1 -72
00B4 0 6300      LDX  3 0
00B5 0 62FD      SB06 LDX  2 -3
00B6 00 C60000B3 SB07 LD  L2 SL+3
00B8 0 D004      STO  SB10
00B9 0 C171      LD   1 INPUT+73 LD INPUT DATA
00BA 0 18D0      RTE  16
00BB 0 C170      LD   1 INPUT+72 LD REST OF WORD
00BC 0 1804      SRA  4
00BD 0 1080      SB10 SLT  *--    REQUIRED SHIFT SET HERE
00BE 0 D328      STO  3 INPUT  STO CONVERTED WORD
00BF 0 7301      MDX  3 1
00C0 0 7101      MDX  1 1
00C1 0 7201      MDX  2 1
00C2 0 70F3      MDX  SB07    CONVERT NEXT WORD
00C3 0 7101      MDX  1 1
00C4 0 70F0      MDX  SB06    SKIP WHEN FINISHED
*
*****
*          CHECK HEADER
*****
* THIS RT DETERMINES WHETHER THE DATA CARD IS
* 1) ABSOLUTE HDR CARD 2) RELOCATABLE HDR CARD
-----
00C5 0 C12A      LD   1 INPUT+2  CK FOR HDR CARDS
00C6 00 4C180156 BSC  L  MLCDE,+  BR TO USER - BLANK CD
00C8 0 E038      AND  LB20
00C9 0 9038      S    LB25
00CA 00 4C180108 BSC  L  ABHED,+  BCH IF ABSOL HEAD CARD
00CC 0 9038      S    LB25
00CD 00 4C180106 BSC  L  RLHED,+  BCH IF RELOC HEAD CARD
*
*****
*          CHECK SUM
*****
* THIS ROUTINE ADDS COLUMNS 0 - 72
* TO CHECK THAT THIS SUM PLUS THE CARD
* SEQUENCE NUMBER EQUALS ZERO.
-----
00CF 0 C129      LD   1 INPUT+1  LD CHECK SUM
00D0 00 4C1800DE BSC  L  CKEOP,+  SKIP CKSUM IF ZERO
*
00D2 0 C030      LD   DCCT
00D3 0 62CA      LDX  2 -54
00D4 0 825E      CKSM1 A  2 INPUT+54  ADD WORDS 1 TO 54
00D5 0 4802      BSC  C
00D6 0 8007      A    ONE
00D7 0 7201      MDX  2 1
00D8 0 70FB      MDX  CKSM1
*
00D9 0 80D4      A    FIVE
00DA 00 4C200158 BSC  L  CKSUM,Z  BR IF CK SUM ERR

```

DATE 01DEC66  
EC NO. 419643

PRG ID 03AA-2  
PAGE 2A



RELOCATING LOADER - 1442

```

00DC 00 74010103      MDX L  CDCT,1  ADD 1 TO CARD CT      3AA02720
*-----*
* CHECK FOR END OF PROGRAM CARD
*-----*
00DE 0  C12A      CKEOP LD  1 INPUT+2  GET WORD COUNT      3AA02730
00DF 0  1008      SRA      8          CLEAR CARD CODE      3AA02740
00E0 0  1808      SRA      8          CLEAR CARD CODE      3AA02750
00E1 0  D12A      STO      1 INPUT+2  SAVE WORD COUNT      3AA02760
00E2 00 4C18010C    BSC L  EOP,+--  EOP IF WC = ZERO      3AA02770
*-----*
* CHECK FOR OVER CORE
*-----*
00E4 0  8128      A      1 INPUT  SUM WC+STO ADDR+RELOC      3AA02780
00E5 0  8077      A      UPPER  RELOCATE STORE ADDR      3AA02790
00E6 0  9078      S      ULIM   RELOCATE STORE ADDR      3AA02800
00E7 00 4C10015A    BSC L  DVCR,-  BR IF OVER CORE      3AA02810
*-----*
*****
* RELOCATE AND STORE
*****
* THIS RT PLACES DATA FIELDS INTO THE CORRECT
* CORE LOCATIONS AND ADDS IN A RELOCATION
* FACTOR IF REQUIRED.
*-----*
00E9 0  C128      LD      1 INPUT  LD STORE ADDR      3AA02820
00EA 0  8072      A      UPPER  RELOCATE STORE ADDR      3AA02830
00EB 0  D00B      STO      STOR4+1  RELOCATE STORE ADDR      3AA02840
00EC 00 6600002B    LDX L2 INPUT+3  XR2 = RELOCATION CODE ADDR      3AA02850
*-----*
00EE 0  63F8      STOR1 LDX  3 -E  XR3= REL BIT CNT      3AA02860
*-----*
00EF 0  C200      LD      2 0      LD NEXT RELOCATION CODE      3AA02870
00F0 0  18D0      RTE      16      STO IN EXT REG      3AA02880
00F1 0  1082      STOR2 SLT  2      BRING IN NEXT REL BIT      3AA02890
00F2 00 4C040101    BSC L  STOR6,E  BR IF RELOCATE      3AA02900
*-----*
00F4 0  1810      SRA      16      LD STORE ADDR      3AA03000
00F5 0  8131      STOR3 A      1 INPUT+9  ADD NEXT WORD      3AA03010
00F6 00 D5000000    STOR4 STO L1 *-#  STO IN PROGRAM      3AA03020
00F8 00 74FF002A    MDX L  INPUT+2,-1  DECR WORD CNT      3AA03030
00FA 0  7001      MDX      STOR5    SKIP WHEN WC = 0      3AA03040
00FB 0  607E      LDX      RDCD    FINISHED      3AA03050
*-----*
00FC 0  7101      STOR5 MDX  1 1      ADV WORD AND STO ADRS      3AA03060
00FD 0  7301      MDX      3 1      ADV REL BIT CNT      3AA03070
00FE 0  70F2      MDX      STOR2    ADV REL BIT CNT      3AA03080
00FF 0  7201      MDX      2 1      ADV TO NXT RELOC WD      3AA03090
0100 0  70ED      MDX      STOR1    ADV TO NXT RELOC WD      3AA03100
*-----*
0101 0  C05B      STOR6 LD      UPPER  LD RELOC FACTOR      3AA03110
0102 0  70F2      MDX      STOR3    GO RELOC ADDR      3AA03120
*-----*
0103 0  0000      CDCT DC  *-#      LD STORE ADDR      3AA03130
0104 0  0F00      LB20 DC  /OF00    GO RELOC ADDR      3AA03140
0105 0  0100      LB25 DC  /0100    GO RELOC ADDR      3AA03150
*****
* HEADER AND EOP CARDS
*****
* THIS RT. HANDLES RELOCATABLE MDR CARDS.
*-----*

```

RELOCATING LOADER - 1442

```

0106 0  C055      RLHED LD  NLOC      COMPUTE RELOC FACTOR      3AA03400
0107 0  9056      S      RLBA      COMPUTE RELOC FACTOR      3AA03410
*-----*
* THIS RT. HANDLES ABSOLUTE MDR CARDS.
*-----*
0108 0  D054      ABHED STO  UPPER  INITIALIZE CARD COUNT      3AA03420
0109 0  7101      MDX  1 1          INITIALIZE CARD COUNT      3AA03430
010A 0  69F8      STX  1 CDCT      INITIALIZE CARD COUNT      3AA03440
010B 0  607E      LDX      RDCD    INITIALIZE CARD COUNT      3AA03450
*-----*
* THIS ROUTINE HANDLES END OF PROGRAM CARDS
*-----*
010C 0  C128      EOP  LD  1 INPUT  SET NEXT AVAIL LOC      3AA03460
010D 0  804F      A      UPPER  SET NEXT AVAIL LOC      3AA03470
010E 0  D04D      STO      NLOC   SET NEXT AVAIL LOC      3AA03480
010F 0  C128      LD      1 INPUT+3  LD TRANSFER ADDR      3AA03490
0110 0  804C      A      UPPER  LD TRANSFER ADDR      3AA03500
0111 0  D001      STO      EOP1+1  LD TRANSFER ADDR      3AA03510
0112 00 4C000000    ECPI BSC L *-#  TRANSFER TO PROGRAM      3AA03520
*-----*
*****
* HEX DATA ROUTINE
*****
* THIS ROUTINE CONVERTS HEX CORRECTION
* DATA TO BINARY AND STORES THE DATA.
*-----*
0114 0  D032      HEX  STO  HE22+1  CLEAR STORE ADDR      3AA03530
0115 00 448000AF    HEX1 LD  1 DATA  LOAD RELOCATION BIT      3AA03540
0117 0  D038      STO      RLREQ   LOAD RELOCATION BIT      3AA03550
0118 00 740100AF    MDX L  DATA,1  ADV TO NXT WORD      3AA03560
*-----*
* CONVERT HEX TO BINARY
*-----*
011A 0  C031      HE10 LD      FOUR  CLEAR STORE ADDR      3AA03570
011B 0  D032      STO      INDX1   LOAD RELOCATION BIT      3AA03580
011C 0  1810      SRA      16      STO RELOCATION BIT      3AA03590
011D 0  1004      HE11 SLA      4      ADV TO NXT WORD      3AA03600
011E 0  D02C      STO      TEMP    ADV TO NXT WORD      3AA03610
011F 0  1810      SRA      16      ADV TO NXT WORD      3AA03620
0120 0  D02E      STO      INDX2   ADV TO NXT WORD      3AA03630
0121 00 448000AF    LD  1 DATA      CK FOR BLANK COLUMN      3AA03640
0123 00 4C18007E    BSC L  RDCD,+--  FINISHED - LOAD NEXT CARD      3AA03650
*-----*
0125 00 4C100129    BSC L  HE11A,-  BR IF NOT A-F      3AA03660
0127 00 7409014F    MDX L  INDX2,9  ADD 9 FOR ALPHA      3AA03670
*-----*
0129 0  1003      HE11A SLA      3      ELIMINATE ZONE BITS      3AA03680
012A 00 4C180133    BSC L  HE14,+--  XF+R IF HEX CHAR = 0      3AA03690
012C 00 7401014F    MDX L  INDX2,1  XF+R IF HEX CHAR = 0      3AA03700
012E 00 4C280132    HE12 BSC L  HE13,+2  XFER IF BIT IS FOUND      3AA03710
0130 0  1001      SLA      1      PREPARE TO LK AT NEXT BIT      3AA03720
0131 0  70FA      MDX      HE12-2  PREPARE TO LK AT NEXT BIT      3AA03730
*-----*
0132 0  C01C      HE13 LD      INDX2  LOAD BINARY BITS      3AA03740
0133 0  E817      HE14 OR      TEMP  ADD TO PREVIOUS CHARS      3AA03750
0134 00 740100AF    MDX L  DATA,1  ADD TO PREVIOUS CHARS      3AA03760
0136 00 74FF014E    MDX L  INDX1,-1  ADD TO PREVIOUS CHARS      3AA03770
0138 0  70E4      MDX      HE11    ADD TO PREVIOUS CHARS      3AA03780
*-----*
* FINISHED - CONVERTED WORD IN ACCUMULATOR
*-----*

```

RELOCATING LOADER - 1442

0139 00 74000147	MDX L	HE22+1,0	SKIP IF STO ADRS	3AA04080
0138 0 7007	MDX	HE21	BR IF DATA	3AA04090
013C 0 8020	A	UPPER	RELOC STO ADRS	3AA04100
013D 0 D009	STO	HE22+1		3AA04110
013E 00 C400002E	LD L	INPUT+6	LD NXT WORD	3AA04120
0140 00 4C980147	BSC I	HE22+1,+	BR TO ADRS IF BLANK	3AA04130
0142 0 7002	MDX	HEX1	CONV NXT WORD	3AA04140
0143 00 74000150	HE21 MDX L	RLREQ,0	SKIP IF NO RELOCATE	3AA04150
0145 0 8017	A	UPPER		3AA04160
0146 00 D4000G00	HE22 STO L	*-*	STORE DATA	3AA04170
0148 00 74010147	MDX L	HE22+1,1	ADV STO ADRS	3AA04180
014A 0 70CA	MDX	HEX1	CONV NXT WORD	3AA04190
014B 0 000C	TEMP DC	*-*	TEMP STORAGE	3AA04200
014C 0 C000	BSS E	0		3AA04210
014C 0 0004	FOUR DC	4		3AA04220
014D 0 1402	FDCD DC	/1402	FEED CARD	3AA04230
014E 0 0000	INDX1 DC	*-*		3AA04240
014F 0 0000	INDX2 DC	*-*		3AA04250
0150 0 0000	RLREQ DC	*-*	HEX RELOCATION BITS	3AA04260
0151 0 1000	NOP			3AA04270
0152 0 1000	NOP			3AA04280
0153 0 30F5	W30F5 WAIT	-11	1442 ERROR IND ON	3AA04290
0154 0 607E	LXD	RDCD	TRY REREADING THE CARD	3AA04300
0155 0 08F6	FEED XIO	FDCD-1	FEED OUT LAST CARD	3AA04310
*****				
MONITOR/LOADER INTERFACE				
*****				
* THE FOLLOWING MUST BE LOCATED IN CORE				
* LOCATIONS /0156-/015F.				
*****				
0156 0 30F6	MLCD WAIT	-10	PROGRAM SHOULD PLACE HERE	3AA04320
0157 0 70FE	MDX	MLCD	A XFER ON BLANK CARD	3AA04330
0158 0 30F7	CYSUM WAIT	-9	CHECK SUM ERROR	3AA04340
0159 0 70FE	MDX	CKSUM		3AA04350
015A 0 3000	OVCR WAIT		EXCEEDED CORE SIZE	3AA04360
015B 0 70FE	MDX	OVCR		3AA04370
015C 0 0160	NLOC DC	/150	NEXT AVAILABLE STORAGE LOC	3AA04380
015D 0 0000	UPPER DC	*-*	RELOCATION FACTOR	3AA04390
015E 0 0000	RLBA DC	0000	BASE ADDRESS	3AA04400
015F 0 0001	ULIM BSS	1	CORE SIZE	3AA04410
*****				
0160 0000	END BSS E	0		3AA04420
0160 0078	END	LOAD		3AA04430

RELOCATING LOADER - 1442

CROSS REFERENCE LISTING

SYMBOL	VALUE	REFERENCES
ABHED	0108	00CA
ADRS	00A5	0082
CDC	0103	00D2,00DC,010A
CHKSM	002A	0013,0015,001C
CKEOP	00DE	00D0
CKSM1	00D4	00D8
CKSUM	0158	00DA,0159
CNT	0055	0051
CONT1	0047	0035
CONT2	004C	0049
CON1	0025	0019
CORE	005D	
CORE2	0062	0067
CORE3	0071	006A
CUUNT	0075	0057
DATA	00AF	0084,0115,0118,0121,0134
DSW	00A8	007E,009E
END	0160	0060
ENDCK	001F	
EDP	010C	00E2
EOP1	0112	0111
ERRDR	002E	004E
FDCD	0140	0155
FEED	0155	0080
FIRST	0074	004F,0054
FOUR	014C	011A
HEX	0114	00A1
HEX1	0115	0142,014A
HE10	011A	
HE11	011D	0138
HE11A	0129	0125
HE12	012E	0131
HE13	0132	012E
HE14	0133	012A
HE21	0143	0138
HE22	0146	0114,0139,013D,0140,0148
INDX1	014E	0118,0136
INDX2	014F	0120,0127,012C,0132
INPUT	0028	009F,00A5,00B9,00BB,00BE,00C5,00CF,00D4,00DE,00E1,00E4,00E9,00EC,00F5,00F8,010C,010F,013E
INT	0C20	0C28
INTAD	0028	0010
INTE	00A6	005A,007B
INTO	008B	00A6
K0001	0026	0017,0038,0042
K0003	00AA	008D
K0800	0046	002E
K0803	0023	004C
K8000	0048	0047
K8003	0045	0033
LB20	0104	00C8
LB25	0105	00C9,00CC
LOAD	0078	0160
MLCD	0156	0079,00C6,0157
NLOC	015C	0106,010E
ONE	00AE	00D5,00D9
OPCOM	009D	0094
OVCR	015A	00E7,015B
PACK	0033	0024
RDCD	007E	0073,00FB,010B,0123,0154
RDCD1	0082	
RDCD2	0089	009C
RDCD3	008C	008E,0097
RDIN	002C	0000,0C02,0036,0037,0039
READ	00AC	00B3,0099,009A
DATE	01DEC66	
EC NO.	419643	

IBM MAINTENANCE DIAGNOSTIC PROGRAM FOR THE 1130 SYSTEM

PART NO. 2191281  
PAGE 5

RELOCATING LOADER - 1442

RESET	0029	000B,000C,0021,0031
RLBA	015E	0107
RLHED	0106	00CD
RLREQ	0150	0117,0143
SB05	00B3	00A3
SB06	00B5	00C4
SB07	00B6	00C2
SB10	00BD	00B8
SENSE	002B	000B,000A,0032
SENSN	00A7	007B,008C
SENSR	00A9	00B5,00B9,009D
SL	00B0	00B6
SRTRD	0030	001F,003B,0044,004A
STAKT	0000	003C,003E
STOPE	003F	0006,0007,000E,00CF,0041,0043
STOR1	00EE	0100
STOR2	00F1	00FE
STOR3	00F5	0102
STOR4	00F6	00E8
STOR5	00FC	00FA
STOR6	0101	00F2
STRD	0027	0003,0005,0030
STRD1	00AB	00B8
STRT	0013	0017,0018,001B
TEMP	014B	011E,0133
ULIM	015F	006C,006E,0071,00E6
UPPER	015D	00E5,00EA,0101,0108,010D,0110,013C,0145
W30F4	0070	
W20F5	0153	0091

RELOCATING DIAGNOSTIC LOADER - 2501

```

028C      ABS
          ORG      0
          *
          START LD   BOUTSTRAP CARD =1
                SRA 1      8800
                EDR 1      4400
                STO PACK C400
                EDR 1      C400 LD L
                STO D400
                EDR 1      D400 STO L
                STO D400
                EDR 1      F400 EDR L
                SRA 4      /F40
                EDR K4001 4F41
                STO RESET 4F41 RESET IOCC
                MDX **1    4F41
                DC      INT
                SRA 8      004F
                STO STRD 004F START DR ADDRS
                A      ONE 0050
                STO EDR+1 0050 1ST HALF WD ADDRS
                STO STO+1 0050 1ST STORE ADDRS
                A      ONE 0051
                STO PACK+1 0051 2ND HALF WD ADDRS
                A      LAST 0078
                STO LAST 0078 END STORE ADDRS
                LD K4001 4001
                SRA 6      0100
                EDR RESET 4E01
                STO STRD+1 4E01 START RD 1/CC
                SRA 8      004E
                STO WC 004E WORD CGUNT
          *
          STRT LD   CHKSM FORM CHECK SUM, THIS CARD
                A    **6    FKDM 0024 THRU 004F
                STO CHKSM
                LD   STRT+1
                A    ONE    MODIFY ADD INST
                STO STRT+1
                EDR CON1    CHECK THAT LAST LOC CHKD
                BSC Z      SKIP IF FINISHED
                LDY  STRT   GO GET NEXT WORD
                LD   CHKSM GET SUM OF 0013 THRU 004F
                BSC Z      SKIP IF CHKSM ZERO
                WAIT -15    CHECK SUM ERROR
                LDX  SRTRD  START LOADING
          *
          *
          K0800 DC   /0800
          STRD DC   0
          K4001 DC   /4001
          LAST DC   /0026
          RESET DC   *-
          CHKSM DC   /0018
          CON1 A    X 80-STRT-3
          ONE DC    /0001
          *
          INT DC    0
          LDX  PACK
          *
          SRTRD XIO STRD
                XIO RESET-1
                BSC E
                LDX SRTRD+1
          *
          PACK DC   /8000 LD L RDIN+1
                DC   /8800
                SRA 8
                EDR DC   /2000 EDR L RDIN

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

0038 0 0000
003C 0 1000 STO DC /1090 STO L RDIN
003D 0 0000 DC 0
003E 0 C0F9 LD PACK+1
003F 0 80F0 A ONE
0040 0 D0FA STO EDR+1
0041 0 80FE A ONE
0042 0 D0F5 STO PACK+1
0043 0 C0F9 LD STO+1
0044 0 80EB A ONE
0045 0 D0F7 STO STO+1
0046 0 F0E5 EDR LAST
0047 0 4820 BSC Z
0048 0 6037 LDX PACK
0049 0 7006 END1 MDX CARD2
004A 0 004E ORG 79
004F 0 004E WC DC 78
0050 0 0000 RDIN BSS 0
0050 0 C0FE CARD2 LD WC
0051 00 D400009D STOWC STO L WC+78
0053 00 74+E002A MDX L STRD,+78 UPDATE START RD ADDR
0055 00 744E0052 MDX L STOWC+1,+78
0057 00 7427092C MDX L LAST,+39
0059 00 74FF005D MDX L COUNT,-1
005B 0 70D7 MDX SRTRD
005C 0 605E LDX CORE
005D 0 0007 COUNT DC 7
*
*
INPUT EQU /29 INPUT AREA
*
* NOTE..... THIS SECTION OVERLAYED.....
* BY READING THE FIRST CARD..
* OF FIRST PROGRAM.....
*
CORE STX L /7FFF
LD INTE INTERRUPT TR VECTOR
STO /C
LDX L3 END-4
CORE2 MDX 3 2
SLT 32
STD 3 2
MDX L /0000,0 SKIP IF ZERO CLEARED
MDX CORE2
006A 00 C4007FFF LD L /7FFF CK IF LAST ADR CLEARED
006C 00 4C180073 BSC L CORE3,+ BR IF CLEARED
006E 00 6F00015F STX L3 ULIM FETCH THE LAST ADR
0070 00 C400015F LD L ULIM A = BIT THAT FAILED
0072 0 30F4 W30F4 WAIT -12 CAN NOT CLEAR CORE
* ADDRESSING BIT FAILED
CORE3 STX L3 ULIM SET THE CORE SIZE
MDX RDCD GO LOAD PROGRAM
*
ORG /78
*****
2501 LOADER
*****
* THIS LOADER READS DATA FROM A CARD AND STORES
* THAT DATA IN LOCATIONS /40 TO /77
*
LOAD XIO SENSN-1
BSC L MLCDE
LD INTE SET INTEKRUPT XFER
STO 3 /3D
*
RDCD LD DSW CK LAST CARD SW
SLA 3

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

007F 00 4C280156      BSC L  MLC0,+Z  GO TO MONITOR IF ON      3AB01360
0081 0  C01B          LD      K004F                      3AB01370
C082 00 04000028      STO L  INPUT-1  WORD COUNT                3AB01380
0084 0  081B          XIO    SENSR-1  SENSE DSW FOR READY    3AB01390
0085 0  4844          ROSC   E          SKIP IF READY      3A=01400
0086 0  30F8          WAIT  -8        WAIT (30F8) IF NOT READY  3AB01410
*
0087 0  081A          *      XIO    STRD1-1  START READ          3AB01420
*
0088 0  0617          RDCD2 XIO  SENSR-1  RESET DSW          3AB01430
0089 0  4878          *      BOSC   +-Z                      3AB01440
*
008A 0  0000          INTO  DC  **      SERVICE COLUMN INTERRUPT  3AB01480
008B 0  0612          RDCD3 XIO  SENSR-1  SENSE DSW FOR READY    3AB01490
008C 0  F017          EOR    K0003                      3AB01500
008D 00 4C18008B      *      BSC L  RDC03,+  LOOP UNTIL CHANGE IN DSW  3AB01510
*
008F 0  18CD          RTE    13          LOOK AT ERR IND          3AB01530
0090 0C 4C04014F      BSC L  W30F5,E    BK IF ON                3AB01540
0092 0  1091          SLT   17          3AB01550
0093 00 4C10008B      *      BSC L  RDCD3,-  BR IF OP COMPLETE  3AB01560
*
0095 0  080A          OPCOM XIO  SENSR-1  RESET DSW          3AB01570
0096 0  0009          STO   DSW          SAVE DSW            3AB01580
0097 0  C091          LD    INPUT        3AB01590
0098 0  1001          SLA   1            3AB01600
0099 00 4C5A010E      *      BOSC L  HEX,C+-  BR IF HEX PATCH CARD  3AB01620
009B 00 4C4000AB      *      BOSC L  SB05                      3AB01630
*
009D 0  004F          K004F DC  /004F          3AB01640
009E 0  0000          BSS   E            3AB01650
009E 0  008A          INTE  DC  INTO     INTERRUPT TRANSFER ADDRS  3AB01670
009F 0  4F00          SENSN DC  /4F00      SENSE - NO RESET        3AB01680
00A0 0  0000          DSW   DC  **      OP COMPLETE DSW          3AB01690
00A1 0  4F03          SENSR DC  /4F03      SENSE RESET             3AB01700
00A2 0  0028          DC    INPUT-1      3AB01710
00A3 0  4E00          STRD1 DC  /4E00     INITIATE READ          3AB01720
00A4 C  0003          K0003 DC  /0003     3AB01730
*
00A5 0  0001          *      ONE2  DC  1            3AB01740
00A6 0  0029          ADRS  DC  INPUT    3AB01750
00A7 0  0000          DATA DC  **      3AB01760
00A8 0  1084          SL    SLT  4       3AB01770
00A9 0  1088          SL    SLT  8       3AB01780
00AA 0  108C          SL    SLT  12      3AB01790
*
*****
*      CONVERT
*****
* THIS RT PACKS BINARY DATA AND LEAVES IT IN
* LOCATIONS 0028 - 0078.
*****
00AB 0  6188          SB05  LDX  1 -72     3AB01800
00AC 0  6300          LDX  3 0            3AB01810
00AD 0  62FD          SB06  LDX  2 -3     3AB01820
00AE 00 C60000AB      SB07  LD   L2 SL+3   3AB01830
00B0 0  D004          STO   SB10          3AB01840
00B1 0  C172          LD    1 INPUT+73   LD INPUT DATA          3AB01850
00B2 0  18D0          RTE   16           3AB01860
00B3 0  C171          LD    1 INPUT+72   LD REST OF WORD        3AB01870
00B4 0  1804          SRA   4            3AB01880
00B5 0  1080          SB10  SLT  **      REQUIRED SHIFT SET HERE  3AB01890
00B6 0  D329          STO   3 INPUT      STO CONVERTED WORD     3AB01900
00B7 0  7301          MDX   3 1          3AB01910
00B8 0  7101          MDX   1 1          3AB01920
00B9 0  7201          MDX   2 1          3AB01930
00BA 0  70F3          MDX   SB07         CONVER3 NEXT WORD     3AB01940

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

00B5 0  7101          MDX   1 1          3AB02040
00B6 0  70F0          MDX   SB06         SKIP WHEN FINISHED  3AB02050
*
*****
*      CHECK HEADER
*****
* THIS RT DETERMINES WHETHER THE DATA CARD IS
* 1) ABSOLUTE HDR CARD 2) RELOCATABLE HDR CARD
*****
00BD 0  C12B          LD    1 INPUT+2     CK FOR HDR CARDS      3AB02140
00BE 00 4C180156      BSC L  MLC0,+  BK TO USER - BLANK CD  3AB02150
00C0 0  E03D          AND   LB20          3AB02160
00C1 0  903D          S     LB25          3AB02170
00C2 00 4C180102      BSC L  ABHD,+  BCH IF ABSOL HEAD CARD  3AB02180
00C4 0  903A          S     LB25          3AB02190
00C5 00 4C180100      BSC L  RLHD,+  BCH IF RELOC HEAD CARD  3AB02200
*
*****
*      CHECK SUM
*****
* THIS ROUTINE ADDS COLUMNS 0 - 72
* TO CHECK THAT THIS SUM PLUS THE CARD
* SEQUENCE NUMBER EQUALS ZERO.
*****
00C7 0  C12A          LD    1 INPUT+1     LD CHECK SUM          3AB02300
00C8 00 4C1800D6      BSC L  CKENP,+  SKIP CKSUM IF ZERO  3AB02310
*
00CA 0  C032          LD    CDCT          3AB02320
00CB 0  62CA          LDX  2 -54         3AB02330
00CC 0  825F          CKSM1 A  2 INPUT+54  ADD WORDS 1 TO 54     3AB02340
00CD 0  4802          BSC   C            3AB02350
00CE 0  80D6          A     ONE2         3AB02360
00CF 0  7201          MDX  2 1           3AB02370
00D0 0  70FB          MDX   CKSM1        3AB02380
*
00D1 0  80D3          A     ONE2         3AB02390
00D2 00 4C200158      BSC L  CKSUM,Z    BR IF CK SUM ERR    3AB02400
*
00D4 00 74010GFD      MDX L  CDCT+1     ADD 1 TO CARD CT      3AB02410
*
*****
*      CHECK FOR END OF PROGRAM CARD
*****
00D6 0  C12B          CKEOP LD  1 INPUT+2  GET WORD COUNT        3AB02500
00D7 0  1008          SLA   8            CLEAR CARD CODE       3AB02510
00D8 0  1608          SRA   8            3AB02520
00D9 0  D12B          STO   1 INPUT+2    SAVE WORD COUNT       3AB02530
00DA 00 4C180106      BSC L  EOP,+     EUP IF WC = ZERO     3AB02540
*
*****
*      CHECK FOR OVER CORE
*****
00DC 0  8129          A     1 INPUT      SUM WC+STO ADTRS+RELOC  3AB02550
00DD 00 8400015D      A     L UPPER      3AB02560
00DF 00 9400015F      S     L ULIM       3AB02570
00E1 00 4C10015A      BSC L  OVCR,-     BR IF OVER CORE      3AB02580
*
*****
*      RELOCATE AND STORE
*****
* THIS RT PLACES DATA FIELDS INTO THE CORRECT
* CORE LOCATIONS AND ADDS IN A RELOCATION
* FACTOR IF REQUIRED.
*****

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

00E3 0 C129      LD 1 INPUT      LD STORE ADDR      3AB02720
00E4 0 8078      A          UPPER      RELOCATE STORE ADDR 3AB02730
00E5 0 D00B      STO        STOR4+1    3AB02740
00E6 00 6600002C LDX L2 INPUT+3     XR2 = RELOCATION CODE ADDR 3AB02750
*
00E8 0 63F8      *
STOR1 LDX 3 -8     XR3 = REL BIT CNT      3AB02760
*
00E9 0 C200      LD 2 0          LD NEXT RELOCATION CODE 3AB02770
00EA 0 18D0      RTE 16         STO IN EXT REG        3AB02780
00EB 0 1082      STOR2 SLT 2     BRING IN NEXT REL BIT 3AB02800
00EC 00 4C0400FB * BSC L STOR6,E   BR IF RELOCATE        3AB02810
*
00EE 0 1810      SRA 16         3AB02820
00EF 0 8132      STOR3 A 1 INPUT+9 ADD NEXT WORD        3AB02830
00F0 00 D5000000 STOR4 STO L1 *--  STO IN PROGRAM        3AB02840
00F1 00 74FF002B MDX L INPUT+2,-1 DECR WORD CNT 3AB02850
00F2 0 7001      MDX STOR5     SKIP WHEN WC = 0 3AB02860
00F3 0 607D      LDX RDCD      FINISHED          3AB02870
*
00F4 0 7101      *
STOR5 MDX 1 1     ADV WORD AND STO ADRS 3AB02880
00F5 0 7301      MDX 3 1     ADV REL BIT CNT 3AB02890
00F6 0 70F2      MDX STOR2    3AB02900
00F7 0 7201      MDX 2 1     ADV TO NXT RELOC WD 3AB02910
00FA 0 70ED      MDX STOR1    3AB02920
*
00FB 0 C061      STOR6 LD      UPPER      LD RELOC FACTOR      3AB02930
00FC 0 70F2      MDX STOR3    GO RELOC ADDR 3AB02940
*
00FD 0 0000      CDCT DC *--    3AB02950
00FE 0 0F00      LB20 DC /0F00 3AB02960
00FF 0 0100      LB25 DC /0100 3AB02970
*
*****
*          HEADER AND EOP CARDS
*****
*
* THIS RT. HANDLES RELOCATABLE HDR CARDS.
*
0100 0 C05B      RLHED LD      NLOC      COMPUTE RELOC FACTOR 3AB02980
0101 0 905C      S          RLBA      3AB02990
*
* THIS RT. HANDLES ABSOLUTE HDR CARDS.
*
0102 0 D0        ABHED STO     UPPER      3AB03000
0103 0 7101      MDX 1 1     INITIALIZE CARD COUNT 3AB03010
0104 0 69F8      STX 1 CDCT   3AB03020
0105 0 607D      LDX RDCD    3AB03030
*
* THIS ROUTINE HANDLES END OF PROGRAM CARDS
*
0106 0 C129      EOP LD 1 INPUT  LD 3AB03040
0107 0 8055      A          UPPER      3AB03050
0108 0 D053      STO        NLOC      SET NEXT AVAIL LOC 3AB03060
0109 0 C12C      LD 1 INPUT+3 LD TRANSFER ADDR 3AB03070
010A 0 8052      A          UPPER      3AB03080
010B 0 D001      STO        EOP1+1    3AB03090
010C 00 4C000000 EOP1 BSC L *--  TRANSFER TO PROGRAM 3AB03100
*
*****
*          HEX DATA ROUTINE
*****
* THIS ROUTINE CONVERTS HEX CORRECTION

