PCL11-A,B

PCL11 EXERCISER V02C

CZPLAC0

AH-E260C-MC

FICHE 1 OF 1

SEP 1979

digital

MADE IN USA

I D E N T I F I C A T I O N

PROGRAM CODE: AC-E259C-MC

PROGRAM NAME: CZPLACO PCL11 EXERCISER V02C

DATE CREATED: 21-SEP-76

UPDATED (V02): 13-MAR-78

MODIFIED (V02A): 11-SEP-78 BY D.WIENS
(V02C) 07-JUN-79 BY D.WIENS

MAINTAINER: SPECIAL SYSTEMS, KANATA

AUTHOR: DAVID G. WIENS

PARALLEL COMMUNICATIONS LINK (PCL11) EXERCISER

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTD)

'-1

PARALLEL COMMUNICATIONS LINK EXERCISER PROGRAM

' GENERAL

'.1 GENERAL DESCRIPTION

THE PDP-11 PARALLEL COMMUNICATIONS LINK (PCL11) EXERCISER IS WRITTEN TO EXERCISE A FULL SET OR PARTIAL SET OF PCL11 UNITS, EACH ON ITS OWN PDP-11. THE MAXIMUM NUMBER OF UNITS (FULL SET) WHICH MAY BE EXERCISED IS 31, ALTHOUGH THE NORMAL MAXIMUM CONFIGURATION INCLUDES ONLY 16 PCL11'S. THE MINIMUM NUMBER OF UNITS IS 1.

THE EXERCISER IS OPERATED BY MEANS OF THE OPERATOR ENTERING COMMANDS AT EACH PDP-11 CONSOLE TERMINAL WHICH WILL DESCRIBE TO THE EXERCISER THE NUMBER OF TARGET RECEIVERS TO BE COMMUNICATED WITH AND THEIR T.D.M. BUS ADDRESSES. STATUS AND ERROR REPORTS ARE ACHIEVED ALSO IN RESPONSE TO OPERATOR COMMANDS.

EACH PCL11 TRANSMITTER IN THE 'CHAIN' OF PDP-11'S MUST BE TOLD HOW MANY OF THE RECEIVERS IN THE 'CHAIN' IT SHOULD SEND DATA TO. THEN, UPON THE OPERATOR'S COMMAND, ALL PCL11 TRANSMITTERS INVOLVED WILL BEGIN SENDING RANDOM DATA PATTERNS TO EACH RECEIVER THEY HAVE BEEN TOLD TO COMMUNICATE WITH.

RECEIVERS IN THE CHAIN ARE ALWAYS RECEPTIVE TO DATA FROM ANY ONE TRANSMITTER AT A TIME AND WILL CHECK THAT THE DATA RECEIVED IS CORRECT. THE RECEIVER HANDLER PORTION OF THE EXERCISER WILL ALSO GENERATE A TABLE OF ERRORS THAT MAY BE EXAMINED UPON ISSUANCE OF THE 'ERRORS' COMMAND.

TRANSMITTERS IN THE CHAIN ARE ACTIVATED OR DE-ACTIVATED BY THE OPERATOR AND ARE DE-ACTIVATED BY A 'MASTER-DOWN'' ERROR OR A UNIBUS TIMEOUT (TRAP TO 4). THE TRANSMITTER HANDLER PORTION OF THE EXERCISER WILL LOOK AFTER THE TRANSMISSION OF RANDOM DATA TO EACH RECEIVER ON ITS 'LIST' AND GENERATE A STATUS TABLE AND AN ERROR TABLE. THE STATUS TABLE WILL INFORM THE OPERATOR OF THE SUCCESS OR FAILURE OF THAT TRANSMITTER TO COMUNICATE WITH EACH OF THE RECEIVERS ON ITS LIST. THE ERROR TABLE WILL SHOW THE OPERATOR ANY HARDWARE ERRORS ENCOUNTERED DURING COMMUNICATION.

A SPECIAL ''SUMMARY'' COMMAND WILL GIVE A CONDENSED ERROR TABLE INDICATING ONLY THE ERROR NUMBER, THE ADDRESS IN THE LISTING OF THE ERROR, AND THE TOTAL NUMBER OF OCCURRENCES OF THAT ERROR WITHOUT REGARD TO WHICH RECEIVER AND TRANSMITTER WERE CONNECTED.

1.2 REVISION HISTORY

1.2.1 VERSION V02

1.2.1.1 STARTING AND RESTARTING (SEE SEC. 4.4)
V-02 OF THE EXERCISER STARTS AT LOCATION 200. WHEN RESTARTED AT LOCATION 204, THE STATUS AND ERRORS ARE PRESERVED.

1.2.1.2 UPPER/LOWER CASE INPUT (SEE SEC.5.4)
EITHER UPPER OR LOWER CASE ALPHA CHARACTERS ARE ACCEPTED BY V- 2 OF THE EXERCISER. WHICHEVER IS TYPED IN, THE UPPER CASE VERSION IS ECHOED.

1.2.1.3 ERRORS COMMAND (SEE SEC.3.2.12)
THE "ERRORS" COMMAND HAS BEEN ADDED TO THE EXERCISER TO GIVE MORE DETAIL ABOUT THE ERROR CONDITIONS.

1.2.1.4 RIB COMMAND (SEE SEC. 3.2.4)
THE "RIB" COMMAND HAS BEEN ADDED TO THE EXERCISER TO ALLOW RUNNING IN A RE-TRY - IF - BUSY MODE, WHICH BRINGS THE ATTEMPTS AND SUCCESSES COUNTS MORE IN LINE.

1.2.1.5 ASSIGN COMMAND (SEE SEC. 3.2.13)
THE "ASSIGN" COMMAND WAS ADDED TO ALLOW CONSOLE CHANGING OF THE PCL11 UNIBUS ADDRESSES AND VECTORS WITHIN THE EXERCISER.

1.2.1.6 GO COMMAND (SEE SEC.3.2.14)
THE "GO" COMMAND HAS BEEN ALTERED SO THAT THE RECEIVER ADDRESS QUEUE, STATUS OF ATTEMPTS AND SUCCESSES, AND THE ERROR TABLE ARE ALL CLEARED.

1.2.1.7 ADDRESS QUEUE EVENT (SEE SEC. 6.4)
THIS EVENT WAS CHANGED SO THAT IF THE EXERCISER WAS TOLD TO "GO" OR "CONTINUE", AND NOTHING WAS LOADED INTO THE "RECEIVER ADDRESS QUEUE", THE EXERCISER WOULD NOT GO.

1.2.1.8 LOADING OF THE SILO (SEE SEC. 3.2.1)
IN THE PREVIOUS EXERCISER, THE SILO COULD BE LOADED IN SUCH A WAY AS TO CAUSE "HARD TO TRACE" HARDWARE ERROR INDICATIONS. NOW A "PAD" VALUE HAS BEEN INSERTED BETWEEN SIMILAR ENTRIES SO THAT THIS CANNOT HAPPEN.

1.2.1.9 NEW CONTROL CHARACTERS (SEE SEC. 3.1)
V-02 OF THE EXERCISER HAS THROWN OUT THE "ESCAPE/ALTMODE" CHARACTER IN FAVOUR OF "CNTRL-C". ALSO, CNTRL-S AND CNTRL-Q WERE ADDED TO CONTROL THE PRINTOUT ON VIDEO TERMINALS.

1.2.1.10 CARRIAGE RETURN (SEE SEC. 3.1.6)
THIS VERSION OF THE EXERCISER WILL NOT GO INTO "LIMBO" IF A CARRIAGE RETURN IS ENTERED BY ITSELF. IT SIMPLY ECHOES ANOTHER $.

1.2.1.11 RUBOUT (SEE SEC. 3.1.8)
THE EXERCISER WILL NOW INDICATE WHEN ALL HAS BEEN RUBBED OUT BY RETURNING A <CR-LF> AND "$" WHEN THE COMMAND INPUT BUFFER IS EMPTY.

1.2.1.12 STATUS TABLE (SEE SEC. 2.1)
THE "ATTEMPTS" AND "SUCCESSES" COUNTS IN THE STATUS TABLE HAVE BEEN BEEFED UP TO DOUBLE PRECISION DECIMAL NUMBERS. ALSO, AN INDICATION HAS BEEN ADDED TO THE TABLE OF WHETHER THE UNIT IS MASTER OR SECONDARY, AND WHETHER "RIB" IS SET OR CLEAR. ALSO, THE ELAPSED TIME OF THE RUN SO-FAR IS PRINTED PROVIDING THAT THE PDP-11 HAS A LINE CLOCK.

1.2.1.13 SUMMARY TABLE (SEE SEC. 2.2)
THE "NO. OF OCCURRENCES" COUNT IN THE SUMMARY TABLE HAS BEEN GIVEN A CEILING OF 65,528 (DECIMAL) AND WILL PRINT "MXCNT" AFTER THAT. THE COUNT IS PRINTED IN DECIMAL.

1.2.2 VERSION V02A

THIS REVISION WAS BROUGHT ABOUT BY THE CHANGE OF THE PROMPT PRINTED WHILE IN COMMAND MODE. THE PROMPT WAS SIMPLY "$". IT HAS BEEN CHANGED TO "PCL>" TO REDUCE CONFUSION ON SYSTEMS WHERE PDP-11'S ARE ATTACHED TO VAX/1170'S. (IT IS ONLY THE DIAGNOSTIC SUPERVISOR ON THE "VAX" THAT IS ALLOWED TO USE THE PROMPT "$".

1.2.3 VERSION V02C

1.2.3.1 ALIGN THE VERSION LETTER WITH THE ONE IN THE PRODUCT CODE, MAKING THIS V02C RATHER THAN V02B.

1.2.3.2 CHANGE THE HANDLING OF RECEIVER "BYTE COUNT OVERFLOW" INTERRUPT SO THAT A SUCCESSFUL TRANSFER INTERRUPT ISN'T LOST. THIS PREVENTS THE EXERCISER FROM HANGING UP A RECEIVER.

1.2.3.3 MODIFY THE COUNT IN THE DATA GENERATION TO ACCOUNT FOR THE FIRST WORD IN THE RECEIVER NOT BEING "NPR'D". THE COUNT (MAX WITHOUT EXPEXTING A TRUNCATE) IS NOW #602 OCTAL.

1.2.3.4 MODIFY THA MESSAGE PRINTED WHEN THERE ARE NO ERRORS TO REMOVE THE WORD "YET". MESSAGE NOW READS:
"NO ERRORS TO REPORT"

1.2.3.5 MODIFY THE MESSAGE PRINTED IF THE EXERCISER IS STARTED BEFORE BEING GIVEN ANY RECEIVER ADDRESSES. MESSAGE NOW READS:
"NO RECEIVERS SELECTED"
ALSO, DO A <CR>,<LF> BEFORE IT SO THAT THE "GO" DOESN'T GET OVERWRITTEN.

1.2.3.6 REMOVE THE CODE WHICH USED TO COMPUTE THE RESTART ADDRESS BASED ON WHETHER THE DEVICE ADDRESSES WERE VALID. RESTART ADDRESS IS NOW ALWAYS "204".

EXERCISER TABLES

THE PCL11 EXERCISER MAINTAINS THREE TABLES: A 'STATUS' TABLE AN 'ERRORS' TABLE, AND A 'SUMMARY' TABLE.

2.1 STATUS TABLE

THE STATUS TABLE IS DESIGNED TO SHOW THE OPERATOR THE NUMBER OF SUCCESSFUL TRANSMISSIONS TO EACH RECEIVER RELATIVE TO THE NUMBER OF ATTEMPTS TO TRANSFER DATA TO EACH RECEIVER. IF THE TRANSMITTER HAS BEEN TOLD TO 'RE-TRY - IF - BUSY'', THE NUMBER OF SUCCESSFUL TRANSFERS WILL BE VERY MUCH CLOSER TO THE NUMBER OF ATTEMPTS THAN WOULD BE THE CASE IF 'RE-TRY - IF - BUSY'' WERE NOT THE ORDER. HOWEVER, THE TOTAL NUMBER OF ATTEMPTS WOULD BE GREATER WITH 'RIB'' CLEAR.
THE INFORMATION GIVEN IN THE TABLE INCLUDES:

RUN STATE - STATE OF MASTER, SECONDARY, AND 'RIB''

RECEIVER ADDRESS - THAT THIS TRANSMITTER HAS BEEN INSTRUCTED TO COMMUNICATE WITH.

CONNECTION ATTEMPTS - NUMBER OF TIMES THIS XMTR TRIED CONNECTING TO THE RECEIVER OF THE ABOVE ADDRESS.

SUCCESSFUL CONNECTIONS- NUMBER OF TIMES THAT THE ABOVE ATTEMPTS WERE SUCCESSFUL.

```
RIB IS -SET- (OR CLEAR)
THIS UNIT IS -MASTER- (OR SECONDARY)
ELAPSED TIME (HRS,MIN,SEC,TIK)...0:0:4:35
RCVR ADDRESS  CONNECTION ATTEMPTS  SUCCESSFUL CONNECTIONS
     1                     X                    Y
     2                     X                    Y
     3                     X                    Y
     .                     .                    .
     .                     .                    .
    37                     X                    Y
```

THE ENTRIES UNDER 'RCVR ADDRESS'' INCLUDE ONLY THOSE ADDRESSES ENTERED BY THE OPERATOR USING THE 'RANGE'' OR ''ADD'' COMMANDS.

THE ENTRIES UNDER ''CONNECTION ATTEMPTS'' AND ''SUCCESSFUL CONNECTIONS'' ARE DOUBLE PRECISION DECIMAL NUMBERS. THEY ARE CAPABLE OF INCREASING TO A VALUE OF 655,359,999. IF THERE ARE A LARGER NUMBER OF ATTEMPTS OR SUCCESSES FOR A PARTICULAR RECEIVER THAN THAT AMOUNT, THE ENTRY IN THE TABLE WILL APPEAR AS : ''*********''.

SINCE ANY DATA RECEIVED BY A PCL11 RECEIVER MUST BE CHECKED, THE DATA PATTERN, WHICH IS RANDOM, MUST BE RECREATED BY THE RECEIVING EXERCISER BASED ON THE FIRST WORD RECEIVED. THIS MUST THEN BE COMPARED WORD-FOR-WORD WITH THE RECEIVED DATA. IT IS READILY APPARENT THAT THIS TAKES TIME AND WILL MAKE THAT PARTICULAR RECEIVER UNAVAILABLE DURING THAT TIME. ON A LARGE SYSTEM, WITH MANY PCL11'S IT MIGHT BE QUITE A WHILE BETWEEN SUCCESIVE SUCCESSFUL CONNECTIONS OF A GIVEN TRANSMITTER TO THE SAME RECEIVER, SINCE ALL TRANSMITTERS MAY VERY WELL BE TRYING FOR THE SAME RECEIVERS AT THE SAME TIME. IT IS FOR THIS REASON, THEN, THAT IT NEED NOT BE ALARMING THAT

THERE IS A DIFFERENCE BETWEEN THE NUMBER OF SUCCESSES AND THE NUMBER I 1
OF ATTEMPTS.

F4 0008

THE PRINTING OF THE STATUS TABLE MUST BE INITIATED BY THE OPERATOR ON EACH PDP-11 INVOLVED DURING THE COURSE OF AN EXERCISE RUN. THE STATUS TABLE OUTPUT IS NOT CAUSED BY ANY OTHER MEANS THAN THE OPERATOR ENTERING COMMAND MODE AND TYPING ''STATUS''. ALSO IT MUST BE NOTED THAT ALL TRANSMITTER ACTIVITY IS SUSPENDED ON THE UNIT WHILE THE STATUS IS BEING PRINTED OUT.

2.2 SUMMARY TABLE

THE SUMMARY TABLE IS DESIGNED TO SHOW THE OPERATOR A SUMMARY OF ERRORS WHICH HAVE OCCURRED IN EITHER THE RECEIVER OR TRANSMITTER HARDWARE ON EACH PDP-11.

INCLUDED IN THIS TABLE ARE:

ERROR NUMBER - FOR REFERENCE LATER IN THIS DOCUMENT TO INDICATE THE TYPE OF FAILURE.

ERROR ADDRESS - FOR REFERENCE IN THE PCL11 EXERCISER LISTING.

NO. OF OCCURRENCES - TO INDICATE HOW OFTEN THIS ERROR HAS OCCURRED DURING RUN TIME.

THE SUMMARY TABLE IS PRINTED IN THE FORM:

ERROR NUMBER	ERROR ADDRESS	NO. OF OCCURRENCES
1	XXX	YYYY
2	XXX	YYYY
3	XXX	YYYY
.	.	.
.	.	.
30	XXX	YYYY

THE ENTRIES UNDER ''ERROR NUMBER'' INCLUDE ONLY THOSE ERRORS WHICH HAVE A ''NO. OF OCCURRENCES'' GREATER THAN 0.

THE ENTRIES UNDER ''ERROR ADDRESS'' REPRESENT THE OCTAL MEMORY ADDRESS OF THE OCCURRENCE OF THE ERROR WHICH OCCURRED. THE OPERATOR MAY FIND THIS USEFUL FOR LOCATING, IN THE LISTING, THE PORTION OF THE PROGRAM WHERE THE ERROR OCCURRED.

THE ENTRIES UNDER ''NO. OF OCCURRENCES'' IS THE TOTAL (OR SUMMARY VALUE) OF ALL THE ERRORS OF THAT ERROR NUMBER WHICH OCCURRED AT THIS TERMINAL REGARDLESS OF WHAT TRANSMITTER WAS CONNECTED TO WHAT RECEIVER.

THE PRINTING OF THE SUMMARY TABLE MUST BE INITIATED BY THE OPERATOR ON EACH PDP-11 INVOLVED DURING THE COURSE OF AN EXERCISE RUN. THE SUMMARY TABLE OUTPUT IS NOT CAUSED BY ANY OTHER MEANS THAN THE OPERATOR ENTERING COMMAND MODE AND TYPING ''SUMMARY''. ALSO IT MUST BE NOTED THAT ALL TRANSMITTER ACTIVITY IS SUSPENDED ON THE UNIT WHILE THE SUMMARY TABLE IS BEING PRINTED OUT.

2.3 ERROR TABLES

THE ERROR TABLES ARE DESIGNED TO SHOW THE OPERATOR THE ERRORS WHICH OCCURRED DURING AN EXERCISE RUN IN ENOUGH DETAIL SO THAT HE CAN DETERMINE AS CLOSELY AS POSSIBLE WHICH TRANSMITTER AND/OR WHICH RECEIVER WERE CONNECTED AT THE TIME OF THE ERROR. THERE ARE 15 OCTAL TRANSMITTER TYPE ERRORS AND 13 OCTAL RECEIVER TYPE ERRORS. FOR EACH OF THE TRANSMITTER ERRORS, ANY ONE OF 31 POSSIBLE RECEIVERS MAY HAVE BEEN CONNECTED TO THAT TRANSMITTER AT THE TIME OF THE ERROR. THE ONE THAT ACTUALLY WAS CONNECTED IS LISTED IN THIS ERROR TABLE ALONGSIDE THE ERROR NUMBER AND THE COUNT OF THAT OCCURRENCE. FOR EACH OF THE RECEIVER-TYPE ERRORS, ANY ONE OF 31 TRANSMITTERS MAY HAVE BEEN TALKING TO THIS RECEIVER AT THE TIME OF THE ERROR. AGAIN, THE ACTUAL ONE IS LISTED IN THE RECEIVER ERRORS TABLE.

THE ENTRIES IN THESE TABLES THEN, INCLUDE:

ERROR NUMBER - FOR REFERENCE LATER IN THIS DOCUMENT TO INDICATE THE TYPE OF FAILURE.

CONNECTED XMTR (RCVR) - TO INDICATE THE ACTUAL ERRONEOUS LINK.

ERROR COUNT - TO SHOW THE NUMBER OF OCCURRENCES OF THIS ERROR WITH THIS CONNECTION.

THE TABLES ARE ALWAYS PRINTED TOGETHER AND ARE PRINTED IN THE FOLLOWING FORMAT:

TRANSMITTER ERRORS:

ERROR NO.	CONCTD RCVR	ERROR COUNT
1	1	63
1	2	7
1	6	48
15	37	2

RECEIVER ERRORS:

ERROR NO.	CONCTD XMTR	ERROR COUNT
16	1	22
16	2	22
27	37	12
30	6	2

IF THE COMMAND IS GIVEN TO DUMP THESE ERROR TABLES AND IT IS FOUND THAT NO ERRORS EXIST, THE EXERCISER RESPONDS IN THE FOLLOWING WAY:

** NO ERRORS TO REPORT **

IF THERE ARE, FOR EXAMPLE, TRANSMITTER ERRORS BUT NO RECEIVER ERRORS, THEN UNDER THE HEADING 'ERROR NO.' OF THE RECEIVER ERRORS WILL BE PRINTED: (NONE).

2.4 CLEARING OF THE TABLES

THE ''STATUS TABLE'' IS NORMALLY CREATED BY MEANS OF THE OPERATOR ENTERING, VIA THE 'RANGE, OR ADD'' COMMANDS, SOME RECEIVER ADDRESSES.
FOR EXAMPLE, IF THE ''ADD'' COMMAND WERE USED TO ENTER RECEIVER ADDRESSES 1, 2, 3, 4, &5, AND IMMEDIATELY THEREAFTER THE STATUS WAS REQUESTED, THE STATUS TABLE WOULD INDICATE ALL OF THE SELECTED RECEIVERS BUT THE ATTEMPTED AND SUCCESSFUL CONNECTION COUNTS FOR EACH OF THOSE RECEIVERS WOULD BE 0.

THE STATUS TABLE IS COMPLETELY CLEARED BY USING THE ''CLEAR'' COMMAND, OR BY USING THE 'DELETE'' COMMAND AND DELETING ALL THOSE ADDRESSES INDICATED IN THE STATUS TABLE. THE ''INITIALIZE'' COMMAND ALSO CLEARS THE STATUS TABLE COMPLETELY BY INTERNALLY CALLING THE ''CLEAR'' COMMAND.

THE ''SUMMARY TABLE'' IS NOT AFFECTED BY THE ''CLEAR'' COMMAND BUT IS ENTIRELY CLEARED BY THE ''INITIALIZE'' COMMAND.

LIKEWISE, THE 'ERRORS TABLE'' IS COMPLETELY CLEARED BY THE ''INITIALIZE'' COMMAND BUT IS NOT AFFECTED BY THE ''CLEAR'' COMMAND.

WHEN THE OPERATOR ''STARTS'' THE EXERCISER BY USING THE ''GO'' COMMAND, THE FOLLOWING RESULTS CAN BE EXPECTED ON THE TABLES:

THE ADDRESS ENTRIES OF THE ''STATUS'' TABLE ARE UNAFFECTED.

THE ATTEMPTS AND SUCCESSES ENTRIES OF THE STATUS TABLE ARE CLR'D

THE ENTIRE SUMMARY TABLE IS CLEARED.

THE ENTIRE ERRORS TABLE IS CLEARED.

CAUTION

IT IS IMPORTANT TO NOTE THAT THE RECEIVER ERRORS ARE CLEARED ALONG WITH THE TRANSMITTER ERRORS UPON THE ISSUANCE OF THE ''GO'' COMMAND. THEREFORE, THE OPERATOR MUST NOTE THAT THE NUMBER OF RECEIVER ERRORS INDICATED IN EITHER THE SUMMARY TABLE OR THE RECEIVER ERRORS TABLE IS ONLY THE NUMBER ACCUMULATED SINCE ''GO'' WAS TYPED.

3 EXERCISE COMMANDS

3.1 CONTROL CHARACTERS

3.1.1 CONTROL-C

THIS CHARACTER IS USED TO GET THE EXERCISER INTO ''COMMAND'' MODE SO THAT ANY OF THE CONTROLLING COMMANDS MAY BE ENTERED.

TYPING CONTROL-C (^C) ECHOS ''^C'' AND TERMINATES ANY OTHER FUNCTIONS WHICH MAY BE TAKING PLACE AT THE TIME. IF ANY OF THE TABLES ARE BEING PRINTED, THE CURRENT LINE WILL BE COMPLETED BEFORE ''^C'' IS ECHOED. THEN THE TABLE WILL BE TRUNCATED. THE EXERCISER WILL THEN ENTER ''COMMAND'' MODE AND INDICATE THIS BY PRINTING 'PCL>''.

'PCL>'' IS THE PROMPT WHICH INDICATES THAT A COMMAND MAY BE ENTERED BY THE OPERATOR.

TYPING ''CNTRL-C'' WHILE THE EXERCISER IS 'EXERCISING'' WILL CAUSE TERMINATION OF TRANSMITTER ACTIVITY ON THAT PDP-11, ECHO ''^C'', AND PRINT THE COMMAND MODE PROMPT 'PCL>''.

3.1.2 CONTROL-O

THIS CHARACTER IS USED TO TERMINATE THE CURRENT PRINTOUT. IT CAUSES THE ENTIRE CONTENTS OF THE OUTPUT QUEUE TO BE THROWN AWAY. WHEN TABLES ARE BEING PRINTED, ONLY A SINGLE LINE IS IN THE OUTPUT QUEUE AT ANY GIVEN TIME. THIS ALLOWS ''SKIPPING'' OF LINES DURING TABLE OUTPUT TO SAVE TIME.

WHEN CONTROL-O IS TYPED, THE EXERCISER WILL ECHO ''^O'', CLEAR THE OUTPUT QUEUE, AND THEN RETURN TO WHICHEVER MODF IT WAS IN WHEN ''CNTRL-O'' WAS TYPED IN. THEREFORE, IF ^O IS TYPED IN DURING EXERCISE TIME, THE ''^O'' WILL STILL BE ECHOED. IF ''CNTRL-O'' IS TYPED ARBITRARILY WHILE IN COMMAND MODE, THE ''^O'' WILL BE ECHOED BUT THERE WILL NOT BE ANOTHER 'PCL>'' GIVEN (YET THE EXERCISER WILL ACCEPT COMMANDS).

3.1.3 CONTROL-U

THIS CHARACTER IS USED (AS IT IS IN MOST D.E.C. SOFTWARE) TO THROW AWAY THE INPUT TYPED THUS FAR. THIS CONTROL CHARACTER SHOULD BE USED IF A MISTAKE IS NOTICED IN THE COMMAND STRING BEFORE CARRIAGE RETURN IS HIT. IF CARRIAGE RETURN IS HIT, THE ERRONECJS COMMAND WILL BE EXECUTED UP TO THE POINT OF THE ERROR, THEN A SYNTAX ERROR MESSAGE WILL BE PRINTED.

WHEN CONTROL-U IS TYPED, THF EXERCISER WILL ECHO ''^U'', CLEAR THE INPUT QUEUE, THEN ISSUE A NEW 'PCL>'' AND AWAIT A FURTHER COMMAND.

3.1.4 CONTROL-S

THIS CHARACTER IS USED TO SUSPEND PRINTER OUTPUT. NOTHING IS LOST WHEN IT IS USED, BUT THE PRINTOUT IS "HELD" UNTIL THE ISSUANCE OF CONTROL-Q OR SIMPLY TYPING ANY OTHER CHARACTER. THIS FEATURE IS USEFUL WHEN GETTING TABLE PRINTOUTS ON A VIDEO TERMINAL AND IT IS DESIREOUS TO STOP THE DISPLAY. IT IS ALSO USEFUL TO STOP THE PRINTOUT IN ORDER TO FIX THE PAPER OR ADD MORE PAPER ON ANY HARD COPY PRINTER.

NOTHING IS ECHOED UPON , OR AFTER, THE ISSUANCE OF A CONTROL-S.

3.1.5 CONTROL-Q

THIS CHARACTER IS USED TO RESUME PRINTOUT AFTER IT WAS STOPPED BY USE OF CONTROL-S. TYPING ANY OTHER CHARACTER WILL ALSO RESUME OUTPUT BUT THE CONTROL-Q CHARACTER IS NOT ENTERED INTO THE INPUT QUEUE, AND SIMPLY RESUMES PRINTOUT.

NOTHING IS ECHOED AS A RESULT OF TYPING CONTROL-Q EXCEPT WHATEVER PRINTOUT HAD BEEN SUSPENDED BY THE CONTROL-S CHARACTER.

3.1.6 CARRIAGE RETURN

THE CARRIAGE RETURN CHARACTER IS USED TO TERMINATE COMMAND STRINGS AND SIGNIFIES ENTRANCE OF A COMMAND. ALL COMMANDS MUST BE TERMINATED WITH EITHER A CARRIAGE RETURN OR A LINE FEED. IF ONLY CARRIAGE RETURN IS TYPED, (BLANK COMMAND) THE EXERCISER SIMPLY ISSUES A NEW "PCL>".

WHEN CARRIAGE RETURN <CR> IS TYPED, BOTH CARRIAGE RETURN AN LINE FEED ARE ECHOED.

3.1.7 LINE FEED

THE LINE FEED CHARACTER IS TREATED EXACTLY AS THE CARRIAGE RETURN CHARACTER. IF ONLY A LINE FEED IS TYPED, (BLANK COMMAND) THE EXERCISER ISSUES A NEW "PCL>".

3.1.8 RUBOUT

THIS CHARACTER IS USED TO 'EDIT' THE COMMAND STRING WHILE IT IS BEING TYPED. EACH RUBOUT TYPED WILL REMOVE A CHARACTER FROM THE COMMAND BUFFER. WHEN ALL THE CHARACTERS HAVE BEEN REMOVED, THE EXERCISER WILL ISSUE A NEW "PCL>" TO INDICATE THE ENTIRE COMMAND HAS BEEN ERASED.

EACH TIME A RUBOUT IS TYPED, A "\" IS ECHOED AND THE LAST CHARACTER WHICH WAS INPUT IS REMOVED.

C..) OMMANDS

THIS SECTION DEALS WITH THE FULL COMMANDS OF THE EXERCISER WHICH AL.OW THE SETTING UP OF CONDITIONS TO BEST SUIT THE SYSTEM BEING TESTED. IN THE COMMANDS SHOWN, THE MINIMUM AMMOUNT REQUIRED TO BE TYPED IS SET IN SQUARE BRACKETS []. FOR EXAMPLE, IN THE COMMAND:

[I]NITIALIZE

AL. OF THE FOLLOWING ARE ACCEPTABLE:

I, IN, INI, INIT, INITI, INITIA, INITIAL, INITIALI, INITIALIZ AND INITIALIZE

WHEREAS THE FOLLOWING ARE NOT ACCEPTABLE:

INT, INITL, INITIALIZES

THERE ARE SOME COMMANDS WHICH MAY BE EXECUTED WITH OR WITHOUT ARGUMENTS. IN THESE CASES,THE <ARGUMENTS> WILL BE SET IN ANGLE BRACKETS. ARGUMENTS FOR COMMANDS MUST BE NUMERIC, EXCEPT FOR THE COMMANDS 'MASTER'', 'SECONDARY'', AND 'RIB'', AND IF THEY ARE TYPED IN AS DECIMAL NUMBERS, THY MUST BE IMMEDIATELY FOL!OWED BY A DECIMAL POINT (PERIOD). FOR THE 'RANGE'' AND ''ASSIGN'' COMMANDS, THE ORDER OF THE INPUT OF ARGUMENTS MUST ADHERE TO THE DESCRIPTIONS (SECT 3.2.5, & 3.2.13)

3.2.1 SILO

[SI]LO <A B C ....N>
SILO -- CLEAR THE MASTER ADDRESS SILO AND RETURN TO ''AUTO ADDRESS MODE''.

SILO A B C . . N
LOAD THE MASTER ADDRESS SILO WITH THE SEQUENCE ''ABC..N'' AS MANY TIMES AS THE WHOLE SEQUENCE WILL FIT INTO THE SILO'S 50 LOCATIONS.

THIS COMMAND MONITORS THE ARGUMENTS CAREFULLY CHECKING FOR SEQUENTIAL ARGUMENTS WHICH ARE THE SAME, ARGUMENTS WHICH ARE 0, OR ARE GREATER THAN 37 OCTAL, AND CHECKING THAT THE LAST ARGUMENT IS NOT THE SAME AS THE FIRST.
IF ANY TWO SEQUENTIAL ARGUMENTS ARE FOUND TO BE THE SAME,INCLUDING THE FIRST AND THE LAST,A ''PAD'' VALUE IS INSERTED BETWEEN THEM. THE ''PAD'' VALUE IS 0. THIS ALSO INCREASES THE NUMBER OF ARGUMENTS IN THE SEQUENCE.
IF ANY OF THE ARGUMENTS ARE ''0'', OR GREATER THAN ''37'' OR NON-NUMERIC, OR IF THERE ARE MORE THAN 50. ARGUMENTS, THE COMMAND WILL BE ABORTED AND THE ''**SYNTAX ERROR**'' MESSAGE WILL BE PRINTED.
WHEN THE SILO HAS BEEN SUCCESSFULLY LOADED, THERE IS THE POSSIBILITY THAT ONE OR TWO MESSAGES WILL BE PRINTED, OR NONE MAY BE PRINTED.

IF THIS UNIT IS NOT MASTER, THE FOLLOWING WILL BE PRINTED:

''THIS UNIT IS NOT MASTER BUT HAS BEEN MADE SECONDARY
THE SILO YOU HAVE JUST LOADED WILL BE USED IF YOU CLEAR
THE CURRENT MASTER.''

IF IT WAS REQUIRED FOR THE EXERCISER TO ''PAD'' THE SILO WITH 0'S

'HE FOLLOWING WILL BE PRINTED:

'THE SILO HAS BEEN PADDED WITH THE ADDRESS '0''.''

SEQ 0015

3.2 2 MASTER

[M]ASTER [S]ET OR [C]LEAR

MASTER SET - WRITE A (1) INTO THE TMMR REGISTER BIT 08 IN THIS TRANSMITTER TO ATTEMPT TO SET MASTER. ANOTHER UNIT HAVING 'MASTER'' SET, WILL DIS-ALLOW THIS BIT TO BE SET.

MASTER CLEAR - WRITE A (0) INTO THE TMMR REGISTER BIT 08 IN THIS TRANSMITTER TO CLEAR 'MASTER''. IF NO OTHER UNIT HAS BEEN SET TO BE ''SECONDARY'', THIS ACTION WILL CAUSE 'MASTER DOWN'' ERRORS ON ALL UNITS ATTEMPTING TO TRANSMIT.

THIS COMMAND IS USED TO ''SOFTWARE SET'' ONE OF THE PCL11 UNITS TO BE MASTER OF THE T.D.M. BUS. THE ONLY ARGUMENTS ALLOWED ARE ''[S]ET'' AND ''[C]LEAR''. NO ARGUMENTS, OR ANY OTHER ARGUMENTS THAN THESE, WILL RESULT IN THE MESSAGE ''**SYNTAX ERROR**''.

SINCE THE HARDWARE DESIGN WILL NOT ALLOW MASTER TO BE SET ON MORE THAN ONE UNIT CONNECTED TO THE SAME T.D.M. BUS, THIS COMMAND MAY NOT WORK IN SETTING MASTER. THERE IS NOT, HOWEVER, ANY INDICATION THAT THE 'MASTER SET'' COMMAND WAS OR WASN'T SUCCESSFUL.

3.2.3 SECONDARY

[SE]CONDARY [S]ET OR [C]LEAR

SECONDARY SET - WRITE A (1) INTO THE TMMR REGISTER BIT 09 IN THIS TRANSMITTER TO ATTEMPT TO SET ''SECONDARY''. THIS UNIT BEING 'MASTER'' WILL DIS-ALLOW THIS BIT TO BE SET.

SECONDARY CLEAR - WRITE A (0) INTO THE TMMR REGISTER BIT 09 IN THIS TRANSMITTER TO CLEAR ''SECONDARY''.

THIS COMMAND IS USED TO ''SOFTWARE SET'' ONE OF THE PCL11 UNITS TO BE SECONDARY MASTER OF THE T.D.M. BUS. IF THE PCL11 UNIT WHICH CURRENTLY HAS MASTER SET CLEARS IT, AND THIS UNIT HAS SECONDARY SET, THEN THIS UNIT WILL BECOME THE NEW MASTER. IF THIS UNIT IS IN COMMUNICATION WITH A RECEIVER AT THE TIME IT BECOMES NEW MASTER, THE MESSAGE :

** THIS UNIT HAS BECOME 'NEW MASTER'' **

IS PRINTED ON THE CONSOLE AND THE EXERCISER AUTOMATICALLY CONTINUES EXERCISING.

3.2.4 RIB

[R]IB [S]ET OR [C]LEAR

RIB SET - WRITE A (1) INTO THE MEMORY LOCATION (BIT 15) WHICH IS USED AS THE TRANSMITTER COMMAND WORD. THE LOCATION IS TAGGED: "TXMST:". THIS WILL CAUSE THE NEXT AND SUBSEQUENT TRANSMITTER EVENTS TO OCCUR IN THE "RE-TRY - IF - BUSY" MODE.

RIB CLEAR - WRITE A (0) INTO THE MEMORY LOCATION (BIT 15) WHICH IS USED AS THE TRANSMITTER COMMAND WORD TAGGED "TXMST". THIS WILL CAUSE "BUSY" INTERRUPTS TO OCCUR WHENEVER THE RECEIVER BEING ADDRESSED IS NOT READY OR NOT THERE.

THIS COMMAND IS USED TO "SOFTWARE SET" THE "RIB" OR RE-TRY IF BUSY FEATURE IN THE PCL11 TRANSMITTER. WITH "RIB" SET, THE TRANSMITTER HARDWARE WILL CONTINUOUSLY RE-TRY TO CONNECT TO THE SELECTED RECEIVER UNTIL EITHER A "TIME-OUT" OCCURS, OR THE CONNECTION IS MADE AND A "SUCCESSFUL TRANSFER" OCCURS.

WITH "RIB" CLEAR, THE HARDWARE PRODUCES AN INTERRUPT IMMEDIATELY UPON FINDING THE ADDRESSED RECEIVER BUSY OR NOT THERE. THE USE OF THIS COMMAND WILL CAUSE A DIFFERENCE IN THE "SUCCESSFUL CONNECTION" COUNT RELATIVE TO THE "ATTEMPTED CONNECTION" COUNT IN THE STATUS PRINTOUT.

3.2.5 RANGE

[RA]NGE [LOW HIGH]

RANGE A B PRODUCE A LIST, IN THE STATUS TABLE, OF ADDRESSES FROM ADDRESS "A" TO "B" AND MAKE THEM "ACTIVE" RECEIVER ADDRESSES. "B" MUST BE HIGHER THAN "A". BOTH "A" AND "B" ARE REPRESENTATIVE OF NUMERICAL VALUES WITHIN THE RANGE OF 1-37 OCTAL.

EXAMPLE:

THE COMMAND "RANGE 12. 16." WILL ACTIVATE RECEIVER ADDRESSES 12 UP TO AND INCLUDING 16 (DECIMAL), OR 14 UP TO AND INCLUDING 20 (OCTAL). THESE ADDRESSES WILL BE USED SEQUENTIALLY BY THE TRANSMITTER MODULE IN THE EXERCISER AS TARGET RECEIVER ADDRESSES.

3.2.6 ADD

[AD]D [1] 2 3 . . 37

ADD A B C — ADD THE NUMERIC VALUES OF A, B C TO THE STATUS TABLE AND MAKE THEM ''ACTIVE RECEIVER ADDRESSES'. THIS COMMAND MAY HAVE ANY NUMBER OF ARGUMENTS PROVIDING THAT THERE IS AT LEAST ONE. THERE IS NO RESTRICTION ON THE ORDER OF THE ARGUMENTS AT ALL. THE SAME NUMBER MAY BE ADDED AS MANY TIMES AS YOU LIKE. THE ARGUMENTS MUST BE NUMERICAL AND BETWEEN THE VALUES 1 AND 31. (DECIMAL) OR 1 AND 37 (OCTAL).

EXAMPLE:

THE COMMAND ''ADD 1 12 3 3 22 8. '' WILL ACTIVATE RECEIVER ADDRESSES 1 3 10 12 AND 22 (OCTAL) OR 1 3 8. 10. AND 18. (DECIMAL) THIS COMMAND AFFECTS ONLY THOSE ADDRESSES INCLUDED IN THE ARGUMENTS OF THE COMMAND. IT MAY BE USED EFFECTIVELY IN CONJUNCTION WITH THE ''RANGE'' COMMAND TO PRODUCE A AN EFFICIENT LIST OF THOSE RECEIVER ADDRESSES WHICH ARE TO BE COMMUNICATED WITH BY THIS TRANSMITTER.

3.2.7 DELETE

[D]ELETE [1] 2 3 . . 37

DELETE A B C — DELETE THE NUMERIC VALUES OF A, B, C FROM THE STATUS TABLE AND MAKE THEM ''INACTIVE RECEIVER ADDRESSES''. THIS COMMAND MAY HAVE ANY NUMBER OF ARGUMENTS PROVIDING THERE IS AT LEAST ONE. THERE IS NO RESTRICTION ON THE ORDER OF THE ARGUMENTS AT ALL. AND REDUNDANCIES ARE ACCEPTABLE. THE ARGUMENTS MUST BE NUMERIC AND BETWEEN THE VALUES 1 AND 31. (DECIMAL) OR 1 AND 37 (OCTAL).

EXAMPLE:

THE COMMAND 'DELETE 12 22 '' WILL MAKE RECEIVER ADDRESSES 12 AND 22 ''INACTIVE'' SO THAT THIS TRANSMITTER WILL NOT ATTEMPT TO COMMUNICATE WITH THEM. THIS COMMAND MAY BE USED EFFECTIVELY IN CONJUNCTION WITH THE 'RANGE'' COMMAND AS FOLLOWS:

```
RANGE 1 7
DELETE 2 5
```

WILL PRODUCE A LIST OF ''ACTIVE'' RECEIVERS WITH THE ADDRESSES:

1, 3, 4, 6, AND 7

3.2.8 CLEAR

[C]LEAR

CLEAR - CLEAR THE ENTIRE STATUS TABLE. REMOVE ALL RECEIVER ACTIVE FLAGS AND CLEAR ALL CONNECTION ATTEMPS AND SUCCESSES FROM EVERY TABLE LOCATION. THIS COMMAND IS MOSTLY USEFUL WHEN IT IS DESIRED TO "RE-START" THE EXERCISER WITH A FRESH SLATE BUT NOT DISTURB THE ERROR SUMMARY TABLE NOR THE ERRORS TABLES.
THE "CLEAR" COMMAND HAS NO AFFECT ON ANY OTHER TABLES IN THE EXERCISER BUT IT DOES ALSO CLEAR THE INTERNAL (SOFTWARE) QUEUE OF RECEIVER ADDRESSES.

EXAMPLE:

CONSIDER THE FOLLOWING STRING OF COMMANDS:

```
RANGE 1 21
ADD 27 30 31 32 33 34
DELETE 17 20
CLEAR
RANGE 1 16
ADD 21 37
```

THE RESULT WOULD BE THE "ACTIVATION" OF RECEIVER ADDRESSES:

1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 21, 37

3.2.9 INITIALIZE

[I]NITIALIZE

INITIALIZE - INITIALIZE EVERYTHING ABOUT THIS TRANSMITTER AND CLEAR ALL TABLES ASSOCIATED WITH THIS PDP-11. INITIALIZE PERFORMS A HARDWARE CLEAR OF THE TRANSMITTER REGISTERS, RESETS ALL OF THE EVENT FLAGS ASSOCIATED WITH THE SENDING OF DATA, CLEARS THE SOFTWARE QUEUE OF RECEIVER ADDRESSES, INITIALIZES THE SEEDS USED FOR GENERATION OF RANDOM DATA, DOES A "CLEAR" COMMAND AS ABOVE AND CLEARS BOTH ERROR TABLES.

3.2.10 STATUS

[S*]ATUS

STATUS - SET THE EVENT FLAG WHICH CAUSES THE CURRENT STATUS TABLE TO BE PRINTED. EVERY TIME THE STATUS COMMAND IS ISSUED, A STATUS HEADER IS PRINTED (SEE 2.1). THEN, IN NUMERICAL ORDER THE "ACTIVE" RECEIVER ADDRESSES, NUMBER OF ATTEMPTED CONNECTIONS, AND NUMBER OF SUCCESSFUL CONNECTIONS IS PRINTED ON A LINE-BY-LINE BASIS. THE PRINTING OF THE STATUS TABLE DOES (LIKE ALL COMMANDS) INHIBIT ANY ACTION BY THE TRANSMITTER IN THAT PDP-11 BUT THE RECEIVER IS ALWAYS ACTIVE.

