PCL11-A,B

PCL11 EXCERCISER V020

CZPLAB0

AH-E260B-MC

FICHE 1 OF 1

DEC 1978

digital

MADE IN USA

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

PROGRAM CODE: AC-E259B-MC

PROGRAM NAME: CZPLAB0 PCL11 EXERCISER V-02A

VERSION: V-02

DATE CREATED: 21-SEP-76

UPDATED: 13-MAR-78

MODIFIED 11-SEP-78

MAINTAINER: SPECIAL SYSTEMS, KANATA

AUTHOR: DAVID G. WIENS

PARALLEL COMMUNICATIONS LINK (PCL11) EXERCISER

TABLE OF CONTENTS

TABLE OF CONTEN'S (CONTD)

PARALLEL COMMUNICATIONS LINK EXERCISER PROGRAM

1 GENERAL

1.1 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, EACH PCL11 TRANSMITTER INVOLVED WILL BEGIN SENDING RANDOM DATA PATTERNS TO EACH RECEIVER IT HAS 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 PON 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 DIFFERENCES IN EXERCISER V-02

1.2.1 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.

1.2.2 UPPER/LOWER CASE INPUT (SEE SEC.5.4)

EITHER UPPER OR LOWER CASE ALPHA CHARACTERS ARE ACCEPTED BY V-02 OF THE EXERCISER. WHICHEVER IS TYPED IN, THE UPPER CASE VERSION IS ECHOED.

1.2.3 ERRORS COMMAND (SEE SEC.3.2.12)

THE ''ERRORS'' COMMAND HAS BEEN ADDED TO THE EXERCISER TO GIVE MORE DETAIL ABOUT THE ERROR CONDITIONS.

1.2.4 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.

1.2.5 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.

1.2.6 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.

1.2.7 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.

1.2.8 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.

1.2.9 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.

1.2.10 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 $.

1.2.11 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.

1.2.12 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.

1.2.13 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.

1.2.14 DIFFERENCES IN V02A

THIS REVISION WAS BROUGHT ABOUT BY THE CHANGE OF THE PROMPT PRINTED WHILE IN COMMAND MODE. THE PROMPT WAS SIMPLY "$". IT HAS BEEN CHANGED TO "PCL>" TO REDUCE THE CONFUSION ON SYSTEMS WHERE PDP-11'S ARE ATTACHED TO VAX11780'S. (IT IS ONLY THE DIAGNOSTIC SUPERVISOR ON THE VAX11780 THAT IS ALLOWED TO USE "$".)

2 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
OF ATTEMPTS.

SEQ 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 YET **

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 MODE 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 ERRONEOUS COMMAND WILL BE EXECUTED UP TO THE POINT OF THE ERROR, THEN A SYNTAX ERROR MESSAGE WILL BE PRINTED.

WHEN CONTROL-U IS TYPED, THE 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 AND 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.

3.2 COMMANDS

THIS SECTION DEALS WITH THE FULL COMMANDS OF THE EXERCISER WHICH ALLOW 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

ALL 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 FOLLOWED 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

THE FOLLOWING WILL BE PRINTED:

SEQ 0015

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

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 IMMEDIATE INDICATION THAT THE 'MASTER SET'' COMMAND WAS OR WASN'T SUCCESSFUL. THE OPERATOR MUST ISSUE THE ''STATUS'' COMMAND TO DETERMINE IF A PARTICULAR UNIT IS MASTER.

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

[RI]B [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.

3.2.4 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 — SELECT TARGET RECEIVER ADDRESSES. 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

[CL]EAR

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 STRINC 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

[ST]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 YET"

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 ABLY TO RUN AND WILL APOLOGIZE THUS:

''* SORRY, I HAVE NO RECEIVER ADDRESSES *''

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 RECEIVER ADDRESSES (VIA RANGE OR ADD), THE RECEIVER ADDRESS QUEUE WOULD EVENTUALLY BECOME EMPTY AND WOULD NEVER BE RE-FILLED. THIS WOULD AGAIN CAUSE THE PRINTOUT:
''* SORRY, I HAVE NO RECEIVER ADDRESSES *''

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 BY 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 (CZPLBA0) 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 IS MORE THAN ONE PCL11 UNIT HOSTED BY ONE PDP-11 AND EACH ONE MUST BE TESTED ''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. (3.2.6 OR 3.2.5)

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).