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

* DATA TO BINARY AND STORES THE DATA.
*
HEX STO HE22+1 CLEAR STORE ADDR
LD ADRS
STO DATA
HEX1 LD I DATA LOAD RELOCATION BIT
STO RLREQ STO RELOCATION BIT
MDX L DATA,1 ADV TO NXT WORD
*
* CONVERT HEX TO BINARY
*
HE10 LD FOUR
STO INDX1
SRA 16
HE11 SLA 4
STO TEMP
SRA 16
STO INDX2
LD I DATA CK FOR BLANK COLUMN
BSC L RDCD,+ FINISHED - LOAD NEXT CARD
*
BSC L HE11A,- BR IF NOT A-F
MDX L INDX2,9 ADD 9 FOR ALPHA
*
HE11A SLA 3 ELIMINATE ZONE BITS
BSC L HE14,+ XF+R IF HEX CHAR = 0
MDX L INDX2,1
HE12 BSC L HE13,+Z XFER IF BIT IS FOUND
SLA 1 PREPARE TO LK AT NEXT BIT
MDX HE12-2
*
HE13 LD INDX2 LOAD BINARY BITS
OR TEMP ADD TO PREVIOUS CHARS
HE14 MDX L DATA,1
MDX L INDX1,-1
MDX HE11
*
* FINISHED - CONVERTED WORD IN ACCUMULATOR
*
MDX L HE22+1,0 SKIP IF STO ADRS
MDX HE21 BR IF DATA
*
A UPPER RELOC STO ADDR
STO HE22+1
LD L INPUT+6 LD NXT WORD
BSC I HE22+1,+ BR TO ADDR IF BLANK
MDX HEX1 CONV NXT WORD
*
HE21 MDX L RLREQ,0 SKIP IF NO RELOCATE
A UPPER
HE22 STO L *-- STORE DATA
MDX L HE22+1,1 ADV STO ADDR
MDX HEX1 CONV NXT WORD
*
* TEMP STORAGE
*
TEMP DC *--
BSS E 0
FOUR DC 4
FDCD DC /1402 FEED CARD
INDX1 DC *--
INDX2 DC *--
RLREQ DC *-- HEX RELOCATION BITS
NGP
NDP

```

RELOCATING DIAGNOSTIC LOADER - 2501

```

*
014F 0 30F5 W30F5 WAIT -11 1442 ERROR IND ON 3AB04080
0150 0 607D LDX RDCD TRY REREADING THE CARD 3AB04090
*
0151 0 08F6 FEED XID FDCD-1 FEED OUT LAST CARD 3AB04100
*
*****
* MONITOR/LOADER INTERFACE
*****
* THE FOLLOWING MUST BE LOCATED IN CORE
* LOCATIONS /0156-/015F.
*****
0152 ORG /0156
0156 0 30F6 MLCB WAIT -10 PROGRAM SHOULD PLACE HERE 3AB04210
0157 0 70FE MDX MLCB A XFER ON BLANK CARD 3AB04220
0158 0 30F7 CKSUM WAIT -9 CHECK SUM ERROR 3AB04230
0159 0 70FE MDX CKSUM 3AB04240
015A 0 3000 OVCR WAIT EXCEEDED CORE SIZE 3AB04250
015B 0 70FE MDX OVCR 3AB04260
015C 0 0160 NLDC DC /160 NEXT AVAILABLE STORAGE LOC 3AB04270
015D 0 0000 UPPER DC *-- RELOCATION FACTOR 3AB04280
015E 0 0000 RLBA DC 0000 BASE ADDRESS 3AB04290
015F 0001 ULIM BSS 1 CORE SIZE 3AB04300
*****
0160 0000 END BSS E 0 3AB04320
0160 0078 END LOAD 3AB04330
3AB04340

```

RELOCATING DIAGNOSTIC LOADER - 2501

CROSS REFERENCE LISTING

```

SYMBOL VALUE REFERENCES
ABFED 0102 00C2
ADRS 00A6 010F
CARD2 0050 0049
CDCT 00FD 00CA,00D4,0104
CHKSM 002E 001C,001E,0025
CKEOP 00D6 00C8
CKSM1 00CC 00D0
CKSUM 0158 00D2,0159
CON1 002F 0022
CORE 005E 005C
CORE2 0064 0069
CORE3 0073 006C
CUUNT 005D 0059
DATA 00A7 0110,0111,0114,011D,0130
DSW 00A0 007D,0096
END 0160 0062
ENDCK 0028
END1 0049
EOP 0106 00DA
EOP1 010C 010B
EGR 003A 0006,0007,0010,0040
FDCD 0149 0151
FEED 0151
FOUR 0148 0116
HEX 010E 0099
HEX1 0111 013E,0146
HE10 0116
HE11 0119 0134
HE11A 0125 0121
HE12 012A 012D
HE13 012E 012A
HE14 012F 0126
HE21 013F 0137
HE22 0142 010E,0135,0139,013C,0144
INDX1 014A 0117,0132
INDX2 014B 011C,0123,0128,012E
INPUT 0029 0082,0097,00A2,00A6,00B1,00B3,00B6,00BD,00C7,00CC,
00D6,00D9,00DC,00E3,00E6,00EF,00F2,0106,0109,013A
000C
INTE 009E 0060,0078
INTD 008A 009E
K0003 00A4 008C
K004F 009D 0081
K0800 0029
K4001 002B 0009,0016
LAST 002C 0014,0015,0046,0057
LB20 00FE 00C0
LB25 00FF 00C1,00C4
LOAD 0078 0160
MLCD 0156 0079,007F,008E,0157
NLOC 015C 0100,0108
ONE 0030 000F,0012,0020,003F,0041,0044
ONE2 00A5 00CE,00D1
OPCOM 0095
OVCR 015A 00E1,0158
PACK 0037 0000,0002,0003,0013,0032,003E,0042,0048
RDCD 007D 0075,00F5,0105,011F,0150
RDCD2 0088
RDCD3 0088 008D,0093
RDIN 0050
RESET 002D 000A,0018,0034
RLBA 015E 0101
RLMED 0100 00C5
RLREQ 014C 0113,013F
SB05 004B 0098

```

IBM MAINTENANCE DIAGNOSTIC PROGRAM FOR THE 1130 SYSTEM

PART NO. 2191284  
PAGE 5

RELOCATING DIAGNOSTIC LOADER - 2501

S606	004D	00BC
S607	00AE	00BA
SE10	00E5	00B0
SENSN	009F	0078,008B
SENSR	00A1	0084,0088,0095
SL	00A8	00AE
SRTRD	0033	0028,0036,005B
START	0000	
STJ	003C	0004,0005,0011,0043,0045
STOR1	00E8	00FA
STOR2	00EB	00F8
STOR3	00EF	00FC
STOR4	00F0	00E5
STOR5	00F6	00F4
STOR6	00FB	00EC
STOWC	0051	0055
STRD	002A	000E,0019,0033,0053
STRD1	00A3	0087
STRT	001C	001F,0021,0024,002F
TEMP	0147	011A,012F
ULIM	015F	006E,0070,0073,00DF
UPPER	015D	00DD,00E4,00FB,0102,0107,010A,013B,0141
WC	004F	001B,0050,0051
W30F4	0072	
W30F5	014F	0090

DATE 15JUN67  
EC NO. 420317

PROG ID 03AB-0  
PAGE 5



TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. PREREQUISITES . . . . .	1
2.1 PROGRAM PREREQUISITES	
2.2 EQUIPMENT REQUIREMENTS	
3. OPERATING PROCEDURE . . . . .	1
3.1 LOADING AND OPERATING	
3.2 WAITS	
4. PRINTOUTS (NONE)	
5. COMMENTS . . . . .	1A
6. APPENDIX (NONE)	

1. PURPOSE

THE 1130 RELOCATING DIAGNOSTIC LOADER IS USED TO LOAD THE DIAGNOSTIC MONITOR AND PROGRAMS WHICH RUN UNDER CONTROL OF THE DIAGNOSTIC MONITOR. THE LOADER ALSO LOADS NON-MONITOR PROGRAMS WHOSE TAPES ARE IN THE PROPER FORMAT. (THE 1130 RELOCATING DIAGNOSTIC LOADER WILL NOT LOAD PROGRAMS WHOSE OBJECT IS COMPATIBLE WITH THE 1130 BASIC DIAGNOSTIC LOADER)

2. PREREQUISITES

2.1 PROGRAM PREREQUISITES.

THIS LOADER USES THE FIRST 340 WORDS OF STORAGE.

2.2 EQUIPMENT PREREQUISITES

- A. 1130 CPU
- B. 1134 TAPE READER

3. USE PROCEDURE

3.1\*\*\* LOADING AND OPERATING

THE 1130 RELOCATING LOADER, TAPE VERSION, IS SUPPLIED AS ONE IPL RECORD AT THE FRONT OF THE TAPE VERSION MONITOR, AND AS A SEPARATE IPL TAPE. THIS LOADER WILL LOAD ANY TAPE PRODUCED IN THE SAME FORMAT AS THE MONITOR TAPE.

TO LOAD THE MONITOR

- A. PLACE THE MONITOR TAPE, WITH LOADER, IN THE TAPE READER.
- B. PRESS RESET, PROGRAM LOAD
- C. MONITOR WILL LOAD AND LOOP IN ITS SUPERVISOR ROUTINE

TO LOAD A PROGRAM.

- A. PLACE PROGRAM TAPE IN THE READER

- B. SET CONSOLE SWITCHES TO 8080 (FUNCTION 2 LOAD CALL. SEE MONITOR DOCUMENTATION)
- C. PRESS INT. PEC KEY
- D. PROGRAM WILL LOAD.

SUBSEQUENT PROGRAMS MAY BE LOADED IN SIMILAR FASHION. FOR OVERLAP LOADING, USE SWITCH SETTING 80C0.

3.2\*\*\* WAITS

HALT NO. (B REG).	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F4	CANNOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START

4. PRINTOUTS (NONE)

5. COMMENTS

5.1\*\*\* THE FOLLOWING ARE THE MAJOR ELEMENTS OF THE 1130 RELOCATABLE DIAGNOSTIC LOADER-

- 5.1.1 READ ROUTINE -- CHECKS READER FOR PROPER STATUS, READS A CARD IMAGE INTO LOCATION /0028 THROUGH /0078, CHECKS FOR SATISFACTORY COMPLETION OF THE READ OPERATION, AND DETERMINES WHETHER THE CARD READ IS A BINARY CARD OR A HEXADECIMAL (CORRECTION CARD).
- 5.1.2 BINARY PACK ROUTINE -- TAKES DATA FOUND IN LOCATIONS /0028 THROUGH /0078 (12 BITS PER CORE WORD) AND PACKS IT INTO LOCATION /0028 THROUGH /0050 (16 BITS PER CORE WORD).
- 5.1.3 CHECKSUM ROUTINE -- COMPUTES CHECKSUM OF A BINARY CARD IMAGE WAITS IF CHECKSUM IS IN ERROR.
- 5.1.4 MOVE ROUTINE -- MOVES DATA FROM /0028 THROUGH /0050 TO PROPER CORE LOCATION. CHECKS FOR EXCEEDING CORE SIZE. ADDS IN RELOCATION FACTOR WHEN REQUIRED.
- 5.1.5 RELOCATABLE HEADER ROUTINE -- ENTERED WHEN A RELOCATABLE HEADER CARD IMAGE IS FOUND. COMPUTES A RELOCATION FACTOR FOR PROGRAM THAT FOLLOWS.
- 5.1.6 ABSOLUTE HEADER ROUTINE -- ENTERED WHEN AN ABSOLUTE HEADER CARD IMAGE IS FOUND. SETS RELOCATION FACTOR TO ZERO.
- 5.1.7 TRANSFER ROUTINE -- ENTERED WHEN A TRANSFER CARD IMAGE IS FOUND. COMPUTES THE NEXT LOCATION AVAILABLE FOR LOADING IF ANOTHER PROGRAM FOLLOWS. TRANSFERS CONTROL TO THE LOCATION SPECIFIED ON THE TRANSFER CARD.

5.1.8 HEX TO BINARY CONVERSION ROUTINE -- CONVERTS A HEXADECI-MAL CARD IMAGE TO BINARY. ADDS IN RELOCATION FACTOR IF REQUIRED.

5.2\*\*\* CARD RECOGNITION

THE FOLLOWING ARE CARD IMAGES WHICH CAN BE LOADED BY THE 1130 RELOCATABLE DIAGNOSTIC LOADER.

5.2.1 ABSOLUTE HEADER CARD HAVE A 1 PUNCH IN COLUMN 4.

5.2.2 RELOCATABLE HEADER CARD HAVE A 0 (ZERO) PUNCH IN COLUMN 4.

5.2.3 NORMAL DATA CARDS HAVE NO PUNCHES IN ROW 12 IN COLUMN 1. AN ADDRESS IN ROWS 11 THROUGH 9 IN COLUMN 1 AND ROWS 12 THROUGH 1 IN COLUMN 2. A CHECKSUM IN ROWS 2 THROUGH 9 OF COLUMN 2 AND ROWS 12 THROUGH 5 OF COLUMN 3. A 12, 0 PUNCH IN COLUMN 4, A WORD COUNT IN ROWS 4 THROUGH 9 OF COLUMN 4. A RELOCATION FIELD (WHICH MAY BE BLANK) IN COLUMN 5 THROUGH 12. DATA IN COLUMNS 13 THROUGH 72. A SEQUENCE NUMBER IN COLUMNS 73-80.

5.2.4 BINARY TRANSFER CARDS HAVE 12, 11, 0, 1 PUNCHES IN COLUMN 4 AND A WORD COUNT OF ZERO (NO PUNCHES IN ROWS 4 THROUGH 9 IN COLUMN 4).

5.2.5 HEXADECI-MAL TRANSFER CARDS HAVE A 12 PUNCH IN COLUMN 1, A TRANSFER ADDRESS IN COLUMN 2 THROUGH 5 AND NO PUNCHES IN COLUMNS 6 AND 7.

5.2.6 HEXADECI-MAL CORRECTION CARDS HAVE A 12 PUNCH IN COLUMN 1. AN ADDRESS IN COLUMN 2 THROUGH 5. DATA IN COLUMNS 6 THROUGH 80. DATA IS GROUPED 5 COLUMNS TO ONE CORE WORD. THE FIRST COLUMN OF EACH GROUP SPECIFIES WHETHER OR NOT THE GROUP REQUIRES A RELOCATION FACTOR. IF THE FIRST COLUMN OF A GROUP IS BLANK A RELOCATION FACTOR WILL NOT BE ADDED. IF THE FIRST COLUMN OF A GROUP CONTAINS AN R (11, 9 PUNCH) A RELOCATION FACTOR WILL BE ADDED TO THE FIELD. LOADING OF THE CARD IS TERMINATED BY TWO SEQUENTIAL BLANK COLUMNS.

6. APPENDIX (NONE)

DATE 15NOV66  
EC NO. 419643

PROG ID 03AC-0  
PAGE 2

IBM CORPORATION  
SYSTEMS DIVISION  
3500 MARKET STREET  
ARMONK, NEW YORK 10504

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. PREREQUISITES . . . . .	1
2.1 PROGRAM PREREQUISITES	
2.2 EQUIPMENT REQUIREMENTS	
3. OPERATING PROCEDURE . . . . .	1
3.1 LOADING AND OPERATING	
3.2 WAITS	
4. PRINTOUTS (NONE)	
5. COMMENTS . . . . .	1A
6. APPENDIX (NONE)	

1. PURPOSE

THE 1130 RELOCATING DIAGNOSTIC LOADER IS USED TO LOAD THE DIAGNOSTIC MONITOR AND PROGRAMS WHICH RUN UNDER CONTROL OF THE DIAGNOSTIC MONITOR. THE LOADER ALSO LOADS NON-MONITOR PROGRAMS WHOSE TAPES ARE IN THE PROPER FORMAT. (THE 1130 RELOCATING DIAGNOSTIC LOADER WILL NOT LOAD PROGRAMS WHOSE OBJECT IS COMPATIBLE WITH THE 1130 BASIC DIAGNOSTIC LOADER)

2. PREREQUISITES

2.1 PROGRAM PREREQUISITES.

THIS LOADER USES THE FIRST 340 WORDS OF STORAGE.

2.2 EQUIPMENT PREREQUISITES

- A. 1130 CPU
- B. 1134 TAPE READER

3. USE PROCEDURE

3.1\*\*\* LOADING AND OPERATING

THE 1130 RELOCATING LOADER, TAPE VERSION, IS SUPPLIED AS ONE IPL RECORD AT THE FRONT OF THE TAPE VERSION MONITOR, AND AS A SEPARATE IPL TAPE. THIS LOADER WILL LOAD ANY TAPE PRODUCED IN THE SAME FORMAT AS THE MONITOR TAPE.

TO LOAD THE MONITOR

- A. PLACE THE MONITOR TAPE, WITH LOADER, IN THE TAPE READER.
- B. PRESS RESET, PROGRAM LOAD
- C. MONITOR WILL LOAD AND LOOP IN IT'S SUPERVISOR ROUTINE

TO LOAD A PROGRAM.

- A. PLACE PROGRAM TAPE IN THE READER

- B. SET CONSOLE SWITCHES TO 8080 (FUNCTION 2 LOAD CALL. SEE MONITOR DOCUMENTATION)
- C. PRESS INT. PCQ KEY
- D. PROGRAM WILL LOAD.

SUBSEQUENT PROGRAMS MAY BE LOADED IN SIMILAR FASHION. FOR OVERLAP LOADING, USE SWITCH SETTING 80C0.

3.2\*\*\* WAITS

HALT NO. (B REG.)	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F4	CANNCT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START

4. PRINTOUTS (NONE)

5. COMMENTS

5.1\*\*\* THE FOLLOWING ARE THE MAJOR ELEMENTS OF THE 1130 RELOCATABLE DIAGNOSTIC LOADER-

5.1.1 READ ROUTINE -- CHECKS READER FOR PROPER STATUS, READS A CARD IMAGE INTO LOCATION /0028 THROUGH /0078, CHECKS FOR SATISFACTORY COMPLETION OF THE READ OPERATION, AND DETERMINES WHETHER THE CARD READ IS A BINARY CARD OR A HEXADECIMAL (CORRECTION CARD).

5.1.2 BINARY PACK ROUTINE -- TAKES DATA FOUND IN LOCATIONS /0028 THROUGH /0078 (12 BITS PER CORE WORD) AND PACKS IT INTO LOCATION /0028 THROUGH /0050 (16 BITS PER CORE WORD).

5.1.3 CHECKSUM ROUTINE -- COMPUTES CHECKSUM OF A BINARY CARD IMAGE WAITS IF CHECKSUM IS IN ERROR.

5.1.4 MOVE ROUTINE -- MOVES DATA FROM /0028 THROUGH /0050 TO PROPER CORE LOCATION. CHECKS FOR EXCEEDING CORE SIZE. ADJS IN RELOCATION FACTOR WHEN REQUIRED.

5.1.5 RELOCATABLE HEADER ROUTINE -- ENTERED WHEN A RELOCATABLE HEADER CARD IMAGE IS FOUND. COMPUTES A RELOCATION FACTOR FOR PROGRAM THAT FOLLOWS.

5.1.6 ABSOLUTE HEADER ROUTINE -- ENTERED WHEN AN ABSOLUTE HEADER CARD IMAGE IS FOUND. SETS RELOCATION FACTOR TO ZERO.

5.1.7 TRANSFER ROUTINE -- ENTERED WHEN A TRANSFER CARD IMAGE IS FOUND. COMPUTES THE NEXT LOCATION AVAILABLE FOR LOADING IF ANOTHER PROGRAM FOLLOWS. TRANSFERS CONTRCL TO THE LOCATION SPECIFIED ON THE TRANSFER CARD.

RELOCATING DIAGNOSTIC LOADER - PAPER TAPE VERSION

5.1.8 HEX TO BINARY CONVERSION ROUTINE -- CONVERTS A HEXADECIMAL CARD IMAGE TO BINARY. ADDS IN RELOCATION FACTOR IF REQUIRED.

5.2... CARD RECOGNITION

THE FOLLOWING ARE CARD IMAGES WHICH CAN BE LOADED BY THE 1130 RELOCATABLE DIAGNOSTIC LOADER.

5.2.1 ABSOLUTE HEADER CARD HAVE A 1 PUNCH IN COLUMN 4.

5.2.2 RELOCATABLE HEADER CARD HAVE A 0 (ZERO) PUNCH IN COLUMN 4.

5.2.3 NORMAL DATA CARDS HAVE NO PUNCHES IN ROW 12 IN COLUMN 1. AN ADDRESS IN ROWS 11 THROUGH 9 IN COLUMN 1 AND ROWS 12 THROUGH 1 IN COLUMN 2. A CHECKSUM IN ROWS 2 THROUGH 9 OF COLUMN 2 AND ROWS 12 THROUGH 5 OF COLUMN 3. A 12, 0 PUNCH IN COLUMN 4, A WORD COUNT IN ROWS 4 THROUGH 9 OF COLUMN 4. A RELOCATION FIELD (WHICH MAY BE BLANK) IN COLUMN 5 THROUGH 12. DATA IN COLUMNS 13 THROUGH 72. A SEQUENCE NUMBER IN COLUMNS 73-80.

5.2.4 BINARY TRANSFER CARDS HAVE 12, 11, 0, 1 PUNCHES IN COLUMN 4 AND A WORD COUNT OF ZERO (NO PUNCHES IN ROWS 4 THROUGH 9 IN COLUMN 4).

5.2.5 HEXADECIMAL TRANSFER CARDS HAVE A 12 PUNCH IN COLUMN 1, A TRANSFER ADDRESS IN COLUMN 2 THROUGH 5 AND NO PUNCHES IN COLUMNS 6 AND 7.

5.2.6 HEXADECIMAL CORRECTION CARDS HAVE A 12 PUNCH IN COLUMN 1. AN ADDRESS IN COLUMN 2 THROUGH 5. DATA IN COLUMNS 6 THROUGH 80. DATA IS GROUPED 5 COLUMNS TO ONE CORE WORD. THE FIRST COLUMN OF EACH GROUP SPECIFIES WHETHER OR NOT THE GROUP REQUIRES A RELOCATION FACTOR. IF THE FIRST COLUMN OF A GROUP IS BLANK A RELOCATION FACTOR WILL NOT BE ADDED. IF THE FIRST COLUMN OF A GROUP CONTAINS AN R (11,9 PUNCH) A RELOCATION FACTOR WILL BE ADDED TO THE FIELD. LOADING OF THE CARD IS TERMINATED BY TWO SEQUENTIAL BLANK COLUMNS.

6. APPENDIX (NONE)



PAPER TAPE RELOCATABLE LOADER

```

0081 0 1010      SLA      16
0082 0 D127     LDRO1 STO 1 /27      CLEAR INPUT AREA
0083 0 71FF     MDX      1 -1
0084 0 70FD     MDX      LDRO1
*
0085 0 081A     XIO      SENSR-1    SENSE DSW
0086 0 E021     AND      K0200      CHECK READY
0087 00 4C2000AF BSC L NRDY,Z      BR IF NOT READY
*
0089 0 402A     LDRO3 BSI      RDRLP      GO READ A CHAR
008A 0 C921     LD      CHAR      _DAD CHAR READ
008B 0 F018     EDR      K7F00      CHACK FOR DELETE
008C 00 4C180089 BSC L LDRO3,+      BR IF DELETE
008E 0 C01D     LD      CHAR      FIRST CHAR
008F 0 1808     SRA      8
0090 0 D01C     STO      WDCNT      WORD COUNT
*
0091 0 4022     LDRO4 BSI      RDRLP      READ A CHAR
0092 0 C019     LD      CHAR      LOAD THE CHAR RAAD
0093 0 1898     SRT      24         SHIFT TO LOW ORDER Q
*
0094 0 401F     BSI      RDRLP      READ SECOND HALF-WORD
0095 0 C016     LD      CHAR      LOAD SECOND HALF
0096 0 18C8     RTE      8         SHIFT TO COMBINE HALVES
0097 00 D4000000 STORE STO L *--      STORE PACKED WORD
*
0099 00 74010098 MDX L STORE+1,1    INCRF STORE ADDRESS
*
009B 00 74FF00AD MDX L WDCNT,-1     SKIP WHEN FINISHED
009D 0 70F3     MDX      LDRO4
009E 0 6100     LDX      1 0       CLEAR XR1
009F 0 701F     MDX      CHHDR      BR TO CHECK HEADER
*
00A0 0 ECFO     KEFCO DC /ECFO
00A1 0 1F01     SENSR DC /1F01      SENSE RESET
00A2 0 0C00     KOC00 DC /OC00
00A3 0 1C10     STRDR DC /1C10      START READER
00A4 0 7F00     K7F00 DC /7F00
00A5 0 1F00     SENSE DC /1F00
00A6 0 00AC     RDIN DC CHAR
00A7 0 1A09     DC /1A00      READ A CHAR
*
00A8 0 0200     K0200 DC /0200
00A9 0 0001     ONE2 DC /0001
00AA 0 0028     ADRS DC INPUT
00AB 0 00B1     INTAD DC INTE
00AC 0 0000     CHAR DC *--
00AD 0 0000     WDCNT DC *--      WORD COUNT
00AE 0 0000     DATA DC *--
*
00AF 0 30F8     NRDY WAIT -8       READER NOT KEADY
00B0 0 70CD     MDX RDCD      TRY AGIN
00B1 0 0000     INTE DC *--
00B2 0 08ED     XIO SENSR-1    SENSE RESET
00B3 0 7007     MDX RDRL1
00B4 0 0000     RDRLP DC *--
00B5 0 08EC     XIO STRDR-1    START READER
00B6 0 08ED     XIO SENSE-1    SENSE DSW
00B7 0 E0E8     AND KEFCO      MASK PUNCH BITS
00B8 0 F0E9     EDR KOC00      CHECK FOR BUSY,NRDY
00B9 0 4818     BSC +-        SKIP IF ANY CHANGE
00BA 0 70FB     MDX RDRLP+2    LOOP UNTIL CHANGES
*
00BB 0 08EA     RDRL1 XIO RDIN  READ A CHAR
00BC 0 08E3     XIO SENSR-1
00BD 00 4CC000B4 BOSC 1 RDRLP

```

DATE 15NOV66 15JUN67  
EC NO. 419643 420317

PROG ID 03AC-2  
PAGE 2

PAPER TAPE RELOCATABLE LOADER

```

*
*
* *****
* CHECK HEADER
* THIS RT DETERMINES WHETHER THE DATA CARD IS
* 1) ABSOLUTE HDR CARD 2) RELOCATABLE HDR CARD
* -----
*
00BF 0 C12A     CHHDR LD 1 INPUT+2    CK FOR HDR CARDS
00C0 00 4C180156 BSC L MLC0,+      BR TO USER - BLANK CD
00C2 0 E03F     AND LB20
00C3 0 903F     S LB25
00C4 00 4C180106 BSC L ABHED,+      BCH IF ABSOL HEAD CARD
00C6 0 903C     S LB25
00C7 00 4C180104 BSC L RLHED,+      BCH IF RELOC HEAD CARD
*
* *****
* CHECK SUM
* THIS ROUTINE ADDS COLUMNS 0 - 72
* TO CHECK THAT THIS SUM PLUS THE CARD
* SEQUENCE NUMBER EQUALS ZERO .
* -----
*
00C9 0 C129     LD 1 INPUT+1      LD CHECK SUM
00CA 00 4C1800DB BSC L CKEOP,+      SKIP CKSUM IF ZERO
*
00CC 0 C034     LD CDCT
00CD 0 62CA     LDX 2 -54
00CE 0 825E     CKSM1 A 2 INPUT+54  ADD WORDS 1 TO 54
00CF 0 4802     BSC C
00D0 0 80DB     A ONE2
00D1 0 7201     MDX 2 1
00D2 0 70FB     MDX CKSM1
*
00D3 0 80D5     A ONE2
00D4 00 4C200158 BSC L CKSUM,Z      BR IF CK SUM ERR
*
00D6 00 74010101 MDX L CDCT,1      ADD 1 TO CARD CT
*
* -----
* CHECK FOR END OF PROGRAM CARD
* -----
*
00D8 0 C12A     CKEOP LD 1 INPUT+2  GET WORD COUNT
00D9 0 1008     SLA 8             CLEAR CARD CODE
00DA 0 1808     SRA 8
00DB 0 D12A     STO 1 INPUT+2    SAVE WORD COUNT
00DC 00 4C18010A BSC L EDP,+      EDP IF WC = ZERO
*
* -----
* CHECK FOR OVER CORE
* -----
*
00DE 0 8128     A 1 INPUT        SUM WC+STO ADDR+RELOC
00DF 00 8400015D A L UPPER
00E1 00 9400015F S L ULIM
00E3 00 4C10015A BSC L OVCR,-      BR IF OVER CORE
*
* *****
* RELOCATE AND STORE
* THIS RT PLACES DATA FIELDS INTO THE CORRECT
* CORE LOCATIONS AND ADDS IN A RELOCATION
* FACTOR IF REQUIRED.
* -----
*
00E5 0 C128     LD 1 INPUT        LD STORE ADDR

```

DATE 15NOV66 15JUN67  
EC NO. 419643 420317

PROG ID 03AC-2  
PAGE 2A

PAPER TAPE RELOCATABLE LOADER

```

00E6 00 8400015D      A  L  UPPER      RELOCATE STORC ADDR      3AC02720
00E8 0  D00B          STO  STOR4+1      3AC02730
00E9 00 6600002B      LDX  L2 INPUT+3  XR2 = RELOCATION CODE ADDR 3AC02740
*
00EB 0  63F8          *
STOR1 LDX  3 -8      XR3= REL BIT CNT      3AC02750
*
00EC 0  C200          LD   2 0          LD NEXT RELOCATION CODE    3AC02770
00ED 0  18D0          RTE  16          STO IN EXT REG           3AC02780
00EE 0  1082          STOR2 SLT  2      BRING IN NEXT REL BIT    3AC02790
00EF 00 4C0400FE      *
BSC  L  STOR6,E    BR IF RELOCATE           3AC02800
*
00F1 0  1810          SRA  16          3AC02810
00F2 0  8131          STOR3 A  1 INPUT+9  ADD NEXT WORD             3AC02820
00F3 00 D5000000      STOR4 STO  L1 *-*   STO IN PROGRAM           3AC02830
00F5 00 74FF002A      MDX  L  INPUT+2,-1 DECR WORD CNT          3AC02840
00F7 0  7001          MDX  STOR5       SKIP WHEN WC = 0 3AC02850
00F8 0  607E          LDX  R0CD        FINISHED                    3AC02860
*
00F9 0  7101          STOR5 MDX  1 1     ADV WORD AND STO ADRS    3AC02870
00FA 0  7301          MDX  3 1         ADV REL BIT CNT          3AC02880
00FB 0  70F2          MDX  STOR2       3AC02890
00FC 0  7201          MDX  2 1         ADV TO NXT RELOC WD    3AC02900
00FD 0  70ED          MDX  STOR1       3AC02910
*
00FE 00 C400015D      STOR6 LD  L  UPPER  LD RELOC FACTOR        3AC02920
0100 0  70F1          MDX  STOR3       GO RELOC ADDR      3AC02930
*
0101 0  0000          COCT DC  *-*     3AC02940
0102 0  0F00          LB20 DC  /0F00   3AC02950
0103 0  0100          LB25 DC  /0100   3AC02960
*
*****
*          HEADER AND EOP CARDS
*****
*
*-----
* THIS RT. HANDLES RELOCATABLE HDR CARDS.
*-----
*
0104 0  C057          RLHED LD  NLOC    COMPUTE RELOC FACTOR 3AC02970
0105 0  9058          S      RLBA      3AC02980
*
*-----
* THIS RT. HANDLES ABSOLUTE HDR CARDS.
*-----
*
0106 0  D056          ABHED STO  UPPER  3AC02990
0107 0  7101          MDX  1 1         INITIALIZE CARD COUNT 3AC03000
0108 0  60FB          STX  1 COCT     3AC03010
0109 0  607E          LDX  R0CD        3AC03020
*
*-----
* THIS ROUTINE HANDLES END OF PROGRAM CARDS
*-----
*
010A 0  C128          EOP  LD  1 INPUT  3AC03030
010B 0  8051          A      UPPER     3AC03040
010C 0  D04F          STO  NLJC        3AC03050
010D 0  C128          LD  1 INPUT+3   SET NEXT AVAIL LOC      3AC03060
010E 0  804E          A      UPPER     LD TRANSFER ADDR      3AC03070
010F 0  D001          STO  EOP1+1     3AC03080
0110 00 4C000000      EOP1 BSC  L *-*   TRANSFER TO PROGRAM    3AC03090
*
*****
*          HEX DATA ROUTINE
*****
*-----
* THIS ROUTINE CONVERTS HEX CORRECTION
* DATA TO BINARY AND STORES THE DATA.
3AC03100
3AC03110
3AC03120
3AC03130
3AC03140
3AC03150
3AC03160
3AC03170
3AC03180
3AC03190
3AC03200
3AC03210
3AC03220
3AC03230
3AC03240
3AC03250
3AC03260
3AC03270
3AC03280
3AC03290
3AC03300
3AC03310
3AC03320
3AC03330
3AC03340
3AC03350
3AC03360
3AC03370
3AC03380
3AC03390

```

PAPER TAPE RELOCATABLE LOADER