3.2.11 SUMMARY

[SU]MMARY

SUMMARY - SET THE EVENT FLAG WHICH CAUSES THE CURRENT ERROR SUMMARY TABLE TO BE PRINTED. EVERY TIME THE SUMMARY COMMAND IS ISSUED, A SUMMARY HEADER IS PRINTED (SEE 2.2). THEN, IN NUMERICAL ORDER, THE ERROR NUMBERS, ERROR ADDRESSES, AND NO. OF OCCURRENCES ARE PRINTED (IF ANY) ON A LINE-BY-LINE BASIS. AGAIN, TRANSMITTER ACTIVITY IS SUSPENDED UNTIL THE EXERCISER IS "CONTINUED".

3.2.12 ERRORS

[E]RRORS

ERRORS - SET THE EVENT FLAG WHICH CAUSES THE CURRENT TRANSMITTER ERROR TABLE TO BE PRINTED. WHEN THE TRANSMITTER ERROR EVENT IS FINISHED, IT WILL AUTOMATICALLY SET THE RECEIVER ERROR EVENT FLAG WHEREBY THE RECEIVER ERROR TABLE WILL BE PRINTED (SEE 2.3).

WHEN THE "ERROR" COMMAND IS ISSUED, A CHECK IS MADE OF THE ENTIRE TRANSMITTER AND RECEIVER TABLES TO DETERMINE IF THERE HAD BEEN ANY ERROR OCCURRENCES TO DATE. IF THERE WERE NO ERRORS IN EITHER TABLE, THEN THE MESSAGE:

"NO ERRORS TO REPORT"

IS PRINTED. OTHERWISE, THE ERROR NUMBER (IN NUMERICAL ORDER), THE CONNECTED RCVR/XMTR, AND THE ERROR COUNT ARE PRINTED ON A LINE-BY LINE BASIS. TRANSMITTER ACTIVITY IS AGAIN SUSPENDED UNTIL THE EXERCISER IS "CONTINUED".

3.2.13 ASSIGN

[AS]SIGN <XM ADDR XM VCT RC ADDR RC VCTR>

ASSIGN - GIVE TO THE EXERCISER, THE UNIBUS ADDRESSES AND VECTORS OF THE PCL11 UNIT WHICH THE OPERATOR DESIRES TO EXERCISE.
AS IS INDICATED BY THE ANGLE BRACKETS, THE ASSIGN COMMANDS' ARGUMENTS ARE OPTIONAL. IF THE ASSIGN COMMAND IS ISSUED WITH NO ARGUMENTS, THEN THE 'DEFAULT' (NORMAL) ADDRESSES AND VECTORS ARE ASSIGNED. THESE ARE:

XMTR ADDR	164200
XMTR VECTOR	170
RCVR ADDR	164220
RCVR VECTOR	174

THE ''ASSIGN'' COMMAND MAY ALSO BE USED WITH ANY, OR ALL OF FOUR ARGUMENTS. HOWEVER THE PROPER FIELD MUST BE USED TO ENTER THE DESIRED ADDRESS:

ASSIGN AAAAA BBB CCCCC DDD

TO ENTER ONLY THE TRANSMITTER ADDRESS, ONLY THE FIELD AAAAA NEED BE USED.

```
ASSIGN 166200
```

TO ENTER THE TRANSMITTER VECTOR (BBB), FIRST THE TRANSMITTER ADDRESS, THEN THE VECTOR IS TYPED:

```
ASSIGN 166200 700
```

TO ENTER THE RECEIVER ADDRESS (FIELD CCCCC), FIRST THE TRANSMITTER ADDRESS, THEN THE TRANSMITTER VECTOR, THEN THE RECEIVER ADDRESS IS TYPED:

```
ASSIGN 166200 700 166220
```

FINALLY, TO ASSIGN THE RECEIVER VECTOR, THE XMTR ADDRESS, XMTR VECTOR, RCVR ADDRESS AND THEN THE RECEIVER VECTOR IS TYPED:

```
ASSIGN 166200 700 166220 704
```

NOTE THAT EACH ARGUMENT MUST BE SEPARATED BY A SPACE (NOT A COMMA).
AT THE SUCCESSFUL COMPLETION OF THE ''ASSIGN'' COMMAND, THE EXERCISER WILL BE STARTED OVER JUST AS THOUGH THE OPERATOR HAD STARTED AT 200.
A ''**SYNTAX ERROR**'' WILL OCCUR WITH ANY OF THE FOLLOWING CONDITIONS:

- TOO MANY ARGUMENTS
- ARGUMENT IS NOT NUMERIC
- ADDRESS ARGUMENT NOT IN I/O ADDRESS FIELD
  (I.E. ABOVE 163776)
- VECTOR ARGUMENT IS NOT IN VECTOR FIELD
  (I.E. FROM 0 TO 776)
- ADDRESS ARGUMENT HAS WRONG OFFSET
  (I.E. LAST 4 BITS MUST BE 0)
- VECTOR ARGUMENT HAS WRONG OFFSET
  (I.E. LAST 2 BITS MUST BE 0)
- IMPROPER SPELLING OF ''ASSIGN'' OR IMPROPER USE OF DECIMAL NUMBERS.

3.2.14 GO

[G]O

GO - START THE EXERCISER. ENTER ''EXERCISE MODE''. GO IS ISSUED TO INITIALLY START THE EXERCISER TRANSMITTING TO OTHER RECEIVERS ON THE T.D.M. BUS. ALL TARGET RECEIVER ADDRESS SHOULD HAVE ALREADY BEEN ENTERED VIA THE ''RANGE'' OR ''ADD'' COMMANDS, AND ONE OF THE PCL11 UNITS SHOULD HAVE BEEN SET TO BE T.D.M. BUS MASTER. IF THE ''SILO'' IS BEING USED TO GENERATE TRANSMITTER ADDRESSES, IT SHOULD HAVE BEEN LOADED PRIOR TO ''GO''.

THE ''GO'' COMMAND WILL CAUSE THE CLEARING OF THE ERRORS TABLES AND THE SUMMARY TABLE AND THE ''ATTEMPTS'' AND ''SUCCESSES'' PORTIONS OF THE STATUS TABLE. IT ALSO CAUSES THE CLEARING OF THE RECEIVER ADDRESS QUEUE (SOFTWARE). NOTE THAT THE RECEIVER ERRORS ACCUMULATED UP TO THE POINT OF TYPING ''GO'' ARE LOST. ONLY THOSE ACCUMULATED AFTER ''GO'' IS TYPED CAN BE DISPLAYED IN THE ERRORS TABLE. IT WILL NOT, HOWEVER, CAUSE THE CLEARING OF THE RECEIVER ADDRESSES IN THE STATUS TABLE; SO THAT ALL RECEIVERS SELECTED BY THE RANGE OR ADD COMMANDS WILL STILL BE ACTIVE.

THE TRANSMITTER ''EVENT'' FLAG IS SET BY THE GO COMMAND WHICH BEGINS THE TRANSMISSION OF DATA TO THE DE-QUEUED TARGET RECEIVER ADDRESSES. WHEN ''GO'' IS ISSUED, THE IMMEDIATE RESPONSE SHOULD BE THAT THE EXERCISER PRINTS:

''EXERCISER STARTED''

HOWEVER, IF THE OPERATOR FORGOT TO ENTER THE TARGET RECEIVER ADDRESSES, AND NONE WERE PREVIOUSLY ENTERED, THE EXERCISER WILL NOT BE ABLE TO RUN AND WILL PRINT:

''** NO RECEIVERS SELECTED **''

3.2.15 CONTINUE

[CO]NTINUE

CONTINUE - CONTINUE EXERCISING. RE-ENTER EXERCISE MODE. ''CONTINUE'' IS ISSUED TO CAUSE THE EXERCISER TO CONTINUE AFTER A TABLE HAS FINISHED BEEING PRINTED, OR AFTER SOME OTHER COMMAND HAS BEEN EXECUTED TO POSSIBLY CHANGE THE RUNNING OF THE EXERCISER.

THE ''CONTINUE'' COMMAND DOES NOTHING TO ANYTHING EXCEPT THAT IT RE-STARTS EXERCISE MODE. ALL TABLES ARE LEFT INTACT AND SOFTWARE QUEUES ARE UNTOUCHED. WHEN ''CONTINUE'' IS ISSUED, THE IMMEDIATE RESPONSE SHOULD BE THAT THE EXERCISER PRINTS:

''EXERCISER CONTINUING''

IF, HOWEVER, WHILE THE EXERCISER WAS STOPPED,
THE OPERATOR USED THE INITIALIZE OR CLEAR
COMMANDS AND DID NOT RE-ENTER ANY TARGET REC-
EIVER ADDRESSES (VIA RANGE OR ADD),THE REC-
EIVER ADDRESS QUEUE WOULD EVENTUALLY BECOME
EMPTY AND WOULD NEVER BE RE-FILLED. THIS WOULD
AGAIN CAUSE THE PRINTOUT:
  "** NO RECEIVERS SELECTED **"

3.2.16 **SYNTAX ERROR**

SYNTAX ERROR IS NOT A COMMAND BUT IS RELATED TO ALL OF THE COMMANDS OF SECTION 3.2. IN GENERAL, A SYNTAX ERROR WILL RESULT FOR THE FOLLOWING REASONS:

.1 COMMAND DOES NOT EXIST IN "KEYWORD" TABLE
.2 COMMAND WORD IS MISSPELLED
.3 FEWER THAN THE MINIMUM CHARACTERS WERE USED
   (I.E. LESS THAN THAT ENCLOSED IN SQUARE BRACKETS [CL]EAR)
.4 NO ARGUMENTS GIVEN WHERE REQUIRED
.5 ARGUMENTS GIVEN WHERE NONE REQUIRED
.6 ARGUMENTS IN WRONG ORDER (WHERE ORDER IS IMPORTANT)
.7 NOT ENOUGH ARGUMENTS
.8 TOO MANY ARGUMENTS
.9 ARGUMENTS ARE WRONG CLASS (USUALLY SHOULD BE NUMBERS)
.10 ARGUMENTS ARE OUTSIDE SPECIFIC BOUNDARIES (SEE 3.2.13)
.11 ARGUMENTS SEPARATED BY OTHER THAN A SPACE
.12 COMMAND SEPARATED FROM ARGUMENTS FROM OTHER THAN A SPACE
.13 DECIMAL NUMERIC ARGUMENTS USED WITHOUT THE POINT (.)
.14 ARGUMENTS ARE THE SAME (ONLY IN "RANGE" COMMAND)

4 GETTING THE EXERCISER STARTED

4.1 PREPARATION

BEFORE RUNNING THE PCL11 EXERCISER, THE FOLLOWING MUST HAVE BEEN PREVIOUSLY PERFORMED:

- INSURE ALL PCL11 UNITS ARE CORRECTLY INSTALLED IN EACH PDP-11 PROCESSOR.
- DETERMINE ALL OF THE T.D.M. BUS ADDRESSES OF THE TRANSMITTERS AND THE RECEIVERS OF THE UNITS WHICH IT IS DESIRED TO BE TESTED. INSURE THAT NO TWO TRANSMITTERS AND NO TWO RECEIVERS HAVE BEEN ASSIGNED THE SAME T.D.M. BUS ADDRESSES.
- RUN THE PCL11 ''STANDALONE'' TEST (CZPLBC0) WITHOUT ERRORS BEFORE CONNECTING ALL UNITS TOGETHER VIA THE T.D.M. BUS.

4.2 LOADING

THE PCL11 EXERCISER IS SUPPLIED IN ABSOLUTE BINARY FORMAT AND IS LOADED BY MEANS OF THE STANDARD PDP-11 ABSOLUTE LOADER OR THE ''XXDP'' LOAD COMMAND.

THE EXERCISER IS APPROXIMATELY 7-K LONG. THEREFORE, WHEN USING ''XXDP'', THE PROCESSOR MUST HAVE 16-K OF MEMORY.

THE EXERCISER MUST BE LOADED INTO EACH PDP-11 HOSTING A PCL11 WHICH IS TO BE TESTED.

4.3 DEVICE ADDRESSES

IT MAY BE FOUND THAT THE UNIBUS ADDRESSES OF THE PCL11 UNIT ARE DIFFERENT THAN THE DEFAULT ADDRESSES IN THE EXERCISER (SEE 3.2.13). IF THIS IS THE CASE, AND THERE ARE NO OTHER DEVICES ON THE UNIBUS WITH THE ADDRESSES 164200-164226, THE FOLLOWING PRINTOUT WILL OCCUR:

''DEVICE ADDRESS ERROR. USE ''ASSIGN'' COMMAND''

ALSO, IT MAY BE THE CASE THAT THERE ARE MORE THAN ONE PCL11 UNIT HOSTED BY ONE PDP-11 AND EACH ONE MUST BE TESETD ''ON-LINE'' USING THE EXERCISER.

IN EITHER CASE, THE OPERATOR MUST USE THE ''ASSIGN'' COMMAND AS SHOWN IN SECTION 3.2.13 TO ASSIGN THE CORRECT UNIBUS ADDRESSES TO THE EXERCISER IN ORDER TO EXERCISE THE RIGHT PCL11.

NOTE THAT THE EXERCISER WILL RUN WITH ONLY ONE PCL11 PER PDP-11 AT A TIME.

4.4 STARTING ADDRESSES

THE EXERCISER STARTING ADDRESS IS 200
THIS WILL INITIALIZE EVERYTHING AND START IN COMMAND INPUT MODE.

THE EXERCISER IS RESTARTED AT LOCATION 204
THIS WILL PRESERVE THE ERROR TABLES, STATUS TABLE, AND THE
RECEIVER ADDRESS QUEUE. IT WILL NOT, HOWEVER, PRESERVE THE
STATE OF THE TRANSMITTER ADDRESS SILO (HARDWARE CLEARED BY 'RESET)
NOR THE STATE OF ''MASTER'' OR ''SECONDARY''. THESE UNPRESERVED
STATES MUST BE RE-ESTABLISHED PRIOR TO STARTING THE EXERCISE MODE
WITH THE ''GO'' COMMAND.

START = 200
RESTART = 204

4.5 OPERATING PROCEDURES

A) LOAD THE PROGRAM USING THE PDP-11 ABS LOADER OR THE LOAD COMMAND OF ''XXDP''. (SEE 4.2)
B) LOAD ADDRESS 200; PRESS START. THE TEST WILL IDENTIFY ITSELF AND TEST THE DEVICE ADDRESSES.
C) IF THE PCL11 UNIBUS ADDRESSES OF THE UNIT TO BE TESTED ARE NON-STANDARD, USE THE ''ASSIGN'' COMMAND AS IN 3.2.13
D) DO A) TO C) (ABOVE) ON ALL PDP-11S BEFORE CONTINUING.
E) ASSIGN ONE OF THE UNITS AS MASTER EITHER BY USING THE ''MASTER SET'' COMMAND, (SEC. 3.2.2) OR BY LOADING THE XMTR ADDRESS SILO (SEC. 3.2.1) ON THE SELECTED UNIT.
F) AT EACH UNIT, DECIDE WHETHER IT IS DESIRED TO RUN THE TRANSMITTER IN THE 'RE-TRY - IF - BUSY'' MODE AND SET OR CLEAR 'RIB'' ACCORDINGLY (SEC. 3.2.4).
G) AT EACH UNIT, ENTER THE RECEIVER ADDRESSES OF ALL THE RECEIVERS THAT THIS TRANSMITTER IS TO COMMUNICATE TO, INCLUDING THE ADDRESS OF ITS OWN RECEIVER.
H) ON EACH UNIT, TYPE ''GO'' TO START THE EXERCISER(S).
I) PERIODICALLY, ON EACH UNIT, TYPE CNTRL-C, THEN ISSUE THE ''STATUS'' COMMAND TO INSURE THAT ALL RECEIVERS ARE BEING TALKED TO AND THAT THE CORRECT UNIT IS MASTER.
J) ALSO PERIODICALLY, ON EACH UNIT, ISSUE THE ''SUMMARY'' COMMAND TO DISCOVER IF THERE HAVE BEEN ANY ERRORS.
K) AT ANY TIME , THE ''ERRORS'' COMMAND MAY BE ISSUED TO DETERMINE WHICH ERRORS HAVE OCCURRED BETWEEN WHICH RECEIVER AND TRANSMITTER CONNECTION.
L) TO RESUME EXERCISING AS BEFORE ON ANY UNIT, TYPE ''CONTINUE'' ON THE UNITS WHICH HAD BEEN STOPPED BY CNTRL-C.(OR MASTER DOWN).

4.6 ERRORS

A LIST OF ERROR NUMBERS MAY BE FOUND IN APPENDIX C OF THIS DOCUMENT. THESE 'ERROR NUMBERS' ARE THOSE REFERRED TO IN THE SUMMARY TABLE AND IN THE ERRORS TABLES. A LITTLE MORE DETAIL MAY BE DETERMINED ABOUT THE ERROR BY REFERRING TO THE PROGRAM LISTING AROUND THE ADDRESS SHOWN IN THE SUMMARY TABLE. THE LISTING WILL HAVE, IN THE COMMENT FIELD, THE ERROR IDENTIFIER:

**** XMTR ERROR N ***** OR:
**** RCVR ERROR N *****

WHERE 'N' IS THE ERROR NUMBER. ABOVE THIS IDENTIFIER, WILL BE THE DESCRIPTION OR CAUSE OF THE ERROR WITH THAT NUMBER.

IT MAY BE NOTED THAT THE TRANSMITTER ERRORS ARE NUMBERED FROM 1 TO 15 (OCTAL), AND THE RECEIVER ERRORS ARE NUMBERED FROM 16 TO 30 (OCTAL). THIS IS DONE SO THAT ONLY ONE TABLE IS REQUIRED TO SUMMARIZE ALL THE ERRORS (SUMMARY TABLE).

IF THERE IS AN ALARMING NUMBER OF OCCURRENCES OF ERRORS, THE FOLLOWING STEPS SHOULD BE TAKEN:

A) DETERMINE THE ERROR CAUSE (FROM APPENDIX C)
B) DETERMINE WHICH TRANSMITTER AND/OR RECEIVER ARE SUSPECT (FROM THE ERRORS TABLES)
C) IF THE ERROR WAS NOT 'MASTER DOWN' (ERROR 10), RUN THE PCL11 'STANDALONE TEST' (YC-Z017D-0B) ON THE UNITS WITH THE SUSPECTED RECEIVER OR TRANSMITTER. SEE IF THE SAME ERROR TYPE CAN BE ACHIEVED WITH THE 'STANDALONE TEST'. IF NOT, THEN THE T.D.M. DRIVERS, OR CABLES, OR TERMINATORS ETC. ARE SUSPECT.
D) IF THE ERROR WAS 'ERROR 10' BUT THIS ERROR HAD NOT OCCURRED ON OTHER UNITS, THE CABLE, OR RECEIVER CHIPS ETC. ARE AGAIN SUSPECT.
E) USING EITHER THE EXERCISER, OR THE STANDALONE TEST, A DEFECTIVE SINGLE MODULE SHOULD BE RELATIVELY SIMPLE TO LOCALIZE AND REPLACE, CORRECTING THE PROBLEM.
F) ONCE A MODULE HAS BEEN REPLACED, ALWAYS RUN THE PCL11 STANDALONE TEST (EVEN IF IT WAS A LINE DRIVER MODULE) BEFORE RUNNING THE EXERCISER.

A SMALL NUMBER OF CERTAIN ERRORS IS ACCEPTABLE DURING A LONG EXERCISE RUN. THESE ERRORS WOULD BE ATTRIBUTED TO LINE NOISE, GENERAL SYSTEM NOISE ETC. THESE ERRORS ARE:

ERROR	6	TRANSMITTER CRC ERROR
ERROR	7	TRANSMITTER MISCELLANEOUS TXM ERROR
ERROR	22	RECEIVER CRC ERROR
ERROR	24	RECEIVER PARITY ERROR

5 COMMAND MODE DESCRIPTION

5.1 SHORTENED COMMANDS

ANY OF THE COMMANDS MAY BE TYPED IN AS SHORT A FORM AS WOULD SEEM REASONABLE. THAT IS, ONLY ENOUGH LETTERS NEED BE TYPED SO AS TO DISTINGUISH ONE COMMAND FROM ANOTHER WITH THE SAME FIRST LETTER.
FOR EXAMPLE, SINCE THERE IS ONLY ONE COMMAND BEGINNING WITH THE LETTER "E" (ERRORS), ONLY THE "E" NEED BE TYPED FOR THAT COMMAND. HOWEVER, THERE ARE FOUR COMMANDS BEGINNING WITH THE LETTER "S" (SUMMARY, STATUS, SILO, AND SECONDARY). IN EACH OF THESE COMMANDS, THE SECOND LETTER IS DIFFERENT, SO JUST TWO LETTERS NEED BE TYPED: (SU, ST, SI, AND SE).
ON THE OTHER HAND, THE COMMAND DECODER WILL NOT ACCEPT ANY COMMAND WORDS WITH ANY OF THE LETTERS WRONG. THAT IS, EVERY LETTER TYPED IN FOR A PARTICULAR COMMAND MUST BE AT LEAST ON THE WAY TO SPELLING THE WORD CORRECTLY.

FOR EXAMPLE: FOR THE "INITIALIZE" COMMAND:

INITIAL IS ACCEPTABLE, WHEREAS:
INITL IS UNACCEPTABLE.

5.2 RUBOUT FEATURES

THERE ARE TWO EDITING FEATURES EMPLOYED IN THE COMMAND DECODER. "RUBOUT" (DELETE) CHARACTER WILL DELETE THE LAST CHARACTER WHICH WAS TYPED IN AS PART OF A COMMAND WORD OR ARGUMENT. CONTROL-U CHARACTER WILL REMOVE ALL THAT HAS BEEN TYPED IN SO FAR ON THIS LINE.
ONE OTHER METHOD OF HAVING THE EXERCISER IGNORE EVERYTHING TYPED IN SO FAR IS TO TYPE "CONTROL-C".

A) RUBOUT DELETE LAST CHARACTER
B) CNTRL-U DELETE THIS LINE
C) CNTRL-C DELETE THIS LINE

5.3 ENTERING COMMAND MODE

COMMAND MODE IS AUTOMATICALLY ENTERED AT STARTUP OR RESTART OF THE EXERCISER. THERE ARE TIMES, HOWEVER, WHEN IT IS NOT IN "COMMAND" MODE. THEY ARE:
A) WHEN IN EXERCISE MODE (RUNNING)
B) WHILE PRINTING THE STATUS, SUMMARY, OR ERRORS TABLES

AT ANY TIME THAT IT IS DESIRED TO ENTER COMMAND MODE, THE OPERATOR NEED ONLY TYPE "CONTROL-C" (^C). THIS WILL TERMINATE ALL TRANSMITTER ACTIVITY ON THE UNIT AND ENTER COMMAND MODE. IT WILL ALSO, (AT COMPLETION OF THE CURRENT LINE), TERMINATE ALL TABLE PRINTING AND RETURN TO COMMAND MODE.
THERE IS ANOTHER CHARACTER WHICH WILL PERFORM THE SAME AS CONTROL-C DUE TO ITS FUNCTION; THAT IS CONTROL-U .

UPPER OR LOWER CASE

WHEN IN COMMAND MODE, THE OPERATOR MAY FIND HIMSELF USING A KEYBOARD WHICH DOES NOT HAVE A "CAPS LOCK" KEY. SINCE THE COMMAND DECODER REQUIRES THAT ALL INPUT BE IN CAPITAL LETTERS, THE KEYBOARD INPUT ROUTINE WILL AUTOMATICALLY CONVERT ALL LOWER CASE ALPHA CHARACTERS INTO UPPER CASE ALPHA CHARACTERS BY CLEARING BIT05 IN THE ASCII CODE OF THE INPUT CHARACTER.

6 EXERCISER MODE DESCRIPTION

6.1 TRANSMIT EVENT

AN ''EVENT FLAG'' IS CHECKED IN THE MAIN LOOP OF THE PROGRAM TO DETERMINE WHETHER THE EXERCISER HAS BEEN TOLD TO ''GO''. THIS FLAG IS THE TRANSMIT EVENT FLAG. IT IS SET WHENEVER THE OPERATOR ISSUES THE ''GO'' COMMAND, OR THE ''CONTINUE'' COMMAND TO THE EXERCISER. THIS FLAG IS CLEARED WHENEVER THE OPERATOR TYPES CONTROL-C OR IF A MASTER DOWN ERROR OCCURS.

WHEN THE FLAG IS DETECTED AS BEING SET, THE EXERCISER CALLS THE TRANSMITTER MODULE WHICH TRANSMITS A BLOCK OF DATA THAT HAD BEEN PREVIOUSLY GENERATED BY THE DATA GENERATION MODULE. IF THIS DATA HAS ALREADY BEEN USED FOR THE FIFTH TIME, IT SETS THE DATA GENERATION EVENT FLAG AND NEW RANDOM DATA WILL BE GENERATED. WHEN THE TRANSMIT MODULE IS CALLED, THE TRANSMIT EVENT FLAG IS CLEARED AND IS NOT SET AGAIN UNTIL SOME TYPE OF ''COMPLETION'' INTERRUPT HAS OCCURRED SUCH AS ''SUCCESSFUL TRANSFER'' OR 'ERROR''.

THE TRANSMIT EVENT IS ALSO RESPONSIBLE FOR UPDATING THE ERROR TABLES FOR TRANSMITTER ERRORS, AND ALSO THE STATUS TABLE FOR ATTEMPTS AND SUCCESSES TO EACH RECEIVER IN THE RECEIVER ADDRESS QUEUE.

6.2 RECEIVE EVENT

WHEN THE EXERCISER IS STARTED BY THE OPERATOR STARTING AT LOCATION 200,A SOFTWARE FLAG CALLED 'RECEIVER EVENT FLAG'' IS SET. WHEN THIS IS DETECTED IN THE MAIN LOOP, THE RECEIVER MODULE IS CALLED TO SET UP THE RECEIVER TO RECEIVE UP TO 600 (OCTAL) WORDS FROM A TRANSMITTER THAT TRIES. WHEN THE MODULE IS CALLED, THE FLAG (RCVR EVENT) IS CLEARED AND NOT SET AGAIN UNTIL SOME TYPE OF COMPLETION INTERRUPT IS RECEIVED SUCH AS ''SUCCESSFUL TRANSFER'', 'ERROR'', OR 'REJECT COMPLETED''.

THE RECEIVE EVENT IS ALSO RESPONSIBLE FOR UPDATING THE ERROR TABLES FOR RECEIVER ERRORS, AND FOR CHECKING THE DATA RECEIVED TO DETERMINE ITS CORRECTNESS AND THAT THE RIGHT NUMBER OF WORDS WERE RECEIVED.

UNLIKE THE TRANSMIT EVENT,THE RECEIVE EVENT CANNOT BE SUSPENDED BY THE OPERATOR ISSUING ANY COMMANDS OR CONTROL CHARACTERS. HALTING THE EXERCISER, OR A HARDWARE FAILURE TO INTERRUPT ARE THE ONLY WAYS TO PREVENT THE RECEIVE EVENT FROM OCCURRING.

6.3 DATA GENERATION EVENT

ANOTHER EVENT WHICH OCCURS IN EXERCISE MODE IS 'DATA GENERATION. A NEW BUFFER FULL OF RANDOM DATA IS GENERATED AFTER 5 PASSES WITH THE OLD DATA ARE COMPLETED. THE LENGTH OF THE DATA BUFFER ALSO RANDOMLY VARIES FROM 1 TO 1000 (OCTAL) WORDS. IF THE BUFFER IS LONGER THAN 600 WORDS, THE RECEIVER WILL BE EXPECTED TO TRUNCATE THE MESSAGE AFTER RECEIVING THE 600TH WORD.

ALSO, IF THE FIRST WORD OF THE BUFFER (''FLAGS'' WORD) HAS THE FOUR MOST SIGNIFICANT BITS SET, THE RECEIVER WILL REJECT THE MESSAGE ENTIRELY. ALL OF THESE 'REJECT'' AND ''TRUNCATE'' OCCURRENCES ARE EXPECTED BY THE TRANSMITTER EVENT AND CHECKS ARE MADE THAT THEY OCCUR AS THEY SHOULD.

ADDRESS QUEUE EVENT

WHEN THE USER HAS COMPLETED GENERATING THE LIST OF RECEIVER ADDRESSES HE WISHES A PARTICULAR TRANSMITTER TO COMMUNICATE WITH, THE STATUS TABLE HAS THOSE ENTRIES ''ACTIVATED''. DURING EXERCISE MODE ACTIVE ADDRESSES IN THE STATUS TABLE ARE LOADED INTO A SOFTWARE QUEUE TO AWAIT THEIR TURN WITH THE TRANSMIT MODULE. ADDRESSES ARE DEQUEUED ON EVERY ENTRANCE TO THE TRANSMIT MODULE AND THE ''ATTEMPTS'' ENTRY OF THE STATUS TABLE IS UPDATED. WHEN THE QUEUE IS EMPTY, THE ADDRESS QUEUE EVENT IS CALLED TO RE-FILL IT FROM THE STATUS TABLE.
NOTE THAT ADDRESSES ARE ALWAYS QUEUED AND DEQUEUED IN NUMERICAL ORDER REGARDLESS OF THE ORDER IN WHICH THEY WERE ENTERED.

6.5 ERRORS UPDATE EVENT

AT ANY TIME, WHETHER THE EXERCISER IS IN ''EXERCISE'' MODE OR COMMAND MODE, THE RECEIVE EVENT IS ACTIVE. THEREFORE, RECEIVER ERRORS CAN OCCUR AT ANY TIME. HOWEVER, TRANSMITTER ERRORS CAN ONLY OCCUR WHEN THE EXERCISER IS ''GOING''. WHEN AN ERROR OCCURS, OF THE TRANSMITTER TYPE, THE TRANSMITTER COMMAND REGISTER (TCR) IS READ TO GET THE ADDRESS OF THE CONNECTED RECEIVER. THIS IS USED TO DECIDE WHICH TRANSMITTER ERROR TABLE LOCATION TO INCREMENT ALONG WITH THE ERROR NUMBER. THIS UPDATE IS ACCOMPLISHED AT THE ACTUAL TIME THAT THE ERROR IS DISCOVERED. WHEN AN ERROR OF THE RECEIVE TYPE OCCURS THE RECEIVER COMMAND REGISTER (RCR) IS READ TO GET THE ADDRESS OF THE CONNECTED TRANSMITTER. AGAIN, THIS IS USED TO DETERMINE THE CORRECT TABLE LOCATION TO INCREMENT.

AGAIN, PLEASE NOTE THAT THE ''GO'' COMMAND CLEARS BOTH THE TRANSMITTER ERROR TABLES AND THE RECEIVER ERROR TABLES.

ANOTHER FUNCTION OF THE ERRORS EVENT IS TO UPDATE THE ''SUMMARY'' TABLE ACCORDING ONLY TO THE ERROR NUMBER.
THERE ARE, THEN, ACTUALLY TWO ERROR EVENTS: ONE FOR TRANSMITTER ERRORS, AND ONE FOR RECEIVER ERRORS.

6.6 SPECIAL EVENTS

DURING AN EXERCISE RUN, THERE ARE TWO INTERRUPTS THAT CAN CAUSE PRINTOUTS ON THE CONSOLE. ONE OF THEM IS MASTER GOING DOWN. THIS WILL CAUSE THE CALLING OF A ''SPECIAL'' EVENT TO PRINT THE MESSAGE:

*** MASTER DOWN ***

AND RETURN THE EXERCISER TO COMMAND MODE. THE OTHER INTERRUPT IS CAUSED BY THIS UNIT BECOMING ''MASTER'' AFTER HAVING BEEN ''SECONDARY''. THIS WILL CAUSE THE CALLING OF ANOTHER ''SPECIAL'' EVENT TO PRINT THE MESSAGE:

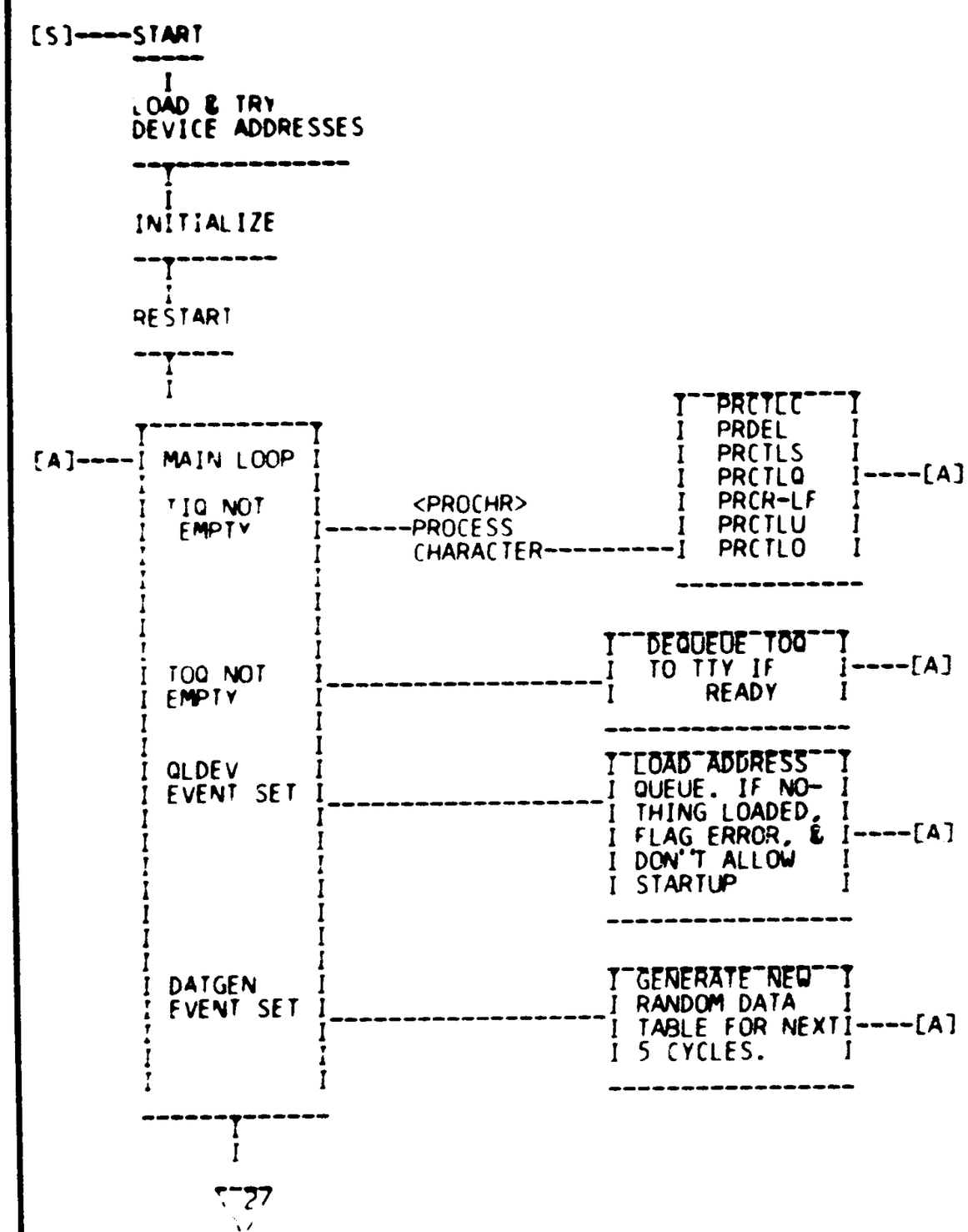
** THIS UNIT HAS BECOME ''NEW MASTER'' **

AND LEAVE THE EXERCISER IN ''EXERCISE MODE.

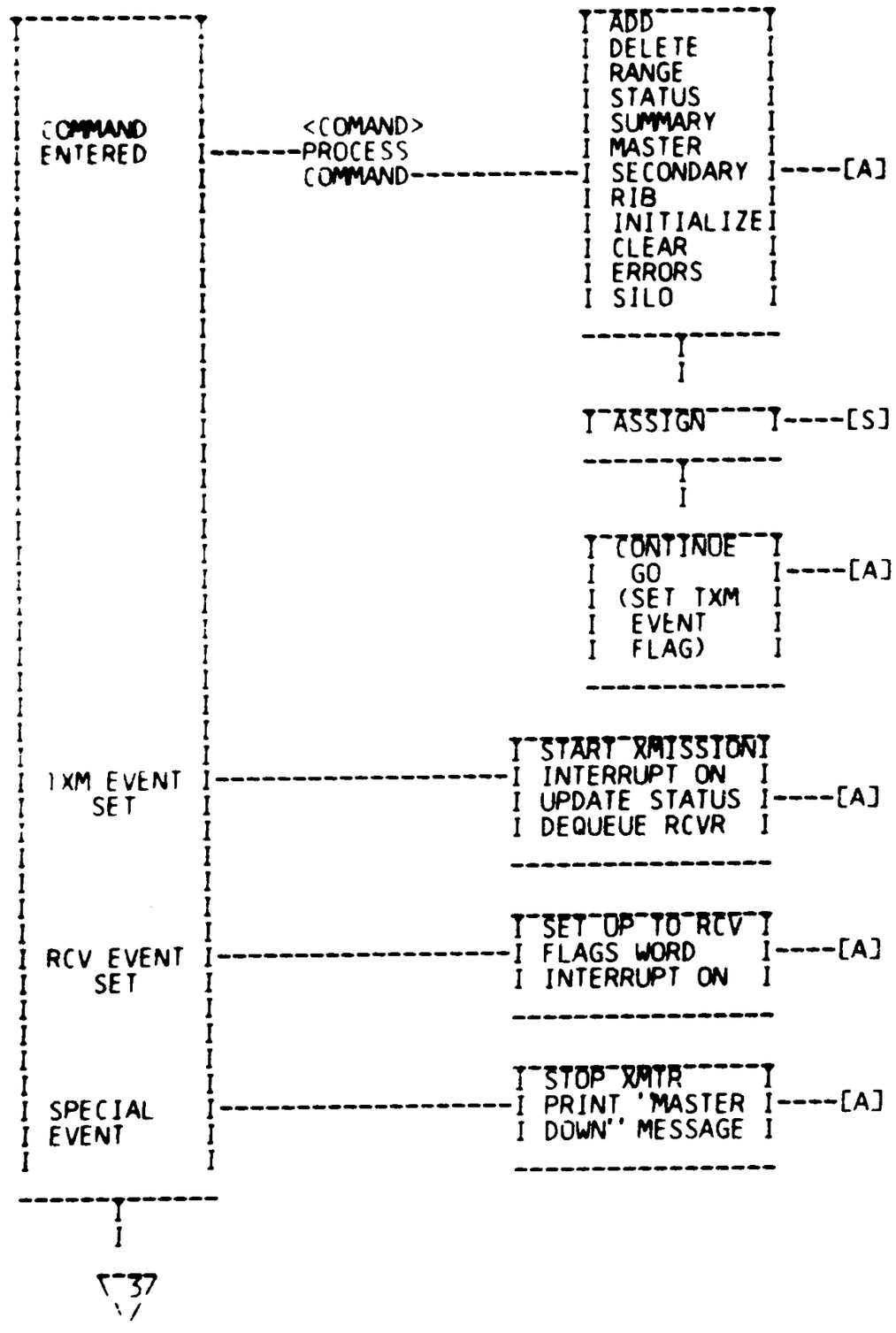
APPENDIX A, B, C, D

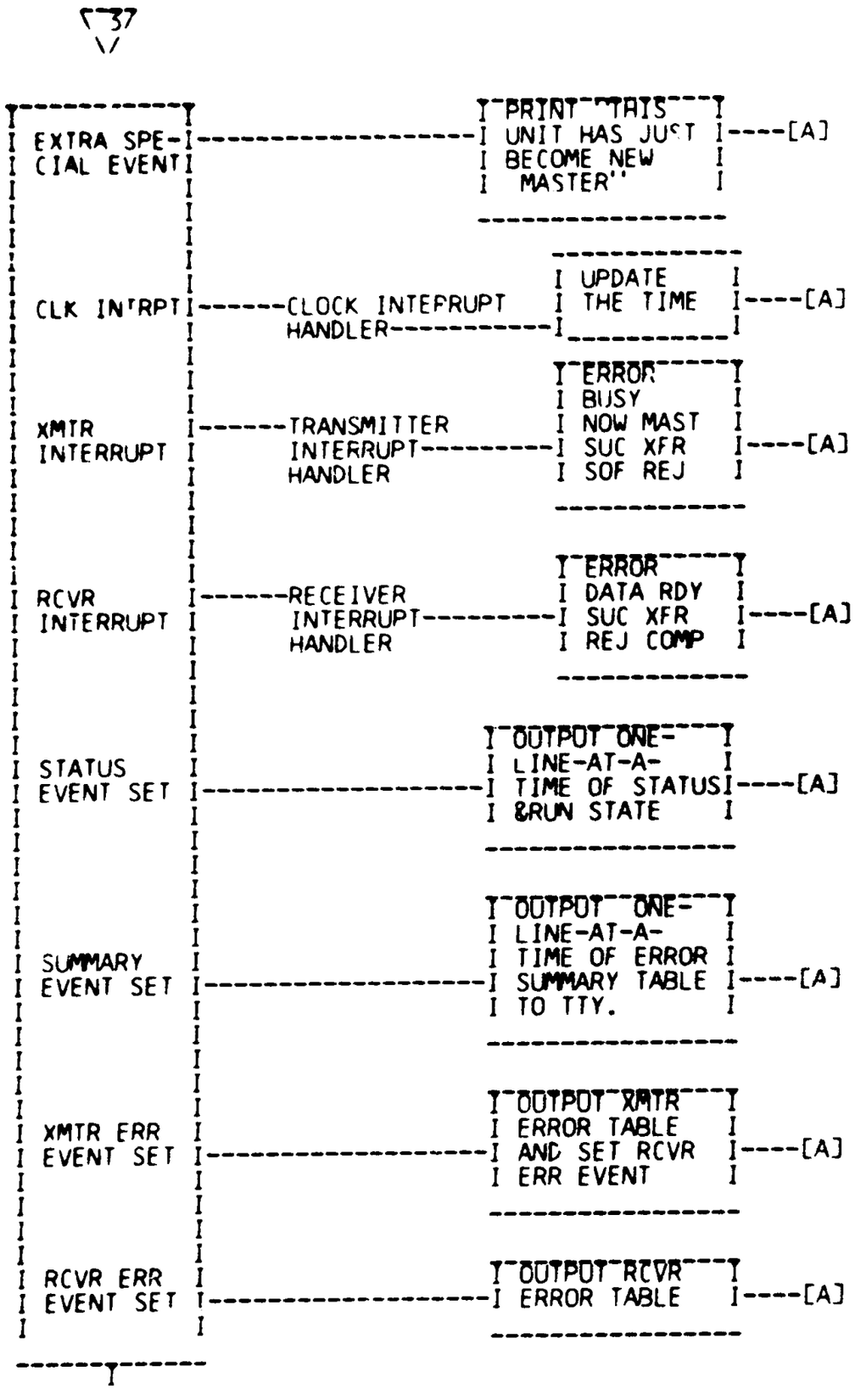
A-1

OVERALL FLOW



A-2


COMMAND ENTERED
<COMAND> PROCESS COMMAND
ADD
DELETE
RANGE
STATUS
SUMMARY
MASTER
SECONDARY
RIB
INITIALIZE
CLEAR
ERRORS
SILO
[A]
ASSIGN
[S]
CONTINUE GO (SET TXM EVENT FLAG)
[A]
IXM EVENT SET
START XMISSION
INTERRUPT ON
UPDATE STATUS
DEQUEUE RCVR
[A]
RCV EVENT SET
SET UP TO RCV
FLAGS WORD
INTERRUPT ON
[A]
SPECIAL EVENT
STOP XMTR
PRINT 'MASTER DOWN'' MESSAGE
[A]


37
EXTRA SPECIAL EVENT
PRINT "THIS UNIT HAS JUST BECOME NEW MASTER"
[A]
CLK INTRPT
CLOCK INTERRUPT HANDLER
UPDATE THE TIME
[A]
XMTR INTERRUPT
TRANSMITTER INTERRUPT HANDLER
ERROR BUSY NOW MAST SUC XFR SOF REJ
[A]
RCVR INTERRUPT
RECEIVER INTERRUPT HANDLER
ERROR DATA RDY SUC XFR REJ COMP
[A]
STATUS EVENT SET
OUTPUT ONE-LINE-AT-A-TIME OF STATUS &RUN STATE
[A]
SUMMARY EVENT SET
OUTPUT ONE-LINE-AT-A-TIME OF ERROR SUMMARY TABLE TO TTY.
[A]
XMTR ERR EVENT SET
OUTPUT XMTR ERROR TABLE AND SET RCVR ERR EVENT
[A]
RCVR ERR EVENT SET
OUTPUT RCVR ERROR TABLE
[A]

[$^{i}_{A}$]

SEQ 0035

EXERCISER COMMANDS

CONTROL CHARACTERS

CHARACTER	ECHOES	EFFECT
CNTRL-C	^C	ENTER COMMAND INPUT MODE
CNTRL-O	^O	THROW AWAY TTY OUTPUT
CNTRL-U	^U	DISCARD CURRENT INPUT LINE
CNTRL-S		SUSPEND TTY OUTPUT
CNTRL-Q		RESUME TTY OUTPUT
RUBOUT	\	DELETE LAST INPUT CHARACTER
CAR RET	<CR, LF>	PERFORM COMMAND JUST ENTERED
LINE FEED	<CR, LF>	SAME AS CAR RET

COMMANDS

COMMAND	ARGUMENTS	MINIMUM	EFFECT
[AD]D	A B C - N	AD A	ADD ADDRESSES A B C - N
[AS]SIGN	ADR VCT ADR VCT	AS	ASSIGN UNIBUS ADDRESSES AND VECTORS FOR TRANSMITTER AND RECEIVER.
[CL]EAR	-	CL	CLEAR THE STATUS TABLE
[CO]NTINUE	-	CO	CONTINUE EXERCISING
[D]ELETE	A B C - N	D A	DELETE ADDRS A B C - N
[E]RRORS	-	E	PRINT ERRORS TABLES
[G]O	-	G	START THE EXERCISER
[I]NITIALIZE	-	I	INIT THE EXERCISER
[M]ASTER	SET	M S	SET 'MASTER''
[M]ASTER	CLEAR	M C	CLEAR 'MASTER
[RA]NGE	LOW HI	RA L H	ADD RANGE OF ADDRESSES FROM LOW TO HIGH INCLUSIVE.
[RI]B	SET	RI S	SET 'RIB''
[RI]B	CLEAR	RI C	CLEAR 'RIB''
[SE]CONDARY	SET	SE S	SET ''SECONDARY''
[SE]CONDARY	CLEAR	SE C	CLEAR ''SECONDARY''
[SI]LO	-	SI	CLEAR SILO; SET AUTO ADDR
[SI]LO	A B C - N	SI A	LOAD SILO WITH A B C - N
[ST]ATUS	-	ST	PRINT STATUS TABLE
[SU]MMARY	-	SU	PRINT SUMMARY TABLE

ERROR DESCRIPTIONS

ERROR NUMBER	DESCRIPTION
1	ERRONEOUS INTERRUPT FROM TRANSMITTER
2	NON EXISTANT LOC. ERROR IN XMTR
3	MEM OVERFLOW ERROR IN TRANSMITTER
4	XMTR TXM ERROR: RCVR ACCEPTED A NULL
5	XMTR TXM ERROR: RCVR HAS GONE OFF-LINE
6	XMTR TXM ERROR: WORD OR C.R.C. REJECTED
7	XMTR TXM ERROR: MISCELLANEOUS TXM ERROR
10	MASTER DOWN
11	TRANSMITTER TIMED OUT
12	SILO OVERRUN ERROR IN TRANSMITTER
13	MESSAGE TRUNCATED UNEXPECTEDLY
14	MESSAGE FAILED TO BE TRUNCATED
15	ERRONEOUS REJECT BY RECEIVER
16	UNKNOWN RECEIVER INTERRUPT OCCURRED
17	NON-EXISTANT LOC. ERROR IN XMTR
20	MEM OVERFLOW ERROR IN RECEIVER
21	RCVR TXM ERROR: XMTR HAS GONE OFF-LINE
22	RCVR TXM ERROR: RCVR C.R.C. ERROR
23	RCVR TXM ERROR: FIRST WORD INVALID
24	RECEIVER DETECTED INVALID PARITY.
25	RECEIVER TIMEOUT ERROR OCCURRED
26	RECEIVER GOT TOO MANY WORDS
27	DATA WORD RECEIVED WAS BAD
30	RECEIVER GOT TOO FEW WORDS

D-1

SEQ 0039

STATISTICAL INFORMATION

STARTING ADDRESS	200
RESTARTING ADDRESS	204
SWITCH OPTIONS	NONE
PROGRAM SIZE	APPROX 7K
MEMORY OCCUPIED	
LOW BOUNDARY	00000
HIGH BOUNDARY	34456

LOCATIONS TO CHANGE FOR DIFFERENT DEVICE ADDRESSES

DEVICE	CHANGE LOCATION
KEYBOARD STATUS	16146
KEYBOARD DATA	16150
TTY STATUS	16152
TTY DATA	16154
KEYBOARD VECTOR	16156
LINE CLK STATUS	16160
XMTR PRIORITY	2326
RCVR PRIORITY	2340
KBD PRIORITY	2352