```

*-----
*
0112 0  D036          HEX  STO  HE22+1  CLEAR STORE ADDR      3AC03400
0113 00 C40000AA      LD  L  ADRS      3AC03410
0115 00 D40000AE      STO  L  DATA    3AC03420
0117 00 C48000AE      HEX1 LD  I  DATA  LOAD RELOCATION BIT    3AC03430
0119 0  D038          STO  RLREQ      STO RELOCATION BIT      3AC03440
011A 00 740100AE      MDX  L  DATA,1  ADV TO NXT WORD       3AC03450
*
*-----
*          CONVERT HEX TO BINARY
*-----
*
011C 0  C031          HE10 LD  FOUR     3AC03460
011D 0  D032          STO  INDX1      3AC03470
011E 0  1810          SRA  16         3AC03480
011F 0  1004          HE11 SLA  4      3AC03490
0120 0  D02C          STO  TEMP       3AC03500
0121 0  1810          SRA  16         3AC03510
0122 0  D02E          STO  INDX2      3AC03520
0123 00 C48000AE      LD  I  DATA    CK FOR BLANK COLUMN 3AC03530
0125 00 4C18007E      BSC  L  R0CD,+  FINISHED - LOAD NEXT CARD 3AC03540
*
0127 00 4C10012B      *
0129 00 74090151      BSC  L  HE11A,- BR IF NOT A-F      3AC03550
*
0128 0  1003          HE11A SLA  3     ELIMINATE ZONE BITS    3AC03560
012C 00 4C180135      BSC  L  HE14,+  XF+R IF HEX CHAR = 0 3AC03570
012E 00 74010151      MDX  L  INDX2,1 3AC03580
0130 00 4C280134      HE12 BSC  L  HE13,+2 XFER IF BIT IS FOUND 3AC03590
0132 0  1001          SLA  1          PREPARE TO LK AT NEXT BIT 3AC03600
0133 0  70FA          MDX  HE12-2    3AC03610
*
0134 0  C01C          HE13 LD  INDX2   LOAD BINARY BITS      3AC03620
0135 0  E817          HE14 DR  TEMP    ADD TO PREVIOUS CHARS 3AC03630
0136 00 740100AE      MDX  L  DATA,1 3AC03640
0138 00 74FF0150      MDX  L  INDX1,-1 3AC03650
013A 0  70E4          MDX  HE11      3AC03660
*
*-----
*          FINISHED - CONVERTED WORD IN ACCUMULATOR
*-----
*
013B 00 74000149      MDX  L  HE22+1,0 SKIP IF STO ADRS     3AC03670
013D 0  7007          MDX  HE21      BR IF DATA          3AC03680
*
013E 0  801E          A      UPPER    RELOC STO ADRS     3AC03690
013F 0  D009          STO  HE22+1    3AC03700
0140 00 C400002E      LD  L  INPUT+6  LD NXT WORD          3AC03710
0142 00 4C980149      BSC  I  HE22+1,+ BR TO ADRS IF BLANK 3AC03720
0144 0  70D2          MDX  HEX1      CONV NXT WORD       3AC03730
*
0145 00 74000152      HE21 MDX  L  RLREQ,0 SKIP IF NO RELOCATE 3AC03740
0147 0  8015          A      UPPER    3AC03750
0148 00 D4000000      HE22 STO  L *-*  STORE DATA          3AC03760
014A 00 74010149      MDX  L  HE22+1,1 ADV STO ADRS        3AC03770
014C 0  70CA          MDX  HEX1      CONV NXT WORD       3AC03780
*
014D 0  0000          TEMP DC  *-*    TEMP STORAGE        3AC03790
014E 0  0004          FOUR DC  4      3AC03800
014F 0  1402          FDOD DC  /1402  FEED CARD            3AC03810
0150 0  0000          INDX1 DC  *-*   3AC03820
0151 0  0000          INDX2 DC  *-*   3AC03830
0152 0  0000          RLREQ DC  *-*   HEX RELOCATION SITS 3AC03840
0153 0  1000          NOP           3AC03850
0154 0  1000          NOP           3AC03860
0155 0  FFFF          DC  /FFFF     TAPE LOADER SWITCH 3AC03870
*

```

PAPER TAPE RELOCATABLE LOADER

```

*
*
*****
* MONITOR/LOADER INTERFACE
*****
* THE FOLLOWING MUST BE LOCATED IN CORE
* LOCATIONS /0156-/015F.
*-----*
0156 0 30F6      MLCD  WAIT  -10      PROGRAM SHOULD PLACE HERE
0157 0 70FE      MDX   MLCD   A XFER ON BLANK CARD
0158 0 30F7      CKSUM WAIT  -9      CHECK SUM ERROR
0159 0 70FE      MDX   CKSUM
015A 0 3000      OVCR  WAIT  EXCEEDED CORE SIZE
015B 0 70FE      MDX   OVCR
015C 0 0160      NLDC  DC    /160     NEXT AVAILABLE STORAGE LOC
015D 0 0000      UPPER DC  *- *     RELOCATION FACTOR
015E 0 0000      RLBA  DC    0000     BASE ADDRESS
015F 0 0001      ULIM  BSS   1      CORE SIZE
*****
0160 0 0000      END1  DC    0
0161 0 0000      DC
0162           ORG   END1
0160 0000      END   BSS   E  0
0160 0078      END   END   LOAD
3AC04080
3AC04090
3AC04100
3AC04110
3AC04120
3AC04130
3AC04140
3AC04150
3AC04160
3AC04170
3AC04180
3AC04190
3AC04200
3AC04210
3AC04220
3AC04230
3AC04240
3AC04250
3AC04260
3AC04270
3AC04280
3AC04290
3AC04300
3AC04310
3AC04320

```

PAPER TAPE RELOCATABLE LOADER

CROSS REFERENCE LISTING

SYMBOL	VALUE	REFERENCES
ABHED	0106	00C4
ADRS	00AA	0113
CDCT	0101	00CC,00D6,0108
CHAR	00AC	008A,008E,0092,0095,00A6
CHHDR	00BF	009F
CHKSM	0021	0012,0015,001C
CKEUP	00D8	00CA
CKSM1	00CE	00D2
CKSUM	0158	00D4,0159
CORE	0028	001D,0020
CORE2	0030	0035
CORE3	003F	0038
CSADD	0013	0016,0018
DATA	00AE	0115,0117,011A,0123,0136
END	0160	0022,002E
END1	0160	0162
EOP	010A	00DC
EOP1	0110	010F
FDCD	014F	
FOUR	014E	011C
HEX	0112	
HEX1	0117	0144,014C
HE10	011C	
HE11	011F	013A
HE11A	0128	0127
HE12	0130	0133
HE13	0134	0130
HE14	0135	012C
HE21	0145	0130
HE22	0148	0112,0138,013F,0142,014A
INDX1	0150	0110,0138
INDX2	0151	0122,0129,012E,0134
INPUT	0028	0050,00AA,00BF,00C9,00CE,00D8,00DB,00DE,00E5,00E9,00F2,00F5,010A,010D,0140
INTAD	00AB	007C
INTE	00B1	000C,002A,00AB
KECFO	00A0	00B7
KOC00	00A2	0088
K0001	0023	0017
K0200	00A8	0079,0086
K7F00	00A4	0088
LAST	0022	0019
LB20	0102	00C2
LB25	0103	00C3,00C6
LDRO1	0082	0084
LDRO3	0089	008C
LDRO4	0091	009D
LOAD	0078	0160
MLCD	0156	007A,00C0,0157
NLOC	015C	0104,010C
NRDY	00AF	0087
ONE2	00A9	00D0,00D3
OVCR	015A	00E3,015B
RDCD	007E	0041,0080,00F8,0109,0125
RDIN	00A6	008B
RDRLP	00B4	0089,0091,0094,00BA,00BD
RDRL1	00B8	0083
RLBA	015E	0105
RLHED	0104	00C7
RLREQ	0152	0119,0145
SENSE	00A5	0086
SENSR	00A1	0078,0085,0082,008C
STORE	0097	007F,0099
STOR1	00EB	00FD
STOR2	00EE	00FB



IBM MAINTENANCE DIAGNOSTIC PROGRAM FOR THE 1130 SYSTEM

PART NO. 2191286  
PAGE 5

PAPER TAPE RELOCATABLE LOADER

STOR3	00F2	0100
STOR4	00F3	00E8
STOR5	00F9	00F7
STOR6	00FE	00EF
STRDR	00A3	00B5
STRT	0012	0000,001B
TEMP	014D	0120,0135
ULIM	015F	003A,003C,003F,00E1
UPPER	015D	00DF,00E6,00FE,0106,010B,010E,013E,0147
WDCNT	00AD	0090,009B
W30F4	003E	

DATE 15NOV66 15JUN67  
EC NO. 419643 420317

PROG ID 03AC-2  
PAGE 5

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. REQUIREMENTS . . . . .	1
2.1 PROGRAM REQUIREMENTS	
2.2 EQUIPMENT REQUIREMENTS	
3. OPERATING PROCEDURE . . . . .	1A
3.1 PROGRAM LOADING	
3.2 PROGRAM OPERATION	
3.2.1 PROGRAM CONTROL - FUNCTION 0	
3.2.2 ROUTINE SELECTION - FUNCTION 1	
3.2.3 1231 OMPR SETUP.	
3.3 HALTS	
3.3.1 NORMAL HALTS	
3.3.2 ERROR HALTS	
3.4 PROGRAM TERMINATION	
3.5 RESTART	
4. PRINTOUTS . . . . .	3
4.1 STATUS MESSAGES	
4.2 ERROR MESSAGES	
5. COMMENTS . . . . .	4A

1. PURPOSE  
THE 1231 OMPR DIAGNOSTIC PROGRAM CHECKS THE OPERATING PERFORMANCE OF THE 1231 OMPR.

2. PREREQUISITES

2.1 PROGRAM PREREQUISITES  
THE 1231 OMPR DIAGNOSTIC PROGRAM MUST BE RUN UNDER CONTROL OF THE 1130 SYSTEM DIAGNOSTIC MONITOR.

2.2 EQUIPMENT PREREQUISITES  
THE FOLLOWING EQUIPMENT IS REQUIRED.

- A. 1131 CENTRAL PROCESSING UNIT (CPU)
- B. 1231 OPTICAL MARK PAGE READER
- C. PROGRAM INPUT DEVICE.

\*\*\*\*\*

3. OPERATING PROCEDURE

3.1\*\*\* PROGRAM LOADING

STANDARD MONITOR LOADING PROCEDURES APPLY

THESE PROCEDURES ARE SUMMARIZED HERE. SEE DM USE PROCEDURE FOR DETAILS.

1. SET FIRST TYPEWRITER TAB 20 CHARACTERS FROM LEFT MARGIN.
2. SET BIT SWITCH 15 OFF - LOAD AND GO  
    ON - IF HALTED AFTER LOADING, SELECT PROGRAM OPTIONS THEN TURN OFF HALT SWITCH OR FOLLOW NORMAL RESTART PROCEDURE (SECTION 3.5)

3. LOAD DIAGNOSTIC MONITOR.
4. SELECT PROGRAM OPTIONS, IF DESIRED.

\*\*\*\*\*

3.2\*\*\* PROGRAM OPERATION.

THESE OPERATING PROCEDURES APPLY TO SINGLE PROGRAM OPERATION ONLY. FOR OVERLAP OPERATION REFER TO SECTION 3.2.3 OF THE 1130 DIAGNOSTIC MONITOR II DOCUMENTATION.

3.2.1 PROGRAM CONTROL - FUNCTION 0

1. SET SWITCHES 0-1 OFF.
2. SET PROGRAM LOADING SEQUENCE NUMBER IN SWITCHES 4-7. (ONE UNLESS RUNNING OVERLAP)
3. SET SWITCHES 8-15 AS DESIRED.

SW	FUNCTION
8	RESTART
9	ROUTINE START MESSAGE
10	LOCK ON FUNCTION
11	LOOP PROGRAM
12	LOOP ON ERROR
13	BYPASS ERROR PRINTOUT
14	HALT ON ERROR
15	HALT

4. PRESS INT RED KEY ON CONSOLE.

\*\*

3.2.2 ROUTINE SELECTION - FUNCTION 1

THE SELECTED ROUTINE WILL LOOP UNTIL A NEW ROUTINE IS SELECTED.

1. SET SWITCHES 0-1 TO 01.
2. SET PROGRAM LOADING SEQUENCE NUMBER IN SWITCHES 4-7. (ONE UNLESS RUNNING OVERLAP)

3. SET ROUTINE NUMBER IN SWITCHES 12-15.

RTN	DESCRIPTION	
1	READ DOCUMENTS AND COMPARE	NORMAL ROUTINE
2	** MASTER MARK TEST	SELECTIVE ROUTINES
3	* ALTERNATE SELECT	THESE ROUTINES RUN ONLY IF SELECTED.
4	* READ DOCUMENTS, NO COMPARE	

\* = REFER TO SECTION 3.2.3 FOR SPECIAL INSTRUCTIONS.  
\*\* = OPTIONAL FEATURE TEST

4. PRESS INT REQ KEY ON CONSOLE.  
\*\*\*\*\*

3.2.3 1231 OMPR SETUP

THE 1231 MUST BE SETUP AS FOLLOWS.

1. ROUTINE 1 USES THE 1231 ONLINE TEST DOCUMENTS P/N 648746.
2. A CONTROL SHEET WITH COLUMNS 8 AND 18 MARKED IN ALL ROWS SHOULD BE USED.
3. SET 1231 SWITCHES AS FOLLOWS.\*

```
*****
* SWITCH      . RTN . SET      *
*-----*
* FIELD CHK   . ALL . AS REQ'D *
*
* TIMING MARK . 1   . 6       *
* CHECK       . 2-4 . AS REQ'D *
*
* FEED MODE   . ALL . ON-DEMAND *
*
* CONTRL      . 1   . YES      *
* TIMING      .    .          *
* MARKS       . 2-4 . AS REQ'D *
*****
```

\* SEE 1231 SRL FOR DETAILED DESCRIPTION.

3.3\*\*\* PROGRAM HALTS

3.3.1 NORMAL HALTS

HALT NO. (B REG).	DESCRIPTION	RESTART ACTION
3001	PROGRAM STOP OR ADDRESS STOP	PRESS START
3002	HALT ON ERROR	DISPLAY MODE-PRESS START. RUN MODE - PRESS START

\*\*

3.3.2 ERROR HALTS

HALT NO. (B REG).	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON FIRST CARD OF LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F3	CARD 2 OF LOADER DID NOT LOAD	RELOAD
30F4	CAN NOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F5	READER CHECK WHEN LOADING MONITOR OR TEST PROGRAM	NPRO THEN PLACE CARDS RUN OUT IN FRONT OF REMAINING DECK AND PRESS START.
30F6	MONITOR DID NOT LOAD	RELOAD
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START
30FA	CONSOLE PRINTER HANG UP - BUSY WILL NOT GO OFF	FIX THE CONSOLE PRINTER OR NOP THIS WAIT

\*\*\*\*\*

3.4\*\*\* PROGRAM TERMINATION

IF LOOP PROGRAM HAS NOT BEEN SPECIFIED THE PROGRAM WILL TERMINATE AT THE END OF ROUTINE 1. ROUTINE 2-4 WILL ONLY RUN IF SELECTED.

IF ANY ROUTINE IS SELECTED THAT ROUTINE WILL LOOP AND WILL NOT TERMINATE.

\*\*\*\*\*

3.5\*\*\* RESTART

1. SET SWITCHES 0-7 TO 01.
2. TURN ON SWITCH 8.
3. SET DESIRED CONTROL IN SWITCHES 9-14.
4. PRESS INTERRUPT REQUEST KEY.

\*\*\*\*\*

4. PRINTOUTS

ALL PRINTOUTS ARE IN THE STANDARD FORMAT.

APPNN ODRR AAAA (MESSAGE)  
OR  
EPPNN ODRR XXXX (MESSAGE)

WHERE A IDENTIFIES STATUS MESSAGES  
E IDENTIFIES ERROR MESSAGES  
PP IS THE PID OF THE PROGRAM CAUSING THE MESSAGE  
THIS WILL BE EITHER 00 FOR MESSAGES  
ORIGINATED BY THE MONITOR OR 03 FOR  
MESSAGES ORIGINATED BY THIS PROGRAM.  
NN IS THE MESSAGE SEQUENCE NUMBER  
RR IS THE ROUTINE NUMBER  
AAAA IS THE ADDRESS OF THE ROUTINE  
MESSAGE IS ANY VARIABLE INFORMATION

4.1\*\*\* STATUS MESSAGES

A0000 NUM PID ADRS RELF LD  
XXXX XXXX XXXX XXXX

THIS MESSAGE IS PRINTED FOLLOWING THE LOADING OF ANY PROGRAM  
(EXCEPT MONITOR). THE MESSAGE GIVES THE LOAD SEQUENCE NUMBER, THE  
PROGRAM ID, THE ADDRESS INTO WHICH THE PROGRAM WAS LOADED, AND THE  
RELOCATION FACTOR.

A0001 SMS PID  
XXXX XXXX

THIS MESSAGE IS PRINTED EACH TIME A VALID SWITCH ENTRY IS READ BY THE  
MONITOR. THE MESSAGE CONTAINS THE SWITCH SETTING READ TOGETHER WITH  
THE PROGRAM ID OF THE PROGRAM INTO WHICH THE CONTENTS OF SWITCHES  
8-15 WERE STORED. IF THE SWITCH ENTRY CALLED FOR HALT OR RESTART OF  
ANY PROGRAM THE WORD HALT OR RESTART WILL FOLLOW THE MESSAGE.

A0300 000R AAAA

ROUTINE START MESSAGE - IF SWITCH 9, FUNCTION 0, IS TURNED ON,  
THIS MESSAGE WILL BE PRINTED BEFORE THE START OF EACH ROUTINE.  
R IS THE NUMBER OF THE NEXT ROUTINE AND AAAA IS THE STARTING  
ADDRESS.

A0301 000R AAAA 1231 NOT READY

THE 1231 IS NOT READY. MESSAGE WILL BE REPEATED EVERY 15 SEC. AP-  
PROXIMATELY.

A0302 000R AAAA 1231 HOPPER EMPTY

BEFORE STARTING AN OPERATION, THE PROGRAM HAS FOUND THE HOPPER EMPTY.  
MESSAGE WILL BE REPEATED EVERY 15 SEC. APPROXIMATELY.

A0303 0001 AAAA TOO LATE TO SELECT

IN ROUTINE ONE, THE DOCUMENT JUST READ HAD THE SELECT BIT IN WORD  
91 MARKED, BUT THE OK TO SELECT BIT IN THE DSW HAS TIMED OUT. THIS  
COULD BE DUE TO ERROR MESSAGES DURING THE READING, OR DELAYS DUE TO  
OTHER PROGRAMS OVERLAPPING. IF THESE DELAYS HAVE NOT OCCURRED, AND  
THIS MESSAGE PRINTS, CHECK THE OK TO SEL SINGLE SHOT.

4.2\*\*\* ERROR MESSAGES

```
*****  
* THE 1231 DSW *  
*-----*  
* *  
* BIT *  
* 0 READ RESPONSE *  
* 1 TM MARK ERROR *  
* 2 READ ERROR *  
* 3 MASTE DATA *  
* 4 OPERATION COMPLETE *  
* 5 OK TO SELECT *  
* 6 FEED BUSY *  
* 7 DOCUMENT SELECTED BY OMPR *  
* 8 TM CHECK BUSY *  
* 9 HOPPER EMPTY *  
* 10 NOT USED *  
* 11 NOT USED *  
* 12 NOT USED *  
* 13 READER BUSY *  
* 14 BUSY *  
* 15 NOT READY *  
* *  
*****
```

E0001 SWS INVLD  
XXXX

THE SETTING OF SWITCHES 4-7 DID NOT EQUAL THE LOAD SEQUENCE  
NUMBER OF ANY PROGRAM IN CORE.

E0003 OVR CORE

THE PROGRAM WHICH THE LOADER WAS ATTEMPTING TO LOAD  
EXCEEDED AVAILABLE CORE. LOADING WAS TERMINATED.

E0004 CKSUM

A CHECK SUM ERROR WAS DETECTED WHILE LOADING A TEST PROGRAM.  
THIS ERROR OCCURS UNDER ANY OF THE FOLLOWING CONDITIONS.

1. A CARD IS MISSING OR IS OUT OF SEQUENCE.
2. THERE IS AN EXTRA CARD IN THE DECK.
3. THE PUNCHED INFORMATION ON THE CARD IS NOT CORRECT.
4. DATA WAS LOST OR PICKED UP DUE TO A MACHINE MALFUNCTION.
5. DUE TO A CPU MALFUNCTION, THE CHECK SUM WAS NOT  
CORRECTLY CALCULATED.

WHEN THIS ERROR OCCURS ATTEMPT TO RELOAD THE PROGRAM.

E0005 000N XXXX

THIS ERROR WILL OCCUR IS AN INTERRUPT OCCURS, BUT THE ILSW  
WAS NOT CORRECT. N IS THE INTERRUPT LEVEL AND XXXX IS THE  
ILSW. THIS PRINTOUT WILL ONLY OCCUR IF THE INTERRUPT IS RESET  
BY A ROSI. NO ATTEMPT IS MADE BY THE ERROR ROUTINE TO RESET  
THE REQUEST BIT.

E0301 000R AAAA WAS - S/B DSW ERROR INITIALIZING  
XXXX XXXX

MODIFIERS SHOW DSW SENSED BEFORE ANY OPERATION WAS STARTED.

E0302 0001 AAAA WAS - S/B DATA ERROR  
XXXX XXXX DDDDD

DATA FROM DOCUMENT DID NOT COMPARE EQUAL.  
MODIFIERS (XXXX) SHOW HEX REPRESENTATION  
OF THE WORD IN ERROR. MODIFIER (DDDD)  
IS THE WORD NUMBER. THE MESSAGE WILL  
BE PRINTED FOR EACH WORD IN ERROR.

E0303 0001 AAAA WAS - S/B LATE READ CALL DSW  
XXXX XXXX

A LATE READ CALL WAS GENERATED AS A RESULT OF A MARK IN WD 92, AND  
THE DSW IS NOT CORRECT. THE MACHINE SHOULD GO NOT READY WITH REFEED  
SEL DOC LIGHT.

E0304 000R AAAA WAS - S/B INTERRUPT DSW ERROR  
XXXX XXXX

THE DSW SENSED IN THE INTERRUPT ROUTINE DID NOT APPEAR AS EXPECTED.

E0305 0002 AAAA WAS - S/B DSW, MASTER MARK NOT INDICATED  
XXXX XXXX

THIS MESSAGE INDICATES THE DSW IN WHICH THE MASTER MARK BIT WAS EX-  
PECTED BUT NOT RECEIVED. THIS MAY BE DUE TO CIRCUIT FAILURE, OR THE  
MASTER MARK SWITCH BEING OFF.

E0306 000R AAAA NO INTERRUPT DSW  
XXXX

A READ WAS INITIATED, BUT NO INTERRUPT OCCURRED WITHIN 2 SECONDS.  
THIS MAY BE DUE TO READY CIRCUITS, INTERRUPT GENERATION, OR FEED  
PROBLEMS.

5. COMMENTS

5.1 ROUTINE 1 - READ DOCUMENTS AND COMPARE

THIS ROUTINE READS THE ON-LINE PREPRINTED DOCUMENTS AND COMPARES EACH  
WORD PROGRAMMED IN THE 1231 TO A TABLE. ANY COMPARE ERRORS GIVE AN  
ERROR MESSAGE. IF THE CPU SELECT MARK IN WORD 91 IS PRESENT, AND NO  
DELAYS HAVE OCCURED FROM MESSAGES ON OVERLAP CONDITIONS, THE DOCUMENT  
IS SELECTED. IF THE O.D. MARKS IN WORD 92 ARE PRESENT, A LATE READ  
CALL CONDITION IS CREATED. ALL DSW'S ARE CHECKED FOR EXPECTED RE-  
SPONSE.

5.2 ROUTINE 2 - MASTER MARK TEST

THIS ROUTINE CHECK THE ABILITY OF THE 1231 TO RECOGNIZE THE MASTER  
MARK (OPTIONAL FEATURE). SPECIAL MASTER MARK DOCUMENTS ARE USED.  
THE ROUTINE MUST BE RUN WITH THE MASTER MARK SWITCH ON.

5.3 ROUTINE 3 - ALTERNATE SELECT

THIS ROUTINE FEEDS DOCUMENTS AND SELECTS EVERY OTHER ONE. THIS MAY  
BE USED FOR ADJUSTMENT AND CHECKING THE SELECT MECHANISM AND FEED.

5.4 ROUTINE 4 - READ DOCUMENTS, NO COMPARE

THIS ROUTINE READS DOCUMENTS INTO CORE. IT MAY BE USED FOR SCOPING  
THE DATA LINES, CHECKING THE FIELD SWITCHES, ETC.

1130-1231 OMPR F. T.

```

***** 0314 - 1231 OMPR FUNCTION TEST ***** 31400010
*****
***** EQUATE TABLE *****
***** THIS TABLE EQUATES TEST PROGRAM LABELS *****
***** TO THEIR EQUIVALENT DIAGNOSTIC MONITOR *****
***** ADDRESSES. *****
*****
-----
* MONITOR ENTRY ADDRESSES
-----
0160 BEGIN EQU /0160 BEGIN ROUTINE 31400120
0161 START EQU BEGIN+1 SUPERVISOR ROUTINE 31400130
0162 ERROR EQU START+1 ERROR LOG ROUTINE 31400140
0163 LOG EQU ERROR+1 STATUS LOG ROUTINE 31400150
0164 END EQU LOG+1 END ROUTINE 31400160
*
-----
* MONITOR CONTROL WORD ADDRESSES
-----
0165 RTNSW EQU END+1 ROUTINE START SW 31400210
0166 ERLCK EQU END+2 LOCK ON ERROR CONTROL 31400220
0167 LOGBY EQU END+3 I/O BUSY SW ADDR 31400230
0168 RLCF EQU END+4 RELOCATION FACTOR ADDR 31400240
*
-----
* INTERRUPT TRANSFER VECTOR ADDRESSES
-----
017A IL0 EQU /017A INTERRUPT LEVEL ZERO 31400290
018A IL1 EQU IL0+16 INTERRUPT LEVEL ONE 31400300
019A IL2 EQU IL1+16 INTERRUPT LEVEL TWO 31400310
01AA IL3 EQU IL2+16 INTERRUPT LEVEL THREE 31400320
01BA IL4 EQU IL3+16 INTERRUPT LEVEL FOUR 31400330
01BB RQTY EQU IL4+1 CONSOLE PRINTER REQUEST 31400340
01BC RQKB EQU RQTY+1 USE KEYBOARD REQUEST 31400350
01BD SVKB EQU RQKB+1 KB SERVICE REQUEST 31400360
*
*****
***** ORG **/05DC *****
*****
* THE MONITOR USES CORE LOCATIONS 0-05DC.
* FOR CONTENTS OF THESE ADDRESSES REFER
* TO THE DIAGNOSTIC MONITOR LISTING.
*****
***** PROGRAM CONTROL TABLE *****
*****
05DC 0 0314 PID DC /0314 PROGRAM ID 31400500
05DD 0 0000 RID DC ** ROUTINE ID 31400510
05DE 0 0000 RAD DC ** ROUTINE ADDR 31400520
05DF 0 0000 SW0 DC ** PROGRAM CONTROL 31400530
05E0 0 0000 SW1 DC ** ROUTINE SELECTION 31400540
05E1 0 0000 SW2 DC ** 31400550
05E2 0 0000 SW3 DC ** 31400560
05E3 1 05F0 DC LPTST LOOP ADDRESS 31400570
05E4 1 05EB DC STRT RESTART ADDRESS 31400580
05E5 0 0000 MLSCF DC ** ENTRY ONE 31400590
05E6 0 0000 DC ** ENTRY TWO 31400600
05E7 0 FFFF DC /FFFF TERMINATOR 31400610
*
*****
***** INITIALIZATION AND START *****
*****
05EB 00 44800160 BGIN BSI I BGIN 31400660
05EA 1 05DC DC PID PCT ADDRESS 31400670
31400680

```

1130-1231 OMPR F. T.

```

05EB 0 1000 * STRT NOP 31400690
*..... 31400700
* DFIN INT4,5,INT4 31400710
*..... 31400720
*..... 31400730
05EC 01 65000614 LDX L1 INT4 31400740
05EE 00 6D000185 STX L1 IL4-5 31400750
* 31400760
05F0 0 6100 LPTST LDX 1 0 31400770
05F1 0 69EB STX 1 RID SET TO START AT RTN ONE 31400780
05F2 0 4000 BSI CNTRL 31400790
* 31400800
***** SEQUENCE CONTROL ROUTINE *****
***** THIS ROUTINE CHECKS SW1 AND CONTROLS *****
***** THE SEQUENCE IN WHICH TEST ROUTINES *****
***** ARE RUN. *****
-----
*
05F3 0 0006 CNTRL DC ** 31400890
05F4 0 C0EB LD SW1 31400900
05F5 01 4C0805FE BSC L CN20,+ BR IF NO RTN SELECTD 31400910
* 31400920
05F7 0 D0E5 CN10 STO RID SAVE NEW RTN NUMBER 31400930
05F8 0 9015 S RIDCK 31400940
05F9 01 4C080604 BSC L CN30,+ BR IF VALID RTN 31400950
05FB 0 1810 SRA 16 31400960
05FC 0 D0E3 STO SW1 IF INVALID RTN GO 31400970
05FD 0 D0DF STO RID TO RTN ONE 31400980
* 31400990
05FE 01 740105DD CN20 MDX L RID,1 ADV TO NEXT RTN 31401000
0600 0 C0DC LD RID CHECK FOR END OF 31401010
0601 0 900D S RTNOM NORMAL SEQUENCE 31401020
0602 00 44800164 BSI I END,-Z END OF PROGRAM 31401030
* 31401040
0604 01 658035DD CN30 LDX I1 RID XRI=NEW ROUTINE NUMBER 31401050
0606 01 C500060F LD L1 RTTBL-1 FETCH RETURN ADRS 31401060
0608 0 D0D5 STJ RAD STORE NEW RTN ADDR 31401070
0609 0 D0DC STO MLSCF+1 SET MLSCF FOR RETURN 31401080
060A 00 D4000165 STO L RTNSW SET RTN START SW 31401090
060C 00 44800161 BSI I START GO TO MONITOR 31401100
* 31401110
060E 0 0004 RIDCK DC LRTN-RTTBL+1 31401120
060F 0 0001 RTNOM DC NRTN-RTTBL+1 31401130
* 31401140
----- ROUTINE ADDRESS TABLE -----
* 31401150
* 31401160
* 31401170
* 31401180
* DFIN RTN,RTN1,,RTN2,RTN3,RTN4 31401190
*..... 31401200
0610 1 0647 RTTBL DC PTM1 31401210
0610 NRTN EQU RTTBL 31401220
0611 1 06D0 DC RTN2 31401230
0612 1 06FD DC RTN3 31401240
0613 1 071C LRTN DC RTN4 31401250
***** INTERRUPT SERVICE SUBROUTINE *****
***** INT4 DC ** 31401260
***** INT4 DC ** 31401270
0614 0 0000 XIO SENSR SENSE RESET DSW 31401280
0615 0 0808 STO HDSWI STORE INTERRUPT DW 31401290
0616 0 D029 NOP 31401300
0617 0 1000 NOP USE FOR TRAP 70FF 31401310
* 31401320
* 31401330
*..... 31401340
* CALL BROUT,INT4 BRANCH OUT OF INT 31401350
*..... 31401360

```

1130-1231 OMPR F. T.