USEFUL LOCATIONS	ADDRESS
XMTR DATA BUFFER	20676
RCVR DATA BUFFER	22736
DATA SEED IN TRANSMITTER	17736
RANDOM MULTIPLIER	17756
RANDOM INCREMENT	17760

```
                        .TITLE  CZPLAC0 PCL11 EXERCISER V02C
                        .IDENT  /0003/          ;DRCMAC.MAC  6-JAN-76
                        .NLIST  TTM
                        .NLIST  ME


                    ; KEYWD MACRO

                    ;   THIS MACRO DETERMINES THE ROUTINE ADDRESS
                    ;   ASSOCIATED WITH THE SYNTACTIC OBJECT OBJ, ACCORDING
                    ;   TO A KEYWORD TABLE POINTED TO BY KWTABL.

                    ;   ON RETURN, IF OBJ WAS IN THE TABLE, THEN THE PS
                    ;   C BIT = 0 & THE ROUTINE ADDRESS IS AT THE TOP OF THE
                    ;   STACK.  IF NOT, THEN C=1 & @SP = 0.

                    ;   EACH KEYWORD IN THE TABLE HAS ASSOCIATED WITH IT A
                    ;   MINIMUM LENGTH SPECIFYING THE MINIMUM NUMBER
                    ;   OF CHARACTERS IN THE KEYWORD THAT MUST MATCH
                    ;   THOSE IN OBJ FOR A MATCH TO HAVE BEEN DEEMED
                    ;   FOUND.  IF OBJ CONTAINS MORE THAN THIS MINIMUM
                    ;   NO. OF CHARACTERS, THOUGH, ALL CHARACTERS IN OBJ MUST
                    ;   CORRESPOND TO THE TABLE KEYWORD FOR A MATCH TO
                    ;   HAVE BEEN DEEMED FOUND.  THUS, FOR EXAMPLE, IF
                    ;   "REPEAT" APPEARS IN THE KEYWORD TABLE WITH A
                    ;   MINIMUM LENGTH OF 3 ASSOCIATED WITH IT, THEN
                    ;   "REP","REPE","REPEA", & "REPEAT" WILL ALL MATCH IT, BUT
                    ;   "R","RPT","REPEET", & "REPEATER" WILL NOT.

                    ;   THE KEYWORD TABLE CONSISTS OF A SET OF ENTRIES AS FOLLOWS:
                    ;       OFFSET  TYPE
                    ;       0       WORD    ROUTINE ADDRESS
                    ;       2       BYTE    MINIMUM LENGTH OF KEYWORD
                    ;       3       BYTE    FULL LENGTH OF KEYWORD
                    ;       4       STRING  KEYWORD

                    ;   ENTRIES ARE STORED CONSECUTIVELY IN THE TABLE.  EACH ENTRY
                    ;   MUST BEGIN ON A WORD BOUNDARY.  THE KEYWORD OF
                    ;   THE PREVIOUS ENTRY MAY HAVE TO HAVE A BYTE (CONTAINING
                    ;   ANYTHING) APPENDED TO IT TO ACCOMPLISH THIS.
                    ;   ENTRIES MUST BE ARRANGED IN ALPHABETICAL
                    ;   ORDER (MORE SPECIFICALLY, IN ASCII COLLATING SEQUENCE).
                    ;   THE TABLE IS ENDED BY AN ENTRY WITH A FULL LENGTH
                    ;   OF 0.
                    ;
                            .MACRO  KEYWD   OBJ,KWTABL
                    .IF     NB      OBJ
                            MOV     OBJ,-(SP)
                      .IF   B       KWTABL
                            .ERROR  ;CANNOT SPECIFY OBJ & NOT TABLE ADDRESS.
                            MOV     #1,-(SP)                ;IF RUN, THIS WILL CAUSE TRAP.
                      .ENDC
                    .ENDC
                    .IF     NB      KWTABL
                            MOV     KWTABL,-(SP)
                    .ENDC
```

```
                        JSR     PC,KEYWD

                        .ENDM   ;KEYWD

                ; ---------------------------------------

                ;       LXSCAN MACRO
                ;
                ; THIS MACRO CALLS THE ROUTINE  LXSCAN  TO CONVERT THE CHARACTER
                ; STRING SPECIFIED BY  STR  INTO ITS CONSTITUENT SYNTACTIC OBJECTS.
                ; STR  IS AN ASSEMBLER EXPRESSION SPECIFYING THE ADDRESS OF THE
                ; BEGINNING OF THE ASCII STRING TO BE CONVERTED.  THE STRING
                ; MUST BE ENDED BY A CARRIAGE RETURN CODE.
                ;
                        .MACRO  LXSCAN  STR
                        MOV     STR,R1
                        JSR     R1,.XSCAN

                        .ENDM   ;LXSCAN

                ; ---------------------------------------

                ;       PROC MACRO

                ;  THIS MACRO DEFINES THE ENTRY POINT FOR A PROCEDURE.  THE PARAMS
                ;  SPECIFIED WILL BE GIVEN THEIR CORRESPONDING OFFSETS RELATIVE TO R5.
                ;  THUS, A PARAMETER  PP  CAN BE REFERENCED IN THE PROCEDURE BY THE
                ;  ASSEMBLER EXPRESSION '@PP(R5)'.  THE PROCEDURE CAN BE CALLED
                ;  USING THE  CALL  MACRO.

                        .MACRO  PROC    PNAME,PARAMS
                ZX1     =       0

                .IRP    ZX2     <PARAMS>
                ZX1     =       ZX1+2
                  .IRP  ZX3     \ZX1
                        .LIST
                ZX2     =       ZX3
                        .NLIST
                  .ENDM
                .ENDM   ;ZX2

                        .LIST

                PNAME:                                  ;**ENTRY POINT**
                        .NLIST
                        .ENDM   ;PROC

                ; ---------------------------------------

                ;       RETURN MACRO

                ;  THIS MACRO RETURNS FROM A PROCEDURE.  IF  ANSWR  IS SPECIFIED IT
                ;  WILL BE LOADED INTO R0 BEFORE RETURNING.

                        .MACRO  RETURN  ANSWR
```

```
                        .IF     NB      ANSWR
                                MOV     ANSWR,R0
                        .ENDC
                                RTS     PC
                                .ENDM   ;RETURN

                        ;-----------------------------------------

                        ;       CALL MACRO
                        ;
                        ; THIS MACRO CALLS A PROCEDURE, SUBR, WITH AN ARGUMENT LIST SPECIFIED
                        ; BY ARGS.  ARGS IS A LIST OF ADDRESSES WHICH WILL BE INCLUDED
                        ; IN THE ASSEMBLED EXPANSION OF THIS MACRO.
                        ;
                        ; THE CALLING SEQUENCE GENERATED IS FORTRAN COMPATIBLE.  R5 IS LEFT
                        ; INTACT THROUGH THE EXECUTION OF THIS MACRO.  OTHER REGISTERS
                        ; ARE DESTROYED.
                        ;
                                .MACRO  CALL    SUBR,ARGS,?PLIST,?ZXCALL
                                JSR     R5,ZXCALL
                        PLIST:  BR      ZXCALL
                        .IF     NB      <ARGS>
                                .WORD   ARGS
                        .ENDC
                        ZXCALL: JSR     PC,SUBR
                                MOV     (SP)+,R5

                                .ENDM   ;CALL

                        ;-----------------------------------------

                        ; MACRO TO MULTIPLY A NUMBER BY A CONSTANT.
                        ; THE NUMBER IN  DST  IS MULTIPLIED BY THE VALUE OF THE EXPRESSION
                        ; CONST; THE RESULT IS LEFT IN  DST.  A TEMPORARY LOCATION MAY
                        ; BE SPECIFIED AT  WORK  WHICH WILL BE USED IN THE MACRO EXPANSION
                        ; IF NECESSARY.  IF  WORK  IS NOT SPECIFIED & A TEMPORARY LOCATION
                        ; IS NEEDED, A STACK ELEMENT WILL BE ALLOCATED (& SUBSEQUENTLY
                        ; DEALLOCATED) FOR THE PURPOSE.  THE MACRO GENERATES A SERIES OF
                        ; SHIFT & ADD INSTRUCTIONS IN-LINE TO ACCOMPLISH THE MULTIPLICATION.
                        ;
                                .MACRO  MULT    CONST,DST,WORK
                        ZX1     =       0                       ;FLAG: 0-->LEAST SIG 1-BIT NOT TESTED
                                                                ;       YET;  1  >OPPOSITE.
                        ZX2     =       CONST                   ;COPY CONSTANT FOR SHIFTING.

                        .IF     Z       ZX2
                                CLR     DST
                                .MEXIT
                        .ENDC

                        .REPT   16.
                        .IF     NZ      ZX2&1                   ;IF BIT 0 = 1
                        ZX2     -       ZX2/2                   ;SHIFT CONSTANT RIGHT 1 POSITION.
                          .IF   Z       ZX1                     ;IF LEAST SIG 1-BIT
                            .IF NZ      ZX2                     ;IF NOT MOST SIG 1-BIT
                              .IF NB    WORK                    ;IF WORK SPECIFIED
```

```
                        MOV     DST,WORK
                    .IFF                            ;ELSE IF WORK BLANK
                        MOV     DST,-(SP)
                    .ENDC                           ;END [.IF NB WORK]
                .IFF                                ;ELSE IF MOST SIG 1-BIT
                    .MEXIT
                .ENDC                               ;END [.IF NZ ZX2]
ZX1             -       1                           ;INDICATE NO LONGER LEAST SIG 1-BI.
            .IFF                                    ;ELSE IF NOT LEAST SIG 1-BIT
                .IF     NZ      ZX2                 ;IF NOT MOST SIG 1-BIT
                .IF     NB      WORK                ;IF WORK SPECIFIED
                    ADD     DST,WORK
                .IFF                                ;ELSE IF WORK BLANK
                    ADD     DST,@SP
                .ENDC                               ;END [.IF NB WORK]
                .IFF                                ;ELSE IF MOST SIG 1-BIT
                .IF     NB      WORK                ;IF WORK SPECIFIED
                    ADD     WORK,DST
                .IFF                                ;ELSE IF WORK BLANK
                    ADD     (SP)+,DST
                .ENDC                               ;END [.IF NB WORK]
                    .MEXIT
            .ENDC                                   ;END [.IF NZ ZX2]
        .ENDC                                       ;END [.IF Z  ZX1]
    .IFF                                            ;ELSE IF BIT 0 = 0
ZX2         =       ZX2/2                           ;SHIFT CONSTANT RIGHT 1 POSITION.
    .ENDC                                           ;END [.IF NZ ZX2&1]
                ASL     DST
.ENDM                                               ;END [.REPT 16.] LOOP.
                .ENDM   ;MULT

; ---------------------------------------

        .MACRO  HEDING  NAM,VER,EDIT,PATCH
        .TITLE  HEDING
        .IDENT  /VER'EDIT'PATCH/
        .CSECT  HEDING
        .GLOBL  HEDING,HEDLEN

HEDING: .ASCII  /NAM'VER'-'EDIT/
.IF     B       PATCH
        .BYTE   40
.IFF
        .ASCII  /PATCH/
.ENDC
HEDLEN          .-HEDING
        .ENDM   ;HEDING

;MULTIPLY MACRO FOR UNSIGNED MULTIPLY ROUTINE

        .MACRO  MULP    A,B
        MOV     A,-(SP)                 ;SAVE A ON STACK
        MOV     B,R4                    ;SAVE B IN R4
        JSR     R4,MLI                  ;PERFORM MULTIPLICATION
        .WORD   .+2
        MOV     (SP)+,B                 ;PUT PRODUCT INTO B
```

```
                                .ENDM   ;MULP


                        ;BOARD INIT MACRO FOR CLEARING PCL HARDWARE
                        ;BOARD INIT RECEIVER OR TRANSMITTER.

                                .MACRO  BDINIT  DEV
                                .NLIST
                        .IF     IDN     <DEV>,<XMTR>
                                BIS     #B01,@TCR
                        .IFF
                            .IF IDN     <DEV>,<RCVR>
                                BIS     #B01,@RCR
                            .IFF
                                .ERROR                          ;BAD ARGUMENT FOR BDINIT
                            .ENDC
                        .ENDC
                                .LIST
                                .ENDM


000001                  N       =       1                       ;INITIAL ERROR NUMBER

                        ;ERROR MACROS

                                .MACRO  ERROT   P
                                ERADR
                                CALL    ERRMOD,<P,ERADR>        ;UPDATE ENTRIES FOR ERROR P
                        N               N+1
                                .LIST

                                                                ;**** XMTR ERROR P *****


                                .NLIST
                                .ENDM


                                .MACRO  ERROR   P
                                ERADR   =
                                CALL    ERRMOR,<P,ERADR>
                            N   =       N+1
                                .LIST

                                                                ;**** RCVR ERROR P *****


                                .NLIST
                                .ENDM


                        ;REGISTER SAVE MACRO

                                .MACRO  REGSAV
                                JSR     R5,REGSAV
                                .ENDM

                        ;REGISTER RESTORE MACRO
```

```
                                        .MACRO  REGRES
                                        JSR     R5,REGRES
                                        .ENDM
```

```
                        .SBTTL DEFINITIONS AND DEVICE INFO

                        .IDENT  '02'

                        ;COPYRIGHT AUGUST, 1975
                        ;COMPUTER SPECIAL SYSTEMS,
                        ;DIGITAL EQUIPMENT OF CANADA LTD.

                        ;  VARIABLE SYMBOL DEFINITIONS.

                        ;  DEVICE DEFAULT INFORMATION.
177560                  TTDEV   =       177560          ;ADDR OF RCSR FOR TTY.
000060                  TTVCTR          60              ;INPUT VECTOR ADDR FOR TTY.
000004                  TTPRIO  =       4               ;PRIORITY LEVEL FOR TTY.
164200                  PCLTXM  =       164200
164220                  PCLRCV          164220
000174                  RCVECT  =       174
000170                  TXVECT          170
000005                  TXPRIO  =       5
000005                  RCPRIO  =       5

                        ;  QUEUE SIZES.
000024                  TISIZE  =       20.             ;TIQ SIZE.
000400                  TOSIZE  =       256.            ;TOQ SIZE.
000040                  AOSIZE  =       32.             ;ADR QUEUE SIZE


000000                  QELEMS  =       0               ;#ELEMENTS PRESENTLY IN QUEUE.
000002                  QSIZE   =       2               ;#ELEMENTS IN QUEUE SPACE.
000004                  QTOP    =       4               ;ADDR OF 1ST WORD OF QUEUE SPACE
000006                  QBOT    =       6               ;=QTOP+(QSIZE*2)
000010                  QFRONT  =       10              ;ADDR OF FRONT ELEMENT OF QUEUE.
000012                  QBACK   =       12              ;ADDR OF BACK ELEMENT OF QUEUE.

                        ;  REGISTER DEFINITIONS.
000000                  R0      =       %0
000001                  R1      =       %1
000002                  R2      =       %2
000003                  R3      =       %3
000004                  R4      =       %4
000005                  R5      =       %5
000006                  SP      =       %6
000007                  PC      =       %7
177776                  PS      =       177776
177570                  SR      =       177570          ;SWITCH REGISTER.

                        ;SPECIAL CHARACTER DEFINITIONS:
000012                  LF.     =       12              ;CR OR LF TO END LINE.
000015                  CR.     =       15              ;
000017                  CTL.O   =       17              ;^O TO THROW AWAY TTY OUTPUT.
000025                  CTL.U   =       25              ;^U TO DELETE LINE.
000003                  CTL.C   =       3               ;CNTRL-C TO START INPUT.(^C)
000021                  CTL.Q   =       21              ;CNTRL-Q TO RESUME PRINTOUT
000023                  CTL.S   =       23              ;CNTRL-S TO SUSPEND PRINTOUT
000177                  RUBOUT          177             ;RUB OUT TO DELETE CHARACTER.
```

```
                                ;  DEVICE BIT DEFINITIONS.
100000                          ERR     =       100000
004000                          BUSY            4000
000200                          DONE            200
000100                          INTENB  -       100
```

```
                                ;BIT DEFINITIONS:

                100000          B15     -       100000
                040000          B14             40000
                020000          B13             20000
                010000          B12     -       10000
                004000          B11             4000
                002000          B10             2000
                001000          B09             1000
                000400          B08             400
                000200          B07     =       200
                000100          B06             100
                000040          B05             40
                000020          B04             20
                000010          B03             10
                000004          B02             4
                000002          B01     -       2
                000001          B00             1
```

```
                        .SBTTL PCL11 EXERCISER MAIN PROCEDURE

                        .ENABL ABS
                        .       0
000000                  .REPT   128.
000200                  .WORD   .+2,0                   ;TRAP CATCHERS
                        .ENDR

                        .       200
000200                  .       200
000200  000167  001574  JMP     PCLEX                   ;PROGRAM STARTS AT 200
000204  012706  002000  MOV     #STKTOP,SP              ;AND RESTARTS AT 204
000210  000167  002342  JMP     PCREST

                        .       2000
002000                  .
002000                  STKTOP  .

002000                  PCLEX:                          ;****START HERE*****
002000  012706  002000  MOV     #STKTOP,SP              ;SET STACK POINTER TO TOP
002004  016700  014166  MOV     TXDEV,R0                ;PREPARE TO GENERATE TXM ADDRS
002010  010067  014072  MOV     R0,TCR                  ;GENERATE TCR ADDRESS
002014  062700  000002  ADD     #2,R0
002020  010067  014064  MOV     R0,TSR                  ;GENERATE TSR ADDRESS
002024  062700  000002  ADD     #2,R0
002030  010067  014056  MOV     R0,TSDB                 ;GENERATE TSDB ADDRESS
002034  062700  000002  ADD     #2,R0
002040  010067  014050  MOV     R0,TSBC                 ;GENERATE TSBC ADDRESS
002044  062700  000002  ADD     #2,R0
002050  010067  014042  MOV     R0,TSBA                 ;GENERATE TSBA ADDRESS
002054  062700  000002  ADD     #2,R0
002060  010067  014034  MOV     R0,TMMR                 ;GENERATE TMMR ADDRESS
002064  005200          INC     R0
002066  010067  014030  MOV     R0,TMMRH                ;AND TMMR HIGH BYTE
002072  005200          INC     R0
002074  010067  014024  MOV     R0,TSCRC                ;GENERATE TSCRC ADDRESS
002100  016767  014074  014034  MOV     TXVEC,TXMVEC    ;GENERATE TXMVEC ADDRESS

002106  016700  014070  MOV     RCDEV,R0                ;PREPARE TO GENERATE RCVR ADDRS
002112  010067  014010  MOV     R0,RCR                  ;GENERATE RCR ADDRESS
002116  062700  000002  ADD     #2,R0
002122  010067  014002  MOV     R0,RSR                  ;GENERATE RSR ADDRESS
002126  062700  000002  ADD     #2,R0
002132  010067  013774  MOV     R0,RDDB                 ;GENERATE RDDB ADDRESS
002136  062700  000002  ADD     #2,R0
002142  010067  013766  MOV     R0,RDBC                 ;GENERATE RDBC ADDRESS
002146  062700  000002  ADD     #2,R0
002152  010067  013760  MOV     R0,RDBA                 ;GENERATE RDBA ADDRESS
002156  062700  000004  ADD     #4,R0
002162  010067  013752  MOV     R0,RDCRC                ;GENERATE RDCRC ADDRESS
002166  016767  014012  013750  MOV     RCVEC,RCVVEC    ;GENERATE RCVVEC ADDR
```

```
  420  002174                                  PCRST:                                  ;*****RESTART HERE ******
  421  002174  012706  002000                         MOV     #STKTOP,SP              ;RESET STACK POINTER
  422  002200  012737  003326  000004                 MOV     #ERTRAP,@#4             ;SET UP VECTOR FOR ADDRESS ERROR
  423  002206  012737  000340  000006                 MOV     #340,@#6
  424  002214  105067  015430                         CLRB    REQINP                  ;CLR INPUT REQUEST
  425  002220  105067  015425                         CLRB    CMDENT                  ;CLR COMMAND ENTERED FLAG
  426  002224  005067  015460                         CLR     PCLGO                   ;CLR XMTR GO FLAG
  427                                          .IRP    LC      <TI,TO>                 ;INITIALIZE IO QUEUES TO EMPT
  428                                          .LIST
  429                                                  CLR     LC'Q
  430                                                  MOV     LC'Q+QTOP,LC'Q+QFRONT
  431                                                  MOV     LC'Q+QTOP,LC'Q+QBACK
  432                                          .NLIST
  433                                          .ENDM
(1)  002230  005067  014314                         CLR     TIQ
(1)  002234  016767  014314  014316                 MOV     TIQ+QTOP,TIQ+QFRONT
(1)  002242  016767  014306  014312                 MOV     TIQ+QTOP,TIQ+QBACK
(1)  002250  005067  014360                         CLR     TOQ
(1)  002254  016767  014360  014362                 MOV     TOQ+QTOP,TOQ+QFRONT
(1)  002262  016767  014352  014356                 MOV     TOQ+QTOP,TOQ+QBACK
  434  002270  012737  000340  177776                MOV     #340,@#PS               ;DISABLE INTERRUPTS
  435  002276  012777  007566  013636                MOV     #XMTINT,@TXMVEC         ;;;SET UP XMTR INTR VECTOR
  436  002304  012777  011112  013632                MOV     #RCVINT,@RCVVEC         ;;;SET UP RCVR INTR VECTOR
  437  002312  012777  015652  013636                MOV     #TTIINT,@TTVECT         ;;;SET UP TTY INTR VECTOR
  438  002320  016700  013616                        MOV     TXMVEC,R0
  439  002324  012760  000240  000002                MOV     #TXPRIO*32.,2(R0)       ;;;SET TXM PRIORITY
  440  002332  016700  013606                        MOV     RCVVEC,R0
  441  002336  012760  000240  000002                MOV     #RCPRIO*32.,2(R0)       ;;;SET RCVR PRIORITY
  442  002344  016700  013606                        MOV     TTVECT,R0
  443  002350  012760  000200  000002                MOV     #TTPRIO*32.,2(R0)       ;;;SET TTY PRIORITY
  444  002356                                       CALL    PNCRLF
  445  002372                                       CALL    PNCRLF
  446  002406                                       CALL    PNTLIN,<PCLEXM>         ;;;PRINT TITLE MESSAGE
  447  002424                                       CALL    PNCRLF                  ;;;<CR>&<LF>
  448  002440                                       CALL    PNTLIN,<RSTMSG>         ;;;ENQUEUE RESTART ADDR MSG
  449  002456                                       CALL    PRINIT                  ;;;INITIALIZE TRANSMITTER
  450  002472  012737  002536  000004  PRST:         MOV     #$TR,@#4                ;;;SET UP TO TEST FOR CLOCK
  451  002500  005067  015154                        CLR     KWFLG                   ;;;CLEAR KW11 FLAG
  452  002504  005777  013450                        TST     @LCS                    ;;;ANY CLOCK?
  453  002510  012767  177777  015142                MOV     #-1,KWFLG               ;;;YES, SET KW11 FLAG
  454  002516  012737  012224  000100                MOV     #CLKINT,@#100           ;;;SET UP CLK VECTOR
  455  002524  012737  000340  000102                MOV     #340,@#102
  456  002532  000167  000020                        JMP     PCPEST                  ;;;AND CONTINUE
  457  002536  022626                  $TR:          CMP     (SP)+,(SP)+             ;;;NO CLOCK, CLEAR STACK
  458  002540  005067  015114                        CLR     KWFLG                   ;;;CLR KW11 FLAG
  459  002544  012737  000102  000100                MOV     #102,@#100              ;;;SET UP TO TRAP HALT
  460  002552  005037  000102                        CLR     @#102
  461  002556  005037  177776          PCREST:       CLR     @#PS                    ;;;ALLOW INTERRUPTS
  462  002562  012737  002670  000004                MOV     #TRAP4,@#4              ;CHANGE TRAP VECTOR FOR ERROR
  463  002570  052767  100000  015114                BIS     #B15,RCVEV              ;SET RCVR EVENT FLAG
  464  002576  052777  000100  013342                BIS     #B06,@TTRCSR            ;SET TTY KBD INTR ENAB
  465  002604  005067  015156                        CLR     ESCFLG                  ;CLEAR CNTRL-C FLAG
  466  002610  005067  015052                        CLR     QLDEV                   ;CLR QUEUE LOAD EVENT FLAG
  467  002614  005067  015130                        CLR     RJCTF                   ;CLEAR REJECT FLAG
  468  002620  005067  015126                        CLR     TRNKF                   ;CLEAR TRUNCATE FLAG
  469  002624  005067  015064                        CLR     SPCEV                   ;CLEAR MST DWN EVENT
```

```
                                                    CLR     XSPCEV                  ;CLR NOW MST EVENT
                                            RTRYA:  CALL    PNCRLF                  ;ENQUEUE CR, & LF
                                                    CALL    PRESC                   ;ENTER COMMAND MODE'
                                                    JMP     PCLOOP                  ;GO TO MAIN LOOP


                                            TRAP4:  MOV     (SP),SUMSV              ;SAVE TRAP ADDRESS
                                                    SUB     #2,SUMSV                ;ALIGN IT -2
                                                    MOV     #TRP4AD,R0              ;SHOW OCT CONV RTN RIGHT ADDRESS
                                                    MOV     SUMSV,R1                ; FOR ASCII CHARS.
                                                    JSR     PC,OCTJSP
                                                    CALL    PNTLIN,<TRPDMG>         ;PRINT TRAP MESSAGE
                                                    MOV     #STKTOP,SP
                                                    CLR     @#PS                    ;LOWER PRIORITY
                                                    JMP     RTRYA
```

```
                                        .SBTTL  MAIN LOOP

                                PCLOOP: TST     TIQ             ;IS TTY INPUT QUEUE EMPTY?
                                        BEQ     NXT0            ;YES, TEST ANOTHER FLAG
                                        CALL    TTINP           ;NO, PROCESS A CHARACTER
                                NXT0:   TSTB    CMDENT          ;HAS A COMMAND BEEN ENTERED?
                                        BPL     NXT1            ;NO, TEST ANOTHER FLAG
                                        CALL    COMENT          ;YES, PROCESS COMMAND
                                NXT1:   TST     TOQ             ;IS TTY OUTPUT QUEUE EMPTY?
                                        BEQ     NXT2            ;YES, TEST ANOTHER FLAG
                                        CALL    TTOUT           ;NO,OUTPUT A CHAR IF DEV RDY
                                NXT2:   TST     QLDEV           ;IS ADDR QUEUE EMPTY?
                                        BPL     NXT3            ;NO, TEST ANOTHER FLAG
                                        CALL    ADQLD           ;YES, LOAD ADDR QUEUE
                                NXT3:   TST     DATGEV          ;IS DATA GEN FLAG SET?
                                        BPL     NXT4            ;NO, TEST ANOTHER FLAG
                                        CALL    DATGEN          ;YES,GENERATE NEW RANDOM DATA
                                NXT4:   TST     TXMEV           ;IS XMTR EVENT FLAG SET?
                                        BPL     NXT5            ;NO, TEST ANOTHER FLAG
                                        CALL    TXMIT           ;YES, ENTER XMIT MODULE
                                NXT5:   TST     RCVEV           ;IS RCVR EVENT FLAG SET?
                                        BPL     NXT6            ;NO, TEST ANOTHER FLAG
                                        CALL    RECV            ;YES, ENTER RECVR MODULE
                                NXT6:   TST     SPCEV           ;IS SPECIAL EVENT FLAG SET?
                                        BPL     NXT7            ;NO, TEST ANOTHER FLAG
                                        CALL    SPEC            ;YES, HANDLE SPECIAL EVENT
                                NXT7:   TST     STSEV           ;IS STATUS EVENT FLAG SET?
                                        BPL     NXT8            ;NO, TEST ANOTHER FLAG
                                        CALL    STATUS          ;YES,OUTPUT STATUS
                                NXT8:   TST     SUMEV           ;IS SUMMARY EVENT FLAG SET?
                                        BPL     NXT9            ;NO, TEST ANOTHER FLAG
                                        CALL    SUMRY           ;YES, OUTPUT ERROR SUMMARY
                                NXT9:   TST     XSPCEV          ;IS EXTRA-SPECIAL EVENT SET?
                                        BPL     NXT10
                                        CALL    XSPEC           ;YES, HANDLE NOW MASTER.
                                NXT10:  TST     TEREV           ;IS XMTR ERROR EVENT SET?
                                        BPL     NXT11           ;NO, TEST ANOTHER FLAG
                                        CALL    TEROS           ;YES OUTPUT XMTR ERROR TABLE
                                NXT11:  TST     REREV           ;IS RCVR ERROR EVENT SET?
                                        BPL     NXT12           ;NO,
                                        CALL    REROS           ;YES, OUTPUT RCVR ERROR TABLE
                                NXT12:  JMP     PCLOOP          ;STAY IN MAIN LOOP

                                ;TRAP TO 4 HANDLER


                                ERTRAP: REGSAV
                                        CALL    PNTLIN,<ERTMSG> ;;;PRINT TRAP MESSAGE
                                        CLR     RCVEV           ;;;CLEAR RCVER EVENT FLAG
                                        CLR     PCLGO           ;;;CLR PCL GO FLAG
                                        BIS     #B06,@TTRCSR    ;;;SET TTY KBD INTR ENAB
                                        CLR     @#PS            ;;;DROP CP PRIORITY
                                        REGRES
                                        MOV     #STKTOP,SP      ;FIX STACK
                                        JMP     RTRYA           ;ENTER COMMAND MODE
```

```
                        .SBTTL COMMAND PROCESSORS:
                        .SBTTL  COMMAND PROC. FOR SILO (LOAD)


                        ;  LOAD OR CLEAR THE TRANSMITTER ADDRESS (HARDWARE) SILO.
                        ; IF THERE ARE NO ARGUMENTS, CLEAR THE ADDRESS SILO AND SET
                        ; AUTO ADDRESS.   IF THERE ARE ARGUMENTS, LOAD THE TOTAL NUMBER
                        ; OF ARGUMENTS (INCLUDING PAD VALUES) INTO THE SILO AS MANY TIMES
                        ; AS THE TOTAL WILL GO INTO 50. LOCATIONS.
                        ;
                        ;  IF ANY 2 SEQUENTIAL ARGUMENTS ARE THE SAME, SEPARATE THEM WITH
                        ; A PAD VALUE OF '0'.   IF THE FIRST ARGUMENT IS THE SAME AS THE LAST
                        ; ARGUMENT, INSERT A PAD VALUE OF '0' AFTER THE LAST ARGUMENT.
                        ;
                        ;  ARGUMENTS HIGHER THAN 37 (OCTAL) WILL NOT BE ACCEPTED
                        ;  ARGUMENTS LOWER THAN 1 WILL NOT BE ACCEPTED.
                        ;
                        ;  ARGUMENTS MUST BE NUMERIC. DECIMAL ARGUMENTS MUST BE FOLLOWED BY
                        ; A DECIMAL POINT. (18.)
                                PROC    CPSILO
003406
(1)
(1)  003406                     CPSILO:                                 ;**ENTRY POINT**
003406  012701  020006                  MOV     #ADSILO,R1              ;SET R1 TO POINT TO SILO BUFFER
003412  016602  000002                  MOV     2(SP),R2
003416  005302                          DEC     R2                      ;GET # OF ARGS. INTO R2
003420  001002                          BNE     11$                     ;O.K IF THERE ARE SOME
003422  000167  000446                  JMP     CLEAV                   ;OTHERWISE, EXIT
003426  010267  014276          11$:    MOV     R2,PADFLG               ;SAVE COUNT FOR LATER USE.
003432  020227  000062                  CMP     R2,#50.                 ;ARE THERE MORE THAN 50 OBJECTS?
003436  101402                          BLOS    1$                      ;NO, CONTINUE
003440  000167  000422                  JMP     SYNRTM                  ;YES, ERROR
003444  010267  014320          1$:     MOV     R2,OBJCNT               ;SAVE OBJECT COUNT
003450  010200                          MOV     R2,R0                   ;TURN LOOK-UP AROUND
003452  005300                          DEC     R0
003454                                  MULT    6,R0,R3
003464  060600                          ADD     SP,R0
003466  062700  000004                  ADD     #4,R0
003472  016067  000004  014220          MOV     4(R0),FIRST             ;SAVE FIRST ITEM
003500  012767  000000  014214          MOV     #0,NEXT                 ;SAVE CURRENT ITEM
003506  021027  000002          CPMLP:  CMP     (R0),#2                 ;IS IT CLASS 2?
003512  001402                          BEQ     2$                      ;YES, O.K.
003514  000167  000346                  JMP     SYNRTM                  ;NO, ERROR
003520  026067  000004  014174  2$:     CMP     4(R0),NEXT              ;IS THIS ITEM SAME AS LAST?
003526  001004                          BNE     3$                      ;NO, O.K.
003530  112721  000000                  MOVB    #0,(R1)+                ;YES, INSERT A PAD VALUE
003534  005267  014230                  INC     OBJCNT                  ;KEEP OBJECT COUNT UP TO DATE
003540  116011  000004          3$:     MOVB    4(R0),(R1)              ;PUT OBJECT INTO BUFFER
003544  001002                          BNE     4$                      ;ERROR IF OBJECT IS 0
003546  000167  000314                  JMP     SYNRTM
003552  122711  000037          4$:     CMPB    #37,(R1)                ;ERROR IF IT WAS > 37
003556  103002                          BHIS    5$
003560  000167  000302                  JMP     SYNRTM
003564  112167  014132          5$:     MOVB    (R1)+,NEXT              ;SAVE REAL ITEM
003570  062700  177772                  ADD     #-6,R0                  ;SET UP TO GET NEXT OBJECT
003574  005302                          DEC     R2                      ;ARE WE DONE LOADING BUFF?
003576  001343                          BNE     CPMLP                   ;NO, KEEP GOING
```

```
 597  003600  026767  014116  014112          CMP     NEXT,FIRST              ;IS LAST OBJECT = FIRST?
 598  003606  001004                          BNE     6$                      ;NO, O.K.
 599  003610  112721  000000                  MOVB    #0,(R1)+                ;YES, INSERT A PAD VALUE
 600  003614  005267  014150                  INC     OBJCNT                  ;AND KEEP OBJECT COUNT UP TO DATE
 601  003620  026727  014144  000062  6$:     CMP     OBJCNT,#50.             ;OBJECT COUNT GOTTEN TOO BIG?
 602  003626  101402                          BLOS    7$                      ;NO, O.K.
 603  003630  000167  000232                  JMP     SYNRTM                  ;YES, ERROR
 604  003634  152777  000060  012260  7$:     BISB    #B05+B04,@TMMRH         ;PREPARE TO LOAD ADDR SILO
 605  003642  012700  000062                  MOV     #50.,R0                 ;HOLD SILO SIZE
 606  003646  026767  014116  014054          CMP     OBJCNT,PADFLG           ;IS OBJECT COUNT DIFFERENT THAN START
 607  003654  001002                          BNE     SILLD                   ;YES, LEAVE SOMETHING IN PADFLG (SET)
 608  003656  005067  014046          8$:     CLR     PADFLG                  ;NO, CLEAR PAD FLAG
 609  003652  016702  014102          SILLD:  MOV     OBJCNT,R2               ;GET NO. OF OBJECTS
 610  003666  012701  020006                  MOV     #ADSILO,R1              ;GET  OBJECT BUFFER
 611  003672  112177  012222          SLOLP:  MOVB    (R1)+,@TMMR             ;GET AN OBJECT INTO SILO
 612  003676  005302                          DEC     R2                      ;LOADED ALL OBJECTS?
 613  003700  001374                          BNE     SLOLP                   ;NO, CONTINUE
 614  003702  166700  014062                  SUB     OBJCNT,R0               ;YES, IS THAT ALL THAT'LL FIT?
 615  003706  020067  014056                  CMP     R0,OBJCNT
 616  003712  002363                          BGE     SILLD                   ;IF NOT LOAD THEM AGAIN
 617  003714  132777  000001  012200          BITB    #1,@TMMRH               ;SEE IF I AM MASTER
 618  003722  001037                          BNE     CPSLV                   ;IF SO TURN ON SILO AND EXIT
 619  003724                                  CALL    PNTLIN,<MSTMG1>         ;PRINT "THIS UNIT IS NOT MASTER
 620  003742                                  CALL    PNTLIN,<MSTMG2>         ; BUT HAS BEEN MADE SECONDARY
 621  003760                                  CALL    PNTLIN,<MSTMG3>         ; THE SILO YOU HAVE JUST FILLED
 622  003776                                  CALL    PNTLIN,<MSTMG4>         ; WILL BE USED IF YOU CLEAR THE
 623  004014  152777  000002  012100          BISB    #B01,@TMMRH             ; THE CURRENT MASTER".
 624  004022  005767  013702          CPSLV:  TST     PADFLG                  ;HAS SILO BEEN PADDED?
 625  004026  001407                          BEQ     9$                      ;NO, CARRY ON.
 626  004030                                  CALL    PNTLIN,<MSTMG5>         ;YES, TELL OPERATOR.
 627  004046  012767  177777  013712  9$:     MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW "PCL>".
 628  004054  142777  000020  012040          BICB    #B04,@TMMRH             ;CLR AUTO ADDR
 629  004062  000241                          CLC                             ;CLEAR "C" FOR NON ERROR RETURN
 630  004064                          CPSEX:  RETURN
 631
 632  004066  000261                  SYNRTM: SEC                             ;SET "C" BIT FOR SYNTAX ERROR
 633  004070  000167  177770                  JMP     CPSEX                   ;AND RETURN
 634
 635
 636  004074  152777  000060  012020  CLEAV:  BISB    #B05+B04,@TMMRH         ;RESET AUTO ADDR & CLR SILO
 637  004102  012767  177777  013656          MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW "PCL>".
 638  004110  000241                          CLC                             ;CLEAR "C" IN CASE IT WAS SET
 639  004112  000167  177746                  JMP     CPSEX                   ;RETURN
```

```
                        .SBTTL  COMMAND PROC. FOR MASTER, SECONDARY AND R.I.B.

                        ;PROCESSOR FOR 'MASTER SET (OR CLEAR)'' COMMAND
                        ; SET MASTER, OR CLEAR MASTER

004116                          PROC    CPMAST
(1)
(1)     004116          CPMAST:                                 ;**ENTRY POINT**
004116  022766  000002  000002          CMP     #2,2(SP)        ;GET # OF ARGUMENTS
004124  001402                          BEQ     CPMOK           ;OKAY IF 2
004126  000261                          SEC
004130  000423                          BR      CPMRET          ;OTHERWISE, SYNTAX ERROR
004132  010603          CPMOK:  MOV     SP,R3                   ;GET 2ND WORD OF COMMAND
004134  062703  000004                  ADD     #4,R3
004140                          KEYWD   R3,#SCTBL               ;PROCESS IT
004152  103446                          BCS     KWDERT          ;SYNTAX ERROR IF 'C' SET
004154  117702  011742                  MOVB    @TMMRH,R2       ;GET OLD TMMR
004160  142702  000001                  BICB    #1,R2           ;REMOVE OLD STATE OF MASTER
004164  052602                          BIS     (SP)+,R2        ;SET NEW STATE OF MASTER
004166  110277  011730                  MOVB    R2,@TMMRH       ;LOAD NEW TMMR
004172  012767  177777  013566          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>'.
004200          CPMRET: RETURN                                  ;EXIT


                        ;PROCESSOR FOR ''SECONDARY SET (OR CLEAR)'' COMMAND
                        ; SET SECONDARY, OR CLEAR SECONDARY

004202                          PROC    CPSEC
(1)
(1)     004202          CPSEC:                                  ;**ENTRY POINT**
004202  022766  000002  000002          CMP     #2,2(SP)        ;ARE THERE 2 ARGUMENTS?
004210  001402                          BEQ     CPSOK           ;IF YES, PROCEED
004212  000261                          SEC
004214  000424                          BR      CPSRET          ;OTHERWISE, SYNTAX ERROR
004216  010603          CPSOK:  MOV     SP,R3                   ;GET 2ND WORD OF COMMAND
004220  062703  000004                  ADD     #4,R3
004224                          KEYWD   R3,#SCTBL               ;PROCESS IT
004236  103414                          BCS     KWDERT          ;SYNTAX ERROR IF 'C' SET
004240  117702  011656                  MOVB    @TMMRH,R2       ;GET OLD TMMR
004244  142702  000002                  BICB    #2,R2           ;REMOVE OLD STATE OF SECONDARY
004250  006316                          ASL     (SP)
004252  052602                          BIS     (SP)+,R2        ;SET NEW STATE OF SECONDARY
004254  110277  011642                  MOVB    R2,@TMMRH       ;LOAD NEW TMMR
004260  012767  177777  013500          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>'.
004266          CPSRET: RETURN                                  ;EXIT


C04270  032600          KWDERT: BIT     (SP)+,R0                ;POP BAD WORD OFF STACK
004272  000775                  BR      CPSRET                  ;EXIT


                        ;PROCESSOR FOR 'RIB SET'' OR 'RIB CLEAR''
                        ; SET 'RIB'' BIT IN XMTR COMMAND WORD LOCATION 'TXMST'', OR CLEAR IT

004274                          PROC    CPRIB
(1)
```

```
(ZPLACO PCL11 EXERCISER V02C     MACY11 30A(1052)  25-JUN-79  08:52  PAGE 8-1
(ZPLAC.P11    08-JUN-79 15:55          COMMAND PROC. FOR MASTER, SECONDARY AND R.I.B.                SEQ 0057

   (1)  004274                        CPRIB:                          ;**ENTRY POINT**
 692    004274  022766  000002  000002          CMP     #2,2(SP)        ;2 ARGUMENTS?
 693    004302  001402                          BEQ     CPROK           ;YES, O.K.
 694    004304  000261                          SEC
 695    004306  000426                          BR      CPRRET          ;OTHERWISE, ERROR RETURN
 696    004310  010603                CPROK:    MOV     SP,R3           ;GET 2ND WORD OF COMMAND
 697    004312  062703  000004                  ADD     #4,R3
 698    004316                                  KEYWD   R3,#SCTBL       ;PROCESS IT
 699    004330  103757                          BCS     KWDERT          ;IF ERROR, SHOW IT.
 700    004332  016702  003030                  MOV     TXMST,R2        ;GET OLD STATE OF TCR
 701    004336  042702  100000                  BIC     #B15,R2         ;CLEAR IMAGE OF RIB
 702    004342  000241                          CLC
 703    004344  006016                          ROR     (SP)            ;GET NEW STATE OF RIB
 704    004346  006016                          ROR     (SP)
 705    004350  052602                          BIS     (SP)+,R2        ;SET IT IN IMAGE
 706    004352  010267  003010                  MOV     R2,TXMST                ;LOAD NEW STATE OF TCR
 707    004356  012767  177777  013402          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>'
 708    004364                     CPRRET: RETURN                          ;BACK TO CALLER
```

```
                        .SBTTL  COMMAND PROC. FOR RANGE


                        ; SET RANGE OF RECEIVER ADDRESSES THAT THIS TRANSMITTER SHOULD TALK TO.
                        ;
                        ;  ACCEPT A LOWER AND UPPER RECEIVER ADDRESS (IN THAT ORDER).
                        ;  GENERATE A LIST (SET ACTIVE FLAGS) OF RCVR ADDRESSES FROM LOW ADDRESS
                        ;    TO HIGH ADDRESS INCLUSIVE.


004366                          PROC    CPRANG
(1)
(1)  004366                     CPRANG:                         ;**ENTRY POINT**
004366  016600  000002                  MOV     2(SP),R0        ;GET # OF OBJECTS
004372  022700  000003                  CMP     #3,R0           ;3 OBJECTS?
004376  001047                          BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004400  016600  000004                  MOV     4(SP),R0        ;GET CLASS OF FIRST OBJECT
004404  022700  000002                  CMP     #2,R0           ;IS IT CLASS 2?
004410  001042                          BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004412  016601  000010                  MOV     10(SP),R1       ;GET "TO" ARGUMENT INTO R1
004416  016600  000012                  MOV     12(SP),R0       ;GET CLASS OF SECOND OBJECT
004422  022700  000002                  CMP     #2,R0           ;IS IT CLASS 2
004426  001033                          BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004430  016602  000016                  MOV     16(SP),R2       ;GET "FROM" ARGUMENT INTO R2
004434  020102                          CMP     R1,R2           ;IS "FROM" LOWER THAN "TO"?
004436  101427                          BLOS    SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004440  022701  000037                  CMP     #37,R1          ;IS "TO">37?
004444  103424                          BLO     SYNRTN          ;IF SO, INDICATE SYNTAX ERROR
004446  010200                          MOV     R2,R0           ;FIND "FROM" ENTRY
004450  001422                          BEQ     SYNRTN          ;IF "FROM" 0, SYNTAX ERROR
004452  006300                          ASL     R0
004454  006300                          ASL     R0
004456  010003                          MOV     R0,R3
004460  006300                          ASL     R0
004462  060300                          ADD     R3,R0
004464  062700  020074                  ADD     #RADB0,R0       ;R0 CONTAINS "FROM" ADDRESS
004470  012710  177777          CPRFIL: MOV     #-1,(R0)        ;SET RCV ADDR ACTIVE FLAG.
004474  062700  000014                  ADD     #14,R0          ;UPDATE TABLE POINTER
004500  005202                          INC     R2              ;INCREMENT "FROM" ADDRESS
004502  020201                          CMP     R2,R1           ;  UNTIL EQUAL TO "TO" ADDRESS
004504  003771                          BLE     CPRFIL          ;EXIT WHEN COMPLETE.
004506  012767  177777  013252          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW "PCL>"
004514                          CPRRTN: RETURN

004516  000261                  SYNRTN: SEC                     ;SET "C" BIT TO INDICATE SYNTAX ERROR
004520  000167  177770                  JMP     CPRRTN
```

```
                                        .SBTTL  COMMAND PROC. FOR ADD AND DELETE

                                ; ADD A RECEIVER ADDRESS TO THE LIST (SET ACTIVE FLAG) OF RCVR ADDRESSES
                                ;
                                ;  ACCEPT ARGUMENTS (AT LEAST 1) CHECK THAT THEY FALL IN A RANGE FROM
                                ;  1 TO 37; THEN SET THE CORRESPONDING ACTIVE FLAGS IN THE STAT
                                ;  TABLE.

004524                                  PROC    CPADD
(1)
(1)  004524                     CPADD:                          ;**ENTRY POINT**
764  004524  016602  000002             MOV     2(SP),R2       ;GET # OF ARGUMENTS INTO R2
765  004530  005302                     DEC     R2             ;IGNORE KEYWORD OBJECT
766  004532  010600                     MOV     SP,R0          ;GENERATE ADDRESS OF ARGUMENTS
767  004534  062700  000004             ADD     #4,R0          ;POINT R0 AT 1ST OBJECT CLASS
768  004540  021027  000002     CPALP:  CMP     (R0),#2        ;IS CLASS = 2?
769  004544  001027                     BNE     SYNTRA         ;NO, INDICATE SYNTAX ERROR
770  004546  016001  000004             MOV     4(R0),R1       ;GET OBJECT INTO R1
771  004552  001424                     BEQ     SYNTRA         ;IF IT'S 0, SYNTAX ERROR
772  004554  022701  000037             CMP     #37,R1
773  004560  103421                     BLO     SYNTRA         ;IF IT'S >37, SYNTAX ERROR
774  004562  006301                     ASL     R1             ;FIND THE TABLE ELEMENT
775  004564  006301                     ASL     R1
776  004566  010103                     MOV     R1,R3
777  004570  006301                     ASL     R1
778  004572  060301                     ADD     R3,R1
779  004574  062701  020074             ADD     #RADB0,R1
780  004600  012711  177777             MOV     #-1,(R1)       ;SET ACTIVE FLAG FOR THIS TABLE ENTRY
781  004604  062700  000006             ADD     #6,R0          ;SET UP FOR NEXT ARGUMENT
782  004610  005302                     DEC     R2             ;ARE WE DONE?
783  004612  001352                     BNE     CPALP          ;NO, CONTINUE.
784  004614  012767  177777  013144     MOV     #-1,ESCFLG     ;FLAG COMENT TO ISSUE NEW 'PCL>'
785  004622                  CPARTN: RETURN
786  004624  000261          SYNTRA: SEC                        ;SET "C" BIT FOR SYNTAX ERROR
787  004626  000167  177770             JMP     CPARTN
788
789                             ;DELETE RECEIVER ADDRESSES AND ASSOCIATED "ATTEMPTS" AND "SUCCESSES".
790
791  004632                             PROC    CPDEL
(1)
(1)  004632                     CPDEL:                          ;**ENTRY POINT**
792  004632  016602  000002             MOV     2(SP),R2       ;GET # OF ARGUMENTS INTO R2
793  004636  005302                     DEC     R2             ;IGNORE KEYWORD OBJECT
794  004640  010600                     MOV     SP,R0          ;GENERATE ADDRESS OF ARGUMENTS
795  004642  062700  000004             ADD     #4,R0          ;POINT R0 AT 1ST OBJECTS CLASS
796  004646  021027  000002     CPDLP:  CMP     (R0),#2        ;IS CLASS =2
797  004652  001034                     BNE     SYNTRD         ;NO,INDICATE SYNTAX ERROR
798  004654  016001  000004             MOV     4(R0),R1       ;GET OBJECT INTO R1
799  004660  001431                     BEQ     SYNTRD         ;IF IT'S 0, SYNTAX ERROR
800  004662  022701  000037             CMP     #37,R1
801  004666  103426                     BLO     SYNTRD         ;IF IT'S >37, SYNTAX ERROR
802  004670  006301                     ASL     R1             ;FIND TABLE ELEMENT
803  004672  006301                     ASL     R1
804  004674  010103                     MOV     R1,R3
805  004676  006301                     ASL     R1
806  004700  060301                     ADD     R3,R1
```

```
 807  004702  062701  020074            ADD     #RADB0,R1
 808  004706  005011                    CLR     (R1)            ;CLEAR ACTIVE FLAG AT TABLE LOCATION
 809  004710  062701  000004            ADD     #4,R1           ;AND OTHER ENTRIES...
 810  004714  005021                    CLR     (R1)+
 811  004716  005021                    CLR     (R1)+
 812  004720  005021                    CLR     (R1)+
 813  004722  005011                    CLR     (R1)
 814  004724  062700  000006            ADD     #6,R0           ;SET UP FOR NEXT ARGUMENT
 815  004730  005302                    DEC     R2              ;ARE WE DONE?
 816  004732  001345                    BNE     CPDLP           ;NO, CONTINUE
 817  004734  012767  177777  013024    MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>'
 818  004742                    CPDRTN: RETURN
 819  004744  000261            SYNTRD: SEC                     ;SET "C" BIT FOR SYNTAX ERROR
 820  004746  000167  177770            JMP     CPDRTN
```

```
                        .SBTTL  COMMAND PROC. FOR CLEAR, STATUS, AND CONTINUE


                        ; CLEAR THE ENTIRE LIST OF RECEIVER ADDRESSES (CLEAR ACTIVE FLAGS)
                        ;
                        ; TO USE THIS ROUTINE  BY A CALL MACRO, ENTER AT PRCLR AS FOLLOWS :
                        ;
                        ;       CALL    PRCLR                   ;CLEAR STATUS TABLE
                        ;       PROC    CPCLR
004752                  (1)
004752                  (1)     CPCLR:                          ;**ENTRY POINT**
                        
004752  016602  000002          MOV     2(SP),R2                ;GET NUMBER OF OBJECTS INTO R2
004756  022702  000001          CMP     #1,R2                   ;ARE THERE ANY ARGUMENTS?
004762  001031                  BNE     SYNTRC                  ;IF SO, INDICATE SYNTAX ERROR
004764  012767  177777  012774  MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>''
004772                          PROC    PRCLR
004772                  (1)
004772                  (1)     PRCLR:                          ;**ENTRY POINT**
004772  012701  000037          MOV     #37,R1                  ;SAVE COUNT OF TABLE ELEMENTS
004776  012700  020110          MOV     #RADB,R0                ;SET UP TO CLR ACTIVE FLAGS
005002  005010          CPCLRC: CLR     (R0)                    ;CLEAR ALL ELEMENTS OF ENTRY
005004  062700  000004          ADD     #4,R0
005010  005020                  CLR     (R0)+
005012  005020                  CLR     (R0)+
005014  005020                  CLR     (R0)+
005016  005020                  CLR     (R0)+
005020  005301                  DEC     R1                      ;DONE?
005022  001367                  BNE     CPCLRC                  ;IF NOT, CONTINUE
005024  005067  011404          CLR     AOQ                     ;INITIALIZE ADDR QUEUE TO EMPTY
005030  016767  011404  011406  MOV     AOQ+QTOP,AOQ+QFRONT
005036  016767  011376  011402  MOV     AOQ+QTOP,AOQ+QBACK
005044          CPCLRT: RETURN                                  ;AND LEAVE
005046  000261          SEC                                     ;SET "C" BIT FOR SYNTAX ERROR
005050  000167  177770          JMP     CPCLRT


                        ;  SET THE STATUS EVENT FLAG SO THAT THE STATUS TABLE WILL BE PRINTED.

005054                          PROC    CPSTAT
005054                  (1)
005054                  (1)     CPSTAT:                         ;**ENTRY POINT**
005054  022766  000001  000002  CMP     #1,2(SP)                ;ARE THERE ANY ARGUMENTS?
005062  001011                  BNE     SYNTRS                  ;IF SO, INDICATE SYNTAX ERROR
005064  052767  100000  012604  BIS     #B15,STSEV              ;SET STATUS EVENT FLAG
005072  005067  012602          CLR     STPNTR                  ;0 STPNTR INDICATES HEADER FIRST
005076  042777  000100  011002  BIC     #B06,@TCR               ;CLR TXM INTERRUPT ENABLE
005104          CPSRTN: RETURN
005106  000261  SYNTRS: SEC                                     ;SET "C" BIT FOR SYNTAX ERROR
005110  000167  177770          JMP     CPSRTN


                        ; CONTINUE EXERCISING. DO NOT AFFECT THE STATUS TABLE
```

```
                        ;  DO NOT AFFECT THE ERROR TABLE
                        ;  DO NOT AFFECT THE RCVR ADDRESS QUEUE
                        ;  SIMPLY SET PCLGO
                        ;
                        ;
                        ;
005114                          PROC    CPCNT
(1)
(1)  005114             CPCNT:                          ;**ENTRY POINT**
005114  022766  000001  000002          CMP     #1,2(SP)        ;ARE THERE ANY ARGUMENTS?
005122  001027                          BNE     SYNTCN          ;IF SO, INDICATE SYNTAX ERROR
005124  012767  100000  012556          MOV     #B15,PCLGO      ;SET PCL GO FLAG
005132                                  CALL    PNCRLF          ;PRINT CR & LF
005146                                  CALL    PNTLIN,<EXCNT>  ;PRINT 'EXERCISER CONTINUING'
005164  005767  012470                  TST     KWFLG           ;GOT A CLOCK?
005170  001403                          BEQ     CPCNRT          ;NO
005172  012777  000100  010760          MOV     #B06,@LCS       ;RE-ENABLE CLOCK.
005200                          CPCNRT: RETURN
005202  000261                  SYNTCN: SEC                     ;SET "C" BIT FOR SYNTAX ERROR
005204  000167  177770                  JMP     CPCNRT
```

```
                        .SBTTL  COMMAND PROC. FOR INIT, SUMMARY, AND GO

                        ;TO USE CALL MACRO, ENTER AT PRINIT AND PARCLR
                        ; CLEAR STATUS TABLE, SUMMARY TABLE, ERROR TABLE
                        ; CLEAR "RIB" IN XMTR COMMAND WORD "TXMST"
                        ; CLEAR CLOCK DATA
                        ; CLEAR TRANSMITTER HARDWARE
                        ; INITIALIZE DATA PATTERN
                        ;


005210                          PROC    CPINIT
(1)
(1)  005210             CPINIT:                         ;**ENTRY POINT**
005210  022766  000001  000002          CMP     #1,2(SP)        ;ARE THERE ANY ARGUMENTS?
005216  001114                          BNE     SYNTIN          ;IF SO, INDICATE SYNTAX ERROR
005220  012767  177777  012540          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW "PCL>"
005226                          PROC    PRINIT
(1)
(1)  005226             PRINIT:                         ;**ENTRY POINT**
005226                          BDINIT  XMTR            ;CLEAR XMITTER.
005234  105067  012413          CLRB    SECNDS          ;CLEAR SECONDS (TIME)
005240  105067  012410          CLRB    MINUTS          ;CLEAR MINUTES
005244  105067  012402          CLRB    TICKS           ;CLEAR TICKS
005250  005067  012402          CLR     HOURS           ;CLEAR HOURS
005254  042767  100000  002104  BIC     #B15,TXMST      ;CLEAR RIB IN COMMAND WORD FOR XMTR
005262                          CALL    PRCLR           ;CLR STATUS TABLE & ADDR QUEUE
005276                          PROC    PARCLR
(1)
(1)  005276             PARCLR:                         ;**ENTRY POINT**
005276  012700  024736          MOV     #ERTBL,R0       ;CLEAR ERROR SUMMARY TABLE
005302  022710  177777  ERTBCL: CMP     #-1,(R0)        ;IS ERR NUM  -1?
005306  001406                  BEQ     ERTCD           ;IF SO, DONE CLEARING
005310  005060  000004          CLR     4(R0)           ;OTHERWISE, CLR OCURRENCES
005314  062700  000006          ADD     #6,R0           ;STEP TO NEXT ENTRY
005320  000167  177756          JMP     ERTBCL          ;AND CONTINUE
005324  012700  001350  ERTCD:  MOV     #37*30,R0       ;ALSO CLEAR ERROR DATA
005330  012701  025160          MOV     #TERTBL,R1      ;FROM DETAILED ERR TABLE
005334  012721  000000  1$:     MOV     #0,(R1)+
005340  005300                  DEC     R0
005342  001374                  BNE     1$
005344  012700  000037          MOV     #37,R0          ;CLEAR ATTEMPS & SUCCESSES
005350  012701  020110          MOV     #RADB,R1        ; FROM STATUS TABLE
005354  062701  000004  2$:     ADD     #4,R1
005360  005021                  CLR     (R1)+
005362  005021                  CLR     (R1)+
005364  005021                  CLR     (R1)+
005366  005021                  CLR     (R1)+
005370  005300                  DEC     R0
005372  001370                  BNE     2$
005374  016767  012334  012334  MOV     ORIGSD,DTSEED   ;RESTORE ORIGINAL DATA SEED
005402  016767  012332  012332  MOV     MLSD,MSGLSD     ;AND MSG LENGTH SEED
005410  005067  012270          CLR     TXMEV           ;CLR XMIT EVENT FLAG
005414  005067  012252          CLR     SUMEV           ;CLR SUMMARY EVENT FLAG
005420  005067  012252          CLR     STSEV           ;CLR STATUS EVENT FLAG
005424  005067  012232          CLR     TEREV
```

```
 941  005430  005067  012230                 CLR     REREV                   ;CLR XMTR & RCVR ERROR EVENTS
 942  005434  005067  012306                 CLR     RCTXPS                  ;CLEAR DATA PASS NO.
 943  005440  052767  100000  012222         BIS     #B15,DATGEV             ;SET DATA GEN FLAG
 944  005446                         CPINRT: RETURN
 945  005450  000261                 SYNTIN: SEC                             ;SET "C" BIT FOR SYNTAX ERROR
 946  005452  000167  177770                 JMP     CPINRT
 947
 948
 949                                 ; SET THE SUMMARY EVENT FLAG SO THAT THE ERROR SUMMARY TABLE WILL BE PRINTED
 950
 951
 952
 953  005456                                 PROC    CPSUM
 (1)
 (1)  005456                         CPSUM:                                  ;**ENTRY POINT**
 954  005456  022766  000001  000002         CMP     #1,2(SP)                ;ARE THERE ANY ARGUMENTS?
 955  005464  001006                         BNE     SYNTSM                  ;IF SO, INDICATE SYNTAX ERROR
 956  005466  012767  100000  012176         MOV     #B15,SUMEV              ;SET SUMMARY EVENT FLAG
 957  005474  005067  012202                 CLR     SUMPNT                  ;0 SUMPNT INDICATES HEADER FIRST
 958  005500                         CPSURT: RETURN
 959  005502  000261                 SYNTSM: SEC                             ;SET "C" BIT FOR SYNTAX ERROR
 960  005504  000167  177770                 JMP     CPSURT
 961
 962
 963
 964
 965                                 ; START THE EXERCISER
 966                                 ;  CLEAR "ATTEMPTS" AND "SUCCESSES" IN STATUS TABLE
 967                                 ;  CLEAR ERROR TABLES
 968                                 ;  CAUSE RCVR ADDRESS QUEUE TO BE LOADED
 969                                 ;  SET PCLGO
 970                                 ;  IF THE RCVR ADDRESS QUEUE IS EMPTY AFTER BEING LOADED,
 971                                 ;    INDICATE THIS BY PRINTING THE FOLLOWING MESSAGE:
 972                                 ;
 973                                 ;       "**NO RECEIVERS SELECTED.**"
 974                                 ;
 975                                 ;  AND RETURN TO COMMAND INPUT MODE.
 976
 977
 978
 979  005510                                 PROC    CPGO
 (1)
 (1)  005510                         CPGO:                                   ;**ENTRY POINT**
 980  005510  022766  000001  000002         CMP     #1,2(SP)                ;ARE THERE ANY ARGUMENTS?
 981  005516  001062                         BNE     SYNTX                   ;IF SO, INDICATE SYNTAX ERROR
 982  005520  005067  012222                 CLR     RCTXPS                  ;CLEAR PASS NO.
 983  005524  005067  010704                 CLR     AOQ                     ;INIT ADDR QUEUE TO EMPTY
 984  005530  105067  012117                 CLRB    SECNDS                  ;CLEAR SECONDS REG.
 985  005534  105067  012114                 CLRB    MINUTS                  ;CLEAR MINUTES REG.
 986  005540  105067  012106                 CLRB    TICKS                   ;CLEAR TICKS REG.
 987  005544  005067  012106                 CLR     HOURS                   ;CLEAR HOURS REG.
 988  005550  016767  010664  010666         MOV     AOQ+QTOP,AOQ+QFRONT
 989  005556  016767  010656  010662         MOV     AOQ+QTOP,AOQ+QBACK
 990  005564  052767  100000  012074         BIS     #B15,QLDEV              ;SET ADDR QUEUE LOAD EVENT
 991  005572  012767  177777  012110         MOV     #-1,PCLGO               ;SET PCL TO GO
 992  005600                                 CALL    PARCLR                  ;CLEAR STATUS & ERRORS (NOT RCVRS)
```

```
 993  005614                                        CALL    PNCRLF                  ;PRINT CR, LF
 994  005630                                        CALL    PNTLIN,<EXRST>          ;PRINT 'EXERCISER STARTED'.
 995  005646  005767  012006                       TST     KWFLG                   ;GOT A CLOCK?
 996  005652  001403                               BEQ     STRTRT                  ;NO.
 997  005654  012777  000100  010276               MOV     #B06,@LCS               ;ENABLE CLOCK INTR.
 998  005662                                STRTRT: RETURN                          ;EXIT
 999
1000  005664  000261                        SYNTX:  SEC                             ;SET 'C'TO INDICATE SYNTAX ERROR
1001  005666  000167  177770                       JMP     STRTRT                  ;AND EXIT
1002
1003
```

```
                        .SBTTL  COMMAND PROC. FOR  ''ASSIGN''

                    ; ASSIGN XMTR ADDR & VECTOR AND RCVR ADDR & VECTOR.
                    ;THE ASSIGN COMMAND MAY HAVE 0, 1, 2, 3 OR 4 ARGUMENTS:
                    ;
                    ;       0 = ASSIGN STANDARD ADDRESSES & VECTORS TO RCVR & XMTR
                    ;       1 = ARGUMENT IS ASSUMED TO BE XMTR ADDRESS
                    ;       2 = ARGUMENTS ARE ASSUMED TO BE : XMTR ADDR, XMTR VECTOR
                    ;       3 = ARGUMENTS ARE ASSUMED TO BE: XMTR ADDR, XMTR VECT, RCVR ADDR
                    ;       4 = ARGUMENTS IN THE FOLLOWING ORDER:
                    ;
                    ;               XMTR ADDRESS, XMTR VECTOR, RCVR ADDRESS, AND RCVR VECTOR.
                    ;  CAUTION MUST BE EXERCISED WHEN ASSIGNING ADDRESSES TO GET THEM
                    ; IN THE CORRECT SEQUENCE AND ORDER.

005672                          PROC    CPASS
(1)
(1)  005672                     CPASS:                          ;**ENTRY POINT**
005672  010600                          MOV     SP,R0          ;COPY LIST POINTER
005674  016001  000002                  MOV     2(R0),R1       ;GET NO. OF ARGUMENTS
005700  020127  000005                  CMP     R1,#5          ;4 ARGUMENTS?
005704  101402                          BLOS    1$
005706  000167  000372                  JMP     SYNXR          ;ERROR TOO MANY ARGUMENTS
005712  001030                  1$:     BNE     PART1          ;LESS THAN 4, PARTIAL ASSIGNMENT
005714  016002  000004                  MOV     4(R0),R2       ;GET CLASS OF OBJECT
005720  022702  000002                  CMP     #2,R2          ;IF IT'S A NUMBER, O.K.
005724  001402                          BEQ     2$
005726  000167  000352                  JMP     SYNXR          ;ERROR WRONG CLASS
005732  062700  000006          2$:     ADD     #6,R0          ;NOW GET ACTUAL NUMBER
005736  016002  000002                  MOV     2(R0),R2
005742  032702  000003                  BIT     #3,R2          ;IS IT ON VECTOR BOUNDARY?
005746  001402                          BEQ     3$             ;YES
005750  000167  000330                  JMP     SYNXR          ;ERROR NOT ON VECTOR BOUNDARY
005754  020227  000774          3$:     CMP     R2,#774        ;IS IT WITHIN VECTOR AREA?
005760  101402                          BLOS    4$             ;YES
005762  000167  000316                  JMP     SYNXR          ;ERROR, OUTSIDE VECTOR AREA
005766  010267  010212          4$:     MOV     R2,RCVEC       ;SAVE AS RCVR VECTOR
005772  005301                          DEC     R1
005774  020127  000004          PART1:  CMP     R1,#4          ;ARE THERE ONLY 3 ARGS?
006000  001030                          BNE     PART2          ;NO, MUST BE FEWER
006002  016002  000004                  MOV     4(R0),R2       ;GET CLASS OF OBJECT
006006  022702  000002                  CMP     #2,R2          ;IF IT'S A NUMBER, O.K.
006012  001402                          BEQ     1$
006014  000167  000264                  JMP     SYNXR          ;ERROR, OBJECT NOT A NUMBER
006020  062700  000006          1$:     ADD     #6,R0          ;NOW, GET THE NUMBER
006024  016002  000002                  MOV     2(R0),R2
006030  032702  000017                  BIT     #17,R2         ;CHECK IF ITS A VALID ADDR
006034  001402                          BEQ     2$             ;IT IS. O.K.
006036  000167  000242                  JMP     SYNXR          ;ERROR INVALID ADDRESS
006042  020227  164000          2$:     CMP     R2,#164000     ;I  IT IN THE ADDRESS FIELD?
006046  103002                          BHIS    3$             ;\_S, O.K.
006050  000167  000230                  JMP     SYNXR          ;ERROR. ADDR = OUTSIDE DEVICE AREA
006054  010267  010122          3$:     MOV     R2,RCDEV       ;SAVE AS RCVR ADDRESS
006060  005301                          DEC     R1
006062  020127  000003          PART2:  CMP     R1,#3          ;ARE THERE ONLY 2 ARGS?
006066  001030                          BNE     PART3          ;NO, MUST BE ONLY '.
```

```
1059  006070  016002  000004                MOV     4(R0),R2        ;GET CLASS OF OBJECT
1060  006074  022702  000002                CMP     #2,R2
1061  006100  001402                        BEQ     1$              ;IF IT'S A NUMBER, O.K.
1062  006102  000167  000176                JMP     SYNXR           ;ERROR, WRONG CLASS
1063  006106  062700  000006        1$:     ADD     #6,R0           ;NOW GET THE NUMBER
1064  006112  016002  000002                MOV     2(R0),R2
1065  006116  032702  000003                BIT     #3,R2           ;IS IT ON VECTOR BOUNDARY?
1066  006122  001402                        BEQ     2$              ;YES
1067  006124  000167  000154                JMP     SYNXR           ;ERROR. NOT ON VECTOR BOUNDARY
1068  006130  020227  000774        2$:     CMP     R2,#774         ;IS IT WITHIN VECTOR AREA?
1069  006134  101402                        BLOS    3$              ;YES
1070  006136  000167  000142                JMP     SYNXR           ;ERROR. OUTSIDE VECTOR AREA
1071  006142  010267  010032        3$:     MOV     R2,TXVEC        ;SAVE AS XMTR VECTOR
1072  006146  005301                        DEC     R1
1073  006150  020127  000002        PART3:  CMP     R1,#2           ;ONLY 1 ARGUMENT?
1074  006154  001027                        BNE     PART4           ;NO, MUST BE NONE.
1075  006156  016002  000004                MOV     4(R0),R2        ;GET CLASS OF OBJECT
1076  006162  022702  000002                CMP     #2,R2
1077  006166  001402                        BEQ     1$              ;IF IT'S A NUMBER, O.K.
1078  006170  000167  000110                JMP     SYNXR           ;ERROR. WRONG CLASS
1079  006174  062700  000006        1$:     ADD     #6,R0           ;NOW GET THE NUMBER
1080  006200  016002  000002                MOV     2(R0),R2
1081  006204  032702  000017                BIT     #17,R2          ;CHECK FOR VALID ADDRESS
1082  006210  001402                        BEQ     2$              ;O.K.
1083  006212  000167  000066                JMP     SYNXR
1084  006216  020227  164000        2$:     CMP     R2,#164000      ;IS IT IN ADDRESS FIELD?
1085  006222  103002                        BHIS    3$              ;YES
1086  006224  000167  000054                JMP     SYNXR           ;ERROR, OUTSIDE ADDRESS FIELD
1087  006230  010267  007742        3$:     MOV     R2,TXDEV        ;SAVE AS XMTR ADDRESS
1088  006234  020127  000001        PART4:  CMP     R1,#1           ;NO ARGUMENTS?
1089  006240  001014                        BNE     ASEXT           ;WE'RE DONE!
1090  006242  012767  164200  007726        MOV     #164200,TXDEV   ;NO ARGUMENTS, LOAD DEFAULTS.
1091  006250  012767  000170  007722        MOV     #170,TXVEC
1092  006256  012767  164220  007716        MOV     #164220,RCDEV
1093  006264  012767  000174  007712        MOV     #174,RCVEC
1094  006272  012737  000340  177776  ASEXT: MOV     #340,@#PS       ;RAISE PROC. PRIORITY
1095  006300  000167  173474                JMP     PCLEX           ;GO START EXERCISER OVER.
1096
1097  006304  000261                SYNXR:  SEC                     ;SET ''C'' TO FLAG ERROR
1098  006306                                RETURN                  ;RETURN
```

```
                        .SBTTL  COMMAND DECODER AND PROCESSOR

                       ;CONVERT COMMAND TO IT'S PROCESSING ROUTINE ADDRESS
                       ;IF C BIT IS SET, INDICATE SYNTAX ERROR
                       ;RETURN TO COMMAND INPUT MODE
                       ;IF ESCFLG <> 0, CALL PRESC (CNTRL-C)

                               PROC    COMENT

                       COMENT:                                 ;**ENTRY POINT**
006310  105067  011335         CLRB    CMDENT                  ;CLEAR COMMAND ENTERED FLAG
006314  005067  011446         CLR     ESCFLG                  ;CLR COMMAND INPUT MODE FLAG
006320                         CALL    COMAND,<CMDBUF,CMDTBL>  ;CONVERT COMMAND
006340  103015                 BCC     CMDRTN                  ;IF C = 0, EXIT
006342                         CALL    PNTLIN,<SYNTAX>         ;PRINT ''SYNTAX ERROR'' IF C SET
006360                         CALL    PRESC                   ;GIVE ANOTHER 'PCL>''
006374  005767  011366         TST     ESCFLG                  ;SHOULD WE FAKE CNTRL-C?
006400  001406                 BEQ     CMDRET                  ;NO, EXIT
006402                         CALL    PRESC                   ;YES, GIVE A 'PCL>''
006416                 CMDRET: RETURN                          ;RETURN TO MAIN LOOP


                       ;PROCESSOR FOR SPECIAL EVENT
                       ;INDICATE ''NO MASTER'' AND CLEAR EVENTS
                       ;THIS EVENT IS CALLED BY THE MAIN LOOP AFTER
                       ;   THE OCCURRENCE OF A MASTER DOWN INTERRUPT
                       ;THE RESULT OF THIS EVENT IS THAT THE PROGRAM WILL
                       ;   PRINT MASTER DOWN IN THE FOLLOWING FASION:
                       ;
                       ;   * * * * * *  MASTER DOWN  * * * * * *
                       ;
                       ;THEN CLEAR ACTIVE EVENT FLAGS BY ENTERING COMMAND MODE.
                       ;TO RECOVER FROM THIS STATE AND RESUME EXERCISING:
                       ;   (A) SET MASTER ON ANY ONE PCL11
                       ;   (B) TYPE THE COMMAND ''CONTINUE''

                               PROC    SPEC

                       SPEC:                                   ;**ENTRY POINT**
                               CALL    PNCRLF                  ;ENQUEUE CR & LF
                               CALL    PNTLIN,<MDNER>          ;ENQUEUE MASTER DOWN MESSAGE
                               CALL    PNCRLF                  ;ANOTHER CR & LF
006466  005067  011222         CLR     SPCEV                   ;CLR SPECIAL EVENT FLAG
006472  005067  011212 SPERTN: CLR     PCLGO                   ;DON'T ALLOW TXM EVENT
006476  005067  011174         CLR     STSEV                   ;DISCONTINUE STATUS EVENT
006502  005067  011164         CLR     SUMEV                   ;DISCONTINUE SUM EVENT
006506                         CALL    PRESC                   ;FAKE CNTRL-C KEY
006522                         RETURN
```

```
                                ;PROCESSOR FOR 'EXTRA' SPECIAL EVENT
                                ;TO INDICATE WHEN THIS UNIT HAS JUST BECOME
                                ;MASTER AS A RESULT OF PREVIOUSLY HAVING
                                ;SECONDARY SET AND THE CURRENT BUS MASTER
                                ;HAVING JUST DROPPED MASTER.
                                ;WHEN 'NOW MASTER' INTERRUPT OCCURS, THE
                                ;CONSOLE DEVICE WILL PRINT
                                ;     '**  THIS UNIT HAS BECOME NEW MASTER **'
                                ;
1154  006524                            PROC    XSPEC

 (1)
 (1)  006524                    XSPEC:                          ;**ENTRY POINT**
1155  006524                            CALL    PNCRLF          ;ENQUEUE CR & LF
1156  006540                            CALL    PNTLIN,<MSCHNG> ;ENQUEUE NEW MASTER MESSAGE
1157  006556                            CALL    PNCRLF          ;ANOTHER CR & LF
1158  006572  005067  011120           CLR     XSPCEV          ;CLR XSPEC EVENT FLAG
1159  006576                            RETURN                  ;RETURN TO MAIN LOOP
1160
1161
1162                                ;PROCESSOR FOR 'DETAILED' ERROR TABLE PRINTOUT COMMAND 'ERRORS'
1163
1164                                ;THIS  ROUTINE WILL INITIALIZE THE TEMPORARY LOCATIONS REQUIRED
1165                                ;BY THE XMTR AND RCVR ERROR EVENTS. IT WILL THEN QUICKLY CHECK THE
1166                                ;RCVR AND XMTR ERROR TABLES FOR ANY ENTRIES. IF NONE EXIST, THE MESSAGE
1167                                ;       '**NO ERRORS TO REPORT**'
1168                                ;WILL BE PRINTED ON THE CONSOLE AND THAT IS ALL.
1169                                ; IF ANY ENTRIES WERE FOUND, THE XMTR ERROR EVENT FLAG 'TEREV' WILL BE SET
1170                                ; WHICH WILL CAUSE THE XMTR ERRORS AND THE RCVR ERRORS (IF ANY) TO
1171                                ; BE PRINTED.
1172
1173
1174  006600                            PROC    CPERR
 (1)
 (1)  006600                    CPERR:                          ;**ENTRY POINT**
1175  006600  022766  000001  000002           CMP     #1,2(SP)        ;ANY ARGUMENTS?
1176  006606  001044                           BNE     SYNRR           ;ERROR IF THERE ARE.
1177  006610  012700  025160                   MOV     #TERTBL,R0      ;POINT AT XMTR TABLE
1178  006614  012701  001350                   MOV     #37*30,R1       ;SET TABLE LENGTH COUNTER
1179  006620  005720                   1$:     TST     (R0)+           ;ANY ENTRIES?
1180  006622  001016                           BNE     SERFL           ;YES, CALL ERROR MODULE
1181  006624  005301                           DEC     R1              ;NO, CHECK WHOLE TABLE
1182  006626  001374                           BNE     1$
1183  006630                                   CALL    PNTLIN,<NOERMG> ;PRINT 'NO ERRORS'
1184  006646  012767  177777  011112           MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE 'PCL>'
1185  006654  000167  000036                   JMP     CPERTN          ;AND RETURN
1186  006660  012767  100000  010774   SERFL:  MOV     #B15,TEREV      ;SET XMTR EVENT FLAG
1187  006666  012767  000001  011104           MOV     #1,ERR0         ;INITIATE ERROR #
1188  006674  012767  000001  011100           MOV     #1,ERR1         ;INITIATE RCVR ADDR
1189  006702  012767  025160  011074           MOV     #TERTBL,ERR2    ;INITIATE TABLE POINTER
1190  006710  052767  100000  011010           BIS     #B15,HDRFLG     ;SET HEADER FLAG
1191  006716                           CPERTN: RETURN
1192
1193  006720  000261                   SYNRR:  SEC                     ;SET 'C' BIT TO INDICATE SYNTAX ERR
1194  006722  000167  177770                   JMP     CPERTN
```

```
                        .SBTTL RECEIVER ADDRESS QUEUE LOADER ROUTINE

                        ; LOAD THE RECEIVER ADDRESS SOFTWARE QUEUE
                        ;
                        ; THE TRANSMITTER EVENT WILL PICK RCVR ADDRESSES FROM THE QUEUE
                        ; UNTIL IT IS EMPTY.
                        ;
                        ; IF THE QUEUE IS LOADED BUT STILL REMAINS EMPTY, AND PCLGO IS SET,
                        ; THIS MEANS THE OPERATOR IS TRYING TO RUN BUT HASN'T ENTERED
                        ; ANY RECEIVER ADDRESSES (USING 'RANGE' OR 'ADD' ETC.)
                        ;

006726                          PROC    ADQLD
 (1)
 (1)  006726            ADQLD:
                                REGSAV                          ;**ENTRY POINT**
1210  006726                                                    ;SAVE R0...R5
1211  006732  005067  010730       CLR     QLDEV                ;CLR QUEUE LOAD EVENT FLAG
1212  006736  012702  000037       MOV     #37,R2               ;POSITION COUNTER
1213  006742  012700  020110       MOV     #RADB,R0             ;R0 POINTS AT TABLE
1214  006746  022710  177777 QCHACT: CMP    #-1,(R0)             ;IS ACTIVE FLAG SET IN TABLE?
1215  006752  001002               BNE     QLOOK                ;NO, LOOK FOR NEXT ONE
1216  006754  004767  000036       JSR     PC,QLOAD             ;YES, LOAD ADDR INTO QUEUE
1217  006760  062700  000014 QLOOK: ADD     #14,R0
1218  006764  005302               DEC     R2                   ;ALL ENTRIES CHECKED?
1219  006766  001367               BNE     QCHACT
1220  006770  005767  007440       TST     ADQ                  ;IS THERE ANYTHING IN QUEUE?
1221  006774  001424               BEQ     QMTRN                ;NO,THIS IS NOT A VALID PASS
1222  006776  005267  010744       INC     RCTXPS               ;YES, INCR PASS NO.
1223  007002  012767  100000 010674 QRTN:  MOV     #B15,TXMEV     ;SET XMTR EVENT FLAG
1224  007010                        REGRES                       ;RESTORE R0...R5
1225  007014                        RETURN
1226
1227
1228  007016  116067  000002 021253 QLOAD: MOVB    2(R0),BKELEM+1       ;GET RCV ADDR INTO HIGH BYTE
1229  007024                        CALL    ENQ,<BKELEM,ADQ>     ;PUT IT IN ADDR QUEUE
1230  007044  000207                RTS     PC
1231
1232
1233  007046  005767  010636 QMTRN: TST     PCLGO                ;ARE WE RUNNING?
1234  007052  001753               BEQ     QRTN                 ;NO, LEAVE
1235  007054                        CALL    PRCTLO               ;STOP PRINTING ANYTHING ELSE
1236  007070                        CALL    PNTLIN,<MTQMSG>      ;TELL OPERATOR NO RCVRS
1237  007106                        CALL    PRESC                ;FAKE CNTRL-C
1238  007122  000727               BR      QRTN                 ;RETURN
```

```
                        .SBTTL DATA GENERATION (RANDOM) ROUTINE

                        ; GENERATE A BUFFER OF DATA AND NOTE THE FOLLOWING CONDITIONS:
                        ;
                        ;       BITS <8:0> OF THE FIRST WORD = MESSAGE LENGTH
                        ;       IF THE FIRST WORD IS 170000 OR HIGHER, EXPECT A REJECT
                        ;       IF THE MESSAGE LENGTH IS > 602, EXPECT A TRUNCATE
                        ;       GENERATE RANDOM DATA USING THE FOLLOWING FORMULA:
                        ;               SN = (SN-1 * RANDOM NO.) + RANDOM INCREMENT
                        ;
                        ;       WHERE S = DATA SEED; AND N = A NUMBER FROM 1 TO MESSAGE LENGTH.
                        ;
                        ;       A MESSAGE LENGTH OF 0 IS NOT ALLOWED AND IS CHANGED TO 1.


007124                          PROC    DATGEN
(1)
(1)  007124                     DATGEN:                         ;**ENTRY POINT**
007124  005067  010616          CLR     RCTXPS                  ;CLEAR DATA PASS FLAG
007130  005067  010614          CLR     RJCTF                   ;CLEAR REJECT SOFTWARE FLAG
007134  005067  010612          CLR     TRNKF                   ;CLR TRUNCATE SOFTWARE FLAG
007140  042767  177000  010574  BIC     #177000,MSGLSD          ;LENGTH NOT TO EXCEED 1000
007146  001003                  BNE     DTCNT                   ;IF NON-0, OKAY, CONTINUE.
007150  012767  000001  010564  MOV     #1,MSGLSD               ;IF 0, MAKE IT AT LEAST 1
007156  026727  010560  000602  DTCNT:  CMP     MSGLSD,#602     ;IS LENGTH LESS THAN 602?
007164  002403                  BLT     DTGNC
007166  012767  177777  010556  MOV     #-1,TRNKF               ;NO,SET TRUNCATE SOFTWARE FLAG
007174  022767  170000  010534  DTGNC:  CMP     #170000,DTSEED  ;IS DATA SEED -10000 OR MORE?
007202  101005                  BHI     DTGNCO
007204  012767  177777  010536  MOV     #-1,RJCTF               ;YES SET REJECT SOFTWARE FLAG
007212  005067  010534          CLR     TRNKF                   ;AND CLR TRUNCATE SFTWR FLAG
007216  012700  020676          DTGNCO: MOV     #DATBUF,R0      ;POINT R0 TO TOP OF BUFFER
007222  016705  010514          MOV     MSGLSD,R5               ;R5 IS MESSAGE LENGTH
007226  006105                  ROL     R5                      ;DOUBLE IT
007230  005405                  NEG     R5                      ;NEGATE IT
007232  010567  010516          MOV     R5,TXML                 ;SAVE IT FOR TRANSMIT MODULE
007236  016705  010500          MOV     MSGLSD,R5               ;R5 IS MSG LENGTH AGAIN
007242  016720  010470          DTCLP:  MOV     DTSEED,(R0)+    ;PUT WORD INTO BUFFER
007246                          MULP    RANM,DTSEED             ;GENERATE NEW RANDOM NUMBER
007270  066767  010464  010440  ADD     RANK,DTSEED
007276  005305                  DEC     R5
007300  003360                  BGT     DTCLP                   ;IF R5 = 0, FINISHED
007302  016767  010430  010432  DTCDON: MOV     DTSEED,MSGLSD  ;GET NEW LENGTH
007310  012767  100000  010366  MOV     #B15,TXMEV              ;SET XMIT EVENT FLAG
007316  005067  010346          CLR     DATGEV                  ;CLEAR DAT GEN EVENT FLAG
007322                          RETURN
```

```
                                        .SBTTL MULTIPLY ROUTINE FOR DATA GENERATION

                                ;UNSIGNED MULTIPLY ROUTINE
                                ;CALLED BY 'JSR R4,MLI'
                                ;WITH MULTIPLICAND IN R4 AND MULTIPLIER ON STACK
                                ;PRODUCT RETURNED ON TOP OF STACK

007324  01260'          MLI:    MOV     (SP)+,R1        ;GET MULTIPLICAND
007326  011603                  MOV     (SP),R3         ;GET MULTIPLIER
007330  010446                  MOV     R4,-(SP)        ;SAVE R4
007332  012704  000020          MOV     #16.,R4         ;SET UP FOR 16 BIT MULTIPLY
007336  005002                  CLR     R2
007340  006002          MUL:    ROR     R2              ;SHIFT PRODUCT
007342  006003                  ROR     R3
007344  103001                  BCC     CYCL            ;CHECK IF MULTIPLIER BIT IS 0
007346  060102                  ADD     R1,R2           ;ADD IN MULTIPLICAND
007350  005304          CYCL:   DEC     R4              ;COUNT LOOP
007352  003372                  BGT     MUL
007354  012604                  MOV     (SP)+,R4        ;RESTORE R4
007356  006202                  ASR     R2              ;ONE LAST SHIFT
007360  006003                  ROR     R3              ;PRODUCT IS IN R3
007362  010316                  MOV     R3,(SP)         ;STORE RESULT ON STACK
007364  000134                  JMP     @(R4)+          ;RETURN


                                ;MACRO FOR THE ABOVE IS AS FOLLOWS:
                                ;       .MACRO  MULP    A,B
                                ;               MOV     A,-(SP)         ;SAVE A ON STACK
                                ;               MOV     B,R4            ;SAVE B IN R4
                                ;               JSR     R4,MLI          ;PERFORM MULTIPLICATION
                                ;               .WORD   .+2
                                ;               MOV     (SP)+,B         ;PUT PRODUCT INTO B
                                ;       .ENDM
```

```
                                        .SBTTL TRANSMIT MODULE



                                ;  CONTROL TRANSMISSION OF RANDOM DATA TO DE-QUEUED RECEIVER ADDRESSES
                                ;
                                ; IF  'PCLGO' IS SET, ('GO' OR 'CONTINUE' SET IT), START TRANSMITTING
                                ; THE DATA CONTAINED IN THE BUFFER 'DATBUF'.
                                ; THE TARGET RECEIVER ADDRESS IS DEQUEUED FROM THE ADDRESS QUEUE.
                                ;  THE MESSAGE LENGTH WAS DETERMINED DURING THE GENERATION OF THE DATA.
                                ;
                                ; NOTE THAT BIT 15 OF 'TXMST' MAY BE 1 OR 0 DEPENDING UPON THE USE
                                ; OF THE 'RIB SET' OR 'RIB CLEAR' COMMANDS.


007366  060101                  TXMST:  .WORD   060101                  ;TRANSMITTER COMMAND WORD

007370                                  PROC    TXMIT

007370                          TXMIT:                                  ;**ENTRY POINT**
007370  005767  010314                  TST     PCLGO                   ;CAN PCL GO?
007374  001455                          BEQ     TXRTN                   ;IF NOT, RETURN
007376  005067  010302                  CLR     TXMEV                   ;CLR XMTR EVENT FLAG
007402                                  BDINIT  XMTR                    ;CLR XMTR HARDWARE
007410  005767  007020                  TST     AOQ                     ;ANY ELEMENTS IN ADDR QUEUE?
007414  001446                          BEQ     TXOUT                   ;IF NOT, FILL IT UP AGAIN
007416  016777  010332  006470          MOV     TXML,@TSBC              ;LOAD BYTE COUNT
007424  012777  020676  006464          MOV     #DATBUF,@TSBA           ;LOAD BUS ADDRESS
007432                                  CALL    DEQ,<FRELEM,AOQ>        ;DEQUEUE RECEIVER ADDRESS
007452  116767  020617  010302          MOVB    FRELEM+1,CURAD          ;GET ADDRESS TABLE OFFSET
007460  016700  010276                  MOV     CURAD,R0                ;FIND ADDRESS TABLE ENTRY
007464  006300                          ASL     R0                      ;       IN ORDER TO UPDATE
007466  006300                          ASL     R0                      ;       "ATTEMPTS"
007470  010005                          MOV     R0,R5
007472  006300                          ASL     R0
007474  060500                          ADD     R5,R0
007476  062700  020074                  ADD     #RADB0,R0
007502  062760  000001  000006          ADD     #1,6(R0)                ;UPDATE DOUBLE WORD
007510  005560  000004                  ADC     4(R0)
007514  016777  020554  006364          MOV     FRELEM,@TCR             ;GIVE RECEIVER ADDR TO XMTR
007522  056777  177640  006356          BIS     TXMST,@TCR              ;START TRANSMISSION
007530                          TXRTN:  RETURN

007532  012767  100000  010126  TXOUT:  MOV     #B15,QLDEV              ;SET QUEUE LOAD EVENT FLAG
007540  026727  010202  000004          CMP     RCTXPS,#4               ;IS THIS THE 5TH PASS?
007546  001402                          BEQ     TXDGEN                  ;IF SO, SET DATA GEN EVENT FLAG
007550  000167  177754                  JMP     TXRTN                   ;RETURN
007554  012767  100000  010106  TXDGEN: MOV     #B15,DATGEV             ;SET DAT GEN EVENT FLAG
007562  000167  177742                  JMP     TXRTN                   ;RETURN
```

```
1367                                    ;;;TRANSMITTER INTERRUPT HANDLER
1368                                    ;;; SORT THE CAUSES OF TRANSMITTER INTERRUPTS
1369                                    ;;; AND GO TO THE PROPER HANDLING ROUTINE.
1370
1371
1372            100000                  TXER    =       100000
1373            000020                  TXBSY   =       20
1374            000200                  TXSUC   =       200
1375            000040                  TXSOR   =       40
1376            000004                  NOWMST  =       4
1377
1378  007566                            XMTINT: REGSAV                          ;;;SAVE R0...R5
1379  007572  042777  000100  006306            BIC     #B06,@TCR               ;;;CLR TXM INTR ENABLE
1380  007600  012737  000200  177776            MOV     #200,@#PS               ;;;ALLOW RCVR INTR BUT NOT TTY
1381  007606  032777  100000  006274            BIT     #TXER,@TSR              ;;;IS THERE A HARDWARE ERROR?
1382  007614  001402                            BEQ     XMTIS                   ;;;NO, CHECK SUC TXF
1383  007616  000167  000114                    JMP     ERRINT                  ;;;HANDLE ERROR
1384  007622  032777  000200  006260  XMTIS:    BIT     #TXSUC,@TSR             ;;;WAS IT A SUC TRANSFER?
1385  007630  001402                            BEQ     XMTSR                   ;;;NO, CHECK SOFTWARE REJECT
1386  007632  000167  000610                    JMP     SUCINT                  ;;;HANDLE SUC TXF
1387  007636  032777  000040  006244  XMTSR:    BIT     #TXSOR,@TSR             ;;;WAS MESSAGE REJECTED?
1388  007644  001402                            BEQ     XMTPNM                  ;;;NO,CHECK NOW MASTER
1389  007646  000167  000772                    JMP     SORINT                  ;;;HANDLE REJECT
1390  007652  132777  000004  006242  XMTPNM:  BITB    #NOWMST,@TMMRH          ;;;IS THIS UNIT NOW MASTER?
1391  007660  001402                            BEQ     XMTIB                   ;;;NO, WAS RECVR BUSY?
1392  007662  000167  001074                    JMP     NMINT                   ;;;YES, HANDLE NOW MASTER INT
1393  007666  032777  000020  006214  XMTIB:    BIT     #TXBSY,@TSR             ;;;WAS RECEIVER BUSY
1394  007674  001402                            BEQ     XMTPRB                  ;;;NO, INTERRUPT WAS ERRONEOUS
1395  007676  000167  000514                    JMP     BSYINT                  ;;;HANDLE BUSY
1396  007702  012767  100000  007774  XMTPRB:  MOV     #B15,TXMEV              ;;;SET XMTR EVENT FLAG
1397  007710                                    ERROT   \N              ;;;ERROR:ERRONEOUS INTERRUPT FROM XMTR
  (1)
  (1)                                                                   ;**** XMTR ERROR 1 *****
  (1)
1398  007730                            REGRES                                ;;;RESTORE R0...R5
1399  007734  000002                    RTI                                   ;;;RETURN
```

```
                                        ;XMTR ERROR INTERRUPT ROUTINE

                                        ; DETERMINE AND REPORT THE CAUSE OF THE TRANSMITTER ERROR.

007736  032777  040000  006144  ERRINT: BIT     #B14,@TSR               ;IS ERROR NON EXISTANT LOCATION?
007744  001412                          BEQ     ERRA
007746                                  ERROT   \N              ;ERROR:NON-EXISTANT LOC ERROR IN XMTR
                                                                        ;**** XMTR ERROR 2 *****

007766  000167  000402                  JMP     ERRX
007772  032777  020000  006110  ERRA:   BIT     #B13,@TSR               ;IS ERROR MEM OFL?
010000  001412                          BEQ     ERRB
010002                                  ERROT   \N              ;ERROR:MEM OFL ERROR IN XMTR
                                                                        ;**** XMTR ERROR 3 *****

010022  000167  000346                  JMP     ERRX
010026  032777  010000  006054  ERRB:   BIT     #B12,@TSR               ;IS ERROR TXM ERROR?
010034  001472                          BEQ     ERRC
010036  017767  006046  007720          MOV     @TSR,RSPC               ;SAVE RESPONSE CODES
010044  042767  177760  007712          BIC     #-20,RSPC               ;REMOVE GARBAGE
010052  022767  000015  007704          CMP     #15,RSPC                ;RSP CODES 11&01?
010060  001012                          BNE     ERRB1
010062                                  ERROT   \N              ;ERROR:TXM ERROR. RCVR ACCEPTED A NULL
                                                                        ;**** XMTR ERROR 4 *****

010102  000167  000266                  JMP     ERRX
010106  042767  000003  007650  ERRB1:  BIC     #3,RSPC                 ;DISCARD RSP A'S
010114  001012                          BNE     ERRB2                   ;RSB B   00?
010116                                  ERROT   \N      ;ERROR:TXM ERROR. RCVR HAS GONE OFF LINE
                                                                        ;**** XMTR ERROR 5 *****

010136  000167  000232                  JMP     ERRX
010142  022767  000010  007614  ERRB2:  CMP     #10,RSPC                ;RSP B = 10?
010150  001012                          BNE     ERRB3
010152                                  ERROT   \N      ;ERROR:TXM ERROR. WORD OR CRC REJECTED
                                                                        ;**** XMTR ERROR 6 *****

010172  000167  000176                  JMP     ERRX
010176                          ERRB3:  ERROT   \N      ;ERROR:MISCELLANEOUS TXM ERROR
                                                                        ;**** XMTR ERROR 7 *****

010216  000167  000152                  JMP     ERRX
010222  032777  004000  005660  ERRC:   BIT     #B11,@TSR               ;MASTER DOWN?
010230  001417                          BEQ     ERRD
010232                                  ERROT   \N      ;ERROR: M A S T E R  D O W N .
                                                                        ;**** XMTR ERROR 10 *****
```

```
1436  010252  005067  007432                CLR     PCLGO                   ;STOP TRANSMITTER
1437  010256  012767  100000  007430        MOV     #B15,SPCEV              ;SET SPECIAL EVENT FLAG
1438  010264  000167  000104                JMP     ERRX
1439  010270  032777  002000  005612  ERRD:  BIT     #B10,@TSR               ;TIMEOUT ERRCR?
1440  010276  001422                        BEQ     ERRE
1441  010300  032777  100000  005600        BIT     #B15,@TCR               ;IS 'R.I.B.' SET?
1442  010306  001404                        BEQ     1$                      ;NO MUST BE WEIRD ERROR.
1443  010310  032777  000020  005572        BIT     #TXBSY,@TSR             ;YES, IS TDM BUS BUSY?
1444  010316  001010                        BNE     2$                      ;YES, RECEIVER WASN'T THERE!
1445  010320                        1$:     ERROT   \N              ;ERROR: ERRONEOUS TIMEOUT IN TRANSMITTER
 (1)                                                                    ;**** XMTR ERROR 11 *****
 (1)
 (1)
1446  010340  000167  000030        2$:     JMP     ERRX
1447  010344  032777  001000  005536  ERRE:  BIT     #B09,@TSR               ;TXM SILO OVERRUN?
1448  010352  001410                        BEQ     ERRX
1449  010354                                ERROT   \N              ;ERROR:SILO OVERRUN ERROR IN XMTR
 (1)                                                                    ;**** XMTR ERROR 12 *****
 (1)
 (1)
1450  010374                        ERRX:   BDINIT  XMTR                    ;CLEAR ALL XMTR HARDWARE
1451  010402  012767  100000  007274        MOV     #B15,TXMEV              ;SET XMTR EVENT FLAG
1452  010410                                REGRES                          ;RESTORE R0...R5
1453  010414  000002                        RTI                             ;RETURN
```

```
                                        ;BUSY INTERRUPT ROUTINE
                                        ;
010416  042777  000020  005464  BSYINT: BIC     #TXBSY,@TSR             ;CLEAR TDM BUS BUSY
010424  012767  100000  007252          MOV     #B15,TXMEV              ;SET TXM EVENT FLAC
010432                                  BDINIT  XMTR                    ;CLEAR HARDWARE
010440                                  REGRES                          ;RESTORE R0...R5
010444  000002                          RTI                             ;RETURN
                                        ;
                                        ;SUCCESSFUL TRANSFER INTERRUPT ROUTINE
                                        ;
010446  042777  000200  005434  SUCINT: BIC     #TXSUC,@TSR             ;CLEAR SUC TXF
010454  005767  007272                  TST     TRNKF                   ;IS SFTWR TRUNCATE FLAG SET?
010460  001047                          BNE     SUCIA                   ;YES, GO CHECK SORE
010462  032777  000040  005420  SUCINY: BIT     #TXSOR,@TSR             ;IS SORE SET?
010470  001414                          BEQ     SUCINP                  ;NO, CONTINUE
010472                                  ERROT   \N              ;ERROR:MESSAGE TRUNCATED UNEXPECTEDLY
 (1)                                                                    ;**** XMTR ERROR 13 *****
 (1)
 (1)
010512  000421                          BR      SHBRT                   ;DON'T INC SUC CONNS IF ERROR
010514  042777  000040  005366          BIC     #TXSOR,@TSR             ;CLR SORE
010522  016700  007234          SUCINP: MOV     CURAD,R0                ;PREPARE TO UPDATE STATUS TABLE
010526  006300                          ASL     R0                      ;GET TABLE ENTRY FOR CURRENT
010530  006300                          ASL     R0                      ;       RECEIVER ADDRESS IN
010532  010005                          MOV     R0,R5                   ;
010534  006300                          ASL     R0                      ;       ORDER TO UPDATE
010536  060500                          ADD     R5,R0                   ;
010540  062700  020074                  ADD     #RADB0,R0               ;       "SUCCESSFUL" ELEMENT
010544  062760  000001  000012          ADD     #1,12(R0)               ;UPDATE DOUBLE WORD
010552  005560  000010                  ADC     10(R0)
010556  012767  100000  007120  SHBRT: MOV     #B15,TXMEV              ;SET TRANSMIT EVENT FLAG
010564                                  BDINIT  XMTR                    ;CLEART HARDWARE
010572                                  REGRES                          ;RESTORE R0...R5
010576  000002                          RTI                             ;RETURN
                                        ;
010600  032777  000040  005302  SUCIA:  BIT     #TXSOR,@TSR             ;IS SORE SET? (SHOULD BE SET)
010606  001011                          BNE     SUCCS
010610                                  ERROT   \N              ;ERROR:MSG FAILED TO BE TRUNCATED
 (1)                                                                    ;**** XMTR ERROR 14 *****
 (1)
 (1)
010630  000752                          BR      SHBRT                   ;DON'T INC SUC CONNS IF ERROR
010632  042777  000040  005250  SUCCS:  BIC     #TXSOR,@TSR             ;CLR SORE
010640  000167  177616                  JMP     SUCINY                  ;PROCESS SUC TXF AS NORMAL
```

```
                                        ;SOFTWARE REJECT INTERRUPT ROUTINE
                                        SORINT: TST     RJCTF                   ;IS SFTWR REJECT FLAG SET?
                                                BEQ     SORIP                   ;IF NOT, SOMETHING'S WRONG
                                                MOV     CURAD,R0                ;PREPARE TO UPDATE STATUS TABLE
                                                ASL     R0                      ;GET TABLE ENTRY FOR CURRENT
                                                ASL     R0                      ;       RECEIVER ADDRESS IN
                                                MOV     R0,R3
                                                ASL     R0                      ;       ORDER TO UPDATE
                                                ADD     R3,R0
                                                ADD     #RADB0,R0               ;       "SUCCESSFUL" ELEMENT
                                                ADD     #1,12(R0)               ;UPDATE ENTRY
                                                ADC     10(R0)
                                        SORCNT: BIC     #TXSOR,@TSR             ;CLR SORE
                                                MOV     #B15,TXMEV              ;SET XMIT EVENT FLAG
                                                BDINIT  XMTR                    ;CLEAR HARDWARE
                                                REGRES                          ;RESTORE R0...R5
                                                RTI                             ;RETURN

                                        SORIP:  ERROT   \N              ;ERROR:ERRONEOUS REJECT BY RECEIVER
  (1)
  (1)                                                                   ;**** XMTR ERROR 15 *****
  (1)
                                                JMP     SORCNT                  ;RETURN


                                        ;NOW MASTER INTERRUPT ROUTINE
                                        ;SET XSPEC EVENT FLAG (XSPEV)
                                        ;AND CLR NOW MST

                                        NMINT:  BICB    #NOWMST,@TMMRH          ;CLEAR NOW MASTER
                                                MOV     #B15,XSPCEV             ;SET EXTRA-SPECIAL EVENT
                                                BIS     #B06,@TCR               ;RE-SET INTERRUPT ENABLE
                                                REGRES                          ;RESTORE R0...R5
                                                RTI                             ;RETURN
```

```
                                        .SBTTL RECEIVER MODULE

                                        ;CONTROL RECEPTION OF DATA.
                                        ; DATA IS STORED IN A BUFFER 'RCBUF'
                                        ; FOR LATER CHECKING.
                                        ; 'RCBUF' BUFFER IS CLEARED BEFORE RECEPTION.

                                        ;ACTIVE IF RCVEV IS -VE

                020000                  RCWD    =       B13

011012                                  RECV    PROC    RECV
 (1)
 (1)    011012                          RECV:
011012  005067  006674                          CLR     RCVEV           ;**ENTRY POINT**
011016  004767  000040                          JSR     PC,DBFCLR       ;CLR RCVR EVENT FLAG
011022                                          BDINIT  RCVR            ;CLEAR RCVR DATA BUFFER FIRST
011030  012777  022736  005100                  MOV     #RCBUF,@RDBA    ;CLR HARDWARE (REGARDLESS)
011036  012777  176400  005070                  MOV     #-1400,@RDBC    ;LOAD RCVR BUS ADDRESS
011044  012777  020000  005054                  MOV     #RCWD,@RCR      ;LOAD BYTE COUNT FOR 600 WORDS
011052  052777  000100  005046                  BIS     #B06,@RCR       ;SET RCV WORD IN RCVR
011060                                          RETURN                  ;SET RCVR INTERRUPT ENABLE


011062                                  DBFCLR: REGSAV                  ;SAVE R0...R5
011066  012700  022736                          MOV     #RCBUF,R0       ;R0 POINTS AT BUFFER AREA
011072  012701  000700                          MOV     #700,R1         ;R1 HOLDS BUFFER SIZE