```

0618 01 670008C6      LDX L3 ZZRO1      31401370
061A 01 6F0005E6      STX L3 MLSCF+     31401380
061C 01 4C800614      BSC 1 INT4        BR OUT OF INTERRUPT 31401390
*                                                                31401400
*                                                                31401410
*          IOCC A EA      31401410
***** I/O C MMANDS ***** 31401420
*          BSS E          31401430
061E 0000              SENS DC 0          31401440
061E 0 0000            DC /4701          SENSE RESET          31401450
061F 0 4701            DC /4701          SENSE NO RESET      31401460
0620 0 0000            DC /4700          FEED DC 0            31401470
0621 0 4700            DC /4700          FEED DC 0            31401480
0622 0 0000            DC /4404          FEED START          31401490
0623 0 4404            DC /4404          DATA LOCATION      31401500
0624 1 0642            DC /4200          READ DC /4200       READ COMMAND         31401510
0625 0 4200            DC /4200          READ DC /4200       READ COMMAND         31401520
0626 0 0000            DC /4480          SELECT DC /4480     SELECT COMMAND       31401530
0627 0 4480            DC /4480          SELECT DC /4480     SELECT COMMAND       31401540
0628 0 0000            DC /4402          DISC DC /4402       DISCONNECT COMMAND  31401550
0629 0 4402            DC /4402          DISC DC /4402       DISCONNECT COMMAND  31401560
*                                                                31401570
***** CONSTANTS ***** 31401570
*                                                                31401580
062A 0 8686            H8686 DC /8686     31401590
062B 0 8486            H8486 DC /8486     NORMAL READ INT DSW 31401600
062C 0 005A            K005A DC /005A     DECIMAL 90          31401610
062D 0 FB7F            KFB7F DC /FB7F     ERROR OCCURRED CONSTANT 31401620
062E 0 E700            KE700 DC /E700     SELECT WORD 91      31401630
062F 0 78C3            K78C3 DC /78C3     LATE READ CALL WD 92 31401640
0630 0 FDFE            KDFE DC /FDFE     31401650
0631 0 EFFF            KEFFF DC /EFFF     31401660
0632 0 0107            H0107 DC /0107     DSW S/B--LATE READ CALL 31401670
0633 0 0001            K0001 DC /0001     ONE                 31401680
0634 0 0C82            K0C82 DC /0C82     OP COMPLETE DSW     31401690
0635 0 9686            H9686 DC /9686     MM DSW S/B          31401700
0636 0 04C0            K04C0 DC /04C0     OP COMP VARIABLE BITS 31401710
0637 0 7939            K7939 DC /7939     ERROR BITS READ RESP 31401720
0638 0 FB39            KFB39 DC /FB39     31401730
0639 0 FB3F            KFB3F DC /FB3F     31401740
063A 0 0200            K0200 DC /0200     31401750
063B 0 0800            K0800 DC /0800     31401760
*                                                                31401770
***** WORK AREA BUCKET ***** 31401780
*                                                                31401790
063C 0 0000            EOPAD DC **--      END OP ADDRESS      31401800
063D 0 0000            FRSW DC **--      FIRST RESPONSE SW   31401810
063E 0 0000            HWASI DC **--     DSW WAS INITIAL     31401820
063F 0 0000            HO2BE DC **--     SHOULD BE           31401830
0640 0 0000            HDSWI DC **--     INTERRUPT DSW       31401840
0641 0 0000            DWCNT DC **--     DEC WORD COUNT      31401850
0642 0 0000            HDWAS DC **--     DATA WORD          31401860
0643 0 0000            EROCC DC 0         ERROR OCCURRED SW   31401870
0644 0 0000            HDSWF DC **--     FEED BUSY DSW       31401880
0645 0 0000            HLRCD DC **--     DSW WAS--LATE READ CALL 31401890
0646 0 0000            HDSWN DC **--     NO INTERRUPT DSW    31401900
***** 31401910
***** ROUTINE ONE ***** 31401920
*                                                                31401930
* THIS ROUTINE READS TEST DOCUMENTS AND COM- * 31401940
* PARES THE DATA RECEIVED TO A TABLE. A ONE * 31401950
* BIT IN WORD 91 OF THE DOCUMENT SHOULD CAUSE * 31401960
* IT TO BE SELECTED. A 1,3,+6 BIT IN WORD 92 * 31401970
* SHOULD CAUSE A FEED CYCLE AND A LATE READ CALL * 31401980
***** 31401990
0647 01 6C00063C      RTN1 STX L EOPAD    SET END OP ADDRESS  31402000
0649 01 44000740      BSI L INIT         INITIALIZE 1231     31402010
064B 0 1010            SLA 16             31402020
064C 0 D0F4            STO DWCNT          CLEAR WORD COUNT    31402030
064D 0 D0F5            STO EROCC          CLEAR ERROR OCCURRED SW 31402040

```

1130-1231 OMPR F. T.

```

064E 01 0C000622      *          XIO L FEED      START FEED      31402050
*                                                                31402060
*          CALL WAIT,NINT   WAIT FOR INTRRUPT 31402070
*          *                                                                31402080
*          *                                                                31402090
0650 01 440008A2      R1A BSI L WAIT      TIMEOUT RETURN ADDR 31402100
0652 1 0731            DC NINT            31402110
*                                                                31402120
0653 01 44000780      *          BSI L CHECK    CHECK DSW FROM INT  31402130
*                                                                31402140
0655 01 0C000624      R1B XIO L READ      READ COMMAND        31402150
0657 01 65800641      LDX I1 DWCNT       WORD COUNT          31402160
0659 0 C0E8            LD HDWAS           DATA WORD           31402170
065A 01 F50007B9      EDR L1 TABLE     COMPARE TO TABLE   31402180
065C 01 4C200661      BSC L R1C,Z       BR IF DATA ERROR   31402190
*                                                                31402200
065E 01 74010641      R1B1 MDX L DWCNT,+1 INCR WORD COUNT     31402210
0660 0 70EF            MDX R1A           LOOP                 31402220
*                                                                31402230
0661 0 C0DF            R1C LD DWCNT       WORD COUNT           31402240
0662 01 9400062C      S L K005A         SUBTRACT 90         31402250
0664 01 4C10068E      BSC L R1D,-       BR IF CONTROL WORD  31402260
*                                                                31402270
0666 01 74010641      R1C1 MDX L DWCNT,+1 INCR WORD COUNT     31402280
0668 01 C50007B9      LD L1 TABLE     PROPER WORD         31402290
066A 01 D400063F      STO L HO2BE       STORE FOR LOG MSG   31402300
066C 01 C4000643      LD L EROCC        ERROR OCCURRED SW   31402310
066E 01 4C200681      BSC L R1C2,Z     BR IF ON             31402320
*                                                                31402330
0670 01 C400062D      LD L KFB7F        ERROR OCCURRED CONSTANT 31402340
0672 01 D4000643      STO L EROCC       SET ERROR OCCURRED SW 31402350
*                                                                31402360
*          *                                                                31402370
*          *                                                                31402380
*          *          CALL LOG,E02,AWSBE,ADERR,HDWAS,HO2BE,DWCNT 31402390
*          *          *                                                                31402400
0674 01 44000908      BSI L ZZERR       31402410
0676 0 0000            DC                31402420
0677 0 0002            DC /02            MSG NO              31402430
0678 0 0004            DC /0004          HEX-DEC SW         31402440
0679 0 0007            DC /0007          DATA ID           31402450
067A 1 081D            DC AWSBE          ALPHA1             31402460
067B 1 0840            DC ADERR          ALPHA2             31402470
067C 1 0642            DC HDWAS          31402480
067D 1 063F            DC HO2BE          31402490
067E 1 0641            DC DWCNT          31402500
*          *          BSC L R1A      BR TO CONTINUE 31402510
*          *          *                                                                31402520
*          *          *                                                                31402530
*          *          *          CALL LOG,E,HDWAS,HO2BE,DWCNT 31402540
*          *          *          *                                                                31402550
0681 01 44000908      R1C2 BSI L ZZERR  31402560
0683 0 0000            DC                31402570
0684 0 0000            DC /00            MSG NO              31402580
0685 0 0004            DC /0004          HEX-DEC SW         31402590
0686 0 8007            DC /8007          DATA ID           31402600
0687 0 0000            DC                ALPHA1             31402610
0688 0 0000            DC                ALPHA2             31402620
0689 1 0642            DC HDWAS          31402630
068A 1 063F            DC HO2BE          31402640
068B 1 0641            DC DWCNT          31402650
*          *          *                                                                31402660
068C 01 4C000650      *          BSC L R1A      BR TO CONTINE 31402670
*          *          *                                                                31402680
068E 01 94000633      R1D S L K001      SUB ONE             31402690
0690 0 4810            BSC -            SKIP IF COL 91     31402700
0691 0 700F            MDX R1D1         BR IF NOT           31402710
0692 01 C4000642      LD L HDWAS       DATA WORD          31402720

```

1130-1231 OMPR F. T.

```

0694 01 F400062E      EOR L KE700      SELECT WORD      31402730
0696 01 4C200666      BSC L RIC1,Z     BR IF DATA ERROR 31402740
*
0698 01 0C000620      XIO L SENSE      SENSE DSW        31402750
069A 0 1005           SLA 5            CHK OK TO SEL    31402760
069B 0 4810           BSC -           SKIP IF OK TO SEL 31402770
069C 0 702A           MDX RID2        BR IF TOO LATE   31402780
*
069D 01 0C000626      XIO L SELCT      SELECT COMMAND    31402790
069F 01 4C00065E      BSC L R1B1      BR TO PROCESS NEXT WD 31402800
*
06A1 01 94000633      RID1 S L K0001   SUBTRACT ONE     31402810
06A3 0 4810           BSC -           SKIP IF COL 92   31402820
06A4 0 70C1           MDX RIC1        BR IF DATA ERROR 31402830
06A5 01 C4000642      LD L HDWAS      LATE READ CALL WORD 31402840
06A7 01 F400062F      EOR L K7BC3     BR IF NOT PROPER 31402850
06A9 01 4C200666      BSC L RIC1,Z     BR IF NOT PROPER 31402860
*
06AB 01 0C000622      XIO L FEED      SET UP LATE READ CALL 31402870
*
*.....
* CALL WAIT,NINT
*.....
06AD 01 440008A2      BSI L WAIT      31402880
06AF 1 0731           DC NINT         31402890
*
*.....
* CALL DELAY,10000 WAIT 10 SECONDS
*.....
06B0 01 440008E2      BSI L DELAY     31402900
06B2 0 2710           DC 10000       DELAY FACTOR     31402910
*
06B3 01 0C000620      XIO L SENSE      SENSE DSW        31402920
06B5 01 D4000645      STO L HLRCD     STORE FOR MSG    31402930
06B7 01 F4000632      EOR L H0107     LATE READ CALL DSW 31402940
06B9 01 4C180647      BSC L RTN1,+    BR IF PROPER DSW 31402950
*
*.....
* CALL LOG,E03,AWSBE,ALRC,HLRCD,H0901
*.....
06BB 01 44000908      BSI L ZZERR     31402960
06BD 0 0000           DC             31402970
06BE 0 0003           DC /03         MSG NO           31402980
06BF 0 0000           DC /0000      HEX-DEC SW      31402990
06C0 0 0003           DC /0003      DATA ID        31403000
06C1 1 081D           DC AWSBE      ALPHA1          31403010
06C2 1 0846           DC ALRC       ALPHA2          31403020
06C3 1 0645           DC HLRCD     31403030
06C4 1 0632           DC H0107     31403040
*
06C5 01 4C000647      BSC L RTN1      BR TO BEGIN AGAIN 31403050
*
*.....
* CALL LOG,A03,ATOLA
*.....
06C7 01 44000900      RID2 BSI L ZZLOG 31403060
06C9 0 000?           DC /03         MSG NO           31403070
06CA 0 0000           DC /0000      HEX-DEC SW      31403080
06CB 0 0000           DC /0000      DATA ID        31403090
06CC 0 0000           DC           ALPHA1          31403100
06CD 1 0864           DC ATOLA      ALPHA2          31403110
*
06CE 01 4C00065E      BSC L R1B1      BR TO CONTINUE   31403120
*
*****
* ROUTINE TWO
*****
31403130
31403140
31403150
31403160
31403170
31403180
31403190
31403200
31403210
31403220
31403230
31403240
31403250
31403260
31403270
31403280
31403290
31403300
31403310
31403320
31403330
31403340
31403350
31403360
31403370
31403380
31403390
31403400

```

1130-1231 OMPR F. T.

```

* MASTER MARK TEST 31403410
* THIS TEST READS DOCUMENTS MARKED 31403420
* AS MASTERS. NO DATA IS COMPARED. THE 31403430
* DSW IS CHECKED TO INSURE THAT THE 31403440
* MASTER MARK IS RECOGNIZED. 31403450
*****
* 31403460
* 31403470
*.....
* CALL LOG,A,AMMTS 31403480
*.....
RTN2 BSI L ZZLOG 31403490
DC /00 31403500
DC /0000 MSG NO 31403510
DC /8000 HEX-DEC SW 31403520
DC /8000 DATA ID 31403530
DC ALPHA1 31403540
DC ALPHA2 31403550
DC AMMTS 31403560
*
06D0 01 44000900 31403570
06D2 0 0000 DC /00 MSG NO 31403580
06D3 0 0000 DC /0000 HEX-DEC SW 31403590
06D4 0 8000 DC /8000 DATA ID 31403600
06D5 0 0000 DC ALPHA1 31403610
06D6 1 0888 DC ALPHA2 31403620
*
06D7 01 6C00063C 31403630
06D9 01 44000740 R2A BSI L INIT SET END OP ADDRESS 31403640
INITIALIZE 1231 31403650
*
06DB 01 0C000622 31403660
* XIO L FEED FEED A DOC 31403670
*
*.....
* CALL WAIT,NINT
*.....
06DD 01 440008A2 31403680
06DF 1 0731 DC NINT TIMEOUT RETURN ADDR 31403690
*
06E0 01 C4000640 31403700
06E2 0 1003 LD L HDSWI INTERRUPT DSW 31403710
06E3 01 4C1006F2 31403720
06E5 01 C4000640 31403730
06E7 01 E4000631 31403740
06E9 01 D4000640 31403750
06EB 01 44000780 31403760
06ED 01 0C000624 31403770
06EF 01 0C000628 31403780
06F1 0 70E7 MDX R2A BR TO READ NEXT DOC 31403790
*
*.....
* CALL LOG,E05,AWSBE,AMMNR,HDSWI,H9686
*.....
06F2 01 44000908 31403800
06F4 0 0000 R2B1 BSI L ZZERR 31403810
06F5 0 0005 DC 31403820
06F6 0 0000 DC /05 MSG NO 31403830
06F7 0 0000 DC /0000 HEX-DEC SW 31403840
06F8 1 081D DC /0003 DATA ID 31403850
06F9 1 0891 DC AWSBE ALPHA1 31403860
06FA 1 0640 DC AMMNR ALPHA2 31403870
06FB 1 0635 DC HDSWI 31403880
DC H9686 31403890
*
06FC 0 70E8 31403900
* MDX R2B BR TO READ AGAIN 31403910
*
*****
* ROUTINE THREE 31403920
*****
* ALTERNATE SELECT TEST 31403930
* THIS TEST SELECTS ALTERNATE DOCUMENTS 31403940
*****
* 31403950
*.....
* CALL LOG,A,AALTS 31403960
*.....
RTN3 BSI L ZZLOG 31403970
DC /00 MSG NO 31403980
DC /0000 HEX-DEC SW 31403990
31404000
31404010
31404020
31404030
31404040
31404050
06FD 01 44000900 31404060
06FF 0 0000 DC /00 MSG NO 31404070
0700 0 0000 DC /0000 HEX-DEC SW 31404080

```



1130-1231 OMPR F. T.

```

0701 0 8000      DC      /8000      DATA ID      31404090
0702 0 0000      DC      ALPHAI      31404100
0703 1 087C      DC      AALTS      ALPHA2      31404110
*              DC      31404120
R3A  LDX  1 2      XRI FOR SELECT ALT SWITCH 31404130
0704 0 6102      STX L  EOPAD      SET END OP ADDRESS 31404140
0705 01 6C00063C BSI L  INIT      INITIALIZE 1231    31404150
0707 01 44000740 XIO L  FEED      FEED A DOC        31404160
0709 01 0C000622 *              31404170
*              31404180
*.....*          31404190
* CALL  WAIT,NINT 31404200
*.....*          31404210
0708 01 440008A2 BSI L  WAIT      TIMEOUT RETURN ADDR 31404220
070D 1 0731      DC      NINT      31404230
*              31404240
070E 01 44000780 *              31404250
*              31404260
*              31404270
*              31404280
*              31404290
*              31404300
*              31404310
*              31404320
*              31404330
*              31404340
0714 01 0C000626 XIO L  SELCT     SELECT COMMAND     31404350
0716 01 0C000628 XIO L  DISC      DISCONNECT COMMAND 31404360
0718 0 70EB      MDX      R3A2     LOOP              31404370
*              31404380
*              31404390
*              31404400
*              31404410
*              31404420
*              31404430
*              31404440
*              31404450
*              31404460
*.....*          31404470
* CALL  LOG,A,ARNC 31404480
*.....*          31404490
071C 01 44000900 RTN4 BSI L  ZZLOG  MSG NO      31404500
071E 0 0000      DC      /00      HEX-DEC SW      31404510
071F 0 0003      DC      /0000     DATA ID      31404520
0720 0 8000      DC      /8000     ALPHA1      31404530
0721 0 0000      DC      ALPHA2     31404540
0722 1 086E      DC      ARNC      31404550
*              31404560
0723 01 6C00063C STX L  EOPAD      SET END OP ADDRESS 31404570
0725 01 44000740 BSI L  INIT      INITIALIZE 1231    31404580
0727 01 0C000622 XIO L  FEED      FEED A DOC        31404590
*              31404600
*.....*          31404610
* CALL  WAIT,NINT 31404620
*.....*          31404630
0729 01 440008A2 R4B  BSI L  WAIT      TIMEOUT RETURN ADDR 31404640
072B 1 0731      DC      NINT      31404650
*              31404660
072C 01 44000780 *              31404670
*              31404680
*              31404690
*              31404700
*              31404710
*              31404720
*              31404730
*              31404740
*              31404750
*              31404760
0731 01 0C000620 NINT XIO L  SENSE  SENSE 1231 STATUS 31404770

```

1130-1231 OMPR F. T.

```

0733 01 D4000646 *      STO L  HDSWN      STORE FOR MSG      31404770
*              31404780
*.....*          31404790
* CALL  LOG,E06,ANINT,HDSWN 31404800
*.....*          31404810
*              31404820
0735 01 44000908 BSI L  ZZERR      31404830
0737 0 0000      DC      31404840
0738 0 0006      DC      /06      MSG NO      31404850
0739 0 0000      DC      /0000     HEX-DEC SW    31404860
073A 0 0001      DC      /0001     DATA ID     31404870
073B 0 0000      DC      ALPHA1     31404880
073C 1 0850      DC      ANINT      ALPHA2      31404890
073D 1 0646      DC      HDSWN     31404900
*              31404910
073E 01 4C000741 *      BSC L  INIT+1   BR TO RETURN VIA INIT 31404920
*              31404930
*.....*          31404940
*              31404950
*.....*          31404960
*              31404970
0740 0 0000      INIT DC      *-*      DISCONNECT COMMAND 31404980
0741 01 0C000628 XIO L  DISC      16      31404990
0743 0 1010      SLA      16      CLEAR FIRST RESP SW 31405000
0744 01 D400063D STO L  FRSW      SENSE DSW      31405010
0746 01 0C00061E XIO L  SENSR     IGNORE SS TIMINGS 31405020
0748 01 E400062D AND L  KFB7F     BR IF DSW CLEAR    31405030
074A 01 D400063E STO L  HWASI     31405040
074C 01 4C980740 BSC I  INIT,+-- 31405050
*              31405060
074E 01 4C040764 BSC L  IN2,E     BR IF NOT READY    31405070
0750 0 1806      SRA      6      CHECK HOPPER EMPTY 31405080
0751 01 4C040775 BSC L  IN3,E     BR IF HOPPER EMPTY 31405090
0753 0 1010      SLA      16      CLEAR ACC          31405100
0754 01 D400063F STO L  HQ2BE     STORE FOR MSG      31405110
*              31405120
*.....*          31405130
* CALL  LOG,E01,AWSBE,ADSWI,HWASI,HQ2BE 31405140
*.....*          31405150
*              31405160
0756 01 44000908 BSI L  ZZERR      31405170
0758 0 0000      DC      31405180
0759 0 0001      DC      /01      MSG NO      31405190
075A 0 0000      DC      /0000     HEX-DEC SW    31405200
075B 0 0003      DC      /0003     DATA ID     31405210
075C 1 081D      DC      AWSBE     ALPHA1      31405220
075D 1 0822      DC      ADSWI     ALPHA2      31405230
075E 1 063E      DC      HWASI     31405240
075F 1 063F      DC      HQ2BE     31405250
*              31405260
*              31405270
*.....*          31405280
* CALL  DELAY,8000 DELAY 8 SECONDS 31405290
*.....*          31405300
0760 01 440008E2 BSI L  DELAY     DELAY FACTOR      31405310
0762 0 1F40      DC      8000     31405320
*              31405330
0763 0 70DD      MDX      INIT+1   TRY AGAIN         31405340
*              31405350
*              31405360
*.....*          31405370
*              31405380
*.....*          31405390
0764 01 440008E2 IN2  BSI L  DELAY     DELAY FACTOR      31405400
0766 0 3A98      DC      15000    31405410
*              31405420
0767 01 0C000620 XIO L  SENSE     SENSE DSW        31405430
0769 01 4C04076D BSC L  INIT4,E   BR IF STILL NOT READY 31405440
076B 0 70DC      MDX      IN1      BR TO CHK DSW

```

1130-1231 OMPR F. T.

```

076C 0 7008      MDX   IN3     BR TO CHK DSW      31405450
*
*.....
*   CALL   LOG,A01,ANRDY      31405460
*.....
*   CALL   LOG,A01,ANRDY      31405470
*.....
076D 01 44000900  INIT4 BSI L ZZLOG      31405480
076F 0 0001      DC     /01     MSG NO      31405490
0770 0 0000      DC     /0000   HEX-DEC SW    31405500
0771 0 0000      DC     /0000   DATA ID      31405510
0772 0 0000      DC     /0000   ALPHA1       31405520
0773 1 082E      DC     ANRDY   ALPHA2       31405530
*
*.....
0774 0 70CC      MDX   INIT+1      31405540
*
*.....
*   CALL   LOG,A02,AHOPE      31405550
*.....
0775 01 44000900  IN3   BSI L ZZLOG      31405560
0777 0 0002      DC     /02     MSG NO      31405570
0778 0 0000      DC     /0000   HEX-DEC SW    31405580
0779 0 0000      DC     /0000   DATA ID      31405590
077A 0 0000      DC     /0000   ALPHA1       31405600
077B 1 0836      DC     AHOPE   ALPHA2       31405610
*
*.....
*   CALL   DELAY,15000        31405620
*.....
077C 01 440008E2  BSI L DELAY      31405630
077E 0 3A98      DC     15000   DELAY FACTOR  31405640
*
*.....
077F 0 70C1      MDX   INIT+1      31405650
*
*.....
*****
*   CHECK DSW SUBROUTINE      31405660
*.....
*   THIS SUBROUTINE CHECKS THE LAST
*   INTERRUPT DSW FOR PROPER RESPONSES.
*****
0780 0 0000      CHECK DC  *--*      31405670
0781 01 C4000640  LD     L HDSWI      31405680
0783 0 4810      BSC   -             31405690
0784 0 7022      MDX   CHK3         BR IF NOT      31405700
*
*.....
0785 01 7400063D  MDX L FRSW,0      SKIP IF FIRST RESP  31405710
0787 0 7007      MDX   CHK1         BR IF NOT      31405720
*
*.....
0788 01 6C00063D  STX L FRSW        SET FIRST RESP SW  31405730
078A 01 F400062A  EDR L H8686       PROPER FIRST RESP  31405740
078C 01 4C980780  BSC I CHECK,+--  BR BACK IF PROPER  31405750
*
*.....
078E 0 7004      MDX   CHK2         CHECK ERRORS    31405760
*
*.....
078F 01 F400062B  CHK1 EDR L H8486   PROPER 2ND RESP    31405770
0791 01 4C980780  BSC I CHECK,+--  BR BACK IF PROPER  31405780
*
*.....
0793 01 E4000638  CHK2 AND L KFB39   ALL ERROR BITS     31405790
0795 01 4C980780  BSC I CHECK,+--  BR BACK IF NO ERRORS 31405800
*
*.....
0797 01 F4000640  EDR L HDSWI       RESTORE S/B DSW    31405810
0799 01 D400063F  STO L H02BE       STORE FOR MSG      31405820
*
*.....
*   CALL   LOG,E04,AWSBE,AINDS,HDSWI,H02BE
*.....

```

1130-1231 OMPR F. T.

```

079B 01 4400090B  RIF3 BSI L ZZERR      31406130
079D 0 0000      DC     /04     MSG NO      31406140
079E 0 0004      DC     /0000   HEX-DEC SW    31406150
079F 0 0000      DC     /0003   DATA ID      31406160
07A0 0 0003      DC     /0003   ALPHA1       31406170
07A1 1 081D      DC     AWSBE   ALPHA2       31406180
07A2 1 0859      DC     AINDS   ALPHA2       31406190
07A3 1 0640      DC     HDSWI   31406200
07A4 1 063F      DC     H02BE   31406210
*
*.....
07A5 01 4C800780  BSC I CHECK        31406220
*
*.....
07A7 01 F4000634  CHK3 EDR L KOC82   NORMAL OP COMP DSW  31406230
07A9 01 E4000639  AND L KFB3F       IGNORE SS TIMINGS AND HE 31406240
07AB 01 4C80063C  BSC I EOPAD       BR IF PROPER END OP  31406250
*
*.....
*   CALL   LOG,E05,AWSBE,AINDS,HDSWI,H9686
*.....
07AD 01 4400090B  BSI L ZZERR      31406260
07AF 0 0000      DC     /05     MSG NO      31406270
07B0 0 0005      DC     /0000   HEX-DEC SW    31406280
07B1 0 0000      DC     /0003   DATA ID      31406290
07B2 0 0003      DC     /0003   ALPHA1       31406300
07B3 1 081D      DC     AWSBE   ALPHA2       31406310
07B4 1 0859      DC     AINDS   ALPHA2       31406320
07B5 1 0640      DC     HDSWI   31406330
07B6 1 0635      DC     H9686   31406340
*
*.....
07B7 01 4C80063C  BSC I EOPAD       31406350
*
*.....
*****
*   DATA TABLE
*****
TABLE DC /4200 1
07B9 0 4200      DC     /4100   2
07BA 0 4100      DC     /1181   3
07BB 0 1181      DC     /2200   4
07BC 0 2200      DC     /F3C3   5
07BD 0 F3C3      DC     /8803   6
07BE 0 8803      DC     /8741   7
07BF 0 8741      DC     /9582   8
07C0 0 9582      DC     /0642   9
07C1 0 0642      DC     /5F81   10
07C2 0 5F81      DC     /8A43   11
07C3 0 8A43      DC     /EB42   12
07C4 0 EB42      DC     /B200   13
07C5 0 B200      DC     /4200   14
07C6 0 4200      DC     /EDC3   15
07C7 0 EDC3      DC     /E241   16
07C8 0 E241      DC     /7882   17
07C9 0 7882      DC     /E803   18
07CA 0 E803      DC     /5EC1   19
07CB 0 5EC1      DC     /0702   20
07CC 0 0702      DC     /7380   21
07CD 0 7380      DC     /5CC0   22
07CE 0 5CC0      DC     /EE42   23
07CF 0 EE42      DC     /0783   24
07D0 0 0783      DC     /5E40   25
07D1 0 5E40      DC     /E803   26
07D2 0 E803      DC     /F2C2   27
07D3 0 F2C2      DC     /4E82   28
07D4 0 4E82      DC     /4200   29
07D5 0 4200      DC

```

1130-1231 OMPR F. T.

07D6	0	0603	DC	/0603	30	31406810
07D7	C	A402	DC	/A402	31	31406820
07D8	0	4100	DC	/4100	32	31406830
07D9	0	1181	DC	/1181	33	31406840
07DA	0	0941	DC	/0941	34	31406850
07DB	0	19C2	DC	/19C2	35	31406860
07DC	0	E680	DC	/E680	36	31406870
07DD	0	F642	DC	/F642	37	31406880
07DE	0	EEC3	DC	/EEC3	38	31406890
07DF	0	F801	DC	/F801	39	31406900
07E0	0	5382	DC	/5382	40	31406910
07E1	0	4842	DC	/4842	41	31406920
07E2	0	58C1	DC	/58C1	42	31406930
07E3	0	9582	DC	/9582	43	31406940
07E4	0	8D42	DC	/8D42	44	31406950
07E5	0	9DC1	DC	/9DC1	45	31406960
07E6	0	4200	DC	/4200	46	31406970
07E7	0	4100	DC	/4100	47	31406980
07E8	0	1181	DC	/1181	48	31406990
07E9	0	2200	DC	/2200	49	31407000
07EA	0	F3C3	DC	/F3C3	50	31407010
07EB	0	8803	DC	/8803	51	31407020
07EC	0	8741	DC	/8741	52	31407030
07ED	0	9582	DC	/9582	53	31407040
07EE	0	0642	DC	/0642	54	31407050
07EF	0	5F81	DC	/5F81	55	31407060
07F0	0	8A43	DC	/8A43	56	31407070
07F1	0	EB42	DC	/EB42	57	31407080
07F2	0	B200	DC	/B200	58	31407090
07F3	0	4200	DC	/4200	59	31407100
07F4	0	EDC3	DC	/EDC3	60	31407110
07F5	0	E241	DC	/E241	61	31407120
07F6	0	7882	DC	/7882	62	31407130
07F7	0	E803	DC	/E803	63	31407140
07F8	0	5EC1	DC	/5EC1	64	31407150
07F9	0	0702	DC	/0702	65	31407160
07FA	0	7380	DC	/7380	66	31407170
07FB	0	5CC0	DC	/5CC0	67	31407180
07FC	0	EE42	DC	/EE42	68	31407190
07FD	0	0783	DC	/0783	69	31407200
07FE	0	5E40	DC	/5E40	70	31407210
07FF	0	E803	DC	/E803	71	31407220
0800	0	F2C2	DC	/F2C2	72	31407230
0801	0	4E82	DC	/4E82	73	31407240
0802	0	4200	DC	/4200	74	31407250
0803	0	0603	DC	/0603	75	31407260
0804	0	A402	DC	/A402	76	31407270
0805	0	4100	DC	/4100	77	31407280
0806	0	1181	DC	/1181	78	31407290
0807	0	0941	DC	/0941	79	31407300
0808	0	19C2	DC	/19C2	80	31407310
0809	0	E680	DC	/E680	81	31407320
080A	0	F642	DC	/F642	82	31407330
080B	0	EEC3	DC	/EEC3	83	31407340
080C	0	F801	DC	/F801	84	31407350
080D	0	5382	DC	/5382	85	31407360
080E	0	4842	DC	/4842	86	31407370
080F	0	58C1	DC	/58C1	87	31407380
0810	0	9582	DC	/9582	88	31407390
0811	0	8D42	DC	/8D42	89	31407400
0812	0	9DC1	DC	/9DC1	90	31407410
0813	0	A702	DC	/A702	91	31407420
0814	0	29C2	DC	/29C2	92	31407430
0815	0	0003	DC	/0003	93	31407440
0816	0	0003	DC	/0003	94	31407450
0817	0	0003	DC	/0003	95	31407460
0818	0	0082	DC	/0082	96	31407470
0819	0	8400	DC	/8400	97	31407480

1130-1231 OMPR F. T.

081A	0	8001	DC	/8001	98	31407490
081B	0	0042	DC	/0042	99	31407500
081C	0	80C1	DC	/80C1	100	31407510
						31407520
*****						31407530
* MESSAGES						31407540
*****						31407550
* .....						31407560
* ALPH WAS S/B						31407570
* .....						31407580
081D	0	903C	AWSBE	DC	/903C	31407590
081E	0	9821	DC	/9821		31407600
081F	0	2198	DC	/2198		31407610
0820	0	8C18	DC	/8C18		31407620
0821	0	FFFF	DC	/FFFF		31407630
* .....						31407640
* .....						31407650
* .....						31407660
* .....						31407670
* ALPH DSW ERROR INITIALIZING						31407680
* .....						31407690
* .....						31407700
0822	0	3098	ADSWI	DC	/3098	31407710
0823	0	9021	DC	/9021		31407720
0824	0	3460	DC	/3460		31407730
0825	0	6050	DC	/6050		31407740
0826	0	6021	DC	/6021		31407750
0827	0	2074	DC	/2074		31407760
0828	0	209C	DC	/209C		31407770
0829	0	203C	DC	/203C		31407780
082A	0	5C20	DC	/5C20		31407790
082B	0	A020	DC	/A020		31407800
082C	0	7414	DC	/7414		31407810
082D	0	FFFF	DC	/FFFF		31407820
* .....						31407830
* .....						31407840
* .....						31407850
* ALPH 1231 NOT READY						31407860
* .....						31407870
082E	0	FCDB	ANRDY	DC	/FCDB	31407880
082F	0	DCFC	DC	/DCFC		31407890
0830	0	2174	DC	/2174		31407900
0831	0	509C	DC	/509C		31407910
0832	0	2160	DC	/2160		31407920
0833	0	343C	DC	/343C		31407930
0834	0	30A4	DC	/30A4		31407940
0835	0	FFFF	DC	/FFFF		31407950
* .....						31407960
* .....						31407970
* .....						31407980
* ALPH 1231 HOPPER EMPTY						31407990
* .....						31408000
0836	0	FCDB	AHOPE	DC	/FCDB	31408010
0837	0	DCFC	DC	/DCFC		31408020
0838	0	2124	DC	/2124		31408030
0839	0	5054	DC	/5054		31408040
083A	0	5434	DC	/5434		31408050
083B	0	6021	DC	/6021		31408060
083C	0	3470	DC	/3470		31408070
083D	0	549C	DC	/549C		31408080
083E	0	A421	DC	/A421		31408090
083F	0	FFFF	DC	/FFFF		31408100
* .....						31408110
* .....						31408120
* .....						31408130
* ALPH DATA ERROR						31408140
* .....						31408150
0840	0	303C	ADERR	DC	/303C	31408160

1130-1231 OMPR F. T.

```

0841 0 9C3C      DC      /9C3C
0842 0 2134      DC      /2134
0843 0 6060      DC      /6060
0844 0 5060      DC      /5060
0845 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  LATE READ CALL DSW
*.....*
0846 0 5C3C      ALRC   DC      /5C3C
0847 0 9C34      DC      /9C34
0848 0 2160      DC      /2160
0849 0 343C      DC      /343C
084A 0 3021      DC      /3021
084B 0 1C3C      DC      /1C3C
084C 0 5C5C      DC      /5C5C
084D 0 2130      DC      /2130
084E 0 9890      DC      /9890
084F 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  NO INTERRUPT DSW
*.....*
0850 0 7450      ANINT  DC      /7450
0851 0 2120      DC      /2120
0852 0 749C      DC      /749C
0853 0 3460      DC      /3460
0854 0 6080      DC      /6080
0855 0 549C      DC      /549C
0856 0 2130      DC      /2130
0857 0 9890      DC      /9890
0858 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  INTERRUPT DSW ERROR
*.....*
0859 0 2074      AINDS  DC      /2074
085A 0 9C34      DC      /9C34
085B 0 6060      DC      /6060
085C 0 8054      DC      /8054
085D 0 9C21      DC      /9C21
085E 0 3098      DC      /3098
085F 0 9021      DC      /9021
0860 0 3460      DC      /3460
0861 0 6050      DC      /6050
0862 0 6021      DC      /6021
0863 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  TOD LATE TO SELECT
*.....*
0864 0 9C50      ATOLA  DC      /9C50
0865 0 5021      DC      /5021
0866 0 5C3C      DC      /5C3C
0867 0 9C34      DC      /9C34
0868 0 219C      DC      /219C
0869 0 5021      DC      /5021
086A 0 9834      DC      /9834
086B 0 5C34      DC      /5C34
086C 0 1C9C      DC      /1C9C
086D 0 FFFF      DC      /FFFF
*
*
*.....*

```

```

31408170
31408180
31408190
31408200
31408210
31408220
31408230
31408240
31408250
31408260
31408270
31408280
31408290
31408300
31408310
31408320
31408330
31408340
31408350
31408360
31408370
31408380
31408390
31408400
31408410
31408420
31408430
31408440
31408450
31408460
31408470
31408480
31408490
31408500
31408510
31408520
31408530
31408540
31408550
31408560
31408570
31408580
31408590
31408600
31408610
31408620
31408630
31408640
31408650
31408660
31408670
31408680
31408690
31408700
31408710
31408720
31408730
31408740
31408750
31408760
31408770
31408780
31408790
31408800
31408810
31408820
31408830
31408840

```

1130-1231 OMPR F. T.

```

* ALPH  READ, NO DATA COMPARE TEST
*.....*
086E 0 6034      ARNC   DC      /6034
086F 0 3C30      DC      /3C30
0870 0 8021      DC      /8021
0871 0 7450      DC      /7450
0872 0 2130      DC      /2130
0873 0 3C9C      DC      /3C9C
0874 0 3C21      DC      /3C21
0875 0 1C50      DC      /1C50
0876 0 7054      DC      /7054
0877 0 3C60      DC      /3C60
0878 0 3421      DC      /3421
0879 0 9C34      DC      /9C34
087A 0 989C      DC      /989C
087B 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  ALTERNATE SELECT TEST
*.....*
087C 0 3C5C      AALTS  DC      /3C5C
087D 0 9C34      DC      /9C34
087E 0 6074      DC      /6074
087F 0 3C9C      DC      /3C9C
0880 0 3421      DC      /3421
0881 0 9834      DC      /9834
0882 0 5C34      DC      /5C34
0883 0 1C9C      DC      /1C9C
0884 0 219C      DC      /219C
0885 0 3498      DC      /3498
0886 0 9C21      DC      /9C21
0887 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  MASTER MARK TEST
*.....*
0888 0 703C      AMHTS  DC      /703C
0889 0 989C      DC      /989C
088A 0 3460      DC      /3460
088B 0 2170      DC      /2170
088C 0 3C60      DC      /3C60
088D 0 5821      DC      /5821
088E 0 9C34      DC      /9C34
088F 0 989C      DC      /989C
0890 0 FFFF      DC      /FFFF
*
*
*.....*
* ALPH  DSW, MASTER MARK NOT INDICATED
*.....*
0891 0 3098      AMNMR  DC      /3098
0892 0 9080      DC      /9080
0893 0 2121      DC      /2121
0894 0 703C      DC      /703C
0895 0 989C      DC      /989C
0896 0 3460      DC      /3460
0897 0 2170      DC      /2170
0898 0 3C60      DC      /3C60
0899 0 5821      DC      /5821
089A 0 7450      DC      /7450
089B 0 9C21      DC      /9C21
089C 0 2074      DC      /2074
089D 0 3020      DC      /3020
089E 0 1C3C      DC      /1C3C
089F 0 9C34      DC      /9C34
08A0 0 3021      DC      /3021

```

```

31408850
31408860
31408870
31408880
31408890
31408900
31408910
31408920
31408930
31408940
31408950
31408960
31408970
31408980
31408990
31409000
31409010
31409020
31409030
31409040
31409050
31409060
31409070
31409080
31409090
31409100
31409110
31409120
31409130
31409140
31409150
31409160
31409170
31409180
31409190
31409200
31409210
31409220
31409230
31409240
31409250
31409260
31409270
31409280
31409290
31409300
31409310
31409320
31409330
31409340
31409350
31409360
31409370
31409380
31409390
31409400
31409410
31409420
31409430
31409440
31409450
31409460
31409470
31409480
31409490
31409500
31409510
31409520

```

1130-1231 OMPR F. T.

```

08A1 0 FFFF          DC      /FFFF          31409530
*
*
*
*****
DMIG SUPPLIED SUBROUTINES
*****
*
*****
INTERRUPT WAIT ROUTINE
*****
*
THIS SUBROUTINE WAITS FOR INTERRUPT.
WHEN THE INTERRUPT IS RECEIVED IT WILL
CHECK THE ROUTINE SELECTION SWITCH.
IF A NEW ROUTINE HAS BEEN SELECTED IT
WILL BRANCH TO THE CONTROL ROUTINE.
*-----*
*
08A2 0 0000          WAIT DC      *--
08A3 01 440008CC      BSI L  ZZSAV      SAVE MACHINE STATUS
08A5 00 65001000      LDX L1 /1000      SET INTERRUPT
08A7 0 691D           STX 1  ZZWCT      WAIT COUNT
*
08A8 01 650008B1      ZZW01 LDX L1 ZZW03
08AA 01 740005E6      MDX L  MLSCF+1,0 CHECK FOR INTERRUPT
08AC 0 700E           MDX  ZZW04      BRANCH IF INTRPT OCCURED
*
08AD 01 6D0005E5      ZZW02 STX L1 MLSCF      SET RETURN ADDRESS
08AF 00 44800161      BSI I  START      GO TO MONITOR
*
08B1 01 74FF08C5      ZZW03 MDX L  ZZWCT,-1 DECREMENT WAIT CNT
08B3 0 70F4           MDX  ZZW01
*
08B4 01 C48008A2      LD  I  WAIT      ENTER HERE IF NO INTRPT
08B6 0 D0EB           STO  WAIT      SET NO INTERPT RETURN
08B7 01 650008C8      LDX L1 ZZR02
08B9 01 6D0005E6      STX L1 MLSCF+1
*
08BB 01 C40005E0      ZZW04 LD  L  SWI
08BD 00 44880161      BSI I  START,+   CK FOR ROUTINE SELECT
08BF 01 940005DD      S  L  RID      CK FOR NEW ROUTINE
08C1 01 442005F3      BSI L  CNTRL,Z  BR IF NEW ROUTINE
08C3 00 44800161      BSI I  START      GO TO MONITOR
*
08C5 0 0000          ZZWCT DC      *--
*
*-----*
*
RETURN HERE AFTER INTERRUPT
*-----*
08C6 01 740108A2      ZZR01 MDX L  WAIT,+1
08C8 01 440008D4      ZZR02 BSI L  ZZRES RESTORE MACHINE STATUS
08CA 01 4C8008A2      BSC I  WAIT      RETURN
*
*****
*****
SAVE MACHINE STATUS
*****
*
08CC 0 0000          ZZSAV DC      *--
08CD 0 D812           STD  ZZSAQ
08CE 0 2807           STS  ZZSTS
08CF 0 6908           STX 1  ZZS1+1
08D0 0 6A09           STX 2  ZZS2+1
08D1 0 660A           STX 3  ZZS3+1
08D2 01 4C8008CC      BSC I  ZZSAV
*
*****
*****
RESTORE MACHINE STATUS
*****

```

1130-1231 OMPR F. T.

```

*****
*****
ZZRES DC      *--
08D4 0 0000          LDD  ZZSAQ
08D5 0 C80A          ZZSTS LDS      *--
08D6 0 2000          ZZS1 LDX L1 *--
08D7 00 65000000     ZZS2 LDX L2 *--
08D9 00 66000000     ZZS3 LDX L3 *--
08DB 00 67000000     BSC I  ZZRES
08DD 01 4C8008D4     ZZSAQ BSS E 2
08E0 0002
*
*
*****
*****
DELAY SUBROUTINE
*****
*
08E2 0 0000          DELAY DC      *--
08E3 01 440008CC      BSI L  ZZSAV
08E5 01 C48008E2      LD  I  DELAY      GET DELAY FACTOR
08E7 0 D017          STO  ZZDFA
08E8 01 740108E2      MDX L  DELAY,+1
08EA 01 650008F8     ZZD02 LDX L1 ZZD01
08EC 01 6D0005E6     STX  L1 MLSCF+1
*
*
08EE 01 C40005E0      LD  L  SWI
08F0 00 44880161     BSI I  START,+   CK FOR ROUTINE SELECT
08F2 01 940005DD     S  L  RID      CK FOR NEW ROUTINE
08F4 01 442005F3     BSI L  CNTRL,Z  BR IF NEW RNT
08F6 00 44800161     BSI I  START
*
08F8 01 74FF08FF     ZZD01 MDX L  ZZDFA,-1 DECRE COUNT
08FA 0 70EF          MDX  ZZD02      BR
08FB 01 440008D4     BSI L  ZZRES      RESTORE SYSTEM
08FD 01 4C8008E2     BSC I  DELAY
08FF 0 0000          ZZDFA DC      *-- DELAY FACTOR
*
*
*****
*****
PRINT STATUS MESSAGE SUBROUTINE
*****
*
0900 0 0000          ZZLOG DC      *--
0901 0 40CA          BSI  ZZSAV      SAVE STATUS
0902 01 67800900     LDX 13 ZZLOG
0904 0 4015          BSI  ZZTCM      SET UP MESSAGE TABLE
0905 00 44800163     BSI I  LOG      PRINT MESSAGE
0907 1 0932          DC  ZZTBL
0908 0 40CB          BSI  ZZRES      RESTORE STATUS
0909 01 4C800900     BSC I  ZZLOG
*
*
*****
*****
PRINT ERROR MESSAGE SUBROUTINE
*****
*
090B 0 0000          ZZERR DC      *--
090C 0 40BF          BSI  ZZSAV      SAVE STATUS
090D 01 67800908     LDX 13 ZZERR
090F 0 C300          LD  3 0
0910 0 D005          STO  ZZLOP
0911 0 7301          MDX  3 1
*
*
0912 0 4007          BSI  ZZTCM      SET UP MESSAGE TABLE
0913 00 44800162     BSI I  ERROR     PRINT MESSAGE
0915 1 0932          DC  ZZTBL
0916 0 0000          ZZLOP DC      *--
*
*
0917 0 408C          BSI  ZZRES      RESTORE STATUS
0918 01 4C800908     BSC I  ZZERR

```

1130-1231 OMPR F. T.

```

*-----*
*          COMMON MESSAGE SET UP SUBROUTINE          *
*-----*
091A 0 0000  ZZTCM DC  *--
091B 0 62FB  LD  2 -5
091C 0 C300  LD  3 0
091D 01 D6000937  STD L2 ZZTBL+5  STD MESSAGE CONTROL
091F 0 7301  MDX  3 1
0920 0 7201  MDX  2 1
0921 0 70FA  MDX  ZZT1
*
0922 0 C3FD  LD  3 -3  LD DATA CONTROL
0923 0 1001  SLA  1
0924 0 1801  SRA  1
0925 0 6BDA  ZZT2 STX  3 ZZLOG  SET RETURN ADDR
0926 0 6BE4  STX  3 ZZERR
0927 01 4C88091A  BSC  1 ZZTCM,+  RETURN IF FINISHED
*
0929 0 18D1  RTE  17
092A 00 C7800000  LD  13 0  LD DATA
092C 01 D6000937  STD L2 ZZTBL+5  STD IN MESSAGE TABLE
092E 0 7201  MDX  2 1
092F 0 7301  MDX  3 1
0930 0 1090  SLT  16
0931 0 70F3  MDX  ZZT2  LOOP UNTIL LAST DATA WORD
*-----*
*          MESSAGE TABLE          *
*-----*
0932 0 0000  ZZTBL DC  *--  MSG ID
0933 0 0000  DC  *--  HEX/DEC
0934 0 0000  DC  *--  DATA ID
0935 0 0000  DC  *--  ALPHA1
0936 0 0000  DC  *--  ALPHA2
*
*          MESSAGE DATA          *
*-----*
0937 0 0000  DC  *--
0938 0 0000  DC  *--
0939 0 0000  DC  *--
093A 0 0000  DC  *--
093B 0 0000  DC  *--
093C 0 0000  DC  *--
093D 0 0000  DC  *--
093E 0 0000  DC  *--
093F 0 0000  DC  *--
0940 0 0000  DC  *--
0941 0 0000  DC  *--
0942 0 0000  DC  *--
0943 0 0000  DC  *--
0944 0 0000  DC  *--
0945 0 6000  DC  *--
*-----*
0946 05E8  END  BGIN

```

1130-1231 OMPR F. T.

CROSS REFERENCE LISTING

SYMBOL	VALUE	REFERENCES
AALTS	087C	0703
ADERR	0840	067B
ADSWI	0822	075D
AHOPE	0836	077B
AINDS	0859	07A2,07B4
ALRC	0846	06C2
AMNMR	0891	06F9
AMMTS	0888	06D6
ANINT	0850	073C
ANRDY	082E	0773
ARNC	086E	0722
ATOLA	0864	06CD
AWSBE	081D	067A,06C1,06F8,075C,07A1,07B3
BEGIN	0160	0000,05E8
BGIN	05E8	0946
CHECK	0780	0653,06EB,070E,072C,078C,0791,0795,07A5
CHK1	078F	0787
CHK2	0793	078E
CHK3	07A7	0784
CNTRL	05F3	05F2,08C1,08F4
CN10	05F7	
CN20	05FE	05F5
CN30	0604	05F9
DELAY	08E2	06B0,0760,0764,077C,08E5,08E8,08FD
DISC	0628	06EF,0716,0719,0741
DWCNT	0641	064C,0657,065E,0661,0666,067E,0688
END	0164	0000,0000,0000,0000,0602
EOPAD	063C	0647,06D7,0705,0723,07AB,07B7
ERLCK	0166	
EROC	0643	064D,066C,0672
ERROR	0162	0000,0913
FEED	0622	064E,06AB,06DB,0709,0727
FRSW	063D	0744,0785,0788
HDSWF	0644	
HDSWI	0640	0616,06E0,06E5,06E9,06FA,0781,0797,07A3,0785
HDSWN	0646	0733,073D
HDWAS	0642	0624,0659,067C,0689,0692,06A5
HLRCD	0645	0685,06C3
HO2BE	063F	066A,067D,068A,0754,075F,0799,07A4
HWASI	063E	074A,075E
HO107	0632	06B7,06C4
H8486	062B	078F
H8686	062A	078A
H9686	0635	06FB,0786
ILO	017A	0000
IL1	018A	0000
IL2	019A	0000
IL3	01AA	0000
IL4	01BA	0000,05EE
INIT	0740	0649,06D9,0707,0725,073E,074C,0763,0774,077F
INIT4	076D	0769
INT4	0614	05EC,061C
IN1	0748	076B
IN2	0764	074E
IN3	0775	0751,076C
KEFFF	0631	06E7
KE700	062E	0694
KFB3F	0639	07A9
KFB39	0638	0793
KFB7F	062D	0670,0748
KFDFF	0630	
KOC82	0634	07A7
K0001	0633	068E,06A1
K005A	062C	0662
K0200	063A	

1130-1231 OMPR F. T.

K04C0	0636	
K0800	0638	
K7BC3	062F	06A7
K7939	0637	
LOG	0163	0000,0905
LOGBY	0167	
LPTST	05F0	05E3
LRTN	0613	060E
MLSCF	05E5	0609,061A,08AA,08AD,08B9,08EC
NINT	0731	0652,06AF,06DF,070D,072B
NRTN	0610	060F
PID	05DC	05EA
RAD	05DE	0608
READ	0624	0655,06ED,0710,072E
RID	05DD	05F1,05F7,05FD,05FE,0600,0604,08BF,08F2
RIDCK	060E	05F8
RLCF	0168	
RQKB	018C	0000
RQTY	018B	0000
RTNDM	060F	0601
RTNSW	0165	060A
RTN1	0647	0610,06B9,06C5
RTN2	06D0	0611
RTN3	06FD	0612
RTN4	071C	0613
RTTBL	0610	0606,060E,060F,0611
R1A	0650	0660,067F,068C
R1B	0655	
R1B1	065E	069F,06CE
R1C	0661	065C
R1C1	0666	0696,06A4,06A9
R1C2	0681	066E
R1D	068E	0664
R1D1	06A1	0691
R1D2	06C7	069C
R1F3	079B	
R2A	06D9	06F1
R2B	06E5	06FC
R2B1	06F2	06E3
R3A	0704	0718
R3A1	0707	0718
R3A2	0719	0713
R4A	0725	
R4B	0729	0730
SELCT	0626	069D,0714
SENSE	0620	0698,06B3,0731,0767
SENSR	061E	0615,0746
START	0161	0000,060C,08AF,08BD,08C3,08F0,08F6
STRT	05EB	05E4
SVKB	018D	
SW0	05DF	
SW1	05E0	05F4,05FC,08BB,08EE
SW2	05E1	
SW3	05E2	
TABLE	07B9	065A,0668
WAIT	08A2	0650,06AD,06DD,070B,0729,08B4,08B6,08C6,08CA
ZZDFA	08FF	08E7,08F8
ZZD01	08F8	08EA
ZZD02	08EA	08FA
ZZERR	0906	0674,0681,06BB,06F2,0735,0756,079B,07AD,090D,0918,0926
ZZLOG	0900	06C7,06D0,06FD,071C,076D,0775,0902,0909,0925
ZZLOP	0914	0910
ZZRES	08D4	08C8,08DD,08FB,0908,0917
ZZR01	08C6	0618
ZZR02	08C8	08B7
ZZSAQ	08E0	08CD,08D5
ZZSAV	08CC	08A3,08D2,08E3,0901,090C

1130-1231 OMPR F. T.

ZZSTS	08D6	08CE
ZZS1	08D7	08CF
ZZS2	08C9	08D0
ZZS3	08DB	08D1
ZZTBL	0932	0907,0915,091D,092C
ZZTCH	091A	0904,0912,0927
ZZT1	091C	0921
ZZT2	0925	0931
ZZWCT	08C5	08A7,08B1
ZZW01	08A8	08B3
ZZW02	08AD	
ZZW03	08B1	08A8
ZZW04	08BB	08AC

TABLE OF CONTENTS

PARAGRAPH	PAGE
1. PURPOSE . . . . .	1
2. REQUIREMENTS . . . . .	1
2.1 PROGRAM REQUIREMENTS	
2.2 EQUIPMENT REQUIREMENTS	
3. OPERATING PROCEDURE . . . . .	1A
3.1 PROGRAM LOADING	
3.2 PROGRAM OPERATION	
3.2.1 PROGRAM CONTROL OPTIONS - FUNCTION 0	
3.2.2 ROUTINE SELECTION - FUNCTION 1	
3.2.3 PRINTER SIZE OPTION - FUNCTION 2	
3.2.4 OPTIONAL ROUTINE SETUP	
3.3 PROGRAM HALTS	
3.3.1 NORMAL HALTS	
3.3.2 ERROR HALTS	
3.4 TERMINATION	
4. PRINTOUTS . . . . .	3A
4.1 STATUS MESSAGES	
4.2 ERROR MESSAGES	
5. COMMENTS . . . . .	6A
5.1 ROUTINE DESCRIPTION	
5.2 1403 OUTPUT CHARACTER CODES	
5.3 CARRIAGE TAPE	
6. APPENDIX	
6.1 SAMPLE PRINTOUT	

1. PURPOSE

THE 1403 FUNCTION TEST IS DESIGNED TO CHECK THE OPERATING PERFORMANCE OF THE 1403 PRINTER, 120 OR 132 PRINT POSITIONS, AND TO AID IN ITS ADJUSTMENT.

2. PREREQUISITES

2.1\*\*\* PROGRAM PREREQUISITES

THIS PROGRAM MUST RUN UNDER CONTROL OF THE DIAGNOSTIC MONITOR. THE DIAGNOSTIC MONITOR PROGRAM USES 1500 STORAGE WORDS, AND THIS PROGRAM USES 1400 STORAGE WORDS.

2.2\*\*\* EQUIPMENT PREREQUISITES

ALL CHANNELS PUNCHED EQUALLY SPACED IN NUMERICAL ORDER IS NECESSARY FOR THE CARRIAGE TEST ROUTINE. THE CHANNEL PUNCHES SHOULD BE SPACED FOUR OR MORE LINES APART ON THE CARRIAGE TAPE. ANY SUCH TAPE WILL WORK WELL. SEE TABLE 4 FOR A SUGGESTED CARRIAGE TAPE.  
\*\*\*\*\*

3. OPERATING PROCEDURE

3.1\*\*\* PROGRAM LOADING

STANDARD MONITOR LOADING PROCEDURES APPLY

THESE PROCEDURES ARE SUMMARIZED HERE. SEE DM USE PROCEDURE FOR DETAILS.

1. SET FIRST TYPEWRITER TAB 20 CHARACTERS FROM LEFT MARGIN.
2. SET BIT SWITCH 15 OFF - LOAD AND GO ON - TO SPECIFY OPTIONS BEFORE RUNNING.
3. LOAD DIAGNOSTIC MONITOR.
4. SELECT PROGRAM OPTIONS, IF DESIRED.

\*\*\*\*\*

3.2\*\*\* PROGRAM OPERATION.

STANDARD MONITOR OPERATING PROCEDURES APPLY. THESE PROCEDURES ARE SUMMARIZED HERE. SEE DM USE PROCEDURE FOR DETAILS.

3.2.1 PROGRAM CONTROL - FUNCTION 0

1. SET SWITCHES 0-1 OFF.
2. SET PROGRAM LOADING SEQUENCE NUMBER IN SWITCHES 4-7. (ONE UNLESS RUNNING OVERLAP)
3. SET SWITCHES 8-15 AS DESIRED.

-----  
SW      FUNCTION  
-----

8	RESTART
9	ROUTINE START MESSAGE
10	LOCK ON FUNCTION
11	LOOP PROGRAM
12	LOOP ON ERROR
13	BYPASS ERROR PRINTOUT
14	HALT ON ERROR
15	HALT

-----

4. PRESS INT REQ KEY ON CONSOLE.  
\*\*

3.2.2 ROUTINE SELECTION - FUNCTION 1

THE SELECTED ROUTINE WILL LOOP UNTIL A NEW ROUTINE IS SELECTED.

1. SET SWITCHES 0-1 TO 01.
2. SET PROGRAM LOADING SEQUENCE NUMBER IN SWITCHES 4-7. (ONE UNLESS RUNNING OVERLAP)



3. SET ROUTINE NUMBER IN SWITCHES 12-15.

RTN	DESCRIPTION
1	PRINTER READY + BUSY TEST
2	CARRIAGE READY + BUSY TEST
3	PARITY CHECK (GOOD PARITY)
4	BIT LINE TEST
5	PARITY TEST (BAD PARITY)
6	CHARACTER COMPLEMENT
7	SPACE AFTER PRINT TEST
8	SPACE BEFORE PRINT TEST
9	* SKIP AFTER PRINT TEST
A	* SKIP IMMEDIATE
B	* PRINTER DATA ENTRY
* = REFER TO SECTION 3.2.3 FOR SPECIAL INSTRUCTIONS.	

4. PRESS INT REQ KEY ON CONSOLE.  
\*\*\*\*\*

3.2.3 PRINTER SIZE OPTION - FUNCTION 2

TO TEST 132 POSITION PRINTERS (RPQ) SET FUNCTION 2, BIT 15 ON.

1. SET SWITCHES 0-1 TO 10.
2. SET PROGRAM LOADING SEQUENCE NUMBER IN SWITCHES 4-7. (ONE UNLESS RUNNING OVERLAP)
3. TURN ON SWITCH 15
4. PRESS INT REQ KEY ON CONSOLE.
5. RESTART THE PROGRAM.

FOLLOW THE SAME PROCEDURE WITH SWITCH 15 OFF TO RESELECT 120 POSITIONS.

\*NOTE\* IF THE 132 POSITION OPTION IS SELECTED ON A 120 POSITION PRINTER, LOST INTERRUPT MESSAGES WILL OCCUR IN ROUTINES 3 AND 5.

3.2.4 OPTIONAL ROUTINE SETUP

1. ROUTINE 9 AND A

THESE ROUTINES REQUIRE A SPECIAL CARRIAGE TAPE OR A TAPE WHICH HAS TWO ADJACENT HOLES IN CHANNELS 1, 9 AND 12. A SAMPLE TAPE IS CALLED OUT IN SECTION 5.3.

2. ROUTINE B

THIS ROUTINE PRINTS THE CHARACTER SET UP IN THE BIT SWITCHES. THIS IS CODING UNIQUE TO THE 1403 AND IS SHOWN IN SECTION 5.2. THE BIT CODE IS SET IN BIT POSITIONS 8-15. BIT 0 SWITCH MUST BE ON OR THE MANUALLY ENTERED CODE WILL NOT BE PRINTED. H'S WILL BE PRINTED IF THE HONOR BIT (B-0) IS NOT ON.

3.3\*\*\* PROGRAM HALTS

3.3.1 NORMAL HALTS

HALT NO. (B REG.)	DESCRIPTION	RESTART ACTION
3001	PROGRAM STOP OR ADDRESS STOP	PRESS START
3002	HALT ON ERROR	DISPLAY MODE-PRESS START. RUN MODE-PRESS START

\*\*

3.3.2 ERROR HALTS

HALT NO. (B REG.)	DESCRIPTION	RESTART ACTION
30F1	CHECK SUM ERROR ON FIRST CARD OF LOADER	RELOAD
30F2	READER DSW ERROR WHEN LOADING LOADER	RELOAD
30F3	CARD 2 OF LOADER DID NOT LOAD	RELOAD
30F4	CAN NOT CLEAR CORE - DUE TO ERROR IN ADDRESSING UPPER CORE.	
30F5	READER CHECK WHEN LOADING MONITOR OR TEST PROGRAM	NPRU THEN PLACE CARDS RUN OUT IN FRONT OF REMAINING DECK AND PRESS START.
30F6	MONITOR DID NOT LOAD	RELOAD
30F7	CHECK SUM WHEN LOADING MONITOR	RELOAD
30F8	READER NOT READY	MAKE READER READY
30F9	INVALID INTERRUPT WHICH WILL NOT RESET	PRESS RESET AND START
30FA	CONSOLE PRINTER HANG UP - BUSY WILL NOT GO OFF	FIX THE CONSOLE PRINTER OR NOP THIS WAIT

\*\*\*\*\*

3.4\*\*\* PROGRAM TERMINATION

IF LOOP PROGRAM HAS NOT BEEN SPECIFIED THE PROGRAM WILL TERMINATE AT THE END OF ROUTINE 7. ROUTINE 7,A, AND E WILL ONLY RUN SELECTED.

IF ANY ROUTINE IS SELECTED THAT ROUTINE WILL LOOP AND WILL NOT TERMINATE.

\*\*\*\*\*

3/5\*\*\* RESTART

3.5.1 RESTART ALL PROGRAMS

1. TURN OFF SWITCHES 0-7.
2. TURN ON SWITCH 8.
3. SET DESIRED CONTROL IN SWITCHES 9-14.
4. PRESS INTERRUPT REQUEST KEY.

3.5.2 RESTART THIS PROGRAM ONLY

1. TURN OFF SWITCHES 0-3.
2. SET SWITCHES 4-7 TO PROGRAM LOAD SEQUENCE NUMBER.
3. TURN ON SWITCH 8.
4. SET DESIRED CONTROL IN SWITCHES 9-14.
5. PRESS INTERRUPT REQUEST KEY.

3.5.3 RESTART FROM HANG UP

PRESS IMMEDIATE STOP, RESET, THEN START. THIS WILL RESET ALL MONITOR, CONTROLS AND, THEN RESTART THE MONITOR.

\*\*\*\*\*

4. PRINTOUTS

ALL PRINTOUTS ARE IN THE STANDARD FORMAT.

APPNN OORR AAAA (MESSAGE)

WHERE A IDENTIFIES STATUS MESSAGES  
E IDENTIFIES ERROR MESSAGES  
PP IS THE PID OF THE PROGRAM CAUSING THE MESSAGE  
NM IS THE MESSAGE SEQUENCE NUMBER  
RR IS THE ROUTINE NUMBER  
AAAA IS THE ADDRESS OF THE ROUTINE  
MESSAGE IS ANY VARIABLE INFORMATION

4.1\*\*\* STATUS MESSAGES

A0000 NUM PID ADRS RELF LD  
XXXX XXXX XXXX XXXX

THIS MESSAGE IS PRINTED FOLLOWING THE LOADING OF ANY PROGRAM (EXCEPT MONITOR), THE MESSAGE GIVES THE LOAD SEQUENCE NUMBER, THE PROGRAM ID, THE ADDRESS INTO WHICH THE PROGRAM WAS LOADED, AND THE RELOCATION FACTOR.

A0001 SWS PID  
XXXX XXXX

THIS MESSAGE IS PRINTED EACH TIME A VALID SWITCH ENTRY IS READ BY THE MONITOR. THE MESSAGE CONTAINS THE SWITCH SETTING READ TOGETHER WITH THE PROGRAM ID OF THE PROGRAM INTO WHICH THE CONTENTS OF SWITCHES 8-15 WERE STORED. IF THE SWITCH ENTRY CALLED FOR HALT OF ANY PROGRAM THE WORD HALT WILL FOLLOW THE MESSAGE.

A0000 000R AAAA

ROUTINE START MESSAGE - IF SWITCH 9, FUNCTION 0, IS TURNED ON, THIS MESSAGE WILL BE PRINTED BEFORE THE START OF EACH ROUTINE. R IS THE NUMBER OF THE NEXT ROUTINE AND AAAA IS THE STARTING ADDRESS.

\*\*\*\*\*

4.2\*\*\* ERROR MESSAGES

THE DSW IS CHECKED FOR ABSOLUTE CORRECTNESS AT ALL TIMES. IF AN ERROR IS DETECTED ONE OF THE MESSAGES BELOW WILL INDICATE THE PROBLEM. IT IS LEFT TO THE OPERATOR TO ANALYZE THE DSW FOR THE SPECIFIC PROBLEM AREA.