011076  005020                          DBCLP:  CLR     (R0)+           ;CLEAR BUFF LOC
011100  005301                                  DEC     R1              ;DONE ?
011102  003375                                  BGT     DBCLP           ;NO,LOOP
011104                                          REGRES                  ;YES, RESTORE R0...R5
011110  000207                                  RTS     PC              ;AND RETURN
```

```
                                        ;;;RECEIVER INTERRUPT HANDLER.
                                        
                100000                  RCER    =       100000
                000400                  RCDOR   =       400
                000200                  RCSUC   =       200
                000040                  RCRCM           40
                                        
011112                          RCVINT: REGSAV                  ;;;SAVE R0...R5
011116  042777  000100  005002          BIC     #B06,@RCR       ;;;CLEAR RCVR INTR ENABLE
011124  032777  100000  004776          BIT     #RCER,@RSR      ;;;IS THERE A HARDWARE ERROR?
011132  001402                          BEQ     RCVINA          ;;;NO,CHECK DATA OUTPUT READY
011134  000167  000100                  JMP     RERINT          ;;;HANDLE ERROR INTERRUPT
011140  032777  000400  004762  RCVINA: BIT     #RCDOR,@RSR     ;;;IS DATA OUTPUT READY?
011146  001402                          BEQ     RCVINB          ;;;NO, CHECK SUC TXF
011150  000167  000500                  JMP     RDOINT          ;;;HANDLE DATA OUTPUT RDY INTERRUPT
011154  032777  000200  004746  RCVINB: BIT     #RCSUC,@RSR     ;;;SUCCESSFUL TRANSFER SET?
011162  001402                          BEQ     RCVINC          ;;;NO, CHECK RECECT COMPLETED
011164  000167  000536                  JMP     RSUINT          ;;;HANDLE SUC TXF INTERRUPT
011170  032777  000040  004732  RCVINC: BIT     #RCRCM,@RSR     ;;;REJECT COMPLETED INTERRUPT?
011176  001402                          BEQ     RCVIND          ;;;NO,ERRONEOUS INTERRUPT
011200  000167  000776                  JMP     RRJINT          ;;;HANDLE REJ-COM INTERRUPT
011204                          RCVIND: ERROR   \N              ;;;ERROR:UNKNOWN RECEIVER INTERRUPT OCCURRED
 (1)                                                            ;**** RCVR ERROR 16 *****
 (1)
 (1)
011224  012767  100000  006460          MOV     #B15,RCVEV      ;;;SET RCVR EVENT FLAG
011232                                  REGRES                  ;;;RESTORE R0...R5
011236  000002                          RTI                     ;RETURN
```

```
                                        ;RCVR ERROR INTERRUPT ROUTINE
1589
1590
1591  011240  032777  040000  004662  RERINT: BIT     #B14,@RSR       ;IS ERROR NON EXST LOC ERR?
1592  011246  001412                         BEQ     RERRA
1593  011250                                 ERROR   \N              ;ERROR:NON EXST LOC ERR IN RCVR
 (1)                                                                   ;**** RCVR ERROR 17 *****
 (1)
 (1)
1594  011270  000167  000340                 JMP     RERRX
1595  011274  032777  020000  004626  RERRA: BIT     #B13,@RSR       ;IS ERROR MEM OFLO?
1596  011302  001412                         BEQ     RERRB
1597  011304                                 ERROR   \N              ;ERROR:MEM OFLO ERROR IN RCVR
 (1)                                                                   ;**** RCVR ERROR 20 *****
 (1)
 (1)
1598  011324  000167  000304                 JMP     RERRX
1599  011330  032777  010000  004572  RERRB. BIT     #B12,@RSR       ;ANY TRANSMISSION ERRORS?
1600  011336  001457                         BEQ     RERRC
1601  011340  017767  004564  006416         MOV     @RSR,RSPC       ;GET RESPONSE CODES
1602  011346  042767  177760  006410         BIC     #-20,RSPC       ;REMOVE GARBAGE
1603  011354  032767  000003  006402         BIT     #3,RSPC         ;LOOK AT TXM RSP CODES
1604  011362  001012                         BNE     RERRB1
1605  011364                                 ERROR   \N              ;ERROR:TXM WENT OFF LINE
 (1)                                                                   ;**** RCVR ERROR 21 *****
 (1)
 (1)
1606  011404  000167  000224                 JMP     RERRX
1607  011410  042767  000003  006346  RERRB1: BIC     #3,RSPC         ;REMOVE TXM RSP CODES NOW
1608  011416  022767  000010  006340         CMP     #10,RSPC        ;WAS THERE A CRC ERROR?
1609  011424  001012                         BNE     RERRB2
1610  011426                                 ERROR   \N              ;ERROR:RCVR CRC ERROR OCCURRED
 (1)                                                                   ;**** RCVR ERROR 22 *****
 (1)
 (1)
1611  011446  000167  000162                 JMP     RERRX
1612  011452                         RERRB2: ERROR   \N              ;ERROR:FIRST WORD RECVD NOT VALID
 (1)                                                                   ;**** RCVR ERROR 23 *****
 (1)
 (1)
1613  011472  000167  000136                 JMP     RERRX
1614  011476  032777  004000  004424  RERRC: BIT     #B11,@RSR       ;WAS ERROR A PARITY ERROR?
1615  011504  001412                         BEQ     RERRD
1616  011506                                 ERROR   \N              ;ERROR:RCVR DETECTED INVALID PARITY
 (1)                                                                   ;**** RCVR ERROR 24 *****
 (1)
 (1)
1617  011526  000167  000102                 JMP     RERRX
1618  011532  032777  002000  004370  RERRD: BIT     #B10,@RSR       ;DID RCVR TIME OUT?
1619  011540  001412                         BEQ     RERRE
1620  011542                                 ERROR   \N              ;ERROR:RCVR TIMEOUT ERROR OCCURRED
 (1)                                                                   ;**** RCVR ERROR 25 *****
 (1)
 (1)
1621  011562  000167  000046                 JMP     RERRX
1622  011566  032777  001000  004334  RERRE: BIT     #B09,@RSP       ;WAS IT BYTE COUNT OVERFLOW?
1623  011574  001417                         BEQ     RERRX
```

```
 1624  011576  042777  040000  004322          BIC     #B14,@RCR       ;CLR RCVR NPR                          *NEW*
 1625  011604  042777  001000  004316          BIC     #B09,@RSR       ;CLR RCVR BYTE COUNT OVERFLOW          *NEW*
 1626  011612  052777  100000  004306          BIS     #B15,@RCR       ;SET 'REJECT' IN RECEIVER              *NEW*
 1627  011620  052777  000100  004300          BIS     #B06,@RCR       ;RE-ENABLE INTERRUPT                   *NEW*
 1628  011626                          RERTRN: REGRES                  ;RESTORE R0...R5
 1629  011632  000002                          RTI                     ;RETURN
 1630  011634                          RERRX:  BDINIT  RCVR            ;CLEAR RECVR
 1631  011642  012767  100000  006042          MOV     #B15,RCVEV      ;SET RCV EVENT FLAG
 1632  011650  000167  177752                  JMP     RERTRN
 1633
```

```
                                        ;RCVR DATA OUTPUT RDY INTERRUPT ROUTINE
                                        
011654  017767  004252  006062  RDOINT: MOV     @RDDB,RCSEED        ;GET FLAG-WORD FOR DATA SEED
011662  022767  170000  006054          CMP     #170000,RCSEED      ;LOOK AT DATA SEED
011670  101011                          BHI     RDOINC              ;IF MORE -VE THAN -10000
011672  052777  100000  004226          BIS     #B15,@RCR           ;       SET REJECT IN RCVR
011700                  RDORTN: REGRES                              ;RESTORE R0...R5
011704  052777  000100  004214          BIS     #B06,@RCR           ;RE-SET INTR ENAB
011712  000002                          RTI                         ;RETURN
011714  052777  040001  004204  RDOINC: BIS     #B14+B00,@RCR       ;START RECEIVER GOING
011722  000167  177752                  JMP     RDORTN
                                        
                                        ;RCVR SUC TXF  INTERRUPT ROUTINE
                                        
011726  032777  000040  004174  RSUINT: BIT     #RCRCM,@RSR         ;WAS RECOM SET?
011734  001403                          BEQ     RSUCNT              ;NO, CONTINUE
011736  042777  000040  004164          BIC     #RCRCM,@RSR         ;YES, CLR IT AND CONTINUE
011744  016701  005774  RSUCNT: MOV     RCSEED,R1           ;GET DATA SEED
011750  042701  177000          BIC     #177000,R1          ;MASK MSG LENGTH
011754  001002                  BNE     RSNT0               ;IF NOT 0, OKAY
011756  012701  000001          MOV     #1,R1               ;IF 0, MAKE IT 1
011762  162701  000001  RSNT0:  SUB     #1,R1               ;ACCOUNT FOR FLAG WORD
011766  020127  000600          CMP     R1,#600             ;IS IT GREATER THAN 600?
011772  003402                  BLE     RSFLSZ              ;NO, LEAVE IT ALONE
011774  012701  000600          MOV     #600,R1             ;YES, SET IT TO LIMIT (600)
012000  012700  001400  RSFLSZ: MOV     #1400,R0
012004  067700  004124          ADD     @RDBC,R0            ;DETERMINE # OF BYTES RECEIVED
012010  001451                  BEQ     RSULV               ;NOTHING TO CHECK, EXIT
012012  006200                  ASR     R0                  ;R0 = NO. OF WORDS RECEIVED
012014  020001                  CMP     R0,R1               ;DID WE RECV ALL THE WORDS?
012016  001412                  BEQ     RSUTST              ;YES, CARRY ON
012020  103456                  BLO     ERNE                ;NOT ENOUGH !
012022                          ERROR   \N          ;ERROR:RCVR GOT TOO MANY WORDS
 (1)                                                        ;**** RCVR ERROR 26 *****
 (1)
 (1)                                    BR      RSULV               ;EXIT IF ERROR
012042  000434  012705  022736  RSUTST: MOV     #RCBUF,R5           ;R5 POINTS AT RECVD DATA
012044          012050  RSUGEN: MULP    RANM,RCSEED
012072  066767  005662  005644          ADD     RANK,RCSEED         ;GENERATE CHECK WORD
012100  026725  005640                  CMP     RCSEED,(R5)+        ;COMPARE IT WITH RECV'D WORD
012104  001411                          BEQ     RSUCHK
012106                          ERROR   \N          ;ERROR:DATA WORD RECV'D WAS BAD
 (1)                                                        ;**** RCVR ERROR 27 *****
 (1)
 (1)                                    BR      RSULV               ;EXIT IF ERROR
012126  000402  005300  RSUCHK: DEC     R0                  ;CHECKED ALL WORDS?
012130  001346                  BNE     RSUGEN              ;NO,CONTINUE
012134          RSULV:  BDINIT  RCVR                        ;CLR HARDWARE
012142                  REGRES                              ;RESTORE R0...R5
012146  012767  100000  005536          MOV     #B15,RCVEV          ;SET RCVR EVENT FLAG
012154  000002                          RTI                         ;RETURN
                                        
012156                  ERNE:   ERROR   \N          ;ERROR: RCVR GOT TOO FEW WORDS
 (1)
```

```
                                                                ;**** RCVR ERROR 30 *****
  (1)
  (1)
1684  012176  000167  177732          JMP     RSULV             ;EXIT IF ERROR
1685
1686                                ;RCVR REJECT INTERRUPT ROUTINE
1687
1688  012202                        RRJINT: BDINIT  RCVR        ;CLEAR HARDWARE
1689  012210  012767  100000  005474        MOV     #B15,RCVEV  ;SET RCVR EVENT FLAG
1690  012216                                REGRES                ;RESTORE R0...R5
1691  012222  000002                       RTI                   ;RETURN
1692
1693                                ;KW11 CLOCK INTERRUPT ROUTINE
1694                                ;UPDATE TICKS. IF - 60. UPDATE SECONDS.
1695                                ;IF SECS = 60, UPDATE MINUTS. IF MINUTS = 60
1696                                ;UPDATE HOURS.
1697
1698  012224  042777  000200  003726  CLKINT: BIC     #B07,@LCS   ;JUST IN CASE IT MATTERS.
1699  012232  105267  005414                INCB    TICKS       ;UPDATE TICKS
1700  012236  126727  005410  000074        CMPB    TICKS,#60.  ;GET TO 60 YET?
1701  012244  103424                        BLO     CLKEXT      ;NO
1702  012246  105067  005400                CLRB    TICKS       ;YES, CLR TICKS
1703  012252  105267  005375                INCB    SECNDS      ;& UPDATE SECONDS
1704  012256  126727  005371  000074        CMPB    SECNDS,#60. ;GET 60 SECS?
1705  012264  103414                        BLO     CLKEXT      ;NO
1706  012266  105067  005361                CLRB    SECNDS      ;YES, CLR SECNDS
1707  012272  105267  005356                INCB    MINUTS      ;& UPDATE MINUTES
1708  012276  126727  005352  000074        CMPB    MINUTS,#60. ;GET 60 MINUTES?
1709  012304  103404                        BLO     CLKEXT      ;NO
1710  012306  105067  005342                CLRB    MINUTS      ;YES, CLEAR MINUTS
1711  012312  005267  005340                INC     HOURS       ;& UPDATE HOURS
1712  012316  000002                CLKEXT: RTI                   ;RETURN
```

```
                        .SBTTL STATUS MODULE

                ; DUMP STATUS TABLE INDICATING ACTIVATED RECEIVERS
                ;  SHOW NO. OF ATTEMPTED CONNECTIONS TO EACH ACTIVE RECEIVER
                ;  AND NO. OF SUCCESSFUL CONNECTIONS TO EACH ACTIVE RECEIVER.
                ;  QUANTITIES ARE IN DECIMAL AND ARE CAPABLE OF EXPANDING TO
                ;  655,359,999. ANY QUANTITY HIGHER THAN THAT AMOUNT WILL BE
                ;  REPRESENTED AS ''*********''. RECEIVER ADDRESS  NUMBERS ARE
                ;  PRINTED IN OCTAL
                ;
1718
1719
1720
1721
1722
1723
1724
1725
1726  012320                    PROC    STATUS
(1)
(1)   012320            STATUS:
1727  012320  005767  004310           TST     TOQ             ;**ENTRY POINT**
1728  012324  001402                   BEQ     60$             ;IS TTY OUT QUEUE EMPTY?
1729  012326  000167  000540           JMP     STRTN           ;NO, RETURN TO WHEREVER
1730  012332  005767  005342   60$:    TST     STPNTR          ;IS OUTPUT POINTER AT TABLE?
1731  012336  001504                   BEQ     STHDR           ;NO, GO PRINT HEADER
1732  012340  026727  005334  020674   CMP     STPNTR,#RADEND  ;IS POINTER AT END OF TABLE?
1733  012346  002402                   BLT     59$             ;YES, DONE; EXIT
1734  012350  000167  000476           JMP     STARXT
1735  012354  005777  005320   59$:    TST     @STPNTR         ;CHECK RCV ADDR ACTIVE FLAG
1736  012360  001002                   BNE     58$
1737  012362  000167  000452           JMP     STBADV          ;ADVANCE TABLE IF FLAG CLR
1738  012366  012700  032043   58$:    MOV     #STLIN1,R0      ;OUTPUT POINTER FOR OCTJSP
1739  012372  062767  000002  005300   ADD     #2,STPNTR
1740  012400  017701  005274           MOV     @STPNTR,R1      ;OCTJSP DUMPS DATA IN STPNTR
1741  012404  012702  177777           MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1742  012410  004767  020250           JSR     PC,OCTJSP
1743  012414  062767  000002  005256   ADD     #2,STPNTR
1744  012422  012700  032065           MOV     #STLIN2,R0      ;OUTPUT POINTER FOR CDDMG
1745  012426  016701  005246           MOV     STPNTR,R1       ;OUTPUT WORD FOR CDDMG
1746  012432  012702  177777           MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1747  012436  004767  020422           JSR     PC,CDDMG
1748  012442  020027  032112   STYHR:  CMP     R0,#STLIN2+25   ;BLANK REST OF FIELD OUT
1749  012446  103003                   BHIS    STYON           ;BY INSERTING SPACES
1750  012450  112720  000040           MOVB    #' ,(R0)+
1751  012454  000772                   BR      STYHR
1752  012456  062767  000004  005214   ADD     #4,STPNTR
1753  012464  012700  032113   STYON:  MOV     #STLIN3,R0      ;OUTPUT POINTER FOR CDDMG
1754  012470  016701  005204           MOV     STPNTR,R1       ;OUTPUT WORD FOR CDDMG
1755  012474  012702  177777           MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1756  012500  004767  020360           JSR     PC,CDDMG
1757  012504  020027  032127   STYTHR: CMP     R0,#STLIN3+14   ;BLANK REST OF FIELD OUT
1758  012510  103003                   BHIS    STYONT          ;BY INSERTING SPACES
1759  012512  112720  000040           MOVB    #' ,(R0)+
1760  012516  000772                   BR      STYTHR
1761  012520                   STYONT: CALL    PNTLIN,<STLIN>  ;ENQUEUE STATUS LINE FOR OUTPUT
1762  012536  062767  000004  005134   ADD     #4,STPNTR       ;UPDATE POINTER FOR NEXT LINE
1763  012544  000167  000322           JMP     STRTN           ;RETURN
1764
1765  012550  132777  000001  003344   STHDR:  BITB    #1,@TMMRH       ;IS MASTER SET HERE?
1766  012556  001407                   BEQ     1$              ;NO.
1767  012560                           CALL    PNTLIN,<THUMST> ;YES, TELL OPERATOR
```

```
1768  012576  132777  000002  003316  1$:     BITB    #2,@TMMRH       ;WELL, IS SECONDARY SET?
1769  012604  001407                          BEQ     2$              ;NO.
1770  012606                                  CALL    PNTLIN,<THUSCN> ;YES, TELL OPERATOR
1771  012624  032767  100000  174534  2$:     BIT     #B15,TXMST      ;IS 'RIB' SET?
1772  012632  001410                          BEQ     3$              ;NO, TELL OPERATOR
1773  012634                                  CALL    PNTLIN,<RBSTMG> ;YES, TELL OPERATOR
1774  012652  000407                          BR      4$
1775  012654                          3$:     CALL    PNTLIN,<RBCLMG>
1776  012672  005767  004762          4$:     TST     KWFLG           ;GOT A CLOCK?
1777  012676  001444                          BEQ     5$              ;NO, FORGET ABOUT TIME.
1778  012700  005002                          CLR     R2              ;SUPPRESS LEADING ZEROS
1779  012702  012700  031730                  MOV     #TMLIN1,R0      ;OUTPUT POINTER FOR DECPNT
1780  012706  016701  004744                  MOV     HOURS,R1        ;OUTPUT DATA FOR DECPNT
1781  012712  004767  017770                  JSR     PC,DECPNT       ;ENQUEUE 'HOURS'
1782  012716  005002                          CLR     R2              ;SUPPRESS LEADING ZEROS
1783  012720  112720  000072                  MOVB    #':,(R0)+       ;INSERT COLON
1784  012724  116701  004724                  MOVB    MINUTS,R1       ;GET 'MINUTES' FOR DECPNT
1785  012730  004767  017752                  JSR     PC,DECPNT       ;ENQUEUE 'MINUTES'
1786  012734  005002                          CLR     R2              ;SUPPRESS LEADING ZEROS
1787  012736  112720  000072                  MOVB    #':,(R0)+       ;INSERT ANOTHER COLON
1788  012742  116701  004705                  MOVB    SECNDS,R1       ;GET 'SECONDS' FOR DECPNT
1789  012746  004767  017734                  JSR     PC,DECPNT       ;ENQUEUE 'SECONDS'
1790  012752  005002                          CLR     R2              ;SUPPRESS LEADING ZEROS
1791  012754  112720  000072                  MOVB    #':,(R0)+       ;INSERT ANOTHER COLON
1792  012760  116701  004666                  MOVB    TICKS,R1        ;GET 'TICKS' FOR DECPNT
1793  012764  004767  017716                  JSR     PC,DECPNT       ;ENQUEUE 'TICKS'
1794  012770  105010                          CLRB    (R0)            ;ENQUEUE '0' PRINT TERMINATOR.
1795  012772                                  CALL    PNTLIN,<ELPSTM> ;PRINT 'ELAPSED TIME'
1796  013010                          5$:     CALL    PNTLIN,<STITLE> ;ENQUEUE STATUS HEADER
1797  013026  012767  020110  004644          MOV     #RADB,STPNTR    ;SET POINTER TO TOP OF STAT LIST
1798  013034  000167  000032                  JMP     STRTN           ;RETURN
1799
1800  013040  062767  000014  004632  STBADV: ADD     #14,STPNTR      ;SET POINTER TO NEXT ACTIVE FLAG
1801  013046  000167  000020                  JMP     STRTN
1802
1803  013052  005067  004620          STARXT: CLR     STSEV           ;CLR STATUS EVENT FLAG
1804  013056                                  CALL    PRESC           ;FAKE CNTRL-C KEY
1805  013072                          STRTN:  RETURN                  ;RETURN
```

```
                                        .SBTTL TRANSMITTER ERRORS MODULE

                                ;DUMP A TABLE OF TRANSMITTER ERRORS INCLUDING:
                                ;  ERROR NO.,  CONCT'D RCVR, AND ERROR COUNT FOR
                                ;  EACH OF THE TRANSMITTER ERRORS.
                                ;IF THE ERROR COUNT FOR ANY ENTRY IS 0, THAT ERROR NO. IS NOT REPERTED
                                ;IF, AFTER CHECKING THE ENTIRE XMTR ERROR TABLE, NO ERRORS
                                ; ARE PRINTED, SIMPLY PRINT (NONE).


1817  013074                            PROC    TEROS
 (1)
 (1)  013074                    TEROS:                          ;**ENTRY POINT**
1818  013074  005767  003534           TST     TOQ             ;IS THE TTY OUTPUT QUEUE EMPTY
1819  013100  001402                   BEQ     60$             ;YES, CONTINUE
1820  013102  000167  000324           JMP     TERTN           ;NO, EXIT
1821  013106  005767  004614   60$:    TST     HDRFLG          ;HEADER PRINTED YET?
1822  013112  001422                   BEQ     1$              ;YES, SKIP IT
1823  013114  005067  004606           CLR     HDRFLG          ;NO, PRINT IT NOW.
1824  013120                           CALL    PNTLIN,<TERHDR>
1825  013136                           CALL    PNTLIN,<TRHLIN>
1826  013154  000167  000252           JMP     TERTN           ;AND RETURN
1827  013160  026727  004614  000016 1$:    CMP     ERR0,#16        ;IS THIS ERROR # 16?
1828  013166  001042                   BNE     3$              ;NOT YET.
1829  013170  005067  004466           CLR     TEREV           ;YES, CLEAR XMTR ERROR EVENT
1830  013174  052767  100000  004462   BIS     #B15,REREV      ;SET RCVR ERROR EVENT
1831  013202  052767  100000  004516   BIS     #B15,HDRFLG     ;SET HEADER FLAG
1832  013210  012767  000001  004564   MOV     #1,ERR1         ;LOAD INITIAL XMTR #
1833  013216  012767  026626  004560   MOV     #RERTBL,ERR2    ;POINT RCVR ERR EVENT AT ITS TABLE
1834  013224  005767  004474           TST     PRINTD          ;DID WE PRINT ANYTHING?
1835  013230  001007                   BNE     2$              ;YES, O.K.
1836  013232                           CALL    PNTLIN,<NONMG>  ;NO, PRINT "(NONE)"
1837  013250                   2$:     CALL    PNCRLF          ;NEW LINE
1838  013264  005067  004434           CLR     PRINTD
1839  013270  000167  000136           JMP     TERTN           ;RETURN
1840  013274  005777  004504   3$:     TST     @ERR2           ;ANY ERRORS?
1841  013300  001434                   BEQ     4$              ;NO.
1842  013302  012700  032345           MOV     #TRRNO,R0       ;READY TO PRINT ERROR NO.
1843  013306  016701  004466           MOV     ERR0,R1
1844  013312  004767  017346           JSR     PC,OCTJSP       ;ERROR NO. IS IN OCTAL
1845  013316  012700  032362           MOV     #TRRCN,R0       ;READY TO PRINT RCVR NO.
1846  013322  016701  004454           MOV     ERR1,R1
1847  013326  004767  017332           JSR     PC,OCTJSP       ;RCVR NO. IS ALSO IN OCTAL
1848  013332  012700  032401           MOV     #TRERC,R0       ;NOW PRINT ERROR COUNT
1849  013336  017701  004442           MOV     @ERR2,R1
1850  013342  004767  017332           JSR     PC,DECJSP       ;ERROR COUNT IS IN DECIMAL
1851  013346                           CALL    PNTLIN,<TRELIN> ;ENQUEUE TABLE OUTPUT LINE
1852  013364  052767  100000  004332   BIS     #B15,PRINTD     ;SET 'PRINTED FLAG"
1853  013372  062767  000002  004404 4$:     ADD     #2,ERR2         ;UPDATE TABLE POINTER
1854  013400  026727  004376  000037   CMP     ERR1,#37        ;ALL RCVR'S DONE?
1855  013406  001404                   BEQ     5$              ;YES.
1856  013410  005267  004366           INC     ERR1            ;NO, DO NEXT
1857  013414  000167  000012           JMP     TERTN
1858  013420  012767  000001  004354 5$:     MOV     #1,ERR1         ;NEXT ERROR NO.
1859  013426  005267  004346           INC     ERR0
1860  013432                   TERTN:  RETURN
```

```
                                        .SBTTL RECEIVER ERRORS MODULE

                                ; DUMP A TABLE OF RECEIVER ERRORS, INCLUDING:
                                ;       ERROR NO., CONCT'D XMTR, AND ERROR COUNT
                                ;FOR EACH OF THE RECEIVER ERRORS.
                                ; IF THE ERROR COUNT FOR ANY ENTRY IS 0, THAT ERROR NO. IS NOT
                                ;REPORTED. IF, AFTER CHECKING THE ENTIRE RCVR ERROR TABLE, NO
                                ;ERRORS ARE PRINTED, SIMPLY PRINT ''(NONE)''.

013434                                  PROC    REROS
 (1)
 (1)  013434                    REROS:                                  ;**ENTRY POINT**
013434  005767  003174                  TST     TOQ                     ;IS TTY OUTPUT QUEUE EMPTY?
013440  001402                          BEQ     60$                     ;YES, CONTINUE
013442  000167  000310                  JMP     RERTN                   ;NO, RETURN
013446  005767  004254          60$:    TST     HDRFLG                  ;IS HEADER PRINTED YET?
013452  001422                          BEQ     1$                      ;YES, SKIP IT
013454  005067  004246                  CLR     HDRFLG                  ;NO, CLEAR FLAG
013460                                  CALL    PNTLIN,<RERHDR>         ;AND PRINT HEADER
013476                                  CALL    PNTLIN,<RRHLIN>
013514  000167  000236                  JMP     RERTN                   ;AND RETURN
013520  026727  004254  000031    1$:   CMP     ERR0,#31                ;DONE LAST ERROR?
013526  001034                          BNE     3$                      ;NOT YET
013530  005067  004130                  CLR     REREV                   ;YES, CLEAR RCVR ERROR EVENT
013534  005767  004164                  TST     PRINTD                  ;DID WE PRINT ANYTHING?
013540  001007                          BNE     2$                      ;YES, O.K.
013542                                  CALL    PNTLIN,<NONMG>          ;NO, PRINT ''(NONE)''
013560                          2$:     CALL    PNCRLF
013574  005067  004124                  CLR     PRINTD
013600                                  CALL    PRESC                   ;WRAP IT UP!
013614  000167  000136                  JMP     RERTN
013620  005777  004160          3$:     TST     @ERR2                   ;ANY ERRORS?
013624  001434                          BEQ     4$                      ;NO.
013626  012700  032502                  MOV     #RCRNO,R0               ;READY TO PRINT ERROR NO.
013632  016701  004142                  MOV     ERR0,R1
013636  004767  017022                  JSR     PC,OCTJSP               ;ERROR NO.S ARE IN OCTAL
013642  012700  032517                  MOV     #RCTRN,R0               ;READY TO PRINT XMTR NO.
013646  016701  004130                  MOV     ERR1,R1
013652  004767  017006                  JSR     PC,OCTJSP               ;XMTR NO.S ARE IN OCTAL
013656  012700  032536                  MOV     #RCERC,R0               ;NOW PRINT ERROR COUNT
013662  017701  004116                  MOV     @ERR2,R1
013666  004767  017006                  JSR     PC,DECJSP               ;ERROR COUNTS ARE IN DECIMAL
013672                                  CALL    PNTLIN,<RCELIN>         ;ENQUEUE TABLE OUTPUT LINE
013710  052767  100000  004006          BIS     #B15,PRINTD             ;SET 'PRINTED'' FLAG
013716  062767  000002  004060  4$:     ADD     #2,ERR2                 ;UPDATE TABLE POINTER
013724  026727  004052  000037          CMP     ERR1,#37                ;ALL XMTRS DONE?
013732  001404                          BEQ     5$
013734  005267  004042                  INC     ERR1                    ;NO, DO NEXT
013740  000167  000012                  JMP     RERTN
013744  012767  000001  004030  5$:     MOV     #1,ERR1                 ;YES, NEXT ERROR
013752  005267  004022                  INC     ERR0
013756                          RERTN:  RETURN
```

```
                                        .SBTTL SUMMARY MODULE

                                ; DUMP SUMMARY TABLE INDICATING ERROR NUMBERS  AND THE LOCATIONS
                                ;  OF THE ERRORS AS WELL AS THE NUMBER OF OCCURRENCES OF EACH OF
                                ;  THE ERRORS. ERRORS WITH 0 OCCURRENCES WILL NOT BE LISTED. THE
                                ;  ENTRY UNDER 'NO. OF OCCURRENCES'' IS IN DECIMAL AND IS CAPABLE
                                ;  OF  EXPANDING TO  65,528.  QUANTITIES OF ERRORS IN  EXCESS OF
                                ;  THIS WILL BE LISTED AS 'MXCNT''.
                                ;

013760                                  PROC    SUMRY
(1)
(1)  013760                     SUMRY:
013760  005767  002650                  TST     TOQ             ;**ENTRY POINT**
013764  001126                          BNE     SMRTN           ;IS OUTPUT QUEUE EMPTY?
013766  005767  003710                  TST     SUMPNT          ;NO, RETURN
013772  001472                          BEQ     SUMHDR          ;IS OUTPUT POINTER AT TABLE?
013774  027727  003702  177777  SMCHK:  CMP     @SUMPNT,#-1     ;NO, GO PRINT HEADER
014002  001507                          BEQ     SMARXT          ;IS ERROR # - -1 ?
014004  016704  003672                  MOV     SUMPNT,R4       ;YES,DONE; EXIT
014010  005764  000004                  TST     4(R4)           ;CHECK TOTAL COUNT OF THIS ERR
014014  001475                          BEQ     SMADV           ;IF 0, TRY NEXT ERROR
014016  012700  032557                  MOV     #SMLIN1,R0      ;R0 IS OUTPUT POINTER FOR OCTJSP
014022  011401                          MOV     (R4),R1         ;R1 IS DATA BUFF FOR OCTJSP
014024  012702  177777                  MOV     #-1,R2          ;DON'T COMPRESS BLANKS
014030  004767  016630                  JSR     PC,OCTJSP
014034  012700  032601                  MOV     #SMLIN2,R0
014040  016401  000002                  MOV     2(R4),R1
014044  012702  177777                  MOV     #-1,R2          ;DON'T COMPRESS BLANKS
014050  004767  016610                  JSR     PC,OCTJSP
014054  012700  032625                  MOV     #SMLIN3,R0
014060  016401  000004                  MOV     4(R4),R1
014064  020127  177770                  CMP     R1,#177770      ;REACHED MAX COUNT?
014070  101415                          BLOS    SMPN            ;NO,OK
014072  112720  000115                  MOVB    #'M,(R0)+       ;YES, PRINT 'MXCNT'' IN
014076  112720  000130                  MOVB    #'X,(R0)+       ;  NOTE THAT 'MXCNT''
014102  112720  000103                  MOVB    #'C,(R0)+       ;  ALSO HAS ONLY 5
014106  112720  000116                  MOVB    #'N,(R0)+       ;  ASCII CHARACTERS.
014112  112720  000124                  MOVB    #'T,(R0)+
014116  000404                          BR      SMSPN
014120  012702  177777                  MOV     #-1,R2          ;DON'T COMPRESS BLANKS
014124  004767  016550          SMPN:   JSR     PC,DECJSP
014130                          SMSPN:  CALL    PNTLIN,<SMLIN>  ;ENQUEUE SUM LINE FOR OUTPUT
014146  062767  000006  003526          ADD     #6,SUMPNT       ;UPDATE POINTER FOR NEXT LINE
014154  000167  000062                  JMP     SMRTN           ;RETURN

014160                          SUMHDR: CALL    PNTLIN,<SMTTLE> ;ENQUEUE SUMMARY HEADER
014176  012767  024736  003476          MOV     #ERTBL,SUMPNT   ;SET POINTER TO TOP OF SUM LIST
014204  000167  000032                  JMP     SMRTN           ;RETURN

014210  062767  000006  003464  SMADV:  ADD     #6,SUMPNT       ;SET POINTER TO NEXT ENTRY
014216  000167  177552                  JMP     SMCHK

014222  005067  003444          SMARXT: CLR     SUMEV           ;CLEAR SUMMARY EVENT FLAG
014226                                  CALL    PRESC           ;FAKE CNTRL-C KEY
```

```
  1967  014242                          SMRTN:  RETURN                          ;RETURN
```

```
                        .SBTTL ERROR UPDATE ROUTINES
                        .SBTTL  TRANSMITTER ERRORS

                        ;CALLED BY THE MACRO 'CALL ERRMOD,<P,ERADR>'

                        PROC    ERRMOD,<ERRNUM,ERRADR>
000002          ERRNUM  -       2
000004          ERRADR  =       4

                        ERRMOD:                                 ;**ENTRY POINT**
014244 010046           MOV     R0,-(SP)                        ;SAVE R0 ON STACK
014246 010146           MOV     R1,-(SP)                        ; AND R1
014250 010246           MOV     R2,-(SP)                        ; AND R2
014252 010346           MOV     R3,-(SP)                        ;AND R3
014254 017767 001626 003512     MOV     @TCR,TEMP               ;SAVE ADDRESS OF CONNECTED RCVR
014262 042767 160377 003504     BIC     #160377,TEMP
014270 000367 003500            SWAB    TEMP                    ;GET IT IN THE RIGHT BYTE
014274 016500 000002            MOV     ERRNUM(R5),R0           ;GET ERROR NUMBER
014300 010067 003472            MOV     R0,TEMP1                ;SAVE ERROR NUMBER
014304 006300                   ASL     R0                      ;FIND TABLE ENTRY
014306 010001                   MOV     R0,R1                   ;       IN ORDER TO UPDATE
014310 006300                   ASL     R0                      ;       ADDRESS FIELD
014312 060100                   ADD     R1,R0
014314 062700 024730            ADD     #ERTBL0,R0
014320 016560 000004 000002     MOV     ERRADR(R5),2(R0)        ;ENTER ERROR ADDRESS
014326 026027 000004 177771     CMP     4(R0),#177771           ;AT MAX COUNT YET?
014334 103401                   BLO     ERMIN                   ;NO, INCREMENT COUNT
014336 000444                   BR      ERMIS                   ;YES SKIP UPDATE.
014340 005260 000004            ERMIN:  INC     4(R0)           ;UPDATE OCCURRENCES
014344 005367 003426            DEC     TEMP1                   ; (ERR # - 1)
014350                          MULT    37,TEMP1,R3             ; (E-1)X37
014414                          MULT    2,TEMP1,R3              ; [(E-1)X37]X2
014420 005367 003350            DEC     TEMP                    ; RCVR ADDR - 1
014424                          MULT    2,TEMP,R3               ; (R-1)X2
014430 066767 003342 003336     ADD     TEMP1,TEMP              ; [(E-1)X37]X2 + (R-1)X2
014436 062767 025160 003330     ADD     #ERTBL,TEMP             ;TEMP = #TERTBL + <[(E-1)X37]X2 + (R-1)X2>
014444 005277 003324            INC     @TEMP                   ;UPDATE TABLE ENTRY FOR THIS ERROR
014450 012603                   ERMIS:  MOV     (SP)+,R3        ;RESTORE R3
014452 012602                   MOV     (SP)+,R2                ;RESTORE R2
014454 012601                   MOV     (SP)+,R1                ;RESTORE R1
014456 012600                   MOV     (SP)+,R0                ;RESTORE R0
014460                          RETURN                          ;RETURN
```

```
                                        .SBTTL  RECEIVER ERRORS

                                ;CALLED BY THE MACRO 'CALL      ERRMOR,<P,ERADR>'

                                        PROC    ERRMOR,<ERRNUM,ERRADR>
000002                          ERRNUM  =       2
000004                          ERRADR          4

                                ERRMOR:                                 ;**ENTRY POINT**
                                        MOV     R0,-(SP)                ;SAVE R0 ON THE STACK
                                        MOV     R1,-(SP)                ;AND R1
                                        MOV     R2,-(SP)                ;AND R2
                                        MOV     R3,-(SP)                ;AND R3
014462 010046 001430 003274             MOV     @RCR,TEMP               ;GET ADDR OF CONNECTED XMTR
014500 042767 160377 003266             BIC     #160377,TEMP
014506 000367 003262                    SWAB    TEMP                    ; INTO RIGHT BYTE
014512 016500 000002                    MOV     ERRNUM(R5),R0           ;GET ERROR NUMBER
014516 010067 003254                    MOV     R0,TEMP1                ;AND SAVE IT
014522 006300                           ASL     R0                      ; FIND "SUM" TABLE ENTRY
014524 010001                           MOV     R0,R1                   ; IN ORDER TO UPDATE
014526 006300                           ASL     R0                      ; ADDRESS FIELD
014530 060100                           ADD     R1,R0                   ; AND # OF OCCURRENCES
014532 062700 024730                    ADD     #ERTBL0,R0
014536 016560 000004 000002             MOV     ERRADR(R5),2(R0)        ;ENTER ERROR ADDRESS
014544 026027 000004 177771             CMP     4(R0),#177771           ;AT MAX COUNT YET?
014552 103401                           BLO     ERMINR                  ;NO, UPDATE ERROR COUNT
014554 000447                           BR      ERMISR                  ;YES, SKIP UPDATES
014556 005260 000004                ERMINR: INC     4(R0)                   ;UPDATE OCCURRENCES
014562 162767 000015 003206             SUB     #15,TEMP1               ;ALIGN ERRORS TABLE FOR RCVR
014570 005367 003202                    DEC     TEMP1                   ; ERR # - 1 (E-1)
014574                                  MULT    37,TEMP1,R3             ; (E-1)X37
014640                                  MULT    2,TEMP1,R3              ; [(E-1)X37]X2
014644 005367 003124                    DEC     TEMP                    ; XMTR ADDR - 1  (T-1)
014650                                  MULT    2,TEMP,R3               ; (T-1)X2
014654 066767 003116 003112             ADD     TEMP1,TEMP              ; [(E-1)X37]X2 + (T-1)X2
014662 062767 026626 003104             ADD     #RERTBL,TEMP            ; TEMP - #RERTBL+[(E-1)X37]X2 + (T-1)X2
014670 005277 003100                    INC     @TEMP                   ;UPDATE RCVR ERROR COUNT
014674 012603                       ERMISR: MOV     (SP)+,R3                ;RESTORE R3
014676 012602                           MOV     (SP)+,R2                ;RESTORE R2
014700 012601                           MOV     (SP)+,R1                ;RESTORE R1
014702 012600                           MOV     (SP)+,R0                ;RESTORE R0
014704                                  RETURN
```

```
                        .SBTTL UTILITY ROUTINES
                        .SBTTL  PROCESS AN INPUT CHARACTER FROM THE TTY

014706                  PROC    TTINP
(1)
(1)  014706         TTINP:                                  ;**ENTRY POINT**

014706                  CALL    DEQ,<FRELEM,TIQ>            ;GET TTY INPUT CHARACTER.
014726  142767  000200  013340  BICB    #200,FRELEM         ;CLEAR OFF PARITY BIT.
014734  116700  013334          MOVB    FRELEM,R0
014740  120027  000141          CMPB    R0,#141             ;CONVERT LOWER CASE INPUT
014744  002405                  BLT     TTINH               ;  TO REGULAR (UPPER CASE)
014746  120027  000172          CMPB    R0,#172             ;  LETTERS INSTEAD.
014752  003002                  BGT     TTINH
014754  042700  000040          BIC     #B05,R0             ;CONVERT CHARACTER
014760  010067  013310  TTINH:  MOV     R0,FRELEM           ; IN  'FRELEM'
014764  012701  016044          MOV     #CMCHTB,R1          ; TO ITS PROCESSING
014770  004767  015642          JSR     PC,PROCHR           ; ROUTINE.
014774  001407                  BEQ     CMMISC              ;IF MISC CHAR, CALL MISC PROC.
014776                          CALL    @R1,<FRELEM>        ;CALL PROCESSING ROUTINE.
015012  000407                  BR      TTRET               ;RETURN.

015014              CMMISC: CALL    PRMISC,<FRELEM>         ;PROCESS MISCELLANEOUS CHARACTER
015032              TTRET:  RETURN                          ;RETURN.
```

```
                        .SBTTL  TTY INPUT CHARACTER PROCESSING ROUTINES


                        ;
                        ; PROCESSOR FOR CARRIAGE RETURN
                        ; THIS CHARACTER SIGNIFIES THE END OF A COMMAND LINE.
                        ;
015034                          PROC    PRCR
 (1)
 (1)  015034            PRCR:                                   ;**ENTRY POINT**
2077  015034  105767  002610          TSTB    REQINP            ;IF INPUT NOT BEING TYPED,
2078  015040  002015                  BGE     CRRET             ; IGNORE CHARACTER.
2079  015042  112777  000015  001342  MOVB    #CR.,@CBUFPT      ;PUT CR IN BUFFER.
2080  015050  105367  002575          DECB    CMDENT            ;SET COMMAND ENTERED FLAG.
2081  015054  105067  002570          CLRB    REQINP            ;CLEAR INPUT REQUESTED FLAG.
2082  015060                          CALL    PNCRLF            ;ECHO CR & LF.
2083  015074  000207        CRRET:    RETURN                    ;RETURN.
2084
2085
2086                    ;
2087                    ; PROCESSOR FOR CONTROL-O CHARACTER.
2088                    ; ALL CHARACTERS PRESENTLY IN THE TTY OUTPUT QUEUE ARE THROWN AWAY.
2089                    ;
015076                          PROC    PRCTLO
 (1)
 (1)  015076            PRCTLO:                                 ;**ENTRY POINT**
2090  015076  005067  001532          CLR     TOQ               ;THROW AWAY ALL CHARACTERS
2091  015102  016767  001540  001534  MOV     TOQ+QBACK,TOQ+QFRONT  ; IN TTY OUTPUT QUEUE.
2092  015110                          CALL    PNCRLF            ;PRINT <CR> AND <LF>
2093  015124                          CALL    PNTLIN,<CTLOMG>   ;ECHO "^O".
2094  015142  000207                  RETURN                    ;RETURN.
2095
2096
2097
2098                    ;
2099                    ; PROCESSOR FOR CONTROL S.
2100                    ; TEMPORARILY SUSPEND  PRINTOUT ON CONSOLE DEVICE
2101                    ;
2102                    ; PRINTOUT WILL BE RESUMED UPON RECEIPT OF A CNTRL-Q
2103                    ; OR ANY OTHER KEYBOARD INTERRUPT.
2104
015144                          PROC    PRCTLS
 (1)
 (1)  015144            PRCTLS:                                 ;**ENTRY POINT**
2106  015144  052767  100000  002560  BIS     #B15,THLTFL       ;SET TTY HALT FLAG
2107  015152  000207                  RETURN
2108
2109
2110                    ;
2111                    ; PROCESSOR FOR CONTROL Q.
2112                    ; RESUME PRINTOUT ON CONSOLE IF TTOUT QUEUE HAS NOT BEEN
2113                    ; CLEARED BY SOME OTHER MEANS.
2114
015154                          PROC    PRCTLQ
 (1)
 (1)  015154            PRCTLQ:                                 ;**ENTRY POINT**
2116  015154  042767  100000  002550  BIC     #B15,THLTFL       ;CLEAR TTY HALT FLAG
2117  015162  000207                  RETURN
```

```
                                ;
                                ;PROCESSOR FOR CONTROL-C.
                                ;ECHO ''^C'' AND GO TO PRESC ROUTINE.

015164                                  PROC    PRCTLC
  (1)
  (1) 015164                    PRCTLC:                         ;**ENTRY POINT**
015164                                  CALL    PNTLIN,<CTLCMG> ;ECHO ''^C''
015202  000167  000016                  JMP     PRESC           ;GO TO PRESC ROUTINE.


                                ;
                                ;  PROCESSOR FOR CONTROL-U.
                                ;  INPUT COMMAND BEING TYPED IS DISCARDED.

015206                                  PROC    PRCTLU
  (1)
  (1) 015206                    PRCTLU:                         ;**ENTRY POINT**
015206                                  CALL    PNTLIN,<CTLUMG> ;ECHO ''^U''.
                                                                ;DROP INTO PRESC


                                ;
                                ;  PROCESSOR FOR CNTRL-C KEY.
                                ;  THIS KEY IS USED TO REQUEST INPUT.  ALL OTHER EVENTS WILL CEASE
                                ;  AFTER THE CURRENT LINE.  A 'PCL>'' WILL BE ECHOED ON THE TTY.  THE
                                ;  OPERATOR IS NOW FREE TO TYPE IN A COMMAND.
                                ;

015224                                  PROC    PRESC
  (1)
  (1) 015224                    PRESC:                          ;**ENTRY POINT**
015224  005767  002430                  TST     KWFLG           ;GOT A CLOCK?
015230  001402                          BEQ     14$             ;NO.
015232  005077  000722                  CLR     @LCS            ;STOP THE CLOCK
015236  105367  002406          14$:    DECB    REQINP          ;SET INPUT REQUESTED FLAG.
015242  005067  002430                  CLR     STSEV           ;CLR STATUS EVENT
015246  005067  002420                  CLR     SUMEV           ;CLR SUMMARY EVENT
015252  005067  002404                  CLR     TEREV           ;CLR XMTR ERROR EVENT
015256  005067  002402                  CLR     REREV           ;AND RCVR ERROR EVENT
015262  005067  002422                  CLR     PCLGO           ;STOP XMTR
015266                                  CALL    ENQ,<A.P,TOQ>   ;ECHO 'PCL>''.
015306                                  CALL    ENQ,<A.C,TOQ>
015326                                  CALL    ENQ,<A.L,TOQ>
015346                                  CALL    ENQ,<A.PR,TOQ>
015366  012767  016206  001016          MOV     #CMDBUF,CBUFPT  ;INITIALIZE COMMAND BUF PTR.
015374  105067  002251                  CLRB    CMDENT          ;CLEAR COMMAND ENTERED FLAG.
015400                                  RETURN                  ;RETURN.


                                ;
                                ;  PROCESS RUB OUT KEY.
                                ;  LAST CHARACTER IN COMMAND BUFFER IS DELETED.  A ''\'' IS ECHOED.
```

```
                                        ;
                                        ;
015402                                          PROC    PRDEL
                                        ;
015402                                  PRDEL:                          ;**ENTRY POINT**
015402  105767  002242                          TSTB    REQINP          ;IF INPUT NOT REQUESTED,
015406  002034                                  BGE     PDRET           ; THEN IGNORE RUBOUT CHAR.
015410  026727  000776  016206                  CMP     CBUFPT,#CMDBUF  ;IF AT BEGINNING OF BUFFER,
015416  003016                                  BGT     1$              ; THEN ISSUE NEW 'PCL>'
015420                                          CALL    PNCRLF
015434                                          CALL    PRESC
015450  000167  000024                          JMP     PDRET
015454  005367  000732                  1$:     DEC     CBUFPT          ;DECREMENT BUFFER POINTER.
015460                                          CALL    ENQ,<A.BKSL,TOQ>;ECHO A '\'.
015500                                  PDRET:  RETURN                  ;RETURN.
                                        ;
                                        ;  PROCESSOR FOR MISCELLANEOUS CHARACTERS.
                                        ;  THE CHARACTER IS APPENDED TO THE COMMAND BUFFER.
                                        ;
015502                                          PROC    PRMISC,<CHAR>
        000002                          CHAR    =       2
                                        ;
015502                                  PRMISC:                         ;**ENTRY POINT**
015502  105767  002142                          TSTB    REQINP          ;IF INPUT NOT REQUESTED,
015506  002034                                  BGE     PMRET           ; THEN JUST IGNORE CHARACTER.
015510  117500  000002                          MOVB    @CHAR(R5),R0
015514  120027  000040                          CMPB    R0,#' '         ;IF CHARACTER
015520  002403                                  BLT     PM176           ; IS NONPRINTING
015522  120027  000140                          CMPB    R0,#140         ; THEN CHANGE
015526  002402                                  BLT     PM187           ; IT TO
015530  012700  000077                  PM176:  MOV     #'?,R0          ; ASCII '?'.
015534  026727  000652  016411          PM187:  CMP     CBUFPT,#CBFEND-1 ;IF WE ARE AT END OF BUFFER,
015542  001416                                  BEQ     PMRET           ; THEN JUST RETURN.
015544  110077  000642                          MOVB    R0,@CBUFPT      ;PUT CHARACTER INTO BUFFER
015550  005267  000636                          INC     CBUFPT          ;UPDATE BUFFER POINTER.
015554  110067  002516                          MOVB    R0,BKELEM
015560                                          CALL    ENQ,<BKELEM,TOQ>;ECHO CHARACTER.
015600                                  PMRET:  RETURN                  ;RETURN.
```

```
                        .SBTTL  TTY OUTPUT HANDLERS


                ;
                ;  OUTPUT A CHARACTER TO THE TELETYPE IF IT IS READY.
                ;
015602                  PROC    TTOUT
(1)
(1)  015602
                TTOUT:                          ;**ENTRY POINT**
2213 015602 105777 000344       TSTB    @TTXCSR ;IF DEVICE IS NOT READY,
2214 015606 002020              BGE     TORET   ; THEN JUST RETURN.
2215 015610 005767 002116       TST     THLTFL  ;IS TTY HALTED (CNTRL-S)?
2216 015614 001402              BEQ     1$      ;NO, O.K. TO PRINT
2217 015616 000167 000026       JMP     TORET   ;YES, DON'T DO ANYTHING.
2218 015622             1$:     CALL    DEQ,<FRELEM,TOQ>        ;GET NEXT CHAR TO TYPE.
2219 015642 116777 012426 000304        MOVB    FRELEM,@TTXBUF  ;OUTPUT IT.
2220 015650             TORET:  RETURN                          ;RETURN.
2221
2222
2223
2224                    .SBTTL  TTY INPUT INTERRUPT PROCESSORS
2225
2226 015652             TTIINT:                         ;;;**INTERRUPT ENTRY POINT**
2227 015652 005067 002054       CLR     THLTFL  ;;;CLEAR TTY HALT FLAG ON INPUT
2228 015656             REGSAV                  ;;;SAVE R0 - R5.
2229 015662 117767 000262 012414        MOVB    @TTRBUF,INTTMP  ;;;GET INPUT CHARACTER.
2230 015670             CALL    ENQ,<INTTMP,TIQ>        ;;;PUT IT IN TTY INPUT QUEUE.
2231 015710             REGRES                  ;;;RESTORE R0 - R5.
2232 015714 000002              RTI             ;;;RETURN.
2233
```

```
                        .SBTTL  MESSAGE PRINT ROUTINE


                        ;
                        ;  ENQUEUE CHARACTERS STARTING AT  MESSAG  IN TTY OUTPUT
                        ;  QUEUE  TOQ  UNTIL A ZERO BYTE IS ENCOUNTERED.  ENQUEUE A CR & LF
                        ;  INSTEAD OF THE ZERO, & THEN EXIT FROM THE ROUTINE.
                        ;

015716                          PROC    PNTLIN,<MESSAG>
(3)     000002          MESSAG  =       2
(1)
(1)     015716          PNTLIN:                                 ;**ENTRY POINT**
2244    015716  016546  000002         MOV     MESSAG(R5),-(SP)        ;GET ADDRESS OF MESSAGE.
2245    015722  117667  000000  012346 PLODEC: MOVB    @(SP),BKELEM            ;GET A CHARACTER.
2246    015730  001412                 BEQ     PLCRLF                  ;IF NULL, APPEND CR & LF.
2247    015732                         CALL    ENQ,<BKELEM,TOQ>        ;ENQUEUE CHAR FOR TTY OUTPUT.
2248    015752  005216                 INC     @SP                     ;POINT TO NEXT CHARACTER.
2249    015754  000762                 BR      PLODEC                  ;PROCESS IT.
2250
2251    015756  005026         PLCRLF: CLR     (SP)+                   ;POP ADDR FROM STACK.
2252    015760  000400                 BR      PL219                   ;APPEND CR & LF.
2253
2254
2255                        ;
2256                        ;  ENQUEUE A CR & LF IN TTY OUTPUT QUEUE.
2257                        ;
2258    015762                      PROC    PNCRLF
(1)
(1)     015762          PNCRLF:                                 ;**ENTRY POINT**
2259    015762          PL219:  CALL    ENQ,<A.CR,TOQ>          ;ENQUEUE A CR.
2260    016002                  CALL    ENQ,<A.LF,TOQ>          ;ENQUEUE A LF.
2261    016022                  CALL    ENQ,<ZERO,TOQ>          ;ENQUEUE A NULL FILL CHAR.
2262    016042                  RETURN                          ;RETURN.
```

```
                                        .SBTTL DATA AREAS

                                ;  TABLE ASSOCIATING SPECIAL TTY INPUT CHARACTERS WITH THEIR
                                ;  PROCESSING ROUTINES.
                                ;
016044                          CMCHTB:
016044  000003  015164                .WORD   CTL.C,PRCTLC
016050  000012  015034        A.LF:   .WORD   LF.,PRCR
016054  000015  015034        A.CR:   .WORD   CR.,PRCR
016060  000017  015076                .WORD   CTL.O,PRCTLO
016064  000021  015154                .WORD   CTL.Q,PRCTLQ
016070  000023  015144                .WORD   CTL.S,PRCTLS
016074  000025  015206                .WORD   CTL.U,PRCTLU
016100  000177  015402                .WORD   RUBOUT,PRDEL