```

*****
* THE 1403 DSW *
*-----*
* BIT *
* 0 PARITY ERROR *
* 1 TRANSFER COMPLETE INTRPT *
* 2 PRINT COMPLETE INTRPT *
* 3 CARRIAGE OPERATION COMPLETE INTRPT *
* 4 *
* 5 RING CHECK *
* 6 SYNCH CHECK *
* 7 *
* 8 *
* 9 *
* 10 *
* 11 CHANNEL 9 *
* 12 CHANNEL 12 *
* 13 CARRIAGE BUSY *
* 14 PRINT BUSY *
* 15 NOT READY *
*****

```

E0001 SWS INVLD  
XXXX  
THE SETTING OF SWITCHES 4-7 DID NOT EQUAL THE LOAD SEQUENCE NUMBER OF ANY PROGRAM IN CORE.

E0903 OVR CORE  
THE PROGRAM WHICH THE LOADER WAS ATTEMPTING TO LOAD EXCEEDED AVAILABLE CORE. LOADING WAS TERMINATED.

E0004 CKSUM  
A CHECK SUM ERROR WAS DETECTED WHILE LOADING A TEST PROGRAM. THIS ERROR OCCURS UNDER ANY OF THE FOLLOWING CONDITIONS.

1. A CARD IS MISSING OR IS OUT OF SEQUENCE.
2. THERE IS AN EXTRA CARD IN THE DECK.
3. THE PUNCHED INFORMATION ON THE CARD IS NOT CORRECT.
4. DATA WAS LOST OR PICKED UP DUE TO A MACHINE MALFUNCTION.
5. DUE TO A CPU MALFUNCTION, THE CHECK SUM WAS NOT CORRECTLY CALCULATED.

WHEN THIS ERROR OCCURS ATTEMPT TO RELOAD THE PROGRAM.

E0005 O00N XXXX  
THIS ERROR WILL OCCUR IS AN INTERRUPT OCCURS, BUT THE ILSW WAS NOT CORRECT. N IS THE INTERRUPT LEVEL AND XXXX IS THE ILSW. THIS PRINTOUT WILL ONLY OCCUR IF THE INTERRUPT IS RESET BY A BOSI. NO ATTEMPT IS MADE BY THE ERROR ROUTINE TO RESET THE REQUEST BIT.

THE ERROR MESSAGES WILL BE IN THE FOLLOWING FORM,

EODNN 00OR AAAA XXXX 00YY  
WHERE NN IS THE MESSAGE NUMBER  
R IS THE ROUTINE NUMBER  
AAAA IS THE ADDRESS OF THE ROUTINE  
XXXX IS THE DSW IN ERROR  
00YY IS THE DSW AS IT SHOULD BE. BECAUSE OF CHAN 9 + 12  
BITS THE YY POSITION MAY VARY DEPENDING ON THE

CARRIAGE TAPE.

EOD01 00OR AAAA WAS S/B  
XXXX 00YY  
PRINTER SHOULD BE READY.

EOD02 00OR AAAA WAS S/B  
XXXX 00YY  
PRINTER SHOULD BE NOT-BUSY.

EOD03 00OR AAAA WAS S/B  
XXXX 00YY  
CARRIAGE SHOULD BE NOT-BUSY.

EOD04 00OR AAAA WAS S/B  
XXXX 00YY  
CARRIAGE SHOULD BE BUSY AND PRINTER SHOULD BE BUSY.

EOD05 00OR AAAA WAS S/B  
XXXX 00YY  
PRINTER SHOULD BE BUSY AND READY.

EOD06 00OR AAAA WAS S/B  
XXXX 00YY  
CARRIAGE SHOULD BE BUSY.

EOD07 00OR AAAA WAS S/B  
XXXX 10YY  
CARRIAGE CHANNEL INDICATOR BITS NOT WORKING PROPERLY.

EOD11 0001 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE DSW)  
XXXX 20YY (PRINT COMPLETE DSW)

ERROR IN INTERRUPT DSW (XXXX). WHEN PRINTING BLANK LINES WITHOUT SPACING THE CARRIAGE. ANY PRINTED CHARACTER ON THE 1403 IS AN ERROR.

EOD12 0002 AAAA WAS S/B  
XXXX 10YY (CARRIAGE OP COMPLETE DSW)

ERROR IN INTERRUPT DSW (XXXX) WHEN SPACING THE CARRIAGE WITHOUT PRINTING.

E0013 0003 AAAA WAS S/B  
XXXX 4002 00NNN (TRANSFER COMPLETE DSW)

XXXX 2000 (PRINT COMPLETE DSW)

ERROR IN INTERRUPT DSW (XXXX). WHEN PRINTING SINGLE-BIT GOOD  
PARITY CHARACTERS. THE LAST WORD INDICATES IN DECIMAL THE PRINT  
POSITION IN ERROR. ONE CHARACTER AT A TIME IS PRINTED.

\*\*\*1122335599VVRR11223\*\*\*

1 = 40 2 = 01  
3 = 02 5 = 04  
4 = 08 V = 10  
R = 20

E0014 0004 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

XXXX 20YY (PRINT COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN PRINTING SINGLE-BIT CHARACTERS

\*\*\*1122335599VVRR11223\*\*\*

1 = 40 2 = 01  
3 = 02 5 = 04  
9 = 08 V = 10  
R = 20

E0015 0005 AAAA WAS S/B  
XXXX 00YY 00NNN (TRANSFER COMPT PARITY ERROR)

XXXX A0YY (PRINT COMPT PARITY ERROR)

ERROR IN INTERRUPT DSW (XXXX) WHEN PRINTING SINGLE BIT BAD PARITY  
CHARACTERS. THE LAST WORD INDICATES IN DECIMAL THE PRINT POSITION  
THAT DID NOT INDICATE A PARITY ERROR.

\*\*\*1122335599VVRR11223\*\*\*

1 = 00 2 = 41  
3 = 42 5 = 44  
9 = 48 V = 50  
R = 60

E0016 0006 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

XXXX 20YY (PRINT COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN PRINT THE FULL CHARACTER  
COMPLIMENT. THIS PORTION OF THE PROGRAM USES ROUTINE 4 TO  
OUTPUT THE DATA AFTER ITS INITIAL SET UP.

E0017 0007 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 20YY (PRINT COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN A SPACE AFTER PRINT COMMAND  
IS GIVEN. TO SPACE AFTER PRINTING A SPACE COMMAND IS ISSUED AFTER  
TRANSFER COMPLETE IS RECEIVED FROM THE PREVIOUS PRINT COMMAND.

E0018 0008 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

XXXX 20YY (PRINT COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN A SPACE COMMAND IS GIVEN.  
THE PRINT SPACE COMMANDS ARE ISSUED AT THE SAME TIME.

E0019 0009 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 20YY (PRINT COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN A SKIP AFTER PRINT COMMAND  
HAS BEEN GIVEN. TO SKIP AFTER PRINTING SKIP COMMAND IS ISSUED  
AFTER TRANSFER COMPLETE IS RECEIVED FROM THE PREVIOUS PRINT  
COMMAND.

E001A 000A AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

XXXX 20YY (PRINT COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN A SKIP IMMEDIATE COMMAND  
IS GIVEN. A PRINT AND A SKIP COMMAND ARE ISSUED TOGETHER.

E001B 000B AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

XXXX 20YY (PRINT COMPLETE)

XXXX 10YY (CARRIAGE OP COMPLETE)

ERROR IN INTERRUPT DSW (XXXX) WHEN PRINTING H'S OR OPTIONAL  
CHARACTERS SET IN THE BIT SWITCHES.

E0021 0001 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)

0000 20YY (PRINT COMPLETE)

ERROR IN INTERRUPT DSW (0000). THIS INDICATE THAT ONE  
OR MORE OF THE EXPECTED INTERRUPTS WAS NOT RECEIVED DURING  
THE PRINTING OF A BLANK LINE WITHOUT CARRIAGE SPACING. ANY  
CHARACTER PRINTED ON THE 1403 IS AN ERROR.

E0D22 0002 AAAA WAS S/B  
0000 10YY (CARRIAGE OP COMPLETE)

THIS ERROR (0000) INDICATES THAT NO INTERRUPT WAS RECEIVED

E0D23 0003 AAAA WAS S/B  
XXXX 40YY 00NNN (TRANSFER COMPLETE)  
0000 20YY (PRINT COMPLETE)

THIS ERROR (0000) INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED WHEN TWO WERE EXPECTED. THIS OCCURRED DURING THE PRINTING OF GOOD PARITY DATA, ONE CHARACTER PER PRINT CYCLE WITH NO CARRIAGE SPACING BETWEEN PRINT CYCLES. THE LAST WORD INDICATES IN DECIMAL THE COLUMN WHERE THE FAILURE OCCURRED.

E0D24 0004 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 10YY (CARRIAGE OP COMPLETE)  
XXXX 20YY (PRINT COMPLETE)

THIS PRINTOUT INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED. THIS ROUTINE SHOULD BE PRINTING LINES OF SINGLE BIT CHARACTERS.

E0D25 0005 AAAA WAS S/B  
XXXX 00YY 00NNN (TRANSFER COMPT PARITY ERROR)  
0000 A0YY (PRINT COMPT PARITY ERROR)

THIS PRINTOUT INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED. TWO INTERRUPTS WERE EXPECTED. IF NO INTERRUPTS WERE RECEIVED ZERO'S WOULD HAVE BEEN PRINTED FOR THE X'S ON BOTH LINES. THIS OCCURRED DURING THE PRINTING OF A BAD PARITY CHARACTER IN THE COLUMN INDICATED BY THE LAST WORD. (00NNN).

E0D26 0006 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 10YY (CARRIAGE OP COMPLETE)  
0000 20YY (PRINT COMPLETE)

THIS PRINTOUT INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED DURING THE CHARACTER COMPLEMENT ROUTINE.

E0D27 0007 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 20YY (PRINT COMPLETE)  
0000 10YY (CARRIAGE OP COMPLETE)

THIS PRINTOUT INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED DURING THE SPACE AFTER PRINT ROUTINE. THIS ROUTINE ISSUES A SPACE COMMAND AFTER THE FIRST INTERRUPT IS RECEIVED. THIS SPACE SHOULD BE COMPLETED AFTER THE LINE OF PRINT.

E0D28.0008 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 10YY (CARRIAGE OP COMPLETE)  
0000 20YY (PRINT COMPLETE)

THIS INDICATES THAT ONE OR MORE INTERRUPTS WERE NOT RECEIVED DURING THE SPACE IMMEDIATE TEST. THE SPACE AND PRINT COMMANDS ARE ISSUED TOGETHER.

E0D29 0009 AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 20YY (PRINT COMPLETE)  
0000 10YY (CARRIAGE OP COMPLETE)

THIS IS IDENTICAL TO E0D27 EXCEPT THAT IT OCCURRED DURING THE SKIP AFTER PRINT ROUTINE.

E0D2A 000A AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 10YY (CARRIAGE OP COMPLETE)  
0000 20YY (PRINT COMPLETE)

THIS IS SIMILAR TO E0D28 EXCEPT THAT IT OCCURRED DURING ROUTINE 10 WHERE A PRINT AND SKIP COMMAND ARE ISSUED TOGETHER. THE PRINT TRANSFER SHOULD BE RECEIVED BEFORE THE SKIP IS COMPLETE. THE PRINT COMPLETE INTERRUPT SHOULD FOLLOW.

E0D2B 000B AAAA WAS S/B  
XXXX 40YY (TRANSFER COMPLETE)  
XXXX 10YY (CARRIAGE OP COMPLETE)  
0000 20YY (PRINT COMPLETE)

THIS IS LIKE E0D26 EXCEPT THAT IT OCCURRED DURING ROUTINE 11 WHERE H'S ARE PRINTED OR THE BIT SWITCH OPTIONAL DATA IS PRINTED.

E0D0C 000R AAAA WAS S/B  
XXXX 0000

UNEXPECTED INTERRUPT RECEIVED

\*\*\*\*\*

## 5. COMMENTS

### 5.1 ROUTINE DESCRIPTION

THE 1403 FUNCTION TEST HAS EIGHT NORMAL ROUTINES AND THREE OPTIONAL ROUTINES TO AID IN THE ADJUSTMENT AND CHECKOUT OF THE PRINTER OPERATION. THE FIRST EIGHT ROUTINES ARE NORMALLY RUN SEQUENTIALLY UNLESS THE OPERATOR HAS SPECIFIED A SPECIFIC ROUTINE TO BE RUN (FCN 1).

A SINGLE LINE OF PRINT MAY BE OBTAINED EACH TIME THE KEYBOARD REQUEST KEY IS PRESSED WHEN THE SWITCHES ARE SET TO 0X40. (X = PROGRAM LOADING SEQUENCE NUMBER).

5.1.1 OPERATION OF NORMAL ROUTINES.

A. ROUTINE 1

THE PRINTER READY-BUSY ROUTINE ASSURES THAT THE 1403 CAN BE READY AND NOT BUSY THEN PRINTS A BLANK LINE WITHOUT SPACING. THE PRINTER SHOULD IMMEDIATELY GO BUSY - NOT READY AND RETURN READY - NOT BUSY WHEN THE PRINTER COMPLETE INTERRUPT IS RECEIVED.

B. ROUTINE 2

THE CARRIAGE READY - BUSY ROUTINE CHECKS THAT THE CARRIAGE CAN BE MADE NOT BUSY AND GOES BUSY IMMEDIATELY AFTER A SPACING COMMAND. AFTER THE SPACE COMPLETE INTERRUPT IS RECEIVED, THE CARRIAGE SHOULD AGAIN BE NOT BUSY. NOTHING IS PRINTED BY THIS ROUTINE.

C. ROUTINE 3

THE GOOD PARITY CHECK ROUTINE CHECKS THE ABILITY OF THE 1403 TO PRINT DATA HAVING VALID PARITY BITS.

A SINGLE GOOD PARITY CHARACTER IS TO BE PRINTED DURING EACH PRINT CYCLE. THIS PROCEDURE IS REPEATED FOR ALL PRINT POSITIONS BEFORE SPACING THE CARRIAGE FOR THE NEXT ROUTINE.

D. ROUTINE 4

THE BIT LINE CHECK ROUTINE PRINTS DATA IN ITS SIMPLEST FORM TO CHECK THE CONTINUITY OF THE PRINT CIRCUITS. ONLY THE FOLLOWING SINGLE BIT CHARACTERS ARE PRINTED.

BINARY WORD	CHARACTER
1000000	1
0000001	2
0000010	3
0000100	5
0001000	9
0010000	V
0100000	R

ANY OPEN OR SHORTED DATA LINE SHOULD BE EASILY SPOTTED BY THE PRINTING OF WRONG CHARACTERS. THIS ROUTINE ISSUES A SPACING COMMAND IMMEDIATELY AFTER THE PRINT COMMAND.

E. ROUTINE 5

THE PARITY CHECK ROUTINE CHECKS THE ABILITY OF THE 1403 TO RESPOND TO CHARACTERS WHICH HAVE INVALID PARITY BITS.

A SINGLE BAD PARITY CHARACTER IS TO BE PRINTED DURING EACH PRINT CYCLE. THIS PROCEDURE IS REPEATED FOR ALL PRINT POSITIONS BEFORE SPACING THE CARRIAGE FOR THE NEXT ROUTINE.

F. ROUTINE 6

THE CHARACTER ROUTINE PRINTS EACH CHARACTER OF THE 48 CHARACTERS SET IN EACH OF THE PRINT POSITIONS. CHARACTERS ARE RIPPLED TO THE LEFT WITH EACH SUCCEEDING LINE OF PRINT. THIS ROUTINE ISSUES A SPACE COMMAND IMMEDIATELY AFTER THE PRINT COMMAND.

G. ROUTINE 7

THE SPACE AFTER PRINT TEST ISSUES A SPACE IMMEDIATELY AFTER A PRINT COMMAND TO CHECK THE CARRIAGE SPACING CIRCUITS.

H. ROUTINE 8

THE SPACE BEFORE PRINT TEST ISSUES A SPACE IMMEDIATELY BEFORE A PRINT COMMAND TO CHECK THE CARRIAGE SPACING CIRCUITS.

5.1.2 OPERATION OF OPTIONAL ROUTINES

A. ROUTINE 9

CARRIAGE SKIP AFTER PRINT ROUTINE FIRST SKIPS CHANNEL 1 THEN AFTER THE INTERRUPT ISSUES A PRINT AND SKIP TO CHANNEL 2. SUCCEEDING CHANNELS ARE CHECKED IN A LIKE MANNER.

A SPECIAL CARRIAGE TAPE WITH ALL CHANNELS PUNCHED EQUALLY SPACED AND IN ORDER IS REQUIRED FOR THIS TEST (6.2)

B. ROUTINE A

CARRIAGE SKIP BEFORE PRINT ROUTINE ISSUES A SKIP TO CHANNEL 1 IMMEDIATELY BEFORE A PRINT COMMAND. OTHER CHANNELS ARE CHECKED IN A LIKE MANNER AFTER WAITING FOR THE INTERRUPTS.

A SPECIAL CARRIAGE TAPE WITH ALL CHANNELS PUNCHED EQUALLY SPACED AND IN ORDER IS REQUIRED FOR THIS TEST (6.2).

C. ROUTINE B

OPTIONAL PRINT ROUTINE PRINTS FULL LINES OF ANY OPTIONAL CHARACTER ENTERED BY SETTING BIT SWITCHES. BIT 0 MUST BE ON OR BIT SWITCHES WILL NOT BE HONORED. SET BIT 0 ON THEN SELECT CHARACTER CODE FROM TABLE 1 (5.2) FOR BIT SWITCHES 1 - 7 and 9 - 15.

IF NO CHARACTER IS ENTERED A LINE OF H'S WILL BE PRINTED.

\*\*\*\*\*

5.2\*\*\* TABLE 1. 1403 OUTPUT CHARACTER CODES

CHARACTER	CODE (BITS 8-15)	CHARACTER	CODE (BITS 8-15)
A	64	Y	13
B	25	Z	54
C	26	1	40
D	67	2	01
E	68	3	02
F	29	4	43
G	2A	5	04
H	6B	6	45
I	2C	7	46
J	58	8	07
K	19	9	08
L	1A	0	49
M	5B	EQUAL SIGN	4A
N	1C	APOSTROPHE	0B
O	5D	SLASH	4C
P	5E	AMPERSAND	15
Q	1F	COMMA	16
R	20	LEFT PARENTHESIS	57
S	0D	DASH	21
T	0E	DOLLAR SIGN	22
U	4F	ASTERISK	63
V	10	PLUS	60
W	51	PERIOD	6E
X	52	RIGHT PARENTHESIS	2F
		BLANK	7F

\*\*\*\*\*



```

*
* MONITOR TRANSFER VECTOR
*
*****
* THIS TABLE EQUATES TEST PROGRAM LABELS
* TO THEIR EQUIVALENT DIAGNOSTIC MONITOR
* ADDRESSES.
*
0160 BEGIN EQU /160 BEGIN ROUTINE 30D00020
0161 START EQU BEGIN+1 SUPERVISOR ROUTINE 30D00030
0162 ERROR EQU START+1 ERROR LOG ROUTINE 30D00040
0163 LOG EQU ERROR+1 STATUS LOG ROUTINE 30D00050
0164 END EQU LOG+1 END ROUTINE 30D00060
*
*-----*
* MONITOR CONTROL WORDS
*
0165 RTNSW EQU END+1 ROUTINE START SWITCH 30D00070
0166 ERLCK EQU END+2 LOCK ON ERROR CONTROL 30D00080
0167 LOGBY EQU END+3 I/O BUSY INDICATOR 30D00090
0168 RLCF EQU END+4 RELOCATION FACTOR ADR 30D00100
*
*-----*
* INTERRUPT TRANSFER VECTOR ADRS
*
017A ILO EQU /17A INTERRUPT LEVEL ZERO 30D00110
018A IL1 EQU ILO+16 ONE 30D00120
019A IL2 EQU IL1+16 TWO 30D00130
01AA IL3 EQU IL2+16 THREE 30D00140
01BA IL4 EQU IL3+16 FOUR 30D00150
01BB RQTY EQU IL4+1 TYPEWRITER SERVICE INTRPT 30D00160
01BC RQKB EQU RQTY+1 KEYBOARD INTRPT REQUEST 30D00170
01BD SVKB EQU RQKB+1 KBD SERVICE INTERRUPT 30D00180
*
*****
*
*-----*
* PROGRAM STATUS TABLE *****
*
0000 ORG **1500 30D00190
*
05DC 0 030D PID DC /030D PROGRAM ID NUMBER 30D00200
05DD 0 0000 RID DC 0 ROUTINE NUMBER 30D00210
05DE 0 0000 RAD DC /0000 ROUTINE ADDRESS 30D00220
05DF 0 0000 SWO DC /0000 FCN 0 - CONTROL 30D00230
05E0 0 0000 SW1 DC /0000 FCN 1 - LOOP ROUTINE 30D00240
05E1 0 0000 SW2 DC NOT USED 30D00250
05E2 0 0000 SW3 DC NOT USED 30D00260
05E3 1 0648 DC RTO LOOP PROGRAM ADDRESS 30D00270
05E4 1 0648 DC RTO INITIAL START ADR 30D00280
05E5 1 0648 MLSCF DC RTO MAIN LINE SEQ CONTROL 30D00290
05E6 0 0000 DC 0 30D00300
05E7 0 0000 DC 0 30D00310
05E8 0 FFFF TERM DC /FFFF TERMINATOR 30D00320
*
*****
*
*-----*
* DSW TABLES
*
05EA 0000 BSS E 30D00330
*
05EA 1 05EA DST DC ENTRY ENTRY NAME DISP 30D00340
05EB 0 0000 ERIND DC DST ENTRY TABLE ADDR 00 30D00350
*
*-----*
*
05EC 0 0000 DSW1 DC /0000 1403 READY DSW 02 30D00360
05ED 0 FFE7 DC /FFE7 DSW EXPECTED MASK 03 30D00370
*
*-----*
*
05EE 0 00C0 DSW2 DC /0000 PRINTER READY DSW 04 30D00380
05EF 0 FFE3 DC /FFE3 DSW EXPECTED MASK 05 30D00390

```

```

*
*-----*
*
05F0 0 0000 DSW3 DC /0000 CARRIAGE READY DSW 06 30D00700
05F1 0 FFE5 DC /FFE5 DSW EXPECTED MASK 07 30D00710
*
*-----*
*
05F2 0 0006 DSW4 DC /0006 1403 BUSY DSW 08 30D00720
05F3 0 FFE7 DC /FFE7 DSW EXPECTED MASK 09 30D00730
*
*-----*
*
05F4 0 0002 DSW5 DC /0002 PRINTER BUSY DSW 0A 30D00740
05F5 0 FFE7 DC /FFE7 DSW EXPECTED MASK 0B 30D00750
*
*-----*
*
05F6 0 0004 DSW6 DC /0004 CARRIAGE BUSY DSW 0C 30D00760
05F7 0 FFE7 DC /FFE7 DSW EXPECTED MASK 0D 30D00770
*
*-----*
*
05F8 0 0000 DSW7 DC /0000 CARRIAGE STATUS 0E 30D00780
05F9 0 FFFF DC /FFFF DSW EXPECTED MASK 0F 30D00790
*
*-----*
*
05FA 0 0000 DSW8 DC SPARE 10 30D00800
05FB 0 0000 DC SPARE 11 30D00810
*
*-----*
*
05FC 0 0000 DSW9 DC SPARE 12 30D00820
05FD 0 0000 DC SPARE 13 30D00830
*
*-----*
*
05FE 0 0000 DSWA DC SPARE 14 30D00840
05FF 0 0000 DC SPARE 15 30D00850
*
*-----*
*
0600 0 0000 DSWB DC SPARE 16 30D00860
0601 0 0000 DC SPARE 17 30D00870
*
*-----*
*
0602 0 0000 DSWC DC FALSE INTERPUPT 18 30D00880
0603 0 0000 DC DSW S/B 19 30D00890
*
*-----*
*
0604 0 0000 CYCNT DC /0000 CYCLE COUNTER 1A 30D00900
0605 0 0000 WACNT DC /0000 WAIT COUNTER 1B 30D00910
*
*-----*
*
0606 0 0000 TEMP DC TEMPORARY STORAGE 1C 30D00920
0607 0 0000 DC 1D 30D00930
0608 0 0000 DC 1E 30D00940
0609 0 0000 DC 1F 30D00950
060A 0 0000 DC 20 30D00960
060B 0 0000 DC 21 30D00970
060C 0 0000 DC 22 30D00980
060D 0 0000 DC 23 30D00990
060E 0 0000 DC 24 30D01000
060F 0 0000 DC 25 30D01010
0610 0 0000 DC 26 30D01020
0611 0 0000 DC 27 30D01030
*
*****
*
*-----*
* PRINTER IOCC AREA
*
0612 0000 BSS E 0 30D01040
0612 0 0000 SPACE DC *- CARRIAGE SPACE CMD 28 30D01250
0613 0 AC00 DC /AC00 CONTROL IOCC 29 30D01260
0614 1 0620 SKIP DC SKPWD CARRIAGE SKIP CMD 2A 30D01270
0615 0 A900 DC /A900 WRITE IOCC 2B 30D01280
0616 1 0AEF PRTIT DC AREA+1 PRINT A LINE CMD 2C 30D01290
0617 0 AD00 DC /AD00 INIT. WRITE IOCC 2D 30D01300
0618 0 0000 SENSN DC SENS DSW - NO RESET 2E 30D01310
0619 0 AF00 DC /AF00 SENSE DSW - RESET 2F 30D01320
061A 0 0000 SENSR DC SENSE DSW - RESET 30 30D01330
061B 0 AF01 DC /AF01 SENSE DSW - RESET 31 30D01340
061C 1 061E BITSW DC SWTCH READ THE BIT SWS 32 30D01350
061D 0 3A00 DC /3A00 33 30D01360

```



1130-1403 FUNCTION TEST

```

061E 0 0000  SWTCH DC /0000  LASTEST SWS 34 30D01380
061F 0 0000  SAVE DC /0000  SAVE SWITCHES 35 30D01390
*****
0620 0 0000  SKPWD DC 0 36 30D01400
*****
*
*          CONSTANTS + SAVE AREA
*
0621 0 0000  AOSAV DC 0  SAVED A + Q REGS. 37 30D01460
0622 0 0000  DC 38 30D01470
0623 0 0000  SWSAV DC 0  BIT SWITCH SAVE 39 30D01480
0624 0 0001  K0001 DC /0001 3A 30D01490
0625 0 0400  K0400 DC /0400 3B 30D01500
0626 0 0800  K0800 DC /0800 3C 30D01510
0627 0 003C  D60 DC 60 3D 30D01520
0628 0 0010  K0010 DC /0010  CONSTANT 3E 30D01530
*****
0629 0 0000  INTEX DC DSW STORAGE POINTER 3F 30D01550
062A 0 0000  DC INTERRUPT DSW - 1ST 40 30D01560
062B 0 0000  DC 2ND 41 30D01570
062C 0 0000  DC 3RD 42 30D01580
062D 0 00BF  K00BF DC /BF  MASK FOR SINGLE CY 43 30D01590
062E 0 0018  K0018 DC /0018 44 30D01600
062F 0 0040  K0040 DC /0040 45 30D01610
0630 0 4000  K4000 DC /4000 46 30D01620
*****
0631 0 4480 0160 PRBGN BSI I BEGIN CALL MONITOR 30D01640
0633 1 05DC DC PID 30D01650
*****
*          INTERRUPT ROUTINE
*
*          THE INTERRUPT ROUTINE IS
*          A STORE AND GO BIT OF
*          LOGIC. ALL INTERRUPT
*          STATUS WORDS ARE STORED
*          IN ORDER OF RECEIPT.
*
*          AN ERROR WILL BE DETECTED
*          IF THE INDICATOR FAILS
*          TO RESET.
*
0634 0 0000  INTRP DC 30D01810
0635 1 6780 0629  LDX 13 INTEX 30D01820
0637 0 08E2  XIO SENSR SENSE DEVICE - RESET 30D01830
*****
0638 0 18D0  RTE 16 30D01840
0639 0 C0EF  LD INTEX 30D01850
063A 1 4C18 0643  BSC L INTR2,+ BR IF FALSE INTRPT 30D01860
*****
063C 0 18D0  RTE 16 30D01870
063D 1 D700 062D  STD L3 INTEX+4 30D01880
*****
063F 1 7401 0629  MDX L INTEX,1 UPDATE STORAGE AREA 30D01890
0641 0 1000  NOP 30D01900
0642 0 7003  MDX INTR3 30D01910
*****
0643 0 C0E4  INTR2 LD K0010 SAVE FALSE INTRPT 30D01920
0644 0 18D0  RTE 16 ERROR 30D01930
0645 0 D8BC  STD DSWC 30D01940
*****
0646 1 4C80 0634  INTR3 BSC I INTRP EXIT 30D01950
*****
*          INITIALIZATION RTN
*
0648 0 10A0  RTO SLT 32 CLEAR ERROR AREAS 30D02000

```

1130-1403 FUNCTION TEST

```

0649 0 6303  LDX 3 3 30D02060
064A 1 6F00 0A1D STX L3 EMESS+2 30D02070
*****
064C 1 6500 0634  LDX L1 INTRP SET INTRPT VECTOR 30D02080
064E 0 6D00 01B6 STX L1 IL4-4 30D02090
0650 1 D400 05DD STD L RID CLEAR ROUTINE NO. 30D02100
*****
0652 0 D0D7  STD INTEX+1 CLEAR INTERRUPT DSW 30D02110
0653 0 D0D7  STD INTEX+2 STORAGE AREAS 30D02120
0654 0 D0D7  STD INTEX+3 30D02130
*****
0655 1 C400 05DF  LD L SWO CLEAR SINGLE CYCLE 30D02140
0657 0 480C  BSC +E IF RESTARTING 30D02150
0658 0 E0D4  AND K00BF 30D02160
0659 1 D400 05DF  STD L SWO 30D02170
065B 1 C400 05E1  LD L SW2 CHECK SIZE OPTION 30D02180
065D 0 4804  BSC E 30D02190
065E 0 7008  MDX RT02 30D02200
065F 0 C005  LD D60C SET TO 120 POS 30D02210
0660 1 D400 0627 RT01 STD L D60 30D02220
0662 1 D400 08C4 STD L RTN85 30D02230
0664 0 7004  MDX PCON BR TO PROG CONTROLLER 30D02240
*****
0665 0 003C  D60C DC 60 30D02250
0666 0 0042  D66C DC 66 30D02260
*****
0667 0 C0FE  RT02 LD D66C SET TO 132 POS 30D02270
0668 0 70F7  MDX RT01 30D02280
*****
*          PROGRAM CONTROLLER
*
*          THE PROGRAM CONTROLLER
*          SELECTS WHICH ROUTINES
*          ARE TO BE DONE NEXT.
*          NORMALLY ROUTINES ONE
*          THROUGH EIGHT ARE RUN
*          SEQUENTIALLY.
*
*          ANY ROUTINE WHICH IS MAN-
*          UALLY SELECTED BY A
*          SETTING OF THE SWITCHES,
*          (4Y0X)
*          WILL RUN UNTIL THOSE
*          SWITCHES ARE AGAIN
*          CLEARED.
*          (4Y00)
*
*          Y=PROGRAM LOANING SEQUENCE NUM.
*          X=ROUTINE NUMBER
*
0669 1 6600 05EA  PCON LDX L2 DST 30D02290
066B 0 C2F6  LD 2 SW1-DST 30D02300
066C 0 D239  STD 2 SWSAV-DST SAVE SWITCH SETTING 30D02310
066D 1 4C08 0677  BSC L PCON2,+ BR IF NO RTN SELECTD 30D02320
*****
066F 0 D2F3  STD 2 RID-DST SAVE NEW RTN NUMBER 30D02330
0670 0 9019  S RIDCK 30D02340
0671 1 4C08 067D  BSC L PCON3,+ BR IF VALID ROUTINE 30D02350
0673 0 1810  SRA 16 30D02360
0674 0 D2F6  STD 2 SW1-DST IF INVALID ROUTINE GO 30D02370
0675 0 D400 FFF3  STD L RID-DST TO RTN ONE 30D02380
*****
0677 1 7401 05DD  PCON2 MDX L RID,1 ADV TO NEXT RTN 30D02390

```

1130-1403 FUNCTION TEST

```

0679 0 C00F          LD      RTNOM          30D02740
067A 0 92F3          S        2 RID-DST          30D02750
067B 0 4488 0164    BSI  1  END,+      END PROG IF ZERO 30D02760
*
*
PCOM3 LDX  11 RID    30D02770
LD      L1 RTTBL-1  UPDATE ROUTINE NO. 30D02780
STO    L  RAD       30D02790
STO    L  MLSCF+1   SET MLSCF FOR RETURN 30D02800
STO    L  RTNSW    SET RTN START SW   30D02810
BSI    1  START    GO TO MONITOR      30D02820
*
RTNOM DC      NRTN-RTTBL+2 30D02830
RIDCK DC     LRTN-RTTBL+1 30D02840
*****
*
ROUTINE TABLE
*
RTTBL DC      RTN1  PRINTER READY - BUSY 30D02850
DC          RTN2  CARRIAGE SPACING      30D02860
DC          RTN3  NON PARITY CHECK      30D02870
DC          RTN4  BIT LINE CHECK        30D02880
DC          RTN5  PARITY CHECK          30D02890
DC          RTN6  CHARACTER COMPLEMENT 30D02900
DC          RTN7  SPACE AFTER PRINT     30D02910
DC          RTN8  SPACE BEFORE PRINT    30D02920
*
OPTIONAL ROUTINES
*
DC          RTN9  SKIP AFTER PRINT      30D02930
DC          RTNA  SKIP BEFORE PRINT     30D02940
LRTN DC     RTNB  OPTIONAL PRINT       30D02950
*****
*
ROUTINE ONE
*
PRINTER READY + BUSY TEST
*
THIS ROUTINE CHECKS THE
ABILITY OF THE 1403 TO
BECOME READY, BUSY, AND
NOT READY, AND THAT A
PRINT COMPLETE IS RE-
TURNED ON COMPLETION OF A
PRINTING OPERATION.
*
TEN BLANK LINES ARE
PRINTED USING A MINIMUM
AMOUNT OF CIRCUITRY WITH-
OUT SPACING THE CARRIAGE.
*
RTN1 BSI  L  SEBIT  CLEAR OUTPUT AREA 30D02960
DC          1      OUTPUT SIZE      30D02970
DC          BLANK  SOURCE           30D02980
DC          1      SOURCE SIZE      30D02990
*
LDX  1 10      SET FOR TEN LINES    30D03000
STX  L1 CYCNT 30D03010
*
RTN12 LDX  1 DSW1-DST CK PRINTER READY 30D03020
BSI  L  PTRDY  ERROR - 1           30D03030
*
XIO  2 PRT17-DST PRINT ONE LINE     30D03040
XIO  2 SENSN-DST SENSE DEVICE       30D03050
*
LDX  1 DSW5-DST CK PRINTER BUSY     30D03060

```

1130-1403 FUNCTION TEST

```

06A4 1 4400 09D7    BSI  L  SBANA  ERROR - 5          30D03420
*
06A6 1 4400 0979    BSI  L  ITDSW  CHECK INTERRUPT DSW    30D03430
06A8 0 0002          DC      2      TWO INTERRUPTS    30D03440
06A9 0 4002          DC     /4002  TRANSFER COMPLETE    30D03450
06AA 0 2000          DC     /2000  PRINT COMPLETE      30D03460
*
06AB 1 4400 0A2B    BSI  L  CKEND  GO CHECK IF END RTN 30D03470
06AD 0 70F0        MDX   RTN12  NO - REPEAT      30D03480
*****
*
ROUTINE TWO
*
CARRIAGE READY AND BUSY
TEST
*
THIS ROUTINE CHECKS THAT
THE 1403 CARRIAGE BUSY
INDICATOR FUNCTIONS
PROPERLY.
*
THE CARRIAGE IS SPACED TEN
LINES, EACH TIME CHECKING
FOR A CARRIAGE READY
BEFORE ISSUING THE SPACE
COMMAND AND A CARRIAGE
BUSY CONDITION RIGHT
AFTER ISSUING THE
COMMAND.
*
RTN2 LDX  1 10      SET CYCLE COUNTER 30D03490
STX  L1 CYCNT 30D03500
*
RTN21 LDX  1 DSW3-DST CK PRINTER READY 30D03510
BSI  L  PTRDY  ERROR - 5          30D03520
*
XIO  2 SPACE-DST SPACE CARRIAGE    30D03530
XIO  2 SENSN-DST SENSE DEVICE     30D03540
*
LDX  1 DSW6-DST CHECK BUSY        30D03550
BSI  L  SBANA  ERROR - 6          30D03560
*
BSI  L  ITDSW  CHECK INTERRUPT DSW 30D03570
DC      1      WAIT FOR ONE INTRPT 30D03580
DC     /1000  CARRIAGE OP COMP    30D03590
*
BSI  L  CKEND  GO CHECK IF END RTN 30D03600
MDX   RTN21  NO - REPEAT      30D03610
*****
*
ROUTINE THREE
*
PARITY CHECK (GOOD PARITY)
*
THIS ROUTINE CHECKS THE
ABILITY OF THE PRINTER TO
RESPOND TO CHARACTERS
HAVING CORRECT PARITY.
*
ROUTINE FIVE CHECKS THE
SAME CHARACTERS HAVING
BAD PARITY.

```



```

*
* RESPOND TO CHARACTERS
* HAVING INVALID PARITY
* BITS.
*
* A SINGLE BAD PARITY
* CHARACTER IS LOADED INTO
* THE OUTPUT BUFFER. THE
* PROGRAM THEN WAITS FOR A
* TRANSFER COMPLETE INTER-
* RUPT. IF ONE IS RECEIVED
* THE MULTIPLEXOR FAILED TO
* RECOGNIZE THE ERROR
* CONDITION. IN THIS CASE
* NO CHARACTER IS PRINTED.
* HOWEVER, IF NO TRANSFER
* COMPLETE INTERRUPT IS
* RECEIVED AND A PRINT
* CHECK OCCURS, THE PARITY
* BIT IS CORRECTED AND THE
* CHARACTER IS PRINTED.
* THIS PROCEDURE IS RE-
* PEATED FOR EACH OF THE
* 120 PRINT POSITIONS. NO
* SPACING IS DONE IN THIS
* ROUTINE.
*
071E 0 C039 RTN5 LD NUM5
071F 1 D400 OA53 STO L PAR+6
0721 1 4400 08F4 BSI L SEBIT SET UP RTN HEADING
0723 0 0007 DC 7 OUTPUT SIZE
0724 1 0A40 DC PAR SOURCE
0725 0 0007 DC 7 SOURCE SIZE
*
0726 1 4400 094F BSI L HDNG PRINT RTN HEADING
*
0728 0 C23D LD 2 D60-DST SET FOR 120 CHARS
0729 0 D21A STO 2 CYCNT-DST
072A 0 1810 SRA 16 INITIALIZE PARITY
072B 0 D007 STO BLNCT
072C 1 D400 OA25 STO L COLND
*
072E 1 6700 OA62 RTN51 LDX L3 PARE
0730 0 6808 STX 3 PACNT
*
0731 1 4400 08F4 RTN52 BSI L SEBIT SET RTN DATA
0733 0 0000 BLNCT DC 0 - 120 OUTPUT SIZE
0734 1 0A3B DC BLANK SOURCE
0735 0 0001 DC 1 SOURCE SIZE
0736 1 4400 08F4 BSI L SEBIT
0738 0 FFFF DC -1 ADD TO BLANK FIELD
0739 0 0000 PACNT DC 0 SOURCE (PARE)
073A 0 0001 DC 1 SOURCE SIZE
*
073B 0 6102 LDX 1 DSW1-DST CK PRINTER READY
073C 1 4400 092E BSI L PTRDY ERROR - 1
*
073E 0 0A2C XIO 2 PRTIT-DST PRINT THE LINE
*
073F 1 4400 09EC BSI L COLUM PRINT COL NO ON NEXT ERR
*
0741 1 4400 0979 BSI L ITDSW CHECK INTERRUPT DSW
0743 0 0002 DC 2 TWO INTERRUPTS
0744 0 C002 DC /CO02 XFER COMP + PARITY CK
0745 0 A000 DC /A000 PRINT COMP + PARITY CK
0746 0 6103 LDX 1 3 DD NOT PRINT COLUMN
0747 1 6D00 OA1D STX L1 EMESS+2 NO ON NEXT ERROR MSG

```

```

0749 0 COEF LD PACNT TAKE NEXT CHARACTER
074A 0 92FE S 2 TERM-DST
074B 0 D0ED STO PACNT
*
074C 0 4804 BSC E SKIP IF NEXT WORD
074D 0 70E3 MDX RTN52
*
074E 1 4400 0A2B BSI L CKEND CHECK IF END ROUTINE
*
0750 1 7401 0733 MDX L BLNCT,1 CHECK END OF DATA
0752 0 COE6 LD PACNT TABLE
0753 0 9003 S EPARE
0754 1 4C28 0731 BSC L RTN52,+Z
0756 0 70D7 MDX RTN51
*
0757 1 0A6F EPARE DC PARE+13
0758 0 7F04 NUM5 DC /7F04 =5 FOR RTN =
*****
*
* ROUTINE SIX
*
* CHARACTER COMPLEMENT
*
* THIS ROUTINE PRINTS EACH
* CHARACTER OF THE 48
* CHARACTER PRINT SET IN
* EACH OF THE 120 PRINT
* POSITIONS.
*
* THE CHARACTERS ARE ROTATED
* ONE POSITION TO THE LEFT
* WITH EACH SUCCEEDINGG
* LINE OF PRINT.
*
0759 1 4400 08F4 RTN6 BSI L SEBIT SET UP RTN HEADING
075B 0 000E DC 14 OUTPUT SIZE
075C 1 0A70 DC CHARC SOURCE
075D 0 000E DC 14 SOURCE SIZE
*
075E 1 4400 094F BSI L HDNG PRINT RTN HEADING
*
0760 1 4400 08F4 BSI L SEBIT SET UP RIPPLE PATTRN
0762 0 004E DC 78 OUTPUT SIZE
0763 1 0A7E DC CHAIN SOURCE
0764 0 001A DC 26 SOURCE SIZE
*
0765 0 C003 LD D54 SET CYCLE COUNTER
0766 0 D21A STO 2 CYCNT-DST
*
0767 1 4C00 070A BSC L RTN41 GO TO PRINT IT
*
0769 0 0036 D54 DC 54
*****
*
* ROUTINE SEVEN
*
* SPACE AFTER PRINT TEST
*
* THIS ROUTINE PROVIDES ONE
* CHECK OF THE SPACING
* CIRCUITRY.

```

```

*           A SPACE COMMAND IS ISSUED          30D06820
*           RIGHT AFTER GIVING A                30D06830
*           PRINT COMMAND. THE DSW             30D06840
*           INDICATORS ARE CHECKED            30D06850
*           FOR VALIDITY AFTER THE            30D06860
*           SPACE COMMAND AND DURING          30D06870
*           INTERRUPT.                        30D06880
*                                           30D06890
*                                           30D06900
*                                           30D06910
076A 1 4400 08F4 RTN7 BSI L SEBIT SET RTN HEADING 30D06920
076C 0 000D          DC    13  OUTPUT SIZE    30D06930
076D 1 0AB1          DC    SPAP  SOURCE        30D06940
076E 0 000D          DC    13  SOURCE SIZE    30D06950
*                                           30D06960
076F 1 4400 094F          BSI L HDNG GO PRINT THE HEADING 30D06970
*                                           30D06980
*                                           30D06990
*                                           30D07000
0771 0 610A          LDX  1 10  SET THE CYCLE COUNT 30D07010
0772 1 6D00 0604          STX LI CYCNT 30D07020
*                                           30D07030
*                                           30D07040
*                                           30D07050
*                                           30D07060
*                                           30D07070
0774 1 4400 08F4          BSI L SEBIT CLEAR OUTPUT TABLE 30D07080
0776 0 0014          DC    20  OUTPUT SIZE    30D07090
0777 1 0A3B          DC    BLANK SOURCE        30D07100
0778 0 0001          DC    1  DATA SIZE    30D07110
*                                           30D07120
*                                           30D07130
*                                           30D07140
*                                           30D07150
0779 1 4400 08F4          BSI L SEBIT SET UP PRINT DATA 30D07160
077B 0 FFD8          DC   -40  OUTPUT SIZE    30D07170
077C 1 0ACF          DC    E  SOURCE        30D07180
077D 0 0091          DC    1  SOURCE SIZE    30D07190
*                                           30D07200
*                                           30D07210
*                                           30D07220
077E 1 CC00 0AE0          LDD  L SLSH          30D07230
0780 1 D400 0B2A          STO  L AREA+60          30D07240
0782 0 18D0          RTE  16          30D07250
0783 1 D400 0B02          STO  L AREA+20          30D07260
*                                           30D07270
*                                           30D07280
0785 1 4400 08E4 RTN72 BSI L ROTA ROTATE OUTPUT CHARS 30D07290
*                                           30D07300
*                                           30D07310
0787 0 6102          LDX  1 DSW1-DST CHECK PRINTER READY 30D07320
0788 1 4400 092E          BSI L PTRDY ERROR - 1 30D07330
*                                           30D07340
*                                           30D07350
078A 0 0A2C          XIO  2 PRIT-DST PRINT THE LINE 30D07360
078B 0 0A2E          XIO  2 SENSN-DST SENSE DEVICE 30D07370
*                                           30D07380
*                                           30D07390
*                                           30D07400
078C 0 610A          LDX  1 DSW5-DST CHECK FOR BUSY 30D07410
078D 1 4400 09D7          BSI L SBANA ERROR - 5 30D07420
*                                           30D07430
*                                           30D07440
078F 1 4400 0979          BSI L ITDSW          30D07450
0791 0 0001          DC    1  ONE INTERRUPT 30D07460
0792 0 4002          DC   /4002 TRANSFER COMPLETE 30D07470
*                                           30D07480
*                                           30D07490
0793 0 0A28          XIO  2 SPACE-DST SPACE CARRIAGE 30D07500
*                                           30D07510
*                                           30D07520
0794 1 4400 0979          BSI L ITDSW          30D07530
0796 0 0003          DC    3  THREE INTRPTS 30D07540
0797 0 4002          DC   /4002 TRANSFER COMPLETE 30D07550
0798 0 2004          DC   /2004 PRINT COMPLETE 30D07560
0799 0 1000          DC   /1000 CARRIAGE OP COMPLETE 30D07570
*                                           30D07580
*                                           30D07590
079A 1 4400 0A2B          BSI L CKEND GO CHECK IF END RTN 30D07600
*                                           30D07610
*                                           30D07620
079C 0 70E8          MDX  RTN72 PRINT NEXT LINE 30D07630
*                                           30D07640
*                                           30D07650
*                                           30D07660
*                                           30D07670
*                                           30D07680
*                                           30D07690
*                                           30D07700
*                                           30D07710
*                                           30D07720
*                                           30D07730
*                                           30D07740
*                                           30D07750
*                                           30D07760
*                                           30D07770
*                                           30D07780
*                                           30D07790
*                                           30D07800
*                                           30D07810
*                                           30D07820
*                                           30D07830
*                                           30D07840
*                                           30D07850
*                                           30D07860
*                                           30D07870
*                                           30D07880
*                                           30D07890
*                                           30D07900
*                                           30D07910
*                                           30D07920
*                                           30D07930
*                                           30D07940
*                                           30D07950
*                                           30D07960
*                                           30D07970
*                                           30D07980
*                                           30D07990
*                                           30D08000
*                                           30D08010
*                                           30D08020
*                                           30D08030
*                                           30D08040
*                                           30D08050
*                                           30D08060
*                                           30D08070
*                                           30D08080
*                                           30D08090
*                                           30D08100
*                                           30D08110
*                                           30D08120
*                                           30D08130
*                                           30D08140
*                                           30D08150
*                                           30D08160
*                                           30D08170

```

```

*           SECOND CHECK OF THE                30D07500
*           SPACING CIRCUITRY.                30D07510
*                                           30D07520
*           A SPACE COMMAND IS ISSUED          30D07530
*           BEFORE GIVING A PRINT             30D07540
*           COMMAND THE DSW INDICA-          30D07550
*           TORS ARE CHECKED FOR             30D07560
*           VALIDITY AFTER THE PRINT          30D07570
*           COMMAND AND DURING                30D07580
*           INTERRUPT.                        30D07590
*                                           30D07600
*                                           30D07610
*                                           30D07620
079D 1 4400 08F4 RTN8 BSI L SEBIT SET UP RTN HEADING 30D07630
079F 0 000D          DC    13  OUTPUT SIZE    30D07640
07A0 1 0ABE          DC    SPBP  SOURCE        30D07650
07A1 0 000D          DC    13  SOURCE SIZE    30D07660
*                                           30D07670
*                                           30D07680
07A2 1 4400 094F          BSI L HDNG GO PRINT THE HEADING 30D07690
*                                           30D07700
*                                           30D07710
*                                           30D07720
07A4 0 610A          LDX  1 10  SET THE CYCLE COUNT 30D07730
07A5 1 6D00 0604          STX LI CYCNT 30D07740
*                                           30D07750
*                                           30D07760
*                                           30D07770
*                                           30D07780
07A7 1 4400 08F4          BSI L SEBIT CLEAR OUTPUT AREA 30D07790
07A9 0 0014          DC    20  OUTPUT SIZE    30D07800
07AA 1 0A3B          DC    BLANK SOURCE        30D07810
07AB 0 0001          DC    1  SOURCE SIZE    30D07820
*                                           30D07830
*                                           30D07840
07AC 1 4400 08F4          BSI L SEBIT SET UP PRINT DATA 30D07850
07AE 0 FFD8          DC   -40  OUTPUT SIZE    30D07860
07AF 1 0ACF          DC    E  SOURCE        30D07870
07B0 0 0001          DC    1  SOURCE SIZE    30D07880
*                                           30D07890
*                                           30D07900
*                                           30D07910
*                                           30D07920
07B1 1 CC00 0AE0          LDD  L SLSH          30D07930
07B3 1 D400 0B2A          STO  L AREA+60          30D07940
07B5 0 18D0          RTE  16          30D07950
07B6 1 D400 0B02          STO  L AREA+20          30D07960
*                                           30D07970
*                                           30D07980
*                                           30D07990
07B8 1 4400 08E4 RTN82 BSI L ROTA ROTATE OUTPUT CHARS 30D08000
*                                           30D08010
*                                           30D08020
07BA 0 6102          LDX  1 DSW1-DST CHECK PRINTER READY 30D08030
07BB 1 4400 092E          BSI L PTRDY ERROR - 1 30D08040
*                                           30D08050
*                                           30D08060
07BD 0 0A28          XIO  2 SPACE-DST SPACE CARRIAGE 30D08070
07BE 0 0A2C          XIO  2 PRIT-DST PRINT THE LINE 30D08080
07BF 0 0A2E          XIO  2 SENSN-DST SENSE DEVICE 30D08090
*                                           30D08100
*                                           30D08110
07C0 0 6108          LDX  1 DSW4-DST CHECK FOR BUSY 30D08120
07C1 1 4400 09D7          BSI L SBANA ERROR - 4 30D08130
*                                           30D08140
*                                           30D08150
*                                           30D08160
07C3 1 4400 0979          BSI L ITDSW          30D08170
07C5 0 0003          DC    3  THREE INTERRUPTS 30D08180
07C6 0 4006          DC   /4006 TRANSFER COMPLETE 30D08190
07C7 0 1002          DC   /1002 CARRIAGE OP COMPLETE 30D08200
07C8 0 2000          DC   /2000 PRINT COMPLETE 30D08210
*                                           30D08220
*                                           30D08230
07C9 1 4400 0A2B          BSI L CKEND CK IF END OF ROUTINE 30D08240
07CB 0 70EC          MDX  RTN82 PRINT NEXT LINE 30D08250
*                                           30D08260
*                                           30D08270
*                                           30D08280
*                                           30D08290
*                                           30D08300
*                                           30D08310
*                                           30D08320
*                                           30D08330
*                                           30D08340
*                                           30D08350
*                                           30D08360
*                                           30D08370
*                                           30D08380
*                                           30D08390
*                                           30D08400
*                                           30D08410
*                                           30D08420
*                                           30D08430
*                                           30D08440
*                                           30D08450
*                                           30D08460
*                                           30D08470
*                                           30D08480
*                                           30D08490
*                                           30D08500
*                                           30D08510
*                                           30D08520
*                                           30D08530
*                                           30D08540
*                                           30D08550
*                                           30D08560
*                                           30D08570
*                                           30D08580
*                                           30D08590
*                                           30D08600
*                                           30D08610
*                                           30D08620
*                                           30D08630
*                                           30D08640
*                                           30D08650
*                                           30D08660
*                                           30D08670
*                                           30D08680
*                                           30D08690
*                                           30D08700
*                                           30D08710
*                                           30D08720
*                                           30D08730
*                                           30D08740
*                                           30D08750
*                                           30D08760
*                                           30D08770
*                                           30D08780
*                                           30D08790
*                                           30D08800
*                                           30D08810
*                                           30D08820
*                                           30D08830
*                                           30D08840
*                                           30D08850
*                                           30D08860
*                                           30D08870
*                                           30D08880
*                                           30D08890
*                                           30D08900
*                                           30D08910
*                                           30D08920
*                                           30D08930
*                                           30D08940
*                                           30D08950
*                                           30D08960
*                                           30D08970
*                                           30D08980
*                                           30D08990
*                                           30D09000
*                                           30D09010
*                                           30D09020
*                                           30D09030
*                                           30D09040
*                                           30D09050
*                                           30D09060
*                                           30D09070
*                                           30D09080
*                                           30D09090
*                                           30D09100
*                                           30D09110
*                                           30D09120
*                                           30D09130
*                                           30D09140
*                                           30D09150
*                                           30D09160
*                                           30D09170

```

```

*
* A SPECIAL CARRIAGE TAPE
* WITH CHANNELS PUNCHED
* EQUALLY SPACED ON THE
* TAPE IS REQUIRED.
*
* A CHANNEL SKIP COMMAND
* IS ISSUED AFTER GIVING
* A PRINT COMMAND. ALL
* DSW INDICATORS ARE
* CHECKED FOR VALIDITY
* BEFORE AND AFTER ISSUING
* THE COMMAND AND DURING
* THE INTERRUPTS.
*
07CC 1 4400 08F4 RTN9 BSI L SEBIT SET UP RTN HEADING
07CE 0 000C DC 12 OUTPUT SIZE
07CF 1 0A98 DC SKAP SOURCE
07D0 0 000C DC 12 SOURCE SIZE
*
07D1 0 61FF LDX 1 -1 SET CHANNEL POINTER
07D2 1 6D00 0887 STX L1 CHAN1
*
07D4 0 C23C LD 2 K0800-DST SET THE CHANNEL SKIP
07D5 0 D236 STO 2 SKPWD-DST
*
07D6 1 6500 0AE2 LDX L1 CHAND
07D8 1 6D00 088D STX L1 CHAN2
*
07DA 0 61E0 LDX 1 -32
07DB 1 6D00 086F STX L1 ELIN1
*
07DD 0 610D LDX 1 13
07DE 1 6D00 0604 STX L1 CYCNT
*
07E0 0 6102 RTN91 LDX 1 DSW1-DST
07E1 1 4400 092E BSI L PTRDY CHECK PRINTER READY
*
07E3 0 0A2C XIO 2 PRTIT-DST PRINT THE LINE
*
07E4 0 0A2E XIO 2 SENSN-DST SENSE DEVICE
*
07E5 0 610A LDX 1 DSW5-DST CHECK IF BUSY
07E6 1 4400 09D7 BSI L SBANA ERROR - 5
*
07E8 1 4400 0979 BSI L ITDSW WAIT FOR XFER CMPLT
07EA 0 0001 DC /0001 ONE INTERRUPT DSW
07EB 0 4002 DC /4002 TRANSFER COMPLETE
*
07EC 0 0A2A XIO 2 SKIP-DST SKIP TO CHANNEL
*
07ED 1 6580 0AEE LDX 11 AREA CHECK IF CHAN SET
07EF 0 71F7 MDX 1 -9
07F0 0 7005 MDX RTN94
*
07F1 1 4400 08F4 BSI L SEBIT SET CHANNEL WORD
07F3 0 0004 DC 4 OUTPUT SIZE
07F4 1 0ACB DC CHAN SOURCE
07F5 0 0004 DC 4 SOURCE SIZE
*
07F6 0 406A RTN94 BSI ELIN SET E LINE
*
07F7 1 4400 0979 BSI L ITDSW WAIT FOR INTRPTS
07F9 0 0003 DC 3 THREE INTERRUPTS
07FA 0 4002 DC /4002 TRANSFER COMPLETE
07FB 0 2004 DC /2004 PRINT COMPLETE
07FC 0 1000 DC /1000 CARRIAGE OP COMPLETE

```

```

30D08180
30D08190
30D08200
30D08210
30D08220
30D08230
30D08240
30D08250
30D08260
30D08270
30D08280
30D08290
30D08300
30D08310
30D08320
30D08330
30D08340
30D08350
30D08360
30D08370
30D08380
30D08390
30D08400
30D08410
30D08420
30D08430
30D08440
30D08450
30D08460
30D08470
30D08480
30D08490
30D08500
30D08510
30D08520
30D08530
30D08540
30D08550
30D08560
30D08570
30D08580
30D08590
30D0860C
30D08610
30D08620
30D08630
30D08640
30D08650
30D08660
30D08670
30D08680
30D08690
30D08700
30D08710
30D08720
30D08730
30D08740
30D08750
30D08760
30D08770
30D08780
30D08790
30D08800
30D08810
30D08820
30D08830
30D08840
30D08850

```

```

07FD 0 6110 LDX 1 16
07FE 1 C400 0AFB LD L AREA+13
0800 1 F400 0A3B EOR L BLANK
0802 0 4818 BSC +- BR IF CHANNEL 9
0803 0 6108 LDX 1 8
0804 1 F400 0AFE EOR L AREA+16
0806 1 F400 0A3B EOR L BLANK
0808 0 4818 BSC +- BR IF CHANNEL C
*
0809 0 6100 LDX 1 0
080A 0 7500 1000 MDX L1 /1000
080C 1 6D00 05F8 STX L1 DSW7 SET DSW EXPECTED
*
080E 0 C242 LD 2 INTEX+3-DST FETCH DSW
*
080F 0 610E LDX 1 DSW7-DST CHECK CHAN IND AND
0810 1 4400 09D7 BSI L SBANA READY ERROR-7
*
0812 1 4400 087F BSI L CHANQ UPDATE CHAN NO
*
0814 1 4400 0A2B BSI L CKEND CHECK IF END OF RTN
0816 0 70C9 MDX RTN91 NO - REPEAT
*****
*
* ROUTINE A
*
* CARRIAGE SKIP IMMEDIATE
* TEST
*
* THIS ROUTINE PROVIDES A
* SECOND CHECK OF THE
* CARRIAGE SKIP CIRCUITRY.
*
* AGAIN, THE SPECIAL
* CARRIAGE TAPE WITH
* CHANNELS PUNCHED EQUALLY
* SPACED IS REQUIRED.
*
* A CHANNEL SKIP COMMAND IS
* ISSUED BEFORE GIVING A
* PRINT COMMAND. ALL INDI-
* CATORS ARE CHECKED FOR
* VALIDITY BEFORE AND AFTER
* THE COMMAND IS ISSUED
* AND DURING THE INTER-
* RUPTS.
*
0817 1 4400 08F4 RTNA BSI L SEBIT SET UP RTN HEADING
0819 0 000D DC 13 OUTPUT SIZE
081A 1 0AA4 DC SKBP SOURCE
081B 0 000D DC 13 SOURCE SIZE
*
081C 1 4400 094F BSI L HDNG PRINT THE HEADING
*
081E 0 C23C LD 2 K0800-DST SET THE CHANNEL SKIP
081F 0 D236 STO 2 SKPWD-DST
*
0820 0 61FF LDX 1 -1
0821 1 6D00 0887 STX L1 CHAN1
*
0823 1 6500 0AE2 LDX L1 CHAND
0825 1 6D00 088D STX L1 CHAN2
*
0827 0 61E0 LDX 1 -32
0828 0 6946 STX 1 ELIN1
0829 0 610C LDX 1 12
082A 1 6D00 0604 STX L1 CYCNT

```

```

30D08860
30D08870
30D08880
30D08890
30D08900
30D08910
30D08920
30D08930
30D08940
30D08950
30D08960
30D08970
30D08980
30D08990
30D09000
30D09010
30D09020
30D09030
30D09040
30D09050
30D09060
30D09070
30D09080
30D09090
30D09100
30D09110
30D09120
30D09130
30D09140
30D09150
30D09160
30D09170
30D09180
30D09190
30D09200
30D09210
30D09220
30D09230
30D09240
30D09250
30D09260
30D09270
30D09280
30D09290
30D09300
30D09310
30D09320
30D09330
30D09340
30D09350
30D09360
30D09370
30D09380
30D09390
30D09400
30D09410
30D09420
30D09430
30D09440
30D09450
30D09460
30D09470
30D09480
30D09490
30D09500
30D09510
30D09520
30D09530

```



1130-1403 FUNCTION TEST

```

*      8-15 WILL NOT BE HONORED.      30D10900
*      REFER TO DOCUMENTATION FOR     30D10910
*      CODING OR CHECK LISTING        30D10920
*      BEGINNING AT LABEL LOCATION    30D10930
*      'CHARC'.                        30D10940
*      30D10950
*      DOUBLE SPACE OPTION IS AVAILABLE 30D10960
*      IF BIT SW 1 IS ON. THIS CHECKS 30D10970
*      THE ABILITY TO STACK TWO SPACE 30D10980
*      INSTRUCTIONS.                   30D10990
*      30D11000
*      30D11010
*      30D11020
*      30D11030
*      30D11040
*      30D11050
*      30D11060
*      30D11070
*      30D11080
*      30D11090
*      30D11100
*      30D11110
*      30D11120
*      30D11130
*      30D11140
*      30D11150
*      30D11160
*      30D11170
*      30D11180
*      30D11190
*      30D11200
*      30D11210
*      30D11220
*      30D11230
*      30D11240
*      30D11250
*      30D11260
*      30D11270
*      30D11280
*      30D11290
*      30D11300
*      30D11310
*      30D11320
*      30D11330
*      30D11340
*      30D11350
*      30D11360
*      30D11370
*      30D11380
*      30D11390
*      30D11400
*      30D11410
*      30D11420
*      30D11430
*      30D11440
*      30D11450
*      30D11460
*      30D11470
*      30D11480
*      30D11490
*      30D11500
*      30D11510
*      30D11520
*      30D11530
*      30D11540
*      30D11550
*      30D11560
*      30D11570

```

1130-1403 FUNCTION TEST

```

08D8 0 4006      DC      /4006      TRANSFER COMPLETE      30D11580
08D9 0 2004      DC      /2004      PRINTER COMPLETE      30D11590
08DA 0 1000      DC      /1000      CARRIAGE OP COMPLETE  30D11600
30D11610
30D11620
30D11630
30D11640
30D11650
30D11660
30D11670
30D11680
30D11690
30D11700
30D11710
30D11720
30D11730
30D11740
30D11750
30D11760
30D11770
30D11780
30D11790
30D11800
30D11810
30D11820
30D11830
30D11840
30D11850
30D11860
30D11870
30D11880
30D11890
30D11900
30D11910
30D11920
30D11930
30D11940
30D11950
30D11960
30D11970
30D11980
30D11990
30D12000
30D12010
30D12020
30D12030
30D12040
30D12050
30D12060
30D12070
30D12080
30D12090
30D12100
30D12110
30D12120
30D12130
30D12140
30D12150
30D12160
30D12170
30D12180
30D12190
30D12200
30D12210
30D12220
30D12230
30D12240
30D12250

```



1130-1403 FUNCTION TEST

```

08FD 0 D01C      STO SEB3+1  SAVE NEG WD COUNT  30D12260
08FE 0 9200      S      2 0      ADD TO EXISTING DATA  30D12270
08FF 0 700B      MDX SEB2                                     30D12280
*                                                       30D12290
0900 1 6580 0627 SEB1 LDX 11 D60      CLEAR OUTPUT AREA  30D12300
0902 1 C400 0A3B LD L BLANK                                     30D12310
0904 1 D500 0AEE SEBC STO L1 AREA                                     30D12320
0906 0 71FF      MDX 1 -1                                     30D12330
0907 0 70FC      MDX SEBC      CLEAR NEXT WORD  30D12340
*                                                       30D12350
0908 0 1010      SLA 16      MAKE WD COUNT NEG  30D12360
0909 0 9300      S      3 0                                     30D12370
090A 0 D00F      STO SEB3+1  SAVE NEG WD COUNT  30D12380
*                                                       30D12390
090B 0 D200      SEB2 STO 2 0      SET TABLE WORD COUNT  30D12400
090C 1 4C10 0927 BSC L SEB8,- BR IF WD CT ZERO  30D12410
090E 0 1010      SLA 16                                     30D12420
090F 0 9200      S      2 0                                     30D12430
0910 0 D200      STO 2 0                                     30D12440
*                                                       30D12450
0911 0 8018      A      SEARA 30D12460
0912 0 D00D      STO SEB6+1 SET STORAGE ADDRESS  30D12470
*                                                       30D12480
0913 0 C301      LD 3 1      SET FETCH ADDRESS  30D12490
0914 0 8302      A      3 2      OF INPUT DATA  30D12500
0915 0 D008      STO SEB5+1                                     30D12510
*                                                       30D12520
0916 0 1010      SLA 16      FETCH SOURCE SIZE  30D12530
0917 0 9302      S      3 2                                     30D12540
0918 0 D003      STO SEB4+1                                     30D12550
*                                                       30D12560
0919 0 6600 0000 SEB3 LDX L2 *--      XR2 = - NO OF XFR WDS  30D12570
*                                                       30D12580
091B 0 6500 0000 SEB4 LDX L1 *--      XR1 = - SOURCE SIZE  30D12590
091D 0 C500 0000 SEB5 LD L1 *--      XFER SOURCE DATA  30D12600
091F 0 D600 0000 SEB6 STO L2 *--      TO OUTPUT AREA  30D12610
0921 0 7101      MDX 1 1                                     30D12620
0922 0 7002      MDX SEB7      TAKE NEXT SOURCE WD  30D12630
*                                                       30D12640
0923 1 6580 091C LDX 11 SEB4+1 RESTORE SOURCE ADDR  30D12650
*                                                       30D12660
0925 0 7201      SEB7 MDX 2 1                                     30D12670
0926 0 70F6      MDX SEB5      GO TRANSFER NEXT WORD  30D12680
*                                                       30D12690
0927 0 6500 0000 SEB8 LDX L1 *--      RESTORE XR1  30D12700
0929 1 6600 05EA LDX L2 DST                                     30D12710
092B 0 4F00 0003 BSC L3 3      RETURN TO CALL  30D12720
*                                                       30D12730
092D 1 0AEF      SEARA DC AREA+1 ADDR OF OUTPUT AREA  30D12740
*****
*                                                       30D12750
*                                                       30D12760
*                                                       30D12770
*                                                       30D12780
*                                                       30D12790
*                                                       30D12800
*                                                       30D12810
*                                                       30D12820
*                                                       30D12830
*                                                       30D12840
*                                                       30D12850
*                                                       30D12860
*                                                       30D12870
*                                                       30D12880
*                                                       30D12890
*                                                       30D12900
*                                                       30D12910
*                                                       30D12920
*                                                       30D12930
092E 0 0000      PTRDY DC                                     30D12940
092F 1 6600 05EA LDX L2 DST                                     30D12950
0931 0 6913      STX 1 PTRY3+1                                     30D12960

```

1130-1403 FUNCTION TEST