016104  177777                        .WORD   -1


                                ;DEVICE ADDRESS TABLES:

016106  164200                TCR:    .WORD   PCLTXM
016110  164202                TSR:    .WORD   PCLTXM+2
016112  164204                TSDB:   .WORD   PCLTXM+4
016114  164206                TSBC:   .WORD   PCLTXM+6
016116  164210                TSBA:   .WORD   PCLTXM+10
016120  164212                TMMR:   .WORD   PCLTXM+12
016122  164213                TMMRH:  .WORD   PCLTXM+13
016124  164214                TSCRC:  .WORD   PCLTXM+14
016126  164220                RCR:    .WORD   PCLRCV
016130  164222                RSR:    .WORD   PCLRCV+2
016132  164224                RDDB:   .WORD   PCLRCV+4
016134  164226                RDBC:   .WORD   PCLRCV+6
016136  164230                RDBA:   .WORD   PCLRCV+10
016140  164234                RDCRC:  .WORD   PCLRCV+14

016142  000170                TXMVEC: .WORD   170
016144  000174                RCVVEC: .WORD   174

016146  177560                TTRCSR: .WORD   TTDEV
016150  177562                TTRBUF: .WORD   TTDEV+2
016152  177564                TTXCSR: .WORD   TTDEV+4
016154  177566                TTXBUF: .WORD   TTDEV+6
016156  000060                TTVECT: .WORD   TTVCTR                  ;VECTOR ADDRESS.
016160  177546                LCS:    .WORD   177546                  ;KW11-L LINE CLOCK ADDR


                                ;  CHARACTER CONSTANTS.  THESE ARE DEFINED AS WORDS SO THAT THEY MAY
                                ;  BE ENQUEUED.
016162  000134                A.BKSL: .WORD   '\                      ;ASCII '\'
016164  000000                ZERO:   .WORD   0                       ;ASCII NULL
016166  000120                A.P:    .WORD   'P                      ;ASCII 'P'
016170  000103                A.C:    .WORD   'C                      ;ASCII 'C'
016172  000114                A.L:    .WORD   'L                      ;ASCII 'L'
016174  000076                A.PR:   .WORD   '>                      ;ASCII '>' (PROMPT)

                                        .EVEN

                                ;DEVICE ADDRESS AND VECTOR VARIABLES
```

```
                                        ;CHANGE THESE LOCATIONS TO MODIFY ALL DEVICE ADDRESSES AND VECTORS
                                        ;FOR PCL11.
                                        
016176  164200                  TXDEV:  .WORD   164200          ;DEFAULT IS 164200
016200  000170                  TXVEC:  .WORD   170             ;DEFAULT IS 170
016202  164220                  RCDEV:  .WORD   164220          ;DEFAULT IS 164220
016204  000174                  RCVEC:  .WORD   174             ;DEFAULT IS 174
                                        
                                        ;  BUFFER & MESSAGE AREAS.
016206  000204                  CMDBUF: .BLKB   132.            ;TTY INPUT COMMAND BUFFER.
        016412                  CBFEND  .
016412  033333                  CBUFPT: .WORD   33333           ;CMDBUF BUFFER POINTER.
                                        
016414  047536     000          CTLOMG: .ASCIZ  '^O'            ;FOR ECHOING CONTROL CHARACTERS.
016417     136  000125          CTLUMG: .ASCIZ  '^U'            ;  "
016422  041536     000          CTLCMG: .ASCIZ  '^C'            ;  "
                                        
016425     040  051105  047522  RPMSG:  .ASCIZ  ' ERROR'
016432  000122
                                        ;  QUEUE DEFINITIONS.
                                        .IRP    LC      <AQ,TI,TO>
                                                .EVEN
                                                .LIST
                                        LC'Q:   .WORD   0               ;QELEMS
                                                .WORD   LC'SIZE         ;QSIZE
                                                .WORD   LC'AREA,LC'END  ;QTOP & QBOT
                                                .WORD   33333,33333     ;QFRONT & QBACK
                                        LC'AREA:        .BLKW   LC'SIZE         ;LC'Q AREA
                                        LC'END          .
                                                .NLIST
                                        .ENDM
016434  000000                  AQ:     .WORD   0               ;QELEMS
016436  000040                          .WORD   AQSIZE          ;QSIZE
016440  016450  016550                   .WORD   AQAREA,AQEND    ;QTOP & QBOT
016444  033333  033333                   .WORD   33333,33333     ;QFRONT & QBACK
016450  000040                  AQAREA: .BLKW   AQSIZE          ;AQ AREA
        016550                  AQEND   =       .
016550  000000                  TIQ:    .WORD   0               ;QELEMS
016552  000024                          .WORD   TISIZE          ;QSIZE
016554  016564  016634                   .WORD   TIAREA,TIEND    ;QTOP & QBOT
016560  033333  033333                   .WORD   33333,33333     ;QFRONT & QBACK
016564  000024                  TIAREA: .BLKW   TISIZE          ;TIQ AREA
        016634                  TIEND   .
016634  000000                  TOQ:    .WORD   0               ;QELEMS
016636  000400                          .WORD   TOSIZE          ;QSIZE
016640  016650  017650                   .WORD   TOAREA,TOEND    ;QTOP & QBOT
016644  033333  033333                   .WORD   33333,33333     ;QFRONT & QBACK
016650  000400                  TOAREA: .BLKW   TOSIZE          ;TOQ AREA
        017650                  TOEND   =       .

                                        ;  FLAG VARIABLES.
                                        ;       < 0 ==> TRUE
                                        ;       >=0 ==> FALSE
                                        ;
017650     333                  REQINP: .BYTE   333             ;INPUT REQUEST IS BEING TYPED.
017651     333                  CMDENT: .BYTE   333             ;COMMAND HAS BEEN ENTERED.
```

```
2357  017652  000          TICKS:  .BYTE   0
2358  017653  000          SECNDS: .BYTE   0
2359  017654  000          MINUTS: .BYTE   0
2360          017656               .EVEN
2361  017656  000000       HOURS:  .WORD   0
2362  017660  000000       KWFLG:  .WORD   0
2363  017662  000000       TEREV:  .WORD   0
2364  017664  000000       REREV:  .WORD   0
2365  017666  000000       QLDEV:  .WORD   0
2366  017670  000000       DATGEV: .WORD   0
2367  017672  000000       SUMEV:  .WORD   0
2368  017674  000000       SUMSV:  .WORD   0
2369  017676  000000       STSEV:  .WORD   0
2370  017700  000000       STPNTR: .WORD   0
2371  017702  000000       SUMPNT: .WORD   0
2372  017704  000000       TXMEV:  .WORD   0
2373  017706  000000       TXMSV:  .WORD   0
2374  017710  000000       PCLGO:  .WORD   0
2375  017712  000000       RCVEV:  .WORD   0
2376  017714  000000       SPCEV:  .WORD   0
2377  017716  000000       XSPCEV: .WORD   0
2378  017720  000000       FIRST:  .WORD   0
2379  017722  000000       NEXT:   .WORD   0
2380  017724  000000       PRINTD: .WORD   0
2381  017726  000000       HDRFLG: .WORD   0
2382  017730  000000       PADFLG: .WORD   0
2383  017732  000000       THLTFL: .WORD   0
2384
2385                       .EVEN
2386
2387                       ;  DATA VARIABLES.
2388  017734  133333       ORIGSD: .WORD   133333
2389  017736  000000       DTSEED: .WORD   0
2390  017740  000333       MLSD:   .WORD   333
2391  017742  000000       MSGLSD: .WORD   0
2392  017744  000000       RCSEED: .WORD   0
2393  017746  000000       RCTXPS: .WORD   0
2394  017750  000000       RJCTF:  .WORD   0
2395  017752  000000       TRNKF:  .WORD   0
2396  017754  000000       TXML:   .WORD   0
2397  017756  037565       RANM:   .WORD   37565
2398  017760  012247       RANK:   .WORD   12247
2399  017762  000000       CURAD:  .WORD   0
2400  017764  000000       RSFC:   .WORD   0
2401  017766  000000       ESCFLG: .WORD   0
2402  017770  000000       OBJCNT: .WORD   0
2403  017772  000000       RSHOLD: .WORD   0
2404  017774  000000       TEMP:   .WORD   0
2405  017776  000000       TEMP1:  .WORD   0
2406  020000  000000       ERR0:   .WORD   0
2407  020002  000000       ERR1:   .WORD   0
2408  020004  000000       ERR2:   .WORD   0
2409
2410
2411                       ;ADDRESS SILO DATA BUFFER AREA
2412
```

```
2413  020006  000102                ADSILO: .BLKB   66.                     ;66. BYTE AREA FOR SILO DATA
2414
2415                                ;RECEIVER ADDRESS TABLE
2416
2417          000001                X       -       1
2418  020110                        RADB:
2419          000037                        .REPT   31.
2420                                        .WORD   0                       ;ACTIVITY FLAG
2421                                        .WORD   X                       ;RECEIVER ADDRESS
2422                                        .WORD   0,0                     ;ATTEMPTS ENTRY
2423                                        .WORD   0,0                     ;SUCCESSES ENTRY
2424
2425                                X       -       X+1
2426                                        .ENDR
2427  020674  020674                RADEND: .WORD   .
2428          020074                RADB0=  RADB-14
2429
2430                                ;TRANSMITTER DATA BUFFER:
2431
2432  020676  001020                DATBUF: .BLKW   1020
2433
2434
2435                                ;RECEIVER DATA BUFFER:
2436
2437  022736  001000                RCBUF:  .BLKW   1000
2438
2439                                ;EXERCISER ERROR TABLE
2440
2441          000001                Y       =       1                       ;INITIAL ERROR NUMBER
2442  024736                        ERTBL:
2443          000030                        .REPT   N-1
2444                                        .WORD   Y                       ;ERROR NUMBER
2445                                        .WORD   0                       ;ERROR ADDRESS
2446                                        .WORD   0                       ;NO. OF OCCURRENCES SINCE INIT
2447
2448                                Y       -       Y+1
2449                                        .ENDR
2450  025156  177777                        .WORD   -1                      ;LAST ERROR # IS -1
2451
2452          024730                ERTBL0  -       ERTBL-6
2453
2454
2455                                ;       DETAILED ERROR TABLES FOR RCVR AND XMTR ERRORS:
2456
2457  025160  000623                TERTBL: .BLKW   37*15                    ;RESERVE SPACE FOR XMTR ERRORS
2458
2459
2460  026626  000623                RERTBL: .BLKW   37*15                    ;RESERVE SPACE FOR RCVR ERRORS
2461
2462
2463
2464
2465  030274  033333                FRELEM: .WORD   33333                   ;STORAGE FOR DEQUEUED ELEMENT.
2466  030276  033333                BKELEM: .WORD   33333                   ;STORAGE FOR ENQUEUED ELEMENT.
2467  030300  033333                TCBFPT: .WORD   33333                   ;CMDBUF POINTER USED DURING SCAN
2468  030302  033333                TCB.N:  .WORD   33333                   ;BINARY VALUE OF INPUT PARAMETER
```

```
2469  030304  033335                          INTTMP: .WORD   33333                   ;TEMP STORAGE FOR INTERRUPT PROC
2470
2471                                                  .SBTTL  KEYWORD TABLE
2472
2473                                          ;KEYWORD TABLE ASSOCIATING A COMMAND WITH ITS PROCESSING ROUTINE
2474
2475  030306  004524                          CMDTBL: .WORD   CPADD
2476  030310     002     003                          .BYTE   2,3
2477  030312  042101  020104                          .ASCII  /ADD /
2478  030316  005672                                  .WORD   CPASS
2479  030320     002     006                          .BYTE   2,6
2480  030322  051501  044523  047107                  .ASCII  /ASSIGN/
2481  030330  004752                                  .WORD   CPCLR
2482  030332     002     005                          .BYTE   2,5
2483  030334  046103  040505  020122                  .ASCII  /CLEAR /
2484  030342  005114                                  .WORD   CPCNT
2485  030344     002     010                          .BYTE   2,8.
2486  030346  047503  052116  047111                  .ASCII  /CONTINUE/
      030354  042525
2487  030356  004632                                  .WORD   CPDEL
2488  030360     001     006                          .BYTE   1,6
2489  030362  042504  042514  042524                  .ASCII  /DELETE/
2490  030370  006600                                  .WORD   CPERR
2491  030372     001     006                          .BYTE   1,6
2492  030374  051105  047522  051522                  .ASCII  /ERRORS/
2493  030402  005510                                  .WORD   CPGO
2494  030404     001     002                          .BYTE   1,2
2495  030406  047507                                  .ASCII  /GO/
2496  030410  005210                                  .WORD   CPINIT
2497  030412     001     012                          .BYTE   1,10.
2498  030414  047111  052111  040511                  .ASCII  /INITIALIZE/
      030422  044514  042532
2499  030426  004116                                  .WORD   CPMAST
2500  030430     001     006                          .BYTE   1,6
2501  030432  040515  052123  051105                  .ASCII  /MASTER/
2502  030440  004366                                  .WORD   CPRANG
2503  030442     002     005                          .BYTE   2,5
2504  030444  040522  043516  020105                  .ASCII  /RANGE /
2505  030452  004274                                  .WORD   CPRIB
2506  030454     002     003                          .BYTE   2,3
2507  030456  044522  020102                          .ASCII  /RIB /
2508  030462  004202                                  .WORD   CPSEC
2509  030464     002     011                          .BYTE   2,9.
2510  030466  042523  047503  042116                  .ASCII  /SECONDARY /
      030474  051101  020131
2511  030500  003406                                  .WORD   CPSILO
2512  030502     002     004                          .BYTE   2,4
2513  030504  044523  047514                          .ASCII  /SILO/
2514  030510  005054                                  .WORD   CPSTAT
2515  030512     002     006                          .BYTE   2,6
2516  030514  052123  052101  051525                  .ASCII  /STATUS/
2517  030522  005456                                  .WORD   CPSUM
2518  030524     002     007                          .BYTE   2,7
2519  030526  052523  046515  051101                  .ASCII  /SUMMARY /
      030534  020131
2520  030536  000000  000000                          .WORD   0,0
```

```
 2521
 2522
 2523  030542  000000                          SCTBL:  .WORD   0
 2524  030544     001      005                        .BYTE   1,5
 2525  030546  046103  040505  020122                .ASCII  /CLEAR /
 2526  030554  000001                                .WORD   1
 2527  030556     001      003                        .BYTE   1,3
 2528  030560  042523  020124                        .ASCII  /SET /
 2529  030564  000000  000000                        .WORD   0,0
 2530
 2531                                                .SBTTL  SOME MORE ASCII STORAGE:
 2532
 2533
 2534  030570  020040  020040  020052  MDNER:  .ASCIZ  /    * * * * * *  MASTER DOWN  * * * * * * */
       030576  020052  020052  020052
       030604  020052  020052  046440
       030612  051501  042524  020122
       030620  047504  047127  020040
       030626  020052  020052  020052
       030634  020052  020052  000052
 2535  030642  020040  025052  052040  MSCHNG: .ASCIZ  /  ** THIS UNIT HAS BECOME 'NEW MASTER'' **/
       030650  044510  020123  047125
       030656  052111  044040  051501
       030664  041040  041505  046517
       030672  020105  047042  053505
       030700  046440  051501  042524
       030706  021122  025040  000052
 2536  030714  020040  025052  051452  SYNTAX: .ASCIZ  /  ***SYNTAX ERROR***/
       030722  047131  040524  020130
       030730  051105  047522  025122
       030736  025052     000
 2537  030741     120  046103  030461  PCLEXM: .ASCIZ  /PCL11 EXERCISER V02C  CZPLAC0  (JUN-79)/
       030746  042440  042530  041522
       030754  051511  051105  053040
       030762  031060  020103  041440
       030770  050132  040514  030103
       030776  020040  045050  047125
       031004  033455  024471     000
 2538  031011     122  051505  040524  RSTMSG: .ASCIZ  /RESTART AT ADDRESS 204/
       031016  052122  040440  020124
       031024  042101  051104  051505
       031032  020123  030062  000064
 2539  031040  025052  047040  020117  MTQMSG: .ASCIZ  /** NO RECEIVERS SELECTED **/
       031046  042522  042503  053111
       031054  051105  020123  042523
       031062  042514  052103  042105
       031070  025040  000052
 2540  031074  051124  050101  042520  TRPDMG: .ASCII  /TRAPPED TO 4 FROM LOCATION /
       031102  020104  047524  032040
       031110  043040  047522  020115
       031116  047514  040503  044524
       031124  047117     040
 2541  031127     116  047116  047116  TRP4AD: .ASCIZ  /NNNNN    /
       031134  020040  000041
 2542  031140  042504  044526  042503  ERTMSG: .ASCIZ  /DEVICE ADDRESS ERROR. USE ''ASSIGN'' COMMAND./
       031146  040440  042104  0425..
```

```
                031154  051523  042440  051122
                031162  051117  020056  051525
                031170  020105  040442  051523
                031176  043511  021116  041440
                031204  046517  040515  042116
                031212  000056
 2543           031214  054105  051105  044503  EXRST:  .ASCIZ  /EXERCISER STARTED/
                031222  042523  020122  052123
                031230  051101  042524  000104
 2544           031236  054105  051105  044503  EXCNT:  .ASCIZ  /EXERCISER CONTINUING/
                031244  042523  020122  047503
                031252  052116  047111  044525
                031260  043516     000
 2545           031263     052  025052  044124  MSTMG1: .ASCIZ  /***THIS UNIT IS NOT MASTER***/
                031270  051511  052440  044516
                031276  020124  051511  047040
                031304  052117  046440  051501
                031312  042524  025122  025052
                031320     000
 2546           031321     102  052125  044040  MSTMG2: .ASCIZ  /BUT HAS NOW BEEN MADE SECONDARY./
                031326  051501  047040  053517
                031334  041040  042505  020116
                031342  040515  042504  051440
                031350  041505  047117  040504
                031356  054522  000056
 2547           031362  044124  020105  044523  MSTMG3: .ASCIZ  /THE SILO YOU HAVE JUST LOADED WILL BE/
                031370  047514  054440  052517
                031376  044040  053101  020105
                031404  052512  052123  046040
                031412  040517  042504  020104
                031420  044527  046114  041040
                031426  000105
 2548           031430  051525  042105  044440  MSTMG4: .ASCIZ  /USED IF YOU CLEAR THE CURRENT MASTER./
                031436  020106  047531  020125
                031444  046103  040505  020122
                031452  044124  020105  052503
                03146C  051122  047105  020124
                031466  040515  052123  051105
                031474  000056
 2549           031476  044124  020105  044523  MSTMG5: .ASCIZ  /THE SILO HAS BEEN PADDED WITH ADDRESS '0'/
                031504  047514  044040  051501
                031512  041040  042505  020116
                031520  040520  042104  042105
                031526  053440  052111  020110
                031534  042101  051104  051505
                031542  020123  030042  000042
 2550           031550  044124  051511  052440  THUMST: .ASCIZ  /THIS UNIT IS -MASTER-/
                031556  044516  020124  051511
                031564  026440  040515  052123
                031572  051105  000055
 2551           031576  044124  051511  052440  THUSCN: .ASCIZ  /THIS UNIT IS -SECONDARY-/
                031604  044516  020124  051511
                031612  026440  042523  047503
                031620  042116  051101  026531
                031626     000
 2552           031627     042  044522  021102  RBSTMG: .ASCIZ  /'RIB' IS -SET-/
```

```
                031634  044440  020123  051455
                031642  052105  000055
2553            031646  051042  041111  020042  RBCLMG: .ASCIZ  /'RIB'' IS -CLEAR-/
                031654  051511  026440  046103
                031662  040505  026522     000
2554            031667     105  040514  051520  ELPSTM: .ASCII  /ELAPSED TIME (HRS:MIN:SEC:TIK).../
                031674  042105  052040  046511
                031702  020105  044050  051522
                031710  046472  047111  051472
                031716  041505  052072  045511
                031724  027051  027056
2555            031730  035060  035060  035060  TMLIN1: .ASCIZ  /0:0:0:0         /
                031736  020060  020040  020040
                031744  020040  020040     000
2556            031751     122  053103  020122  STITLE: .ASCIZ  /RCVR ADDRESS  CONNECTION ATTEMPTS  SUCCESSFUL CONNECTIONS/
                031756  042101  051104  051505
                031764  020123  041440  047117
                031772  042516  052103  047511
                032000  020116  052101  042524
                032006  050115  051524  020040
                032014  052523  041503  051505
                032022  043123  046125  041440
                032030  047117  042516  052103
                032036  047511  051516     000
2557
2558            032043                          STLIN:
2559            032043     116  047116  020116  STLIN1: .ASCII  /NNNN             /
                032050  020040  020040  020040
                032056  020040  020040  020040
                032064     040
2560            032065     116  047116  020116  STLIN2: .ASCII  /NNNN                 /
                032072  020040  020040  020040
                032100  020040  020040  020040
                032106  020040  020040     040
2561            032113     116  047116  020116  STLIN3: .ASCIZ  /NNNN         /
                032120  020040  020040  020040
                032126  020040     000
2562
2563
2564            032131     105  051122  051117  SMTTLE: .ASCIZ  /ERROR NUMBER    ERROR ADDRESS    NO. OF OCCURRENCES/
                032136  047040  046525  042502
                032144  020122  020040  042440
                032152  051122  051117  040440
                032160  042104  042522  051523
                032166  020040  020040  047516
                032174  020056  043117  047440
                032202  041503  051125  042522
                032210  041516  051505     000
2565
2566            032215     040  025040  020052  NOERMG: .ASCIZ  /  ** NO ERRORS TO REPORT **/
                032222  047516  042440  051122
                032230  051117  020123  047524
                032236  051040  050105  051117
                032244  020124  025052     000
2567            032251     124  040522  051516  TERHDR: .ASCIZ  /TRANSMITTER ERRORS:/
                032256  044515  052124  051105
```

```
                032264  042440  051122  051117
                032272  035123     000
 2568           032275     105  051122  051117  TRHLIN: .ASCIZ  /ERROR NO.    CONCTD RCVR    ERROR COUNT/
                032302  047040  027117  020040
                032310  020040  047503  041516
                032316  042124  051040  053103
                032324  020122  020040  042440
                032332  051122  051117  041440
                032340  052517  052116     000
 2569           032345                           TRELIN:
 2570           032345     116  047116  047116  TRRNO:  .ASCII  /NNNNN        /
                032352  020040  020040  020040
                032360  020040
 2571           032362  047116  047116  020116  TRRCN:  .ASCII  /NNNNN          /
                032370  020040  020040  020040
                032376  020040     040
 2572           032401     116  047116  020116  TRERC:  .ASCIZ  /NNNN   /
                032406  020040     000
 2573           032411     122  041505  044505  RERHDR: .ASCIZ  /RECEIVER ERRORS:/
                032416  042526  020122  051105
                032424  047522  051522  000072
 2574           032432  051105  047522  020122  RRHLIN: .ASCIZ  /ERROR NO.    CONCTD XMTR    ERROR COUNT/
                032440  047516  020056  020040
                032446  041440  047117  052103
                032454  020104  046530  051124
                032462  020040  020040  051105
                032470  047522  020122  047503
                032476  047125  000124
 2575           032502                           RCELIN:
 2576           032502  047116  047116  020116  RCRNO:  .ASCII  /NNNNN        /
                032510  020040  020040  020040
                032516     040
 2577           032517     116  047116  047116  RCTRN:  .ASCII  /NNNNN          /
                032524  020040  020040  020040
                032532  020040  020040
 2578           032536  047116  047116  020040  RCERC:  .ASCIZ  /NNNN   /
                032544  000040
 2579           032546  020040  047050  047117  NONMG:  .ASCIZ  /  (NONE)/
                032554  024505     000
 2580           032557                           SMLIN:
 2581           032557     116  047116  020116  SMLIN1: .ASCII  /NNNN             /
                032564  020040  020040  020040
                032572  020040  020040  020040
                032600     040
 2582           032601     116  047116  020116  SMLIN2: .ASCII  /NNNN               /
                032606  020040  020040  020040
                032614  020040  020040  020040
                032622  020040     040
 2583           032625     116  047116  020116  SMLIN3: .ASCIZ  /NNNN    /
                032632  020040  000040
 2584
 2585
```

```
                                .EVEN
                                .SBTTL AUXILIARY ROUTINES
                                .SBTTL  CHARACTER PROCESSOR

                        ;  CONVERT A CHARACTER TO ITS PROCESSING ROUTINE ADDRESS BASED
                        ;  UPON A TABLE OF ENTRIES IN THE FOLLOWING FORM:
                        ;                          I--------------I--------------I
                        ;                          I   (UNUSED)   I  CHARACTER   I
                        ;                          I--------------I--------------I
                        ;                          I   PROCESSING ROUTINE ADDR   I
                        ;                          I--------------I--------------I
                        ;
                        ;  THE TABLE MUST BE ARRANGED IN ASCENDING ORDER OF CHARACTER VALUES.
                        ;
                        ;  THE TABLE ENDS WITH A DUMMY ENTRY FOR CHARACTER FF (HEXADECIMAL).
                        ;
                        ;  ON ENTRY:    R0 - CHARACTER
                        ;               R1 - TABLE ADDRESS
                        ;
                        ;  CALL:        JSR     PC,PROCHR
                        ;
                        ;  ON RETURN:   R0 - CHARACTER
                        ;               R1 = PROCESSING ROUTINE ADDRESS, IF ANY
                        ;               Z  = 1 ==> CHARACTER NOT IN TABLE

032636                  PROCHR:                                 ;**ENTRY POINT**
032636  121100          PCLOOK: CMPB    @R1,R0                  ;COMPARE TABLE CHAR
                                                                ; WITH ARG. CHAR.
032640  001404                  BEQ     PCCALL                  ;IF SAME, RETURN PROC.
                                                                ; ROUTINE ADDR.
032642  101006                  BHI     PCQUIT                  ;IF >, THEN ARG. CHAR
                                                                ; NOT IN TABLE.
032644  062701  000004          ADD     #4,R1                   ;IF <, POINT TO NEXT
                                                                ; TABLE ENTRY.
032650  000772                  BR      PCLOOK                  ;TRY AGAIN.

032652  016101  000002  PCCALL: MOV     2(R1),R1                ;R1 = PROCESSING
                                                                ; ROUTINE ADDRESS.
032656  000207                  RTS     PC                      ;RETURN: Z BIT IS
                                                                ; OFF.

032660  005001  PCQUIT: CLR     R1                              ;R1   0
032662  000207          RTS     PC                              ;RETURN: Z BIT IS ON.
```

```
                        .SBTTL  BINARY TO ASCII CONVERSION


;
; BINARY TO ASCII CONVERSION
;
; LOCAL MACROS
;
; SETUP CONVERSION CONTROL WORD ON STACK
;
; STCVT RADIX,WIDTH,SIGNED,COMPR,BLANKS
;
; WHERE:
;
;       RADIX=NUMERIC VALUE SPECIFYING CONVERSION RADIX
;       WIDTH=NUMERIC VALUE FROM 1 TO 7 SPECIFYING FIELD WIDTH
;       SIGNED=SIGN OR MAGNITUDE FLAG. ASCII STRING 'SIGN' SPECI-
;              FIES SIGNED CONVERSION. ANYTHING ELSE SPECIFIES MAGNI-
;              TUDE.
;
;       COMPR=COMPRESS LEADING ZEROS FLAG. ASCII STRING 'COMPRES' SPE-
;              CIFIES COMPRESSION OF LEADING ZEROS. ANYTHING ELSE MEANS
;              INCLUDE LEADING ZEROS OR SPACES IN CONVERSION.
;
;       BLANK REPLACE LEADING ZEROS WITH BLANKS (SPACES).  ASCII STRING
;              'BLANKS' MEANS BLANK REPLACEMENT IF ZERO  MPRESS IS DIS-
;              ABLED. ANYTHING ELSE SPECIFIES ZERO PADDING.
;
        .MACRO  STCVT   RADIX,WIDTH,SIGN,COMPR,BLANK
$BLK=0
$SGN 0
$SUP=1*1000
        .IF     IDN     <BLANK>,<BLANKS>
$BLK 1*2000
        .ENDC
        .IF     IDN     <SIGN>,<SIGNED>
$SGN 1*400
        .ENDC
        .IF     IDN     <COMPR>,<COMPRES>
$SUP 0*1000
        .ENDC
        MOV     #<WIDTH*4000>!$BLK!$SGN!$SUP!RADIX,-(SP)
        .ENDM
```

```
                        ;
                        ; INPUTS:
                        ;
                        ;       R0=ADDRESS TO STORE FIRST BYTE IN OUTPUT STRING
                        ;       R1-NUMBER TO BE CONVERTED
                        ;       R2 ZERO COMPRESSION INDICATOR
                        ;               IF R2 EQ 0, THEN SUPPRESS ZEROS
                        ;               IF R2 NE 0, THEN DO NOT SUPPRESS ZEROS.
                        ;
                        ; IF CBTA IS CALLED, THEN R2 MUST CONTAIN THE FOLLOWING INFORMATION
                        ;
                        ;       LOW BYTE=CONVERSION RADIX (2-10.)
                        ;       BIT 8=MAGNITUDE/SIGNED CONVERSION (1=SIGNED)
                        ;
                        ;       BIT 9 -ZERO COMPRESS FLAG (0=COMPRESS LEADING ZEROS)
                        ;
                        ;       BIT 10=BLANK FILL FLAG (1-REPLACE LEADING ZEROS WITH BLANKS
                        ;               IF ZERO COMPRESS DISABLED, 0=ZERO FILL).
                        ;
                        ;       BITS 11-15=FIELD WIDTH (1-32)
                        ;
                        ;
                        ; OUTPUTS:
                        ;
                        ;       R0=ADDRESS OF NEXT BYTE AFTER LAST DIGIT STORED.
                        ;
                        ;       IF THE CONVERTED DIGIT EXCEEDS 9, THE RESULT IS BIASED TO FALL
                        ;       IN THE RANGE A - Z
                        ;
                        ;
                        ; CONVERT 6 DIGIT OCTAL TO ASCII MAGNITUDE
                        ;
032664                  OCTJSP:
032664                          STCVT   8.,6.,MAGN,NOCOMP,BLANKS        ;PUSH CONVERSION PARAMETERS
032670  000411                  BR      SETCN                   ;CONVERT TO ASCII
                        ;
                        ; CONVERT 6 DIGIT OCTAL TO ASCII (ZERO COMPR)
                        ;
032672                  OCTPNT:
032672                          STCVT   8.,6.,MAGN,COMPRES,NOBLKANK
032676  000406                  BR      SETCN
                        ;
                        ; CONVERT 5 DIGIT DECIMAL TO ASCII MAGNITUDE
                        ;
032700                  DECJSP:
032700                          STCVT   10.,5.,MAGN,NOCOMP,BLANKS
032704  000403                  BR      SETCN
                        ;
                        ;CONVERT 5 DIGIT DECIMAL TO ASCII (ZERO COMPR)
                        ;
032706                  DECPNT:
032706                          STCVT   10.,5.,MAGN,COMPRES,NOBLANK
032712  000400                  BR      SETCN
                        ;
032714                  SETCN:
032714  005702                  TST     R2                      ;SUPPRESS ZEROS?
032716  001002                  BNE     20$                     ;IF NE, NO
```

```
 2731  032720  042716  001000                 BIC     #1*1000,(SP)            ;ENABLE ZERO SUPPRESS
 2732  032724                         20$:
 2733  032724  012602                         MOV     (SP)+,R2                ;SET CONTROL WORD
```

```
2735
2736
2737  032726                                  .SBTTL  GENERAL BINARY TO ASCII CONVERSION
2738  032726  004567  000406          CBTA:
2739  032732  110205                          JSR     R5,SAVRG        ;SAVE THE NON-VOLATILE REGISTERS
2740  032734  000302                          MOVB    R2,R5           ;COPY RADIX BYTE
2741  032736  106202                          SWAB    R2              ;POSITION REMAINING TO LOW BYTE
2742  032740  103005                          ASRB    R2              ;SHIFT OFF MAG. FLAG
2743  032742  005701                          BCC     10$             ;UNSIGNED IF C IS CLR
2744  032744  100003                          TST     R1              ;POSITIVE VALUE?
2745  032746  005401                          BPL     10$             ;IF PL, YES
2746  032750  112720  000055                  NEG     R1              ;MAKE VALUE POSITIVE
2747  032754                                  MOVB    #'-,(R0)+       ;INSERT A MINUS SIGN
2748  032754  010004          10$:
2749  032756  000241                          MOV     R0,R4           ;COPY STRING POINTER
2750  032760  106002                          CLC                     ;CLEAR CARRY
2751  032762  006002                          RORB    R2              ;SHIFT OFF SUPPR FLAG
2752  032764  006003                          ROR     R2              ;TRANSFER TO R2
2753  032766  105003                          ROR     R3              ;GET BLANK/ZERO PAD FLAG
2754  032770  150203                          CLRB    R3              ;CLEAR COUNT BYTE
2755  032772  105002                          BISB    R2,R3           ;TRANSFER COUNT BYTE
2756  032774  152702  000060                  CLRB    R2              ;CLEAR FILL BYTE
2757  033000  010100                          BISB    #'0,R2          ;SET FILL BYTE
2758  033002                                  MOV     R1,R0           ;DIVIDEND TO R0
2759  033002  010501          1$:
2760  033004  004767  000272                  MOV     R5,R1           ;SET CONVERSION RADIX
2761  033010  020127  000011                  JSR     PC,DIV          ;DIVIDE EM UP
2762  033014  101402                          CMP     R1,#9.          ;RESULT EXCEED NUMERICS
2763  033016  062701  000007                  BLOS    15$             ;IF LOS, NO
2764  033022                                  ADD     #7,R1           ;BIAS TO FALL IN ALPHA
2765  033022  060201          15$:
2766  033024  010146                          ADD     R2,R1           ;ADD CHARACTER BIAS
2767  033026  105303                          MOV     R1,-(SP)        ;SAVE CHARACTER
2768  033030  003412                          DECB    R3              ;DECREMENT CHARACTER COUNT
2769  033032  005700                          BLE     3$              ;IF LE NO DIGITS LEFT
2770  033034  001006                          TST     R0              ;ZERO QUOTIENT
2771  033036  005702                          BNE     2$              ;IF NE, YES, GO AGAIN
2772  033040  100006                          TST     R2              ;SUPPRESS ZEROS
2773  033042  005703                          BPL     3$              ;IF PL, YES, ALL DONE
2774  033044  100002                          TST     R3              ;SUBSTITUTE BLANKS?
2775  033046  042702  000020                  BPL     2$              ;IF PL, NO
2776  033052                                  BIC     #20,R2          ;CONVERT FILL TO BLANK
2777  033052  004767  177724          2$:
2778  033056                                  JSR     PC,1$           ;DIVIDE AGAIN
2779  033056  112624          3$:
2780  033060  010400                          MOVB    (SP)+,(R4)+     ;STORE A DIGIT
2781  033062                                  MOV     R4,R0           ;STORE TERMINAL ADDRESS
2782                                          RETURN
```

```
                                        .SBTTL  DOUBLE PRECISION BINARY TO ASCII
                                ; CONVERT A DOUBLE PRECISION UNSIGNED QUANTITY TO DECIMAL ASCII
                                ;
                                ; LOCAL MACROS
                                ;
                                ; SET ASCII CONVERSION PARAMETERS
                                ;
                                ; CBTAS RADIX,WIDTH,SIGN,BLANK
                                ;
                                ; WHERE:
                                ;
                                ;       RADIX=CONVERSION RADIX
                                ;
                                ;       WIDTH=FIELD WIDTH
                                ;
                                ;       SIGN='SIGNED' FOR SIGNED CONVERSION. ANYTHING ELSE IMPLIES
                                ;            UNSIGNED CONVERSION
                                ;
                                ;       BLANK='BLANKS' TO CONVERT LEADING ZEROS TO BLANKS. ANYTHING ELSE
                                ;            IMPLIES NO CONVERSION OF ZEROS .
                                ;
                                        .MACRO  CBTAS   RADIX,WIDTH,SIGN,BLANK
                                $BLKS 0
                                $SGNS 0
                                        .IF     IDN     <BLANK>,<BLANKS>
                                $BLKS 1*2000
                                        .ENDC
                                        .IF     IDN     <SIGN>,<SIGNED>
                                $SGNS-1*400
                                        .ENDC
                                        MOV     #<WIDTH*400>!$SGNS!$BLKS!RADIX,R5
                                        TST     R2
                                        BEQ     .+4
                                        BIS     #1*1000,R5
                                        .ENDM


                                ; INPUTS:
                                ;
                                ;       R0=POINTER TO ASCII OUTPUT STRING
                                ;       R1-ADDRESS OF DOUBLE PRECISION VALUE
                                ;       R2-ZERO COMPRESS FLAG
                                ;
033064                          CDDMG:
033064  004567  000250                  JSR     R5,SAVRG                ;SAVE THE NON-VOLATILE REGISTERS
033070  010003                          MOV     R0,R3                   ;COPY THE STRING POINTER
033072  012704  023420                  MOV     #10000.,R4              ;SET DIVISOR
033076                                  CBTAS   10.,0,NOSIGN,BLANKS     ;SET CONVERSION PARAMETERS
033112  022104                          CMP     (R1)+,R4                ;TEST FOR OVERFLOW
033114  103042                          BHIS    40$                     ;IF HIS, OVERFLOW
033116  011102                          MOV     (R1),R2                 ;GET LOW PART OF NUMBER
033120  014101                          MOV     -(R1),R1                ;GET HIGH PART OF NUMBER
033122  010400                          MOV     R4,R0                   ;COPY DIVISOR
033124  004767  000110                  JSR     PC,DDIV                 ;DO DOUBLE PREC. DIVIDE
033130  010046                          MOV     R0,-(SP)                ;SAVE REMAINDER
033132  010201                          MOV     R2,R1                   ;COPY QUOTIENT
```

```
2840  033134  001011                    BNE     11$                     ;IF NE, SOMETHING TO PRINT
2841  033136  012702  000005            MOV     #5,R2                   ;OTHERWISE, FILL FIELD WITH BLANKS
2842  033142  112723  000040    21$:    MOVB    #' ,(R3)+
2843  033146  005302                    DEC     R2
2844  033150  001374                    BNE     21$
2845  033152  052705  003000            BIS     #3000,R5                ;DISABLE BLANK SUPPRESSION
2846  033156  000411                    BR      20$
2847  033160  012702  024000    11$:    MOV     #5*4000,R2              ;SET FIELD WIDTH
2848  033164  004767  000020            JSR     PC,30$                  ;OUTPUT HIGH ORDER DIGITS
2849  033170  052705  001000            BIS     #1*1000,R5              ;DISABLE ZERO COMPRESS
2850  033174  042705  002000            BIC     #1*2000,R5              ;DISABLE BLANKS
2851  033200  010003            31$:    MOV     R0,R3                   ;SET STRING POINTER
2852  033202                    20$:
2853  033202  012601                    MOV     (SP)+,R1                ;GET LOW ORDER VALUE
2854  033204  012702  020000            MOV     #4*4000,R2              ;SET FIELD WIDTH
2855  033210                    30$:
2856  033210  010300                    MOV     R3,R0                   ;GET STRING POINTER
2857  033212  050502                    BIS     R5,R2                   ;INCLUDE RADIX & BLANK SUPPRESS
2858  033214  004767  177506            JSR     PC,CBTA                 ;CONVERT TO ASCII
2859  033220  000406                    BR      60$                     ;EXIT
2860  033222                    40$:
2861  033222  012702  000011            MOV     #9.,R2                  ;GET COUNT
2862  033226                    50$:
2863  033226  112720  000052            MOVB    #'*,(R0)+               ;FILL FIELD WITH ASTERISKS
2864  033232  005302                    DEC     R2
2865  033234  001374                    BNE     50$                     ;   ''SOB     R2,50$''
2866  033236                    60$:
2867  033236                            RETURN
2868
2869
2870
2871                                      .SBTTL  DOUBLE PRECISION DIVIDE ROUTINE
2872                                    ;
2873                                    ; INPUTS:
2874                                    ;
2875                                    ; R2=LOW ORDER OF DIVIDEND
2876                                    ; R1-HIHG ORDER OF DIVIDEND
2877                                    ; R0=DIVISOR (15 BITS UNSIGNED)
2878                                    ;
2879                                    ; OUTPUTS:
2880                                    ;
2881                                    ; R2=LOW ORDER OF QUOTIENT
2882                                    ; R1-HIGH ORDER OF QUOTIENT
2883                                    ; R0=REMAINDER
2884                                    ;
2885
2886  033240                    DDIV:
2887  033240  010346                    MOV     R3,-(SP)                ;SAVE R3
2888  033242  012703  000040            MOV     #32.,R3                 ;SET ITERATION COUNT IN R3
2889  033246  010046                    MOV     R0,-(SP)                ;STACK DIVISOR
2890  033250  005000                    CLR     R0                      ;SET REMAINDER TO 0
2891  033252                    1$:
2892  033252  006302                    ASL     R2                      ;SHIFT THE ENTIRE DIVIDEND
2893  033254  006101                    ROL     R1                      ;... ONE BIT TO THE LEFT AND...
2894  033256  006100                    ROL     R0                      ;... INTO THE REMAINDER
2895  033260  020016                    CMP     R0,(SP)                 ;IS REMAINDER GE DIVISOR?
```

```
2896  033262  103402                  BLO     2$                      ;NO, SKIP TO ITERATION CONTROL
2897  033264  161600                  SUB     (SP),R0                 ;YES, SUB DIVISOR OUT
2898  033266  005202                  INC     R2                      ;AND INCR THE QUOTIENT
2899  033270                  2$:
2900  033270  005303                  DEC     R3                      ;REPEAT AS LONG AS NECESSARY
2901  033272  003367                  BGT     1$
2902  033274  005726                  TST     (SP)+                   ;PURGE DIVISOR FROM STACK
2903  033276  012603                  MOV     (SP)+,R3                ;RESTORE R3
2904  033300                          RETURN
2905
2906
2907                                  .SBTTL  INTEGER DIVIDE MAGNITUDE NUMBERS
2908                          ;
2909                          ;
2910                          ; INPUTS:
2911                          ;
2912                          ; R0=DIVIDEND
2913                          ; R1-DIVISOR
2914                          ;
2915                          ; OUTPUTS:
2916                          ;
2917                          ; QUOTIENT IS RETURNED IN R0
2918                          ; REMAINDER IS RETURNED IN R1
2919                          ;
2920                          DIV:
2921  033302                          MOV     #20,-(SP)               ;SET LOOP COUNT
2922  033302  012746  000020          MOV     R1,-(SP)                ;SAVE DIVISOR FOR SUBTRACTS
2923  033306  010146                  CLR     R1                      ;CLEAR REMAINDER
2924  033310  005001          30$:
2925  033312                          ASL     R0                      ;DOUBLE LEFT SHIFT
2926  033312  006300                  ROL     R1
2927  033314  006101                  CMP     R1,(SP)                 ;SUBTRACT OUT DIVISOR
2928  033316  020116                  BLO     40$                     ;IF LO, NO
2929  033320  103402                  SUB     (SP),R1                 ;SUBTRACT OUT DIVISOR
2930  033322  161601                  INC     R0                      ;ADD IN LOW BIT
2931  033324  005200          40$:
2932  033326                          DEC     2(SP)                   ;DECREMENT REPEAT COUNT
2933  033326  005366  000002          BGT     30$                     ;IF GT, MORE TO GO
2934  033332  003367          50$:
2935  033334                          CMP     (SP)+,(SP)+             ;CLEAN STACK
2936  033334  022626                  RETURN
2937  033336
2938                          ; SAVE/RESTORE NONVOLATILE REGISTERS
2939
2940
2941
2942  033340  010446          SAVRG:  MOV     R4,-(SP)                ;SAVE R4 &R3
2943  033342  010346                  MOV     R3,-(SP)
2944  033344  010546                  MOV     R5,-(SP)                ;PUT RETURN ADDRESS ON STACK
2945  033346  016605  000006          MOV     6(SP),R5                ;RETRIEVE REAL R5
2946  033352  004736                  JSR     PC,@(SP)+               ;CALL THE CALLER ...!
2947  033354  012603                  MOV     (SP)+,R3                ;RESTORE NON VOLATILE REGISTERS
2948  033356  012604                  MOV     (SP)+,R4
2949  033360  012605                  MOV     (SP)+,R5
2950  033362                          RETURN
```

```
                                        .SBTTL  QUEUE HANDLING ROUTINES
                                ; THIS MODULE CONTAINS 2 SUBROUTINES, ENQ & DEQ, TO ENQUEUE & DEQUEUE
                                ; WORDS, RESPECTIVELY, IN A FIRST-IN-FIRST-OUT LIST.
                                ;
                                        .LIST   MEB


                                        PROC    ENQ,<ITEM,QUEUE>
                                ITEM    =       2
                                QUEUE   =       4

                                ENQ:                                            ;**ENTRY POINT**
                                ;
                                ;  APPEND  ITEM  (A WORD) TO THE FIRST-IN-FIRST-OUT LIST  QUEUE .
                                ;
                                        MOV     QUEUE(R5),R4                    ;GET QUEUE ADDRESS.
                                        CMP     @R4,QSIZE(R4)                   ;IF QUEUE IS FULL,
                                        BGE     NQFULL                          ; SIGNAL TRAGIC ERROR.
                                        MOV     @ITEM(R5),@QBACK(R4)            ;PUT ITEM AT BACK OF QUEUE.
                                        ADD     #2,QBACK(R4)                    ;UPDATE BACK POINTER.
                                        CMP     QBACK(R4),QBOT(R4)              ; ''
                                        BLO     NQNOWP                          ; ''
                                        MOV     QTOP(R4),QBACK(R4)              ;
                                NQNOWP: INC     @R4                             ;INCREMENT NO. OF ELEMENTS.
                                        CLC                                     ;INDICATE SUCCESSFUL ENQ.
                                NQRET:  RETURN                                  ;RETURN.
                                        RTS     PC

                                NQFULL: SEC                                     ;INDICATE UNSUCCESSFUL ENQ.
                                        BR      NQRET                           ;IGNORE ITEM & RETURN.


                                        PROC    DEQ,<ITEM,QUEUE>
                                ITEM    =       2
                                QUEUE   =       4

                                DEQ:                                            ;**ENTRY POINT**
                                ;
                                ;  REMOVE A WORD ENTRY FROM THE FIRST-IN-FIRST-OUT LIST  QUEUE  &
                                ;  STORE IT AT  ITEM .
                                ;
                                        MOV     QUEUE(R5),R4                    ;GET QUEUE ADDRESS.
                                        TST     @R4                             ;IF QUEUE IS EMPTY,
                                        BEQ     DQEMP                           ; SIGNAL TRAGIC ERROR.
                                        MOV     @QFRONT(R4),@ITEM(R5)           ;RETRIEVE FRONT ELEMENT.
                                        ADD     #2,QFRONT(R4)                   ;UPDATE FRONT POINTER.
                                        CMP     QFRONT(R4),QBOT(R4)             ; ''
                                        BLO     DQNOWP                          ; ''
                                        MOV     QTOP(R4),QFRONT(R4)             ;
                                DQNOWP: DEC     @R4                             ;DECREMENT NO. OF ELEMENTS.
                                        CLC                                     ;INDICATE SUCCESSFUL DEQ.
                                DQRET:  RETURN                                  ;RETURN.
                                        RTS     PC

                                DQEMP:  SEC                                     ;INDICATE UNSUCCESSFUL DEQ
                                        MOV     #-1,@ITEM(R5)                   ;SET ITEM TO ALL ONES.
```

```
 2998  033522  000772                    BR      DQRET                   ;RETURN.
```

```
3000                            ; SUBROUTINE TO SCAN AN INPUT COMMAND & CALL ITS
3001                            ; PROCESSING ROUTINE.
3002
3003                                    .LIST   MEB
3004
3005
3006                                    .MACRO  SPAN    REG,CHAR,?L
3007
3008                            ;  THIS MACRO SCANS THE STRING OF CHARACTERS STARTING AT
3009                            ;  @REG UNTIL IT FINDS ONE NOT EQUAL TO CHAR.  REG IS
3010                            ;  SET POINTING TO THAT CHARACTER.
3011
3012                            L:      CMPB    (REG)+,CHAR
3013                                    BEQ     L
3014                                    DEC     REG
3015
3016                                    .ENDM
3017
3018                                    .MACRO  BREAK   REG,CHRSET,?HH,?JJ
3019
3020                            ;  THIS MACRO SCANS THE STRING STARTING AT @REG UNTIL
3021                            ;  IT FINDS A CHARACTER THAT IS A MEMBER OF CHRSET.
3022                            ;  REG IS SET POINTING TO THAT CHARACTER.
3023
3024                            ;  EACH MEMBER OF CHRSET IS AN ADDRESSABLE QUANTITY.
3025
3026
3027                            HH:
3028                                    .IRP    LS,<CHRSET>
3029                                    CMPB    @REG,LS
3030                                    BEQ     JJ
3031                                    .ENDM
3032                                    INC     REG
3033                                    BR      HH
3034
3035                            JJ:
3036                                    .ENDM
3037
3038
3039                                    .MACRO  SYNCLS  CHAR,CLASS,?CC,?DD,?EE,?FF,?GG
3040
3041                            ;  THIS MACRO DETERMINES THE SYNTACTIC CLASS OF AN
3042                            ;  OBJECT BEGINNING WITH CHAR.  THE CLASS IS RETURNED
3043                            ;  IN CLASS AS FOLLOWS:
3044                            ;       CLASS   =       0 (WORD) IF CHAR = (A,....,Z,-)
3045                            ;                       2(NUMBER)IF CHAR = (0,....,9)
3046                            ;                       6(END OF LINE) IF CHAR = CARRIAGE RETURN
3047                            ;                       4(CHARACTER STRING) OTHERWISE
3048
3049                                    CMPB    CHAR,#'A
3050                                    BLT     EE
3051                                    CMPB    CHAR,#'Z
3052                                    BGT     DD
3053                            CC:     CLR     CLASS
3054                                    BR      GG
3055
```

```
3056                                    DD:     MOV     #4,CLASS
3057                                            BR      GG
3058
3059                                    EE:     CMPB    CHAR,#'0
3060                                            BLT     FF
3061                                            CMPB    CHAR,#'9
3062                                            BGT     DD
3063                                            MOV     #2,CLASS
3064                                            BR      GG
3065
3066                                    FF:     CMPB    CHAR,#'-
3067                                            BEQ     CC
3068                                            CMPB    CHAR,#15
3069                                            BNE     DD
3070                                            MOV     #6,CLASS
3071                                    GG:
3072
3073                                            .ENDM
```

```
                        .SBTTL  COMMAND PROCESSOR INITIATING ROUTINE

                        PROC    COMAND,<INPLIN,KWTABL>
                INPLIN  =       2
                KWTABL  =       4

                COMAND:                                 ;**ENTRY POINT**
                ;
                ; THIS ROUTINE CAUSES THE COMMAND SPECIFIED BY  INPLIN  TO BE
                ; PROCESSED, AS DESCRIBED BELOW.  INPLIN  IS A STRING OF ASCII
                ; CHARACTERS ENDED BY A CARRIAGE RETURN CODE.
                ;
                ;     - INPLIN IS LEXICALLY SCANNED USING THE  LXSCAN  ROUTINE.
                ;     - THE 1ST OBJECT IS CONVERTED TO A PROCESSING ROUTINE ADDRESS
                ;       USING THE  KEYWD  ROUTINE & THE KEYWORD TABLE  KWTABL
                ;       SUPPLIED BY THE CALLING PROGRAM.
                ;     - THE PROCESSING ROUTINE IS CALLED WITH THE OUTPUT FROM  LXSCAN
                ;       ON THE STACK STARTING AT 2(SP).  (THE RETURN ADDRESS OF THIS
                ;       CALL OCCUPIES THE TOP WORD OF THE STACK.)
                ;     - THE  LXSCAN  OUTPUT IS REMOVED FROM THE STACK.
                ;
                ; IF  LXSCAN  OR  KEYWD  OR THE PROCESSING ROUTINE RETURN AN ERROR
                ; CONDITION, THEN C = 1; OTHERWISE, C = 0.
                ;
033524 010546           MOV     R5,-(SP)                ;SAVE PAR LIST POINTER.
033526 010667 000076    MOV     SP,CMMARK               ;SAVE STACK POINTER.
033532                  LXSCAN  INPLIN(R5)              ;LEXICALLY SCAN  INPLIN .
(1) 033532 016501 000002        MOV     INPLIN(R5),R1
(1) 033536 004167 000310        JSR     R1,LXSCAN
033542 103426           BCS     CMRET                   ;IF ERROR, RETURN WITH C=1.
033544 005716           TST     @SP                     ;HAVE WE ANY OBJECTS ?
033546 003005           BGT     1$                      ;NO, IGNORE BLANK COMMAND.
033550 012767 177777 164210     MOV     #-1,ESCFLG
033556 000167 000036    JMP     CMRET
033562 016700 000042    1$:     MOV     CMMARK,R0       ;YES, DETERMINE ADDRESS
033566 162700 000006            SUB     #6,PC           ; OF FIRST OBJECT.
033572 017705 000032            MOV     @CMMARK,R5      ;RESTORE PAR LIST POINTER.
033576                  KEYWD   R0,KWTABL(R5)           ;GET ADDR OF COMMAND PROCESSOR.
(1) 033576 010046               MOV     R0,-(SP)
(1) 033600 016546 000004        MOV     KWTABL(R5),-(SP)
(1) 033604 004767 000024        JSR     PC,KEYWD
033610 103403           BCS     CMRET                   ;IF INVALID COMMAND, MAKE
                                                        ; ERROR RETURN.
033612 012705 033632    MOV     #NULPAR,R5              ;LOAD NULL PAR LIST ADDRESS.
033616 004736           JSR     PC,@(SP)+               ;PROCESS COMMAND & POP ADDR.
033620 016706 000004    CMRET:  MOV     CMMARK,SP       ;RESTORE STACK TO ENTRY STATUS.
033624 030026           BIT     R0,(SP)+                ;THROW AWAY SAVED R5 (WITHOUT
                                                        ; AFFECTING C BIT).
                        RETURN                          ;RETURN TO CALLING PROGRAM.
033626 000207           RTS     PC

                ; DATA AREAS.
033630 033333   CMMARK:  .WORD   33333                  ;STORAGE FOR STACK PTR ON ENTRY.
033632 000000   NULPAR:  .WORD   0                      ;PAR LIST CONTAINING NO PARS.
```

```
                                        .SBTTL  KEYWORD PROCESSING ROUTINE

                                ;  SUBROUTINE TO DETERMINE THE ADDRESS OF THE PROCESSING
                                ;  ROUTINE ASSOCIATED WITH THE KEYWORD REPRESENTED BY
                                ;  THE SYNTACTIC OBJECT POINTED TO BY THE ARGUMENT
                                ;  SRC.  CONVERSION FROM KEYWORD TO ROUTINE ADDRESS
                                ;  IS DONE AS DEFINED IN THE KEYWORD TABLE
                                ;  POINTED TO BY THE ARGUMENT TABLAD.

                                ;  ON ENTRY, THE TOP OF THE STACK IS AS FOLLOWS:
                                ;       (SP):   RETURN ADDRESS
                                ;       2(SP):  TABLAD
                                ;       4(SP):  SRC

                                ;  CALLING INSTRUCTION:                 JSR     PC,KEYWD

                                ;  ON RETURN, THE TOP OF THE STACK IS AS FOLLOWS:
                                ;       (SP):   ROUTINE ADDRESS, IF KEYWORD IN TABLE; 0 IF NOT.

                                ;  IF THE KEYWORD IS IN THE TABLE, C=0 ON RETURN.  IF NOT, C 1.

                                ;  STACK POINTER OFFSETS
                                ;ROUTAD =       0                       ;ROUTINE ADDR FOR
                                                                        ; CURRENT TABLE
                                                                        ; ELEMENT.
000002                          ADRINP  =       2                       ;ADDR OF INPUT WORD
000004                          LENINP  =       4                       ;#CHAR IN INPUT WORD
000022                          RETURN  =       22                      ;SUBROUTINE RETURN ADDR
000024                          TABLAD  =       24                      ;ADDR OF KEYWORD TABLE
000026                          SRC     =       26                      ;ADDR OF INPUT OBJECT
000026                          RESULT  =       26                      ;RESULT RETURNED

033634                          KEYWD:                                  ;**ENTRY POINT**
033634                                  REGSAV                          ;SAVE REGISTERS.
(1) 033634  004567  000160              JSR     R5,REGSAV
033640  162706  000006                  SUB     #6,SP                   ;ALLOCATE STACK SPACE
033644  016602  000026                  MOV     SRC(SP),R2              ;GET ADDR OF INPUT
                                                                        ; OBJECT
033650  005722                          TST     (R2)+                   ;IF OBJECT NOT A WORD
033652  001051                          BNE     NOTHER                  ; THEN EXIT: NOT FOUND
033654  012205                          MOV     (R2)+,R5                ;GET # CHAR IN OBJECT.
033656  010566  000004                  MOV     R5,LENINP(SP)           ;STORE ON STACK.
033662  011204                          MOV     @R2,R4                  ;GET ADDR OF INPUT WORD.
033664  010466  000002                  MOV     R4,ADRINP(SP)           ;STORE ON STACK.
033670  016603  000024                  MOV     TABLAD(SP),R3           ;GET ADDR. OF KEYWORD
                                                                        ; TABLE.
033674  012316                  GTLENS: MOV     (R3)+,@SP               ;SAVE ROUTINE ADDR OF
                                                                        ; 1ST ELEMENT.
033676  112302                          MOVB    (R3)+,R2                ;GET MINIMUM LENGTH
033700  042702  177400                  BIC     #177400,R2              ; OF TABLE WORD.
033704  112301                          MOVB    (R3)+,R1                ;GET FULL LENGTH.
033706  042701  177400                  BIC     #177400,R1              ;
033712  001431                          BEQ     NOTHER                  ;IF 0, THEN NO MORE
                                                                        ; TABLE TO SEARCH
033714  122423                  NXTCH:  CMPB    (R4)+,(R3)+             ;COMPARE INPUT CHAR
                                                                        ; WITH TABLE CHAR.
```

```
3175  033716  002427                  BLT     NOTHER          ;IF<, THEN NO MATCH
3176                                                          ; EXISTS.
3177  033720  003005                  BGT     NXTWD           ;IF>, THEN TRY NEXT
3178                                                          ; TABLE WORD.
3179  033722  005305                  DEC     R5              ;IF INPUT STRING
3180                                                          ; EXHAUSTED,
3181  033724  001413                  BEQ     THFND           ; WE MAY HAVE FOUND
3182                                                          ; MATCH.
3183  033726  005301                  DEC     R1              ;IF MORE CHAR IN TABLE WORD
3184  033730  001371                  BNE     NXTCH           ; TO TEST, GO & TEST THEM.
3185  033732  005201                  INC     R1
3186  033734  060103          NXTWD:  ADD     R1,R3           ;GET ADDR OF NEXT
3187  033736  042703  000001          BIC     #1,R3           ; TABLE ENTRY.
3188  033742  016604  000002          MOV     ADRINP(SP),R4   ;POINT TO BEGINNING
3189                                                          ; OF INPUT WORD.
3190  033746  016605  000004          MOV     LENINP(SP),R5   ;GET LENGTH OF INPUT WORD.
3191  033752  000750                  BR      GTLENS          ;GET LENGTHS OF TABLE WORD.
3192
3193  033754  026602  000004  THFND:  CMP     LENINP(SP),R2   ;IF LEN(INP.WD) < MIN LEN (TABLE
3194  033760  002406                  BLT     NOTHER          ; WORD), WORD IS NOT IN TABLE.
3195  033762  011666  000026          MOV     (SP),RESULT(SP) ;SAVE ROUTINE ADDR. OF
3196                                                          ; MATCH.
3197  033766  062706  000006          ADD     #6,SP           ;FREE LOCAL STACK SPACE.
3198  033772  000241                  CLC                     ;CLEAR CARRY BIT.
3199  033774  000405                  BR      KWEXIT
3200
3201                          ; WORD IS NOT IN TABLE. SET RESULT TO 0 & SET Z BIT ON.
3202  033776  005066  000026  NOTHER: CLR     RESULT(SP)      ;CLEAR RESULT.
3203  034002  062706  000006          ADD     #6,SP           ;FREE LOCAL STACK SPACE.
3204  034006  000261                  SEC                     ;SET CARRY BIT.
3205  034010                  KWEXIT: REGRES                  ;RESTORE REGISTERS.
 (1)  034010  004567  000020          JSR     R5,REGRES
3206  034014  012616                  MOV     (SP)+,@SP       ;POP AN ARGUMENT.
3207  034016  000207                  RTS     PC              ;RETURN.
```

```
                        .SBTTL  REGISTER SAVE & RESTORE ROUTINES

                ;  SUBROUTINE TO SAVE R0 - R5 ON STACK
                ;  CALLING SEQUENCE:    JSR     R5,REGSAV
                ;
034020          REGSAV:                                 ;**ENTRY POINT**
034020  010446          MOV     R4,-(SP)
034022  010346          MOV     R3,-(SP)
034024  010246          MOV     R2,-(SP)
034026  010146          MOV     R1,-(SP)
034030  010046          MOV     R0,-(SP)
034032  000195          JMP     @R5


                ;  SUBROUTINE TO RESTORE R0-R5 FROM STACK
                ;  THE CONDITION CODE BITS IN THE PS ARE DESTROYED,
                ;  EXCEPT FOR THE CARRY BIT, WHICH IS PRESERVED.

                ;  CALLING SEQUENCE:    JSR     R5,REGRES
034034          REGRES:                                 ;**ENTRY POINT**
034034  030025          BIT     R0,(SP)+                ;THROW AWAY OLD R5 VALUE.
034036  012600          MOV     (SP)+,R0
034040  012601          MOV     (SP)+,R1
034042  012602          MOV     (SP)+,R2
034044  012603          MOV     (SP)+,R3
034046  012604          MOV     (SP)+,R4
034050  000205          RTS     R5
```

```
                        .SBTTL  LEXICAL SCAN ROUTINE

                ;
                ; PERFORM LEXICAL SCAN OF INPUT COMMAND IN
                ; BUFFER.
                ;
                ; THREE CLASSES OF SYNTACTIC OBJECTS ARE RECOGNIZED:
                ;       1. WORD: A STRING OF CHARACTERS BEGINNING WITH
                ;                A LETTER & TERMINATED WITH A
                ;                BLANK OR CARRIAGE RETURN.
                ;          2. NUMBER: A STRING OF OCTETS TERMINATED WITH A
                ;                   BLANK OR CARRIAGE RETURN, OR A STRING OF DIGITS
                ;                   TERMINATED WITH A DOT.
                ;          3. CHARACTER STRING: A STRING SURROUNDED BY 2 INSTANCES
                ;                               (1 ON EACH END) OF A SPECIAL CHARACTER.
                ;
                ; SYNTACTIC OBJECTS ARE SEPARATED BY 1 OR MORE BLANKS.
                ; THE COMMAND IS ENDED BY A CARRIAGE RETURN.
                ;
                ; THIS LEXICAL SCANNER DETERMINES THE LOCATIONS OF THE
                ; SYNTACTIC OBJECTS & DETERMINES THEIR CLASSES.
                ; NUMBERS ARE CONVERTED TO THEIR BINARY VALUES.
                ;
                ; AFTER THE LEXICAL SCAN IS PERFORMED, EACH
                ; SYNTACTIC OBJECT WILL BE REPRESENTED ON THE STACK
                ; AS A 3 WORD QUANTITY AS FOLLOWS:
                ;       TOP WORD        =       SYNTACTIC CLASS:
                ;                                       0==>WORD
                ;                                       2==>NUMBER
                ;                                       4==>CHARACTER STRING
                ;       2ND WORD        -       LENGTH OF OBJECT IN CHARACTERS; NOT
                ;                               SIGNIFICANT FOR NUMBERS; DOES NOT
                ;                               INCLUDE SURROUNDING DELIMITERS FOR CHAR.
                ;                               STRINGS.
                ;       3RD WORD        =       VALUE OF NUMBER, OR POINTER TO 1ST
                ;                               LETTER OF KEYWORD OR 1ST SIGNIFICANT
                ;                               CHARACTER OF STRING (IE: NOT SUR-
                ;                               ROUNDING DELIMITER)
                ;
                ; AT THE END OF THE LEXICAL SCAN THE STACK WILL BE
                ; ARRANGED AS FOLLOWS (N - NO. OF SYNTACTIC OBJECTS):
                ;       !--------------!
                ;       !      N       ! <--- SP
                ;       !--------------!
                ;       !  NTH OBJECT  !
                ;       !--------------!
                ;       !      :       !
                ;       !--------------!
                ;       !  1ST OBJECT  !
                ;       !--------------!
                ; WHERE EACH OBJECT IS REPRESENTED AS ABOVE.
                ;
                ;  CALLING SEQUENCE:    R1=BUFFER ADDRESS
                ;                       JSR     R1,LXSCAN
                ;       ON RETURN, REGISTERS 0-5 ARE UNDEFINED.
                ;       EXCEPT FOR THE ABOVE TABLE, THE STACK IS AS IT
                ;       WAS BEFORE ENTRY TO THE ROUTINE.
```

```
                        ;  IF LXSCAN DETECTS AN ERROR CONDITION, IT RETURNS WITH THE CARRY (C)
                        ;  BIT SET; OTHERWISE IT IS CLEAR.  AT THE MOMENT, THE
                        ;  ONLY ERROR CONDITION DETECTED BY LXSCAN IS A STRING WHICH IS MISSING
                        ;  ITS CLOSING DELIMITER.

034052                  LXSCAN:                                 ;**ENTRY POINT**
034052  005002                  CLR     R2                      ;CLEAR #OBJECTS.
034054  012605                  MOV     (SP)+,R5                ;GET ADDR. OF COMMAND BUFFER.
034056                  NXTEL:  SPAN    R5,<#' >                ;R5 = ADDR (1ST NONBLANK CHAR)
034056  122527  00004C  64$:    CMPB    (R5)+,#'
034062  001775                  BEQ     64$
034064  005305                  DEC     R5
034066                          SYNCLS  @R5,R4                  ;DETERMINE SYNTACTIC CLASS.
034066  121527  00010C          CMPB    @R5,#'A
034072  002410                  BLT     67$
034074  121527  000132          CMPB    @R5,#'Z
034100  003002                  BGT     66$
034102  005004          65$:    CLR     R4
034104  000424                  BR      69$
034106  012704  000004  66$:    MOV     #4,R4
034112  000421                  BR      69$
034114  121527  000060  67$:    CMPB    @R5,#'0
034120  002406                  BLT     68$
034122  121527  000071          CMPB    @R5,#'9
034126  003367                  BGT     66$
034130  012704  000002          MOV     #2,R4
034134  000410                  BR      69$
034136  121527  000055  68$:    CMPB    @R5,#'-
034142  001757                  BEQ     65$
034144  121527  000015          CMPB    @R5,#15
034150  001356                  BNE     66$
034152  012704  000006          MOV     #6,R4
034156  000174  034162          JMP     @SJT(R4)                ;PROCESS OBJECT

034162  034172          SJT:    .WORD   SCWORD                  ;WORD PROCESSOR.
034164  034232                  .WORD   SCNO                    ;NUMBER PROCESSOR.
034166  034376                  .WORD   SCCHAR                  ;CHARACTER STRING PROCESSOR.
034170  034436                  .WORD   SCEOL                   ;END OF LINE PROCESSOR.

034172  010503  SCWORD: MOV     R5,R3                           ;R3 = START ADDR OF OBJECT
034174                  BREAK   R5,<<#' >,<#15>>                ;R5   ADDR (NEXT BLANK OR CR)
034174  121527  000040          CMPB    @R5,#'
034200  001405                  BEQ     65$
034202  121527  000015          CMPB    @R5,#15
034206  001402                  BEQ     65$
034210  005205                  INC     R5
034212  000770                  BR      64$
034214  010346                  MOV     R3,-(SP)                ;PUSH ADDR OF OBJECT ONTO STACK.
034216  160503                  SUB     R5,R3                   ;R3 = (-LENGTH OF OBJECT)
034220  005403  PLAC:   NEG     R3                              ;NEGATE TO GET LENGTH.
034222  010346                  MOV     R3,-(SP)                ;PUSH LENGTH ONTO STACK.
034224  010446                  MOV     R4,-(SP)                ;PUSH SYNTACTIC CLASS ONTO
                                                                ;  STACK
034226  005202  LXINCN: INC     R2                              ;INCREMENT
034230  000712                  BR      NXTEL                   ;GO TO SCAN NEXT ELEMENT
```

```
3320
3321  034232  005000            SCNO:   CLR     R0              ;CLEAR ACCUMULATED NO.
3322  034234  010504                    MOV     R5,R4           ;SAVE POINTER TO 1ST DIGIT.
3323  034236  121527  000060    SNXTDG: CMPB    @R5,#'0         ;IF CHAR <'0'
3324  034242  002415                    BLT     SNTDIG          ; THEN TREAT AS DELIMITER.
3325  034244  121527  000071            CMPB    @R5,#'9         ;IF CHAR >'9'
3326  034250  003012                    BGT     SNTDIG          ; THEN TREAT AS DELIMITER.
3327  034252                            MULT    10.,R0,R3       ;MULTIPLY PREVIOUS DIGITS BY 10.
 (2)  034252  006300                    ASL     R0
 (2)  034254  010003                    MOV     R0,R3
 (2)  034256  006300                    ASL     R0
 (2)  034260  006300                    ASL     R0
 (2)  034262  060300                    ADD     R3,R0
3328  034264  112503                    MOVB    (R5)+,R3
3329  034266  142703  000060            BICB    #60,R3          ;CLEAR TOP BITS OF ASCII CODE.
3330  034272  060300                    ADD     R3,R0           ;ADD DIGIT.
3331  034274  000760                    BR      SNXTDG          ;GET NEXT DIGIT.
3332
3333  034276  121527  000040    SNTDIG: CMPB    @R5,#' '        ;IF DELIMITER = SPACE, TRY
3334  034302  001413                    BEQ     TRYOCT          ; CONVERTING OCTAL NO.
3335  034304  121527  000015            CMPB    @R5,#15         ;IF DELIMITER = CR, TRY
3336  034310  001410                    BEQ     TRYOCT          ; CONVERTING OCTAL NO.
3337  034312  122527  000056            CMPB    (R5)+,#'.       ;IF DELIMITER IS NOT DOT,
3338  034316  001052                    BNE     LXERR           ; THEN SIGNAL LXSCAN ERROR.
3339  034320  010046    PUSHNO: MOV     R0,-(SP)        ;PUT CONVERTED NO. ON STACK.
3340  034322  005046            CLR     -(SP)           ;SET OBJECT LENGTH TO ZERO.
3341  034324  012746  000002            MOV     #2,-(SP)        ;SET OBJECT CLASS TO 2
3342                                                            ; (NUMBER).
3343  034330  000736                    BR      LXINCN          ;INCREMENT OBJ. COUNT &
3344                                                            ; SCAN NEXT OBJ.
3345
3346  034332  005000    TRYOCT: CLR     R0              ;CLEAR ACCUMULATED NO.
3347  034334  121427  000060    ONXTDG: CMPB    @R4,#'0         ;IF CHAR < '0',
3348  034340  002413                    BLT     ODELIM          ; THEN TREAT AS DELIMITER.
3349  034342  121427  000067            CMPB    @R4,#'7         ;IF CHAR > '7',
3350  034346  003010                    BGT     ODELIM          ; THEN TREAT AS DELIMITER.
3351  034350                            MULT    8.,R0           ;MULTIPLY PREVIOUS DIGITS BY 8.
 (2)  034350  006300                    ASL     R0
 (2)  034352  006300                    ASL     R0
 (2)  034354  006300                    ASL     R0
3352  034356  112403                    MOVB    (R4)+,R3        ;GET CHARACTER.
3353  034360  142703  000060            BICB    #60,R3          ;CLEAR TOP BITS OF ASCII CODE.
3354  034364  060300                    ADD     R3,R0           ;ADD DIGIT.
3355  034366  000762                    BR      ONXTDG          ;GET NEXT DIGIT.
3356
3357  034370  020405    ODELIM: CMP     R4,R5           ;IF NOT AT END OF NO.(DUE TO
3358  034372  001024                    BNE     LXERR           ; '8' OR '9') , LXSCAN ERROR.
3359  034374  000751                    BR      PUSHNO          ;GENERATE OBJECT FOR NUMBER.
3360
3361  034376  112500    SCCHAR: MOVB    (R5)+,R0        ;R0 = DELIMITER OF STRING.
3362  034400  010503                    MOV     R5,R3           ;R3 = ADDR (1ST CHAR OF
3363                                                            ;  STRING ITSELF)
3364  034402                            BREAK   R5,<R0,#15>     ;R5 - ADDR (NEXT DELIM OR CR).
 (2)  034402  121500                    CMPB    @R5,R0
 (2)  034404  001405                    BEQ     65$
 (2)  034406  121527  000015            CMPB    @R5,#15
```

```
  (2)  034412  001402                    BEQ     65$
  (1)  034414  005205                    INC     R5
  (1)  034416  000771                    BR      64$
3365   034420  121527  000015            CMPB    @R5,#15         ;WAS CR FOUND BEFORE DELIMITER?
3366   034424  001407                    BEQ     LXERR           ;YES, PROCESS ERROR
3367   034426  010346                    MOV     R3,-(SP)        ;PUSH ADDR. OF STRING ONTO
                                                                  ; STACK.
3368
3369   034430  160503                    SUB     R5,R3           ;R3 = (-LENGTH OF STRING EXCL.
                                                                  ; DELIMITER)
3370
3371   034432  005205                    INC     R5              ;R5 = ADDR (CHAR AFTER CLOSING
                                                                  ; DELIMITER).
3372
3373   034434  000671                    BR      PLAC            ;FINISH OFF PROCESSING STRING.
3374
3375   034436  010246            SCEOL:  MOV     R2,-(SP)        ;PUSH NO. OF OBJECTS ONTO STACK
3376   034440  000241                    CLC                     ;INDICATE NO ERROR .
3377   034442  000111            LXIT:   JMP     @R1             ;RETURN TO CALLING PROGRAMME .
3378
3379                              ;  PROCESS LEXICAL SCAN ERROR .
3380   034444  000261            LXERR:  SEC                     ;INDICATE ERROR OCCURRED .
3381   034446  005302                    DEC     R2              ;CLEAN UP STACK & EXIT .
3382   034450  002774                    BLT     LXIT            ; ..
3383   034452  062706  000006            ADD     #6,SP           ; ..
3384   034456  000772                    BR      LXERR           ;
3385
3386
3387           000001                    .END
```

```
ADQLD   006726            500   1209#
ADRINP= 000002           3145#  3162*  3188
ADSILO  020006            563    610   2413#
AQAREA  016450           2349#
AQEND = 016550           2349#
AQQ     016434            848*   849*   850*   983*   988*   989*  1220  1229  1341  1345  2349#
AQSIZE= 000040            310#  2349
ASEXT   006272           1089   1094#
A.BKSL  016162           2179   2310#
A.C     016170           2157   2313#
A.CR    016054           2259   2270#
A.L     016172           2158   2314#
A.LF    01605C           2260   2270#
A.P     016166           2156   2312#
A.PR    016174           2159   2315#
BKELEM  030276           1228*  1229   2202*  2203   2245*  2247  2466#
BSYINT  010416           1395   1457#
BUSY  = 004000            345#
B00   = 000001            366#  1644
B01   = 000002            365#   623    907   1340   1450   1459  1484  1510  1545  1630  1678  1688
B02   = 000004            364#
B03   = 000010            363#
B04   = 000020            362#   604    628    636
B05   = 000040            361#   604    636   2059
B06   = 000100            360#   464    537    864    886    997  1379  1525  1549  1571  1627  1642
B07   = 000200            359#  1698
B08   - 000400            358#
B09   - 001000            357#  1447   1622   1625
B10   = 002000            356#  1439   1618
B11   - 004000            355#  1433   1614
B12     010000            354#  1415   1599
B13   = 020000            353#  1411   1540   1595
B14   - 040000            352#  1407   1591   1624   1644
B15   - 100000            351#   463    701    862    881    912   943   956   990  1186  1190  1223  1281
                         1360   1364   1396   1437   1441   1451  1458  1483  1509  1524  1585  1626  1631
                         1640   1680   1689   1771   1830   1831  1852  1903  2106  2116

CBFEND- 016412           2198   2330#
CBTA    032726           2737#  2858
CBUFPT  016412           2079*  2160*  2173   2178*  2198   2200* 2201* 2331#
CDDMG   033064           1747   1756   2827#
CHAR  = 000002           2189#  2192
CLEAV   004074            567    636#
CLKEXT  012316           1701   1705   1709   1712#
CLKINT  012224            454   1698#
CMCHTB  016044           2061   2268#
CMDBUF  016206           1110   2160   2173   2329#
CMDENT  017651            425*   492   1108*  2080*  2161*  2356#
CMDRET  006416           1115   1117#
CMDRTN  006374           1111   1114#
CMDTBL  030306           1110   2475#
CMMARK  033630           3096*  3103   3105   3111   3117#
CMMISC  015014           2063   2067#
CMRET   033620           3098   3102   3107   3111#
COMAND  033524           1110   3077#
COMENT  006310            494   1107#
CPADD   004524            763#  2475
```

```
CPALP   004540            768#    783
CPARTN  004622            785#    787
CPASS   005672           1020#   2478
CPCLR   004752            831#   2481
CPCLRC  005002            840#    847
CPCLRT  005044            851#    853
CPCNRT  005200            885     887#    889
CPCNT   005114            878#   2484
CPDEL   004632            791#   2487
CPDLP   004646            796#    816
CPDRTN  004742            818#    820
CPERR   006600           1174#   2490
CPERTN  006716           1185    1191#   1194
CPGO    005510            979#   2493
CPINIT  005210            902#   2496
CPINRT  005446            944#    946
CPMAST  004116            646#   2499
CPMLP   003506            580#    596
CPMOK   004132            648     651#
CPMRET  004200            650     660#
CPRANG  004366            720#   2502
CPRFIL  004470            744#    748
CPRIB   004274            691#   2505
CPROK   004310            693     696#
CPRRET  004364            695     708#
CPRRTN  004514            750#    753
CPSEC   004202            666#   2508
CPSEX   004064            630#    633     639
CPSILO  003406            562#   2511
CPSLV   004022            618     624#
CPSOK   004216            668     671#
CPSRET  004266            670     681#    685
CPSRTN  005104            865#    867
CPSTAT  005054            859#   2514
CPSUM   005456            953#   2517
CPSURT  005500            958#    960
CRRET   015074           2078    2083#
CR.     = 000015          335#   2079    2271
CTLCMG  C16422           2125    2335#
CTLOMG  016414           2093    2333#
CTLJMG  016417           2134    2334#
CTL.C   = 000003          338#   2269
CTL.O   = 000017          336#   2272
CTL.J   = 000021          339#   2273
CTL.S   = 000023          340#   2274
CTL.U   = 000025          337#   2275
CURAD   017762           1346*   1347    1474    1499    2399#
CYCL    007350           1299    1301#
DATBUF  020676           1269    1344    2432#
DATGEN  007124            503    1255#
DATGEV  017670            501     943*   1282*   1364*   2366#
DBCLP   011076           1557#   1559
DBFCLR  011062           1544    1554#
DDIV    033240           2837    2886#
DECJSP  032700           1850    190'    1953    2718#
DECPNT  032706           1781    1785    1789    1793    2724#
```

```
DEQ     033442            1345    2052    2218    2979#
DIV     033302            2760    2921#
DONE  = 000200             346#
DQEMP   033512            2986    2996#
DQNOWP  033504            2990    2992#
DQRET   033510            2994#   2998
DTCDON  007302            1280#
DTCLP   007242            1275#   1279
DTCNT   007156            1260    1262#
DTGNC   007174            1263    1265#
DTGNCO  007216            1266    1269#
DTSEED  017736             935*   1265    1275    1276*   1277*   1280    2389#
ELPSTM  031667            1795    2554#
ENQ     033364            1229    2156    2157    2158    2159    2179    2203    2230    2247    2259    2260    2261    2959#
ERADR = 012156            1397#   1409#   1413#   1421#   1425#   1429#   1431#   1435#   1445#   1449#   1471#   1490#   1514#
                          1584#   1593#   1597#   1605#   1610#   1612#   1616#   1620#   1667#  .1674#   1683#
ERMIN   014340            1991    1993#
ERMINR  014556            2029    2031#
ERMIS   014450            1992    2002#
ERMISR  014674            2030    2041#
ERNE    012156            1666    1683#
ERR   = 100000             344#
ERRA    007772            1408    1411#
ERRADR= 000004            1974#   1989    2012#   2027
ERRB    010026            1412    1415#
ERRB1   010106            1420    1423#
ERRB2   010142            1424    1427#
ERRB3   010176            1428    1431#
ERRC    010222            1416    1433#
ERRD    010270            1434    1439#
ERRE    010344            1440    1447#
ERRINT  007736            1383    1407#
ERRMOD  014244            1397    1409    1413    1421    1425    1429    1431    1435    1445    1449    1471    1490    1514
                          1974#
ERRMOR  014462            1584    1593    1597    1605    1610    1612    1616    1620    1667    1674    1683    2012#
ERRNUM= 000002            1974#   1982    2012#   2020
ERRX    010374            1410    1414    1422    1426    1430    1432    1438    1446    1448    1450#
ERR0    020000            1187*   1827    1843    1859*   1881    1894    1910*   2406#
ERR1    020002            1188*   1832*   1846    1854    1856*   1858*   1897    1905    1907*   1909*   2407#
ERR2    020004            1189*   1833*   1840    1849    1853*   1891    1900    1904*   2408#
ERTBCL  005302             916#    920
ERTBL   024736             915    1959    2442#   2452
ERTBLO= 024730            1988    2026    2452#
ERTCD   005324             917     921#
ERTMSG  031140             534    2542#
ERTRAP  003326             422     533#
ESCFLG  017766             465*    627*    637*    659*    680*    707*    749*    784*    817*    836*    905*    1109*   1114
                          1184*   2401#   3101*
EXCNT   031236             883    2544#
EXRST   031214             994    2543#
FIRST   017720             578*    597    2378#
FREELEM 030274            1345    1346    1356    2052    2053*   2054    2060*   2064    2067    2218    2219    2465#
GTLENS  033674            3165#   3191
HDRFLG  017726            1190*   1821    1823*   1831*   1875    1877*   2381#
HOURS   017656             911*    987*   1711*   1780    2361#
INPLIN= 000002            3077#   3097
```

```
INTENB- 000100  347#
INTTMP  030304  2229* 2230  2469#
ITEM  = 000002  2959# 2966  2979# 2987* 2997*
KEYWD   033634  653   673   698   3106  3152#
KWDERT  004270  654   674   684#  699
KWEXIT  034010  3199  3205#
KWFLG   017660  451*  453*  458*  884   995   1776  2147  2362#
KWTABL= 000004  3077# 3106
LCS     016160  452   886*  997*  1698* 2149* 2305#
LENINP- 000004  3146# 3160* 3190  3193
LF.   = 000012  334#  2270
LXERR   034444  3338  3358  3366  3380#  3384
LXINCN  034226  3318# 3343
LXIT    034442  3377# 3382
LXSCAN  034052  3097  3298#
MDNER   030570  1136  2534#
MESSAG- 000002  2243# 2244
MINUTS  017654  909*  985*  1707* 1708  1710* 1784  2359#
MLI     007324  1276  1292#  1670
MLSD    017740  936   2390#
MSCHNG  030642  1156  2535#
MSGLSD  017742  936*  1259* 1261* 1262  1270  1274  1280*  2391#
MSTMG1  031263  619   2545#
MSTMG2  031321  620   2546#
MSTMG3  031362  621   2547#
MSTMG4  031430  622   2548#
MSTMG5  031476  626   2549#
MTQMSG  031040  1236  2539#
MUL     007340  1297# 1302
N     - 000031  246#  1397# 1409# 1413# 1421# 1425# 1429# 1431# 1435# 1445# 1449# 1471# 1490#
                1514# 1584# 1593# 1597# 1605# 1610# 1612# 1616# 1620# 1667# 1674# 1683# 2443
NEXT    017722  579*  583   593*  597   2379#
NMINT   010762  1392  1523#
NOERMG  032215  1183  2566#
NONMG   032546  1836  1886  2579#
NOTHER  033776  3158  3171  3175  3194  3202#
NOWMST- 000004  1376# 1390  1523
NOFULL  033436  2965  2975#
NONOWP  033430  2969  2971#
NORET   033434  2973# 2976
NULPAR  033632  3109  3118#
NXTCH   033714  3173# 3184
NXTEL   034056  3301# 3319
NXTWD   033734  3177  3186#
NXT0    002772  490   492#
NXT1    003014  493   495#
NXT10   003256  520   522#
NXT11   003300  523   525#
NXT12   003322  526   528#
NXT2    003036  496   498#
NXT3    003060  499   501#
NXT4    003102  502   504#
NXT5    003124  505   507#
NXT6    003146  508   510#
NXT7    003170  511   513#
NXT8    003212  514   516#
```

```
NXT9    003234  517  519#
OBJCNT  017770  572* 586* 600* 601  606  609  614  615  2402#
OCTJSP  032664  480  1742 1844 1847 1895 1898 1937 1941 2706#
OCTPNT  032672  2712#
ODELIM  034370  3348 3350 3357#
ONXTDG  034334  3347# 3355
ORIGSD  017734  935  2388#
PADFLG  017730  568* 606  608* 624  2382#
PARCLR  005276  914# 992
PART1   005774  1026 1041#
PART2   006062  1042 1057#
PART3   006150  1058 1073#
PART4   006234  1074 1088#
PCCALL  032652  2615 2623#
PCLEX   002000  379  386# 1095
PCLEXM  030741  446  2537#
PCLGO   017710  426* 536* 881* 991* 1139* 1233 1337 1436* 2155* 2374#
PCLOOK  032636  2613# 2621
PCLOOP  002750  473  489# 528
PCLRCV= 164220  301# 2290 2291 2292 2293 2294 2295
PCLTXM= 164200  300# 2282 2283 2284 2285 2286 2287 2288 2289
PCQUIT  032660  2617 2628#
PCREST  002556  381  456  461#
PCRST   002174  420#
PDRET   015500  2172 2177 2180#
PLAC    034220  3314# 3373
PLCRLF  015756  2246 2251#
PLODEC  015722  2245# 2249
PL219   015762  2252 2259#
PMRET   015600  2191 2199 2204#
PM176   015530  2194 2197#
PM187   015534  2196 2198#
PNCRLF  015762  444  445  447  471  882  993  1135 1137 1155 1157 1837 1887 2082
                2092 2175 2258#
PNTLIN  015716  446  448  481  534  619  620  621  622  626  883  994  1112 1136
                1156 1183 1236 1761 1767 1770 1773 1775 1795 1796 1824 1825 1836
                1851 1878 1879 1886 1902 1954 1958 2093 2125 2134 2243#
PRCLR   004772  837# 913
PRCR    015034  2076# 2270 2271
PRCTLC  015164  2124# 2269
PRCTLO  015076  1235 2089# 2272
PRCTLQ  015154  2115# 2273
PRCTLS  015144  2105# 2274
PRCTLU  015206  2133# 2275
PRDEL   015402  2170# 2276
PRESC   015224  472  1113 1116 1142 1237 1804 1889 1966 2126 2146# 2176
PRINIT  005226  449  906#
PRINTD  017724  1834 1838* 1852* 1884 1888* 1903* 2380#
PRMISC  015502  2067 2189#
PROCHR  032636  2062 2612#
PRST    002472  450#
PS      177776  330# 434* 461* 483* 538* 1094* 1380*
PUSHNO  034320  3339# 3359
QBACK = 000012  319# 433* 850* 989* 2091 2966* 2967* 2968 2970*
QBOT    000006  317# 2968 2989
QCHACT  006746  1214# 1219
```

```
QELEMS= 000000  314#
QFRONT= 000010  318#  433*  849*  988*  2091*  2987  2988*  2989  2991*
QLDEV   017666  466*  498  990*  1211*  1360*  2365#
QLOAD   007016  1216  1228#
QLOOK   006760  1215  1217#
QMTRN   007046  1221  1233#
QRTN    007002  1223#  1234  1238
QSIZE = 000002  315#  2964
QTOP  = 000004  316#  433  849  850  988  989  2970  2991
QUEUE = 000004  2959#  2963  2979#  2984
RADB    020110  839  927  1213  1797  2418#  2428
RADBO - 020074  743  779  807  1353  1480  1505  2428#
RADEND  020674  1732  2427#
RANK    017760  1277  1671  2398#
RANM    017756  1276  1670  2397#
RBCLMG  031646  1775  2553#
RBSTMG  031627  1773  2552#
RCBUF   022736  1546  1555  1669  2437#
RCDEV   016202  406  1055*  1092*  2325#
RCDOR = 000400  1566#  1575
RCELIN  032502  1902  2575#
RCER  = 100000  1565#  1572
RCERC   032536  1899  2578#
RCPRIO= 000005  305#  441
RCR     016126  407*  1545*  1548*  1549*  1571*  1624*  1626*  1627*  1630*  1640*  1642*  1644*  1678*
                1688*  2017  2290#
RCRCM = 000040  1568#  1581  1649  1651
RCRNO   032502  1893  2576#
RCSEED  017744  1637*  1638  1652  1670*  1671*  1672  2392#
RCSUC = 000200  1567#  1578
RCTRN   032517  1896  2577#
RCTXPS  017746  942*  982*  1222*  1256*  1361  2393#
RCVEC   016204  418  1039*  1093*  2326#
RCVECT= 000174  302#
RCVEV   017712  463*  507  535*  1543*  1585*  1631*  1680*  1689*  2375#
RCVINA  011140  1573  1575#
RCVINB  011154  1576  1578#
RCVINC  011170  1579  1581#
RCVIND  011204  1582  1584#
RCVINT  011112  436  1570#
RCVVEC  016144  418*  436*  440  2298#
RCWD  = 020000  1540#  1548
RDBA    016136  415*  1546*  2294#
RDBC    016134  413*  1547*  1661  2293#
RDCRC   016140  417*  2295#
RDDB    016132  411*  1637  2292#
RDOINC  011714  1639  1644#
RDOINT  011654  1577  1637#
RDORTN  011700  1641#  1645
RECV    011012  509  1542#
REGRES  034034  539  1224  1398  1452  1460  1485  1511  1526  1560  1586  1628  1641  1679
                1690  2231  3205  3227#
REGSAV  034020  533  1210  1378  1554  1570  2228  3153  3213#
REQINP  017650  424*  2077  2081*  2150*  2171  2190  2355#
REREV   017664  525  941*  1830*  1883*  2154*  2364#
RERHDR  032411  1878  2573#
```

```
RERINT  011240  1574  1591#
REROS   013434   527  1871#
RERRA   011274  1592  1595#
RERRB   011330  1596  1599#
RERRB1  011410  1604  1607#
RERRB2  011452  1609  1612#
RERRC   011476  1600  1614#
RERRD   011532  1615  1618#
RERRE   011566  1619  1622#
RERRX   011634  1594  1598  1606  1611  1613  1617  1621  1623  1630#
RERTBL  026626  1833  2039  2460#
RERTN   013756  1874  1880  1890  1908  1911#
RERTRN  011626  1628#  1632
RESULT= 000026  3150#  3195*  3202*
RETURN= 000022  3147#
RJCTF   017750   467*  1257*  1267*  1497  2394#
RPMSG   016425  2337#
RRHLIN  032432  1879  2574#
RRJINT  012202  1583  1688#
RSFLSZ  012000  1658  1660#
RSHOLD  017772  2403#
RSNTO   011652  1654  1656#
RSPC    017764  1417*  1418*  1419  1423*  1427  1601*  1602*  1603  1607*  1608  2400#
RSR     016130   409*  1572  1575  1578  1581  1591  1595  1599  1601  1614  1618  1622  1625*
                1649  1651*  2291#
RSTMSG  031011   448  2538#
RSUCHK  012130  1673  1676#
RSUCNT  011744  1650  1652#
RSUGEN  012050  1670#  1677
RSUINT  011726  1580  1649#
RSULV   012134  1662  1668  1675  1678#  1684
RSUTST  012044  1665  1669#
RTRYA   002634   471#  484  541
RUBOUT= 000177   341#  2276
SAVRG   033340  2738  2828  2942#
SCCHAR  034376  3307  3361#
SCEOL   034436  3308  3375#
SCNO    034232  3306  3327#
SCTBL   030542   653  673  698  2523#
SCWORD  034172  3305  3310#
SECNDS  017653   908*  984*  1703*  1704  1706*  1788  2358#
SERFL   006660  1180  1186#
SETCN   032714  2708  2714  2720  2726  2728#
SHBRT   010556  1472  1483#  1491
SILLD   003662   607  609#  616
SJT     034162  3303  3305#
SLOLP   003672   611#  613
SMADV   014210  1933  1962#
SMARXT  014222  1930  1965#
SMCHK   013774  1929#  1963
SMLIN   032557  1954  2580#
SMLIN1  032557  1934  2581#
SMLIN2  032601  1938  2582#
SMLIN3  032625  1942  2583#
SMRN    014124  1945  1953#
SMRTN   014242  1926  1956  1960  1967#
```

```
SMSPN   014130            1951    1954#
SMTTLE  032131            1958    2564#
SNTDIG  034275            3324    3326    3333#
SNXTDG  034236            3323#   3331
SOPCNT  010706            1508#   1515
SORINT  010644            1389    1497#
SORIP   010736            1498    1514#
SPCEV   017714             469*    510    1138*   1437*   2376#
SPEC    006420             512    1134#
SPERTN  006472            1139#
SR    = 177570             331#
SRC   = 000026            3149#   3155
STARXT  013052            1734    1803#
STATUS  012320             515    1726#
STBADV  013040            1737    1800#
STCHK   012340            1732#
STHDR   012550            1731    1765#
STITLE  031751            1796    2556#
STKTOP= 002000             380     384#    387     421     482     540
STLIN   032043            1761    2558#
STLIN1  032043            1738    2559#
STLIN2  032065            1744    1748    2560#
STLIN3  032113            1753    1757    2561#
STPNTR  017700             863*   1730    1732    1735    1739*   1740    1743*   1745    1752*   1754    1762*   1797*   1800*
                          2370#
STRTN   013072            1729    1763    1798    1801    1805#
STRTRT  005662             996     998#   1001
STSEV   017676             513     862*    939*   1140*   1803*   2151*   2369#
STYHR   012442            1748#   1751
STYON   012456            1749    1752#
STYONT  012520            1758    1761#
STYTHR  012504            1757#   1760
SUCCS   010632            1489    1492#
SUCIA   010600            1468    1488#
SUCINP  010522            1470    1474#
SUCINT  010446            1386    1466#
SUCINY  010462            1469#   1493
SUMEV   017672             516     938*    956*   1141*   1965*   2152*   2367#
SUMHDR  014160            1928    1958#
SUMPNT  017702             957*   1927    1929    1931    1955*   1959*   1962*   2371#
SUMRY   013760             518    1924#
SUMSV   017674             476*    477*    479    2368#
SYNRR   006720            1176    1193#
SYNRTM  004066             571     582     589     592     603     632#
SYNRTN  004516             723     726     730     733     735     737     752#
SYNTAX  030714            1112    2536#
SYNTCN  005202             880     888#
SYNTIN  005450             904     945#
SYNTRA  004624             769     771     773     786#
SYNTRC  005046             835     852#
SYNTRD  004744             797     799     801     819#
SYNTRS  005106             861     866#
SYNTSM  005502             955     959#
SYNTX   005664             981    1000#
SYNXR   006304            1025    1030    1035    1038    1046    1051    1054    1062    1067    1070    1078    1083    1097#
```

```
TABLAD= 000024  3148# 3163
 CBFPT  030300  2467#
TCBIN   030302  2468#
TCR     016106   389*  864*  907*  1340*  1356*  1357*  1379*  1441  1450*  1459*  1484*  1510*  1525*
                1979  2282#
TEMP    017774  1979*  1980*  1981*  1997*  1998*  1999*  2000*  2001*  2017*  2018*  2019*  2036*  2037*
                2038*  2039*  2040*  2404#
TEMP1   017776  1983*  1994*  1995*  1996*  1999  2021*  2032*  2033*  2034*  2035*  2038  2405#
TEREV   017662   522  940*  1186*  1829*  2153*  2363#
TERHDR  032251  1824  2567#
TEROS   013074   524  1817#
TERTBL  025160   922  1177  1189  2000  2457#
TERTN   013432  1820  1826  1839  1857  1860#
THFND   033754  3181  3193#
THLTFL  017732  2106*  2116*  2215  2227*  2383#
THUMST  031550  1767  2550#
THUSCN  031576  1770  2551#
TIAREA  016564  2349#
TICKS   017652   910*  986*  1699*  1700  1702*  1792  2357#
TIEND = 016634  2349#
TIQ     016550   433*  489  2052  2230  2349#
TISIZE= 000024   308#  2349
TMLIN1  031730  1779  2555#
TMPR    016120   399*  611*  2287#
TMPRH   016122   401*  604*  617  623*  628*  636*  655  658*  675  679*  1390  1523*  1765
                1768  2288#
TOAREA  016650  2349#
TOEND = 017650  2349#
TOQ     016634   433*  495  1727  1818  1872  1925  2090*  2091*  2156  2157  2158  2159  2179
                2203  2218  2247  2259  2260  2261  2349#
TORET   015650  2214  2217  2220#
TOSIZE= 000400   309#  2349
TRAP4   002670   462  476#
TRELIN  032345  1851  2569#
TRERC   032401  1848  2572#
TRHLIN  032275  1825  2568#
TRNKF   017752   468*  1258*  1264*  1268*  1467  2395#
TRPDMG  031074   481  2540#
TRP4AD  031127   478  2541#
TRRCN   032362  1845  2571#
TRRNO   032345  1842  2570#
TRYOCT  034332  3334  3336  3346#
TSBA    016116   397*  1344*  2286#
TSBC    016114   395*  1343*  2285#
TSCRC   016124   403*  2289#
TSDB    016112   393*  2284#
TSR     016110   391*  1381  1384  1387  1393  1407  1411  1415  1417  1433  1439  1443  1447
                1457*  1466*  1469  1473*  1488  1492*  1508*  2283#
TTDEV = 177560   297#  2300  2301  2302  2303
TTIINT  015052   437  2226#
TTINH   014760  2056  2058  2060#
TTINP   014706   491  2050#
TTOUT   015602   497  2212#
TTPRIO= 000004   299#  443
TTRBUF  016150  2229  2301#
TTRCSR  016146   464*  537*  2300#
```

```
TTRET   015032            2065    2068#
TTVCTR= 000060             298#   2304
TTVECT  016156             437*    442    2304#
TTXBUF  016154            2219*   2303#
TTXCSR  016152            2213    2302#
TXBSY = 000020            1373#   1393    1443    1457
TXDEV   016176             388    1087*   1090*   2323#
TXDGEN  007554            1362    1364#
TXER  = 100000            1372#   1381
TXMEV   017704             504     937*   1223*   1281*   1339*   1396*   1451*   1458*   1483*   1509*   2372#
TXMIT   007370             506    1336#
TXML    017754            1273*   1343    2396#
TXMST   007366             700     706*    912*   1334#   1357    1771
TXMSV   017706            2373#
TXMVEC  016142             404*    435*    438    2297#
TXOUT   007532            1342    1360#
TXPRIO= 000005             304#    439
TXRTN   007530            1338    1358#   1363    1365
TXSOR = 000040            1375#   1387    1469    1473    1488    1492    1508
TXSUC = 000200            1374#   1384    1466
TXVEC   016200             404    1071*   1091*   2324#
TXVECT= 000170             303#
X     = 000040            2417#   2426#
XMTIB   007666            1391    1393#
XMTIP.T 007566             435    1378#
XMTIS   007622            1382    1384#
XMTPNM  007652            1388    1390#
XMTPRB  007702            1394    1396#
XMTSR   007636            1385    1387#
XSPCEV  017716             470*    519    1158*   1524*   2377#
XSPEC   006524             521    1154#
Y     = 000031            2441#   2449#
ZERO    016164            2261    2311#
ZX1   = 000000             562#    575#    646#    666#    691#    720#    763#    791#    831#    837#    859#    878#    902#
                           906#    914#    953#    979#   1020#   1107#   1134#   1154#   1174#   1209#   1255#   1336#   1542#
                          1726#   1817#   1871#   1924#   1974#   1995#   1996#   1998#   2012#   2034#   2035#   2037#   2050#
                          2076#   2089#   2105#   2115#   2124#   2133#   2146#   2170#   2189#   2212#   2243#   2258#   2959#
                          2979#   3077#   3327#   3351#
ZX2     000000             575#   1995#   1996#   1998#   2034#   2035#   2037#   3327#   3351#
$BLK  - 000000            2707#   2713#   2719#   2725#
$BLKS   002000            2831#
$SGN  - 000000            2707#   2713#   2719#   2725#
$SGNS   000000            2831#
$SUP  - 000000            2707#   2713#   2719#   2725#
$TR     002536             450     457#
.       034460             372#    376     378#    383#    384    1276    1397    1409    1413    1421    1425    1429    1431
                          1435    1445    1449    1471    1490    1514    1584    1593    1597    1605    1610    1612    1616
                          1620    1667    1670    1674    1683    2329#   2330    2349#   2360#   2413#   2427    2432#   2437#
                          2457#   2460#   2831
```

```
BDINIT  231#  907  1340  1450  1459  1484  1510  1545  1630  1678  1688
BREAK   3018#  3311  3364
CALC    131#  444  445  446  447  448  449  471  472  481  491  494  497  500  503
        506  509  512  515  518  521  524  527  534  619  620  621  622  626  882
        883  913  992  993  994  1110  1112  1113  1116  1135  1136  1137  1142  1155  1156
        1157  1183  1229  1235  1236  1237  1345  1397  1409  1413  1421  1425  1429  1431  1435
        1445  1449  1471  1490  1514  1584  1593  1597  1605  1610  1612  1616  1620  1667  1674
        1683  1761  1767  1770  1773  1775  1795  1796  1804  1824  1825  1836  1837  1951  1878
        1879  1886  1887  1889  1902  1954  1958  1966  2052  2064  2067  2082  2092  2093  2125
        2134  2156  2157  2158  2159  2175  2176  2179  2203  2218  2230  2247  2259  2260  2261
CBIAS   2805#  2831
ERROR   262#  1584  1593  1597  1605  1610  1612  1616  1620  1667  1674  1683
ERROT   250#  1397  1409  1413  1421  1425  1429  1431  1435  1445  1449  1471  1490  1514
HEDING  202#
KEYWD   46#  653  673  698  3106
LXSCAN  71#  3097
MULP    219#  1276  1670
MULT    153#  575  1995  1996  1998  2034  2035  2037  3327  3351
PROC    87#  562  646  666  691  720  763  791  831  837  859  878  902  906  914
        953  979  1020  1107  1134  1154  1174  1209  1255  1336  1542  1726  1817  1871  1924
        1974  2012  2050  2076  2089  2105  2115  2124  2133  2146  2170  2189  2212  2243  2258
        2959  2979  3077
REGRES  282#  539  1224  1398  1452  1460  1485  1511  1526  1560  1586  1628  1641  1679  1690
        2231  3205
REGSAV  276#  533  1210  1378  1554  1570  2228  3153
RETURN  112#  630  660  681  708  750  785  818  851  865  887  944  958  998  1098
        1117  1143  1159  1191  1225  1283  1358  1550  1805  1860  1911  1967  2006  2045  2068
        2083  2094  2107  2117  2162  2180  2204  2220  2262  2781  2867  2904  2937  2950  2973
        2994  3114
SPAN    3006#  3301
STCVT   2659#  2707  2713  2719  2725
SYNCLS  3039#  3302

. ABS.  034460  000

ERRORS DETECTED:  0

CZPLAC,CZPLAC/CR=CZPLAC
RUN-TIME:  31 43 3 SECONDS
RUN-TIME RATIO:  462/79-5.8
CORE USED:  10K  (19 PAGES)
```