```

0932 1 4400 096E BSI L CLEAR      SET FOR INTERRUPT  30D12940
0934 0 C23A      LD 2 K0001-DST  30D12950
0935 0 D218      PTRY1 STO 2 WACNT-DST SET WAIT COUNTER  30D12960
*                                                       30D12970
0936 1 6600 05EA PTRY2 LDX L2 DST  30D12980
0938 1 6580 0945 LDX 11 PTRY3+1  30D12990
093A 0 0A2E      XIO 2 SENSN-DST SENSE DEVICE  30D13000
*                                                       30D13010
093B 1 F500 05EA EOR L1 DST      CK AGAINST EXPECTED  30D13020
093D 1 E500 05EB AND L1 DST+1     REMOVE UNNEC BITS  30D13030
093F 1 4C98 092E BSC 1 PTRDY,+-- BR IF PRINTER READY  30D13040
*                                                       30D13050
0941 1 74FF 0605 MDX L WACNT,-1   IS ENOUGH TIME  30D13060
0943 0 7007      MDX PTRY9      NO - TRY AGAIN  30D13070
*                                                       30D13080
0944 0 6500 0000 PTRY3 LDX L1 *--  YES - ERROR  30D13090
0946 0 0A2E      XIO 2 SENSN-DST SENSE DEVICE  30D13100
0947 1 4400 09D7 BSI L SBANA      GO PRINT THE ERROR  30D13110
0949 0 C23B      LD 2 K0400-DST  30D13120
094A 0 70EA      MDX PTRY1  30D13130
*                                                       30D13140
094B 1 6600 0936 PTRY9 LDX L2 PTRY2  30D13150
094D 1 4C00 098D BSC L LOPGO      TRY AGAIN - LATER  30D13160
*****
*                                                       30D13170
*                                                       30D13180
*                                                       30D13190
*                                                       30D13200
*                                                       30D13210
*                                                       30D13220
*                                                       30D13230
*                                                       30D13240
*                                                       30D13250
*                                                       30D13260
*                                                       30D13270
*                                                       30D13280
*                                                       30D13290
*                                                       30D13300
*                                                       30D13310
*                                                       30D13320
*                                                       30D13330
*                                                       30D13340
*                                                       30D13350
*                                                       30D13360
*                                                       30D13370
*                                                       30D13380
*                                                       30D13390
*                                                       30D13400
*                                                       30D13410
*                                                       30D13420
*                                                       30D13430
*                                                       30D13440
*                                                       30D13450
*                                                       30D13460
*                                                       30D13470
*                                                       30D13480
*                                                       30D13490
*                                                       30D13500
*                                                       30D13510
*                                                       30D13520
*                                                       30D13530
*                                                       30D13540
*                                                       30D13550
*                                                       30D13560
*                                                       30D13570
*                                                       30D13580
*                                                       30D13590
*                                                       30D13600
*                                                       30D13610
094F 0 0000      HDNG DC 1 3      SET FOR THREE SPACES  30D13620
0950 0 6103      LDX 1 3                                     30D13630
0951 0 6918      STX 1 HDNGA                                     30D13640
0952 0 6102      HDNG1 LDX 1 DSW1-DST CHECK PRINTER READY  30D13650
0953 1 4400 092E BSI L PTRDY      ERROR - 1  30D13660
*                                                       30D13670
0955 0 0A28      XIO 2 SPACE-DST SPACE CARRIAGE  30D13680
*                                                       30D13690
0956 1 4400 0979 BSI L ITDSW      CHECK INTERRUPT DSW  30D13700
0958 0 0001      DC 1      WAIT FOR ONE INTRPT  30D13710
0959 0 1000      DC /1000 CARRIAGE OP COMPLETE  30D13720
*                                                       30D13730
095A 1 74FF 096D MDX L HDNGA,-1   SKIP IF DONE  30D13740
095C 0 70F5      MDX HDNG1                                     30D13750
*                                                       30D13760
095D 0 4010      BSI CLEAR      SET FOR NEXT INTERRUPT  30D13770
*                                                       30D13780
095E 0 0A2C      XIO 2 PRTIT-DST PRINT THE HEADING  30D13790
*                                                       30D13800
095F 1 4400 0979 BSI L ITDSW      CHECK INTERRUPT DSW  30D13810
0961 0 0002      DC 2      WAIT FOR TWO INTRPTS  30D13820
0962 0 4002      DC /4002 TRANSFER COMPLETE  30D13830
0963 0 2000      DC /2000 PRINT COMPLETE  30D13840
*                                                       30D13850
0964 0 6102      LDX 1 DSW1-DST CHECK PRINTER READY  30D13860
0965 0 40C8      BSI PTRDY      ERROR - 1  30D13870
*                                                       30D13880
0966 0 0A28      XIO 2 SPACE-DST SPACE CARRIAGE  30D13890
*                                                       30D13900
*                                                       30D13910
*                                                       30D13920
*                                                       30D13930
0967 1 4400 0979 BSI L ITDSW      CHECK INTERRUPT DSW  30D13940
0969 0 0001      DC 1      WAIT FOR ONE INTERRU  30D13950
096A 0 1000      DC /1000 CARRIAGE OP COMPLPETE  30D13960
*                                                       30D13970
096B 1 4C80 094F BSC 1 HDNG      RETURN TO CALL  30D13980
*                                                       30D13990
096D 0 0000      HDNGA DC 0      SPACE COUNTER  30D14000
*****
*                                                       30D14010
*                                                       30D14020
*                                                       30D14030
*                                                       30D14040
*                                                       30D14050
*                                                       30D14060
*                                                       30D14070
*                                                       30D14080
*                                                       30D14090
*                                                       30D14100
*                                                       30D14110
*                                                       30D14120
*                                                       30D14130
*                                                       30D14140
*                                                       30D14150
*                                                       30D14160
*                                                       30D14170
*                                                       30D14180
*                                                       30D14190
*                                                       30D14200
*                                                       30D14210
*                                                       30D14220
*                                                       30D14230
*                                                       30D14240
*                                                       30D14250
*                                                       30D14260
*                                                       30D14270
*                                                       30D14280
*                                                       30D14290
*                                                       30D14300
*                                                       30D14310
*                                                       30D14320
*                                                       30D14330
*                                                       30D14340
*                                                       30D14350
*                                                       30D14360
*                                                       30D14370
*                                                       30D14380
*                                                       30D14390
*                                                       30D14400
*                                                       30D14410
*                                                       30D14420
*                                                       30D14430
*                                                       30D14440
*                                                       30D14450
*                                                       30D14460
*                                                       30D14470
*                                                       30D14480
*                                                       30D14490
*                                                       30D14500
*                                                       30D14510
*                                                       30D14520
*                                                       30D14530
*                                                       30D14540
*                                                       30D14550
*                                                       30D14560
*                                                       30D14570
*                                                       30D14580
*                                                       30D14590
*                                                       30D14600
*                                                       30D14610

```

```
*
* CLEAR DC
096E 0 0000 LDX 3 -3 CLEAR INTERRUPT
096F 0 63FD STX L3 INTEX STORAGE AREAS
0970 1 6F00 0629 SLA 16
0972 0 1010 CLEA1 STO L3 INTEX+4
0973 1 0700 062D MDX 3 1
0975 0 7301 MDX CLEA1
0976 0 70FC
*
0977 1 4C80 096E BSC 1 CLEAR RETURN TO USER
*****
*
* CHECK INTERRUPT DSW
SUBROUTINE
*
ITDSW DC
0979 0 0000 LD 2 K0400-DST SET WAIT COUNTER
097A 0 C238 STO 2 WACNT-DST
097B 0 D219 LD 1 ITDSW FIND RETURN ADDRESS
097C 1 C480 0979 A ITDSW
097E 0 80FA S 2 TERM-DST
097F 0 92FE STO ITDS9+1 SAVE RETURN ADDRESS
0980 0 D050
*
ITDS1 LDX 11 ITDSW CK ALL INTRPTSRECD
0981 1 6580 0979 LDX L2 DST RESTORE XR2
0983 1 6600 05EA LD 1 0
0985 0 C100 STO ITDS4+1
0986 0 D010 S THREE
0987 0 9048 S 2 INTEX-DST
0988 0 923F BSC L ITDS4,+ BR IF ALL INTRPTS RECD
0989 1 4C08 0996
*
LOPGD LDX L2 ITDS3 WAIT FOR INTERRUPTS
098B 1 6600 0991 STX L2 MLSCF
098D 1 6E00 05E5 BSI 1 START TRY AGAIN - LATER
098F 0 4480 0161
*
ITDS3 MDX L WACNT,-1 WAITED LONG ENOUGH
0991 1 74FF 0605 MDX ITDS1 NO - RETURN
0993 0 70ED
*
LDX 1 /40 YES - LOST INTERRUPT
0994 0 6140 MDX ITDS6 ERROR - 2X
0995 0 7019
*
ITDS4 LDX L3 *-- FETCH NO OF INTRPTS
0996 0 6700 0000 SLT 32
0998 0 10A0 STO ESAVE CLEAR SAVE WORD
0999 0 D03A EOR 1 1 CHECK INTERRUPT DSW
099A 0 F101 EOR 2 INTEX+1-DST
099B 0 F240 STO RSAVE TEMP SAVE
099C 0 D038 OR ESAVE SAVE ANY ERRORS
099D 0 E836 STO ESAVE
099E 0 D035 LD RSAVE RESTORE ACC
099F 0 C035 MDX 1 1
09A0 0 7101 MDX 2 1
09A1 0 7201 MDX 3 -1
09A2 0 73FF MDX ITDS5 ALL EXPECTED INCLUDED
09A3 0 70F6 NO - RETURN
*
MDX 1 1
09A4 0 7101 LDX L2 DST
09A5 1 6600 05EA AND 2 DSW1-DST CLEAR CARRIAGE BITS
09A7 0 E202 BSC L ITDS6-1,2 BR IF DSW ERKOR
09A8 1 4C20 09AE LD ESAVE POSSIBLE ERRORS
09AA 0 C029 AND K8FE7 REMOVE ALL BUT ERR BITS
09AB 0 E02A BSC L1 0,+ BR IF DSW OK
09AC 0 4D18 0000
*
LDX 1 /20 ERROR 1X
09AE 0 6120 MDX 11 RID DSW ERROR IN INTERPT
09AF 1 7580 05DD MDX 11 RID
09B1 1 7500 05DD
```

30013620  
30013630  
30013640  
30013650  
30013660  
30013670  
30013680  
30013690  
30013700  
30013710  
30013720  
30013730  
30013740  
30013750  
30013760  
30013770  
30013780  
30013790  
30013800  
30013810  
30013820  
30013830  
30013840  
30013850  
30013860  
30013870  
30013880  
30013890  
30013900  
30013910  
30013920  
30013930  
30013940  
30013950  
30013960  
30013970  
30013980  
30013990  
30014000  
30014010  
30014020  
30014030  
30014040  
30014050  
30014060  
30014070  
30014080  
30014090  
30014100  
30014110  
30014120  
30014130  
30014140  
30014150  
30014160  
30014170  
30014180  
30014190  
30014200  
30014210  
30014220  
30014230  
30014240  
30014250  
30014260  
30014270  
30014280  
30014290

```
0983 1 6600 05EA LDX L2 DST
0985 0 C0C3 LD ITDSW CALL ADDR
0986 0 823A A 2 K0001-DST ADD ONE
0987 0 D008 STO ITDS8+1 SET DSW S/B POINTER
*
0988 0 1010 SLA 16
0989 0 D018 STO LNCNT CLEAR LINE COUNT
*
ITDS7 LDX I3 LNCNT
098A 1 6780 09D2 LD L3 INTEX+1 WAS DSW
098C 1 C700 062A AND 2 K0018-DST SET CHAN BITS
098E 0 E244 ITDS8 OR L3 *-- BUILD S/B DSW
098F 0 EF00 0000 RTE 16 PLACE IN Q
09C1 0 18D0 LD L3 INTEX+1 WAS DSW
09C2 1 C700 062A BSI ERRIT GO TO LOG MSG
09C4 0 407E
*
09C5 0 6700 8003 LDX L3 /8003 SET MULTI-LINE SW
09C7 0 6855 STX 3 EMESS+2
09C8 1 7401 09D2 MDX L LNCNT+1 SET TO NEXT LINE
09CA 0 C0CC LD ITDS4+1 NO. OF INTERRUPTS
09CB 0 F006 EOR LNCNT CHK IF FINISHED
09CC 0 4820 BSC 2 SKIP IF FINISHED
09CD 0 70EC MDX ITDS7 RR IF NOT
09CE 0 6303 LDX 3 3 RESET MULTI LINE SW
09CF 0 684D STX 3 EMESS+2
*
J9D0 0 6400 0000 ITDS9 LDX L *-- RETURN TO USER
*
LNCNT DC *-- LINE COUNTER
09D2 0 0000 THREE DC 3 CONSTANT
09D3 0 0003 ESAVE DC 0 ERROR SAVE
09D4 0 0000 RSAVE DC 0 TEMP ACL SAVE
09D5 0 0000 K8FE7 DC /8FE7 ERROR BIT MASK
09D6 0 8FE7
*
09D7 0 0000 SBANA DC /0000
09D8 0 690F STX 1 SBANA+1
09D9 1 7500 05EA MDX L1 DST FETCH ANALYSIS MSG
09DB 0 D237 STO 2 AQSASV-DST
09DC 0 18D0 RTE 16
09DD 0 C237 LD 2 AQSASV-DST
09DE 0 F100 EOR 1 0 CHECK AGAINST EXPECTED
09DF 0 E101 AND 1 1 REMOVE UNNEC BITS
09E0 1 4C98 09D7 BSC 1 SBANA,+ EXIT IF DSW OK
```

```
30014300
30014310
30014320
30014330
30014340
30014350
30014360
30014370
30014380
30014390
30014400
30014410
30014420
30014430
30014440
30014450
30014460
30014470
30014480
30014490
30014500
30014510
30014520
30014530
30014540
30014550
30014560
30014570
30014580
30014590
30014600
30014610
30014620
30014630
30014640
30014650
30014660
30014670
30014680
30014690
30014700
30014710
30014720
30014730
30014740
30014750
30014760
30014770
30014780
30014790
30014800
30014810
30014820
30014830
30014840
30014850
30014860
30014870
30014880
30014890
30014900
30014910
30014920
30014930
30014940
30014950
30014960
30014970
```

1130-1403 FUNCTION TEST

```

09E2 0 C101          LD      1 1      SET DSW S/B      30D14980
09E3 0 F2FE          EOR     2 TERM-DST  30D14990
09E4 0 E237          AND    2 AQSAB-DST  30D15000
09E5 0 E900          OR     1 0          30D15010
09E6 0 18D0          RTE     13         30D15020
09E7 0 6500 0000    SBAN1 LDX L1 *-+   RESTORE XR 1  30D15030
09E9 0 4009          BSI    ERRIT      PRINT THE ERROR 30D15040
                                30D15050
09EA 1 4C80 09D7    *      BSC I SBANA  EXIT      30D15060
                                30D15070
                                ***** 30D15080
                                *      30D15090
                                ***** 30D15100
                                *      PRINT POSITION SUBROUTINE 30D15110
                                *      30D15120
                                *      THIS SUBROUTINE COUNTS THE 30D15130
                                *      PRINT MAGNET POSITIONS AS THE 30D15140
                                *      NO PARITY ROUTINE RIPPLE 30D15150
                                *      PRINTS FROM LEFT TO RIGHT 30D15160
                                *      DURING A PRINT CYCLE FOR EACH 30D15170
                                *      CHARACTER PRINTED.149 SCANS PER 30D15180
                                *      PRINT CYCLE( 30D15190
                                *      30D15200
                                *      COLUM DC 30D15210
                                *      MDX L COLNO,1 UPDATE COL NUMBER 30D15220
                                *      LDX 1 /23 SET TO PRINT 30D15230
                                *      STX 1 EMESS+2 30D15240
                                *      BSC I COLUM RETURN 30D15250
                                *      30D15260
                                *      30D15270
                                *      ***** 30D15280
                                *      LOG ERROR SUBROUTINE 30D15290
                                *      30D15300
                                *      30D15310
                                *      30D15320
                                *      ERRIT DC 30D15330
                                *      STD EMESS+5 SAVE MESSAGE 30D15340
                                *      STX 1 EMESS BUILD MESSAGE ID 30D15350
                                *      LD EMESS 30D15360
                                *      SRA 1 30D15370
                                *      STO EMESS 30D15380
                                *      SLA 16 30D15390
                                *      STO EMESS+3 CLEAR ALPHA 30D15400
                                *      LDX L2 AWASB 30D15410
                                *      LD EMESS+2 30D15420
                                *      BSC - SKIP IF 2ND LINE 30D15430
                                *      STX 2 EMESS+3 ALPHA HDNG ADDR 30D15440
                                *      30D15450
                                *      ***** PRINT ERROR MESSAGE 30D15460
                                *      BSI I ERROR * 30D15470
                                *      DC EMESS MSG * 30D15480
                                *      DC * LOOP * 30D15490
                                *      ***** 30D15500
                                *      LDX 2 3 RESTORE WORD COUNT 30D15510
                                *      STX 2 EMESS+2 30D15520
                                *      LDX L2 DST 30D15530
                                *      BSI CKERR CHECK ERROR 30D15540
                                *      LDX I ERRIT RETURN TO USER 30D15550
                                *      ***** 30D15560
                                *      30D15570
                                *      * 30D15580
                                *      * CHECK SPURIOUS INTERRUPTS 30D15590
                                *      * SUBROUTINE 30D15600
                                *      * 30D15610
                                *      * 30D15620
                                *      * CKERR DC CHECK SPURIOUS ER INTRPS 30D15630
                                *      * LDX 3 32 30D15640
                                *      * LDD 2 DSWC-DST 30D15650
                                *      * SLC 3 0

```

1130-1403 FUNCTION TEST

```

0A0F 1 4C90 0A0B    BSC I CKERR,- BR IF NO SPUR INTRPT 30D15660
0A11 0 6118          LDX 1 DSWC-DST ERROR - C 30D15670
0A12 0 CA18          LDD 2 DSWC-DST 30D15680
0A13 0 1890          SRT 16 30D15690
0A14 0 D218          STO 2 DSWC-DST 30D15700
0A15 0 D219          STO 2 DSWC+1-DST 30D15710
0A16 0 1090          SLT 16 30D15720
0A17 0 70DC          MDX ERRIT+1 PRINT THE ERROR 30D15730
0A18 1 6480 0A0B    LDX I CKERR RETURN TO USER 30D15740
                                30D15750
                                *      30D15760
                                *      BSS E 30D15770
                                *      KFFE DC /FFEF CONSTANT 30D15780
                                *      * 30D15790
                                *      EMESS DC /0000 MESSAGE ID NO. 30D15800
                                *      DC /0020 HEX - DEC 30D15810
                                *      DC /0003 WORD COUNT 30D15820
                                *      DC ALPHA OUT ADRS 30D15830
                                *      DC ALPHA OUT ADRS 30D15840
                                *      DC /0000 DSW 30D15850
                                *      DC /0000 DSW SHOULD BE 30D15860
                                *      DC 30D15870
                                *      DC 30D15880
                                *      DC PRINT COLUMN (RTN 3,5) 30D15890
                                *      COLNO DC WAS S/B 30D15900
                                *      AWASB DC /903C S 30D15910
                                *      DC /9821 S 30D15920
                                *      DC /2198 S 30D15930
                                *      DC /BC18 /B 30D15940
                                *      DC /FFFF 30D15950
                                *      ***** 30D15960
                                *      * 30D15970
                                *      * 30D15980
                                *      * CHECK SUBROUTINE END 30D15990
                                *      * 30D16000
                                *      * 30D16010
                                *      * CKEND DC 30D16020
                                *      * HALT1 LDX L2 DST 30D16030
                                *      * BSI CKERR 30D16040
                                *      * LD 2 SWSAB-DST 30D16050
                                *      * EOR 2 SW1-DST 30D16060
                                *      * BSC L PCON,Z BR IF END ROUTINE 30D16070
                                *      * 30D16080
                                *      * MDX L CYCNT,-1 DECR CYCLE COUNTER 30D16090
                                *      * MDX HALT2 30D16100
                                *      * BSC L PCON END THE ROUTINE 30D16110
                                *      * 30D16120
                                *      * HALT2 BSC I CKEND RETURN 30D16130
                                *      * 30D16140
                                *      * 30D16150
                                *      * 30D16160
                                *      * ***** 30D16170
                                *      * 30D16180
                                *      * 30D16190
                                *      * 30D16200
                                *      * BSS E 1 30D16210
                                *      * BLANK DC /FFFF BLANK 30D16220
                                *      * 30D16230
                                *      * BITL DC /252C BIT LINE 30D16240
                                *      * DC /0E7F 30D16250
                                *      * DC /1A2C 30D16260
                                *      * DC /1C68 30D16270
                                *      * DC /7F7F 30D16280
                                *      * DC /200E 30D16290
                                *      * DC /1C7F 30D16300
                                *      * DC /7F43 30D16310
                                *      * 30D16320
                                *      * BLINE DC /4040 1 1 30D16330
                                *      * DC /0101 2 2

```



OAC2 0 2950	DC	/2950	
OAC3 0 2068	DC	/2068	
OAC4 0 7F5E	DC	/7F5E	
OAC5 0 202C	DC	/202C	
OAC6 0 1C0E	DC	/1C0E	
OAC7 0 7F7F	DC	/7F7F	
OAC8 0 200E	DC	/200E	
OAC9 0 1C7F	DC	/1C7F	
OACA 0 7F07	DC	/7F07	
*			
OACB 0 266B	CHAN DC	/266B	CHANNEL
OACC 0 641C	DC	/641C	
OACD 0 1C68	DC	/1C68	
OACE 0 1A7F	DC	/1A7F	
*			
OACF 0 6868	E DC	/6868	E E
OADO 0 4C7F	SLASH DC	/4C7F	/ SP
*			
OAD1 0 5E20	OPENT DC	/5E20	PRINT OPTIONAL DATA
OAD2 0 2C1C	DC	/2C1C	
OAD3 0 0E7F	DC	/0E7F	
OAD4 0 5D5E	DC	/5D5E	
OAD5 0 0E2C	DC	/0E2C	
OAD6 0 5D1C	DC	/5D1C	
OAD7 0 641A	DC	/641A	
OAD8 0 7F67	DC	/7F67	
OAD9 0 640E	DC	/640E	
OADA 0 647F	DC	/647F	
OADB 0 7F7F	DC	/7F7F	
OADC 0 200E	DC	/200E	
OADD 0 1C7F	DC	/1C7F	
OADE 0 7F25	DC	/7F25	
*			
OAE0 0000	BSS E	0	
OAE0 0 684C	SLSH DC	/684C	E/
OAE1 0 4C68	DC	/4C68	/E
*			
OAE2 0 7F40	CHAND DC	/7F40	1
OAE3 0 7F01	DC	/7F01	2
OAE4 0 7F02	DC	/7F02	3
OAE5 0 7F43	DC	/7F43	4
OAE6 0 7F04	DC	/7F04	5
OAE7 0 7F45	DC	/7F45	6
OAE8 0 7F46	DC	/7F46	7
OAE9 0 7F07	DC	/7F07	8
OAEA 0 7F08	DC	/7F08	9
OAEB 0 7F64	DC	/7F64	A
OAEF 0 7F25	DC	/7F25	B
OAEED 0 7F26	DC	/7F26	C
OAAE 0050	AREA BSS E	80	OUTPUT BUFFER AREA
OB3E 0631	END	PRBGN	

NO ERRORS IN ABOVE ASSEMBLY

30D17700  
30D17710  
30D17720  
30D17730  
30D17740  
30D17750  
30D17760  
30D17770  
30D17780  
30D17790  
30D17800  
30D17810  
30D17820  
30D17830  
30D17840  
30D17850  
30D17860  
30D17870  
30D17880  
30D17890  
30D17900  
30D17910  
30D17920  
30D17930  
30D17940  
30D17950  
30D17960  
30D17970  
30D17980  
30D17990  
30D18000  
30D18010  
30D18020  
30D18030  
30D18040  
30D18050  
30D18060  
30D18070  
30D18080  
30D18090  
30D18100  
30D18110  
30D18120  
30D18130  
30D18140  
30D18150  
30D18160  
30D18170  
30D18180  
30D18190  
30D18200  
30D18210

CROSS REFERENCE

NAME	VALUE	REFERENCES
AQSAV	0621	09DB,09DD,09E4
AREA	0AEE	0616,0780,0783,07B3,07B6,07ED,07FE,0804,0849,084F,0863,086B,0874,0876,0880,0883,0898,089C,08E5,08E7,08EA,08ED,08F6,0904,092D
AWASB	0A26	09FB
BEGIN	0160	0631
BITL	0A3C	06FE
BITSW	061C	08A5,08BD
BLANK	0A3B	0699,06D6,0734,0777,07AA,0800,0806,084B,0851,0868,0888,0902
BLINE	0A44	0705
BLNCR	06D5	06CD,06F2
BLNCT	0733	072B,0750
CHADR	089C	
CHAIN	0A7E	0763
CHAN	0ACB	07F4,082F
CHAND	0AE2	07D6,0823,088D
CHANQ	087F	0812,0832,085D,089A
CHAN1	0887	07D2,0821,088F
CHAN2	088D	07D8,0825,0891
CHARC	0A70	075C
CKEND	0A2B	06AB,06BD,06F0,071B,074E,079A,07C9,0814,085E,08DE,0A38
CKERR	0A08	0A08,0A0F,0A18,0A2E
CLEAR	096E	0932,095D,0977
CLEA1	0973	0976
COLNO	0A25	06CE,072C,09ED
COLUM	09EC	06E1,073F,09F1
CYCNT	0604	069C,06AF,06CB,0708,0729,0766,0772,07A5,07DE,082A,08DC,0A33
DATA	08E1	08AC,08B5,08B6,08C5
DST	05EA	05EA,0669,066B,066C,066F,0674,0675,067A,069E,06A1,06A2,06A3,06B1,06B4,06B5,06B6,06CA,06CB,06DD,06ED,06EC,070A,070D,070E,070F,0710,0728,0729,073B,073E,074A,0766,0787,078A,078B,078C,0793,07BA,07BD,07BE,07BF,07C0,07D4,07D5,07E0,07E3,07E4,07E5,07EC,080E,080F,081E,081F,0833,0836,0837,0838,0839,0859,085A,0893,0896,0897,08A5,08A6,08A7,08B2,08B7,08BD,08BE,08BF,08C7,08CA,08CB,08CF,08D0,08D1,08D2,0929,092F,0934,0935,0936,093A,093B,093D,0946,0949,0952,0955,095E,0964,0966,097A,097B,097F,0983,0988,0998,09A5,09A7,0983,0986,098E,09D9,09DB,09DD,09E3,09E4,0A06,0A0D,0A11,0A12,0A14,0A15,0A2C,0A2F,0A30
DSWA	05FE	
DSWB	0600	
DSWC	0602	0645,0A0D,0A11,0A12,0A14,0A15
DSW1	05EC	069E,06DD,070A,0738,0787,07BA,07E0,0833,08C7,0952,0964,09A7
DSW2	05EE	
DSW3	05F0	06B1
DSW4	05F2	0710,07C0,0839,08D2
DSW5	05F4	06A3,078C,07E5
DSW6	05F6	06B6
DSW7	05F8	080C,080F,0857,085A
DSW8	05FA	
DSW9	05FC	
D54	0769	0765
D60	0627	0660,06CA,0728,0900
D60C	0665	065F
D66C	0666	0667
E	0ACF	077C,07AF,0870
ELIN	0861	07F6,0831,0841,087D
ELIN1	086F	07DB,0828,0878,087C
ELIN2	087D	087A
EMESS	0A1E	064A,06E9,0747,09C7,09CF,09F0,09F4,09F5,09F6,09F8,09FA,09FD,09FF,0A02,0A05
END	0164	067B
EPARE	0757	0753
EPARR	06F9	06F5
ERIND	05EB	
ERLCK	0166	
ERRIT	09F3	09C4,09E9,0A09,0A17
ERROR	0162	0A00

1130-1403 FUNCTION TEST

ESAVE 09D4 0999,099D,099E,09AA  
 HALT1 0A2C  
 HALT2 0A38 0A35  
 HDNG 094F 06C8,0700,0726,075E,076F,07A2,081C,08A3,096B  
 HDNGA 096D 0951,095A  
 HDNG1 0952 095C  
 HH 08E2 08A9  
 H2D 089D  
 ILO 017A  
 IL1 018A  
 IL2 019A  
 IL3 01AA  
 IL4 01BA 064E  
 INTEX 0629 0635,0639,063D,063F,0652,0653,0654,080E,0859,0970,0973,0988,0998  
 098C,09C2  
 INTRP 0634 0646,064C  
 INTR2 0643 063A  
 INTR3 0646 0642  
 ITDSW 0979 06A6,06B9,06E3,0713,0741,078F,0794,07C3,07E8,07F7,083C,0842,08D5  
 0956,095F,0967,097C,097E,0981,0985  
 ITDS1 0981 0993  
 ITDS3 0991 098B  
 ITDS4 0996 0986,0989,09CA  
 ITDS5 099A 09A3  
 ITDS6 09AF 0995,09A8  
 ITDS7 09BA 09CD  
 ITDS8 09BF 09B7  
 ITDS9 09D0 0980  
 KFFEF 0A1A  
 K00BF 062D 0658  
 K0001 0624 0934,09B6  
 K0010 0628 0643  
 K0018 062E 09BE  
 K0040 062F  
 K0400 0625 0949,097A  
 K0800 0626 07D4,081E,0896  
 K3F3F 08E3 08AA  
 K4000 0630 08B3  
 K8FE7 09D6 09AB  
 LNCNT 09D2 09B9,09BA,09C8,09C8  
 LOG 0163  
 LOGBY 0167  
 LOPGO 098D 094D  
 LRTN 0695 068A  
 MLSCF 05E5 0683,098D  
 NOPAR 0A48 06C6  
 NRTN 0692 0689  
 NUM3 06FA 06C0  
 NUM5 0758 071E  
 OPENT 0AD1 08A1  
 PACNR 06DB 06D2,06EB,06ED,06F4  
 PACNT 0739 0730,0749,0748,0752  
 PAR 0A4D 06C1,071F,0724  
 PARE 0A62 072E,0757  
 PARN 0A54 06D0,06F9  
 PCON 0669 0664,0A31,0A36  
 PCON2 0677 066D  
 PCON3 067D 0671  
 PID 05DC 0633  
 PRBGN 0631 083E  
 PRT1T 0616 06A1,06E0,070E,073E,078A,07BE,07E3,0837,08CA,095E  
 PTRDY 092E 069F,06B2,06DE,0708,073C,0788,07BB,07E1,0834,08C8,093F,0953,0965  
 PTRY1 0935 094A  
 PTRY2 0936 094B  
 PTRY3 0944 0931,0938  
 PTRY9 0948 0943  
 RAD 05DE 0681  
 RID 05D9 0650,066F,0675,0677,067A,067D,09AF,09B1

1130-1403 FUNCTION TEST

RIDCK 068A 0670  
 RLCF 0168  
 ROTA 08E4 0719,0785,07B8,08F2  
 ROTA1 08FA 08F1  
 RQKB 018C  
 RQTY 018B  
 RSAVE 09D5 099C,099F  
 RTNA 0817 0694  
 RTNA1 0833 0860  
 RTNB 089E 0695  
 RTNB1 08A5 08C0  
 RTNB2 088D 08E0  
 RTNB3 08C7  
 RTNB4 08D0 08CD  
 RTNB5 08C4 0662  
 RTNOM 0689 0679  
 RTNSW 0165 0685  
 RTN1 0696 0688  
 RTN12 069E 06AD  
 RTN2 06AE 068C  
 RTN21 06B1 06BF  
 RTN3 06C0 068D  
 RTN31 06D0 06F6  
 RTN32 06D3 06EF,06F6  
 RTN4 06FB 068E  
 RTN41 07GA 071D,0767  
 RTN5 071E 068F  
 RTN51 072E 0756  
 RTN52 0731 074D,0754  
 RTN6 0759 0690  
 RTN7 076A 0691  
 RTN72 0785 079C  
 RTN8 079D 0692  
 RTN82 0788 07CB  
 RTN9 07CC 0693  
 RTN91 07E0 0816  
 RTN94 07F6 07F0  
 RTTBL 0688 067F,0689,068A  
 RTO 0648 05E3,05E4,05E5  
 RTD1 0660 0668  
 RTD2 0667 065E  
 SAVE 061F 08A7,08BF  
 SBANA 09D7 06A4,06B7,0711,078D,07C1,07E6,0910,083A,085B,08D3,0947,09E0,09EA  
 SBAN1 09E7 09DB  
 SEARA 092D 0911  
 SEBC 0904 0907  
 SEBIT 08F4 0696,0603,06D3,06D8,06FB,0702,0721,0731,0736,0759,0760,076A,0774  
 0779,079D,07A7,07AC,07CC,07F1,0817,082C,0865,086D,0885,088A,089E  
 08C2,08F8  
 SEB1 0900 08FB  
 SEB2 090B 08FF  
 SEB3 0919 08FD,090A  
 SEB4 0918 0918,0923  
 SEB5 091D 0915,0926  
 SEB6 091F 0912  
 SEB7 0925 0922  
 SEB8 0927 08F5,090C  
 SENSN 0618 06A2,06B5,070F,0788,07BF,07E4,0836,08D1,093A,0946  
 SENSR 061A 0637  
 SKAP 0A98 07CF  
 SKAP 0A98 07CF  
 SKBP 0AA4 081A  
 SKIP 0614 07EC,0836  
 SKPWD 0620 0614,07D5,081F,0893,0897  
 SLA 08AF 08B0,08B8,08BC  
 SLASH 0AD0 0872  
 SLSH 0AEO 077E,07B1  
 SPACE 0612 0684,070D,0793,078D,08CF,08D0,0955,0966

1130-1403 FUNCTION TEST

SPAP 0AB1 076D  
SPBP 0ABE 07A0  
START 0161 0687,098F  
SVKB 01BD  
SWSAV 0623 066C,0A2F  
SWTCH 061E 061C,08A6,08BE,08C8  
SWO 05DF 0655,0659  
SW1 05E0 066B,0674,0A30  
SW2 05E1 065B  
SW3 05E2  
TEMP 0606 08B2,08B7  
TERM 05E8 06EC,074A,097F,09E3  
THREE 09D3 0987  
WACNT 0605 0935,0941,097B,0991  
AQSAV UNDEF 09E4

END OF ASSEMBLY

----- LAST PAGE -----

DATE 15JUN67 20SEP67  
EC NO. 420317 420317C

PROG ID 0300-0  
PAGE 16