SEQ 0026

ER

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 HAVIN 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 .

SEQ 0029

5.4 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.

6.4 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.

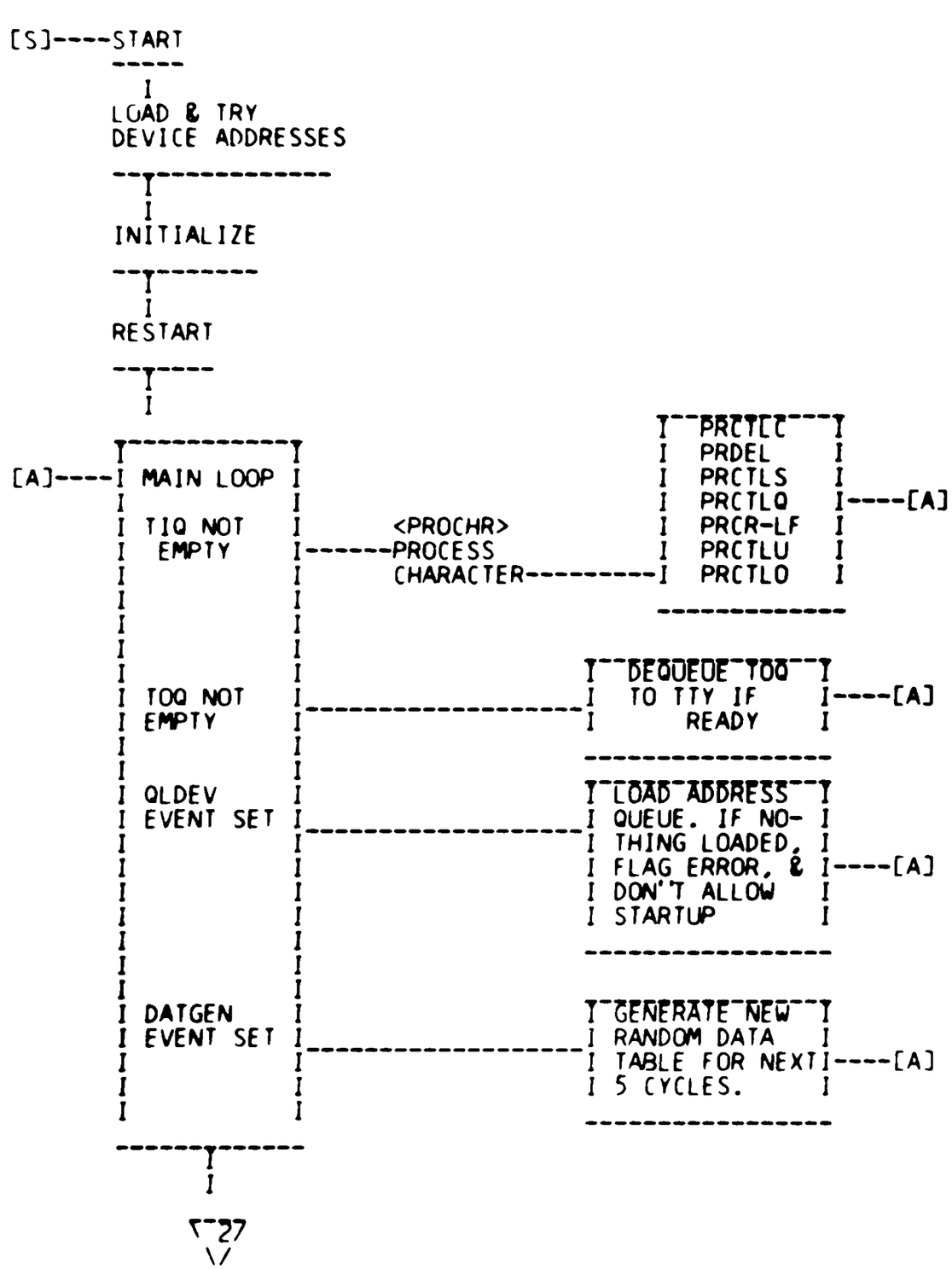
SEQ 0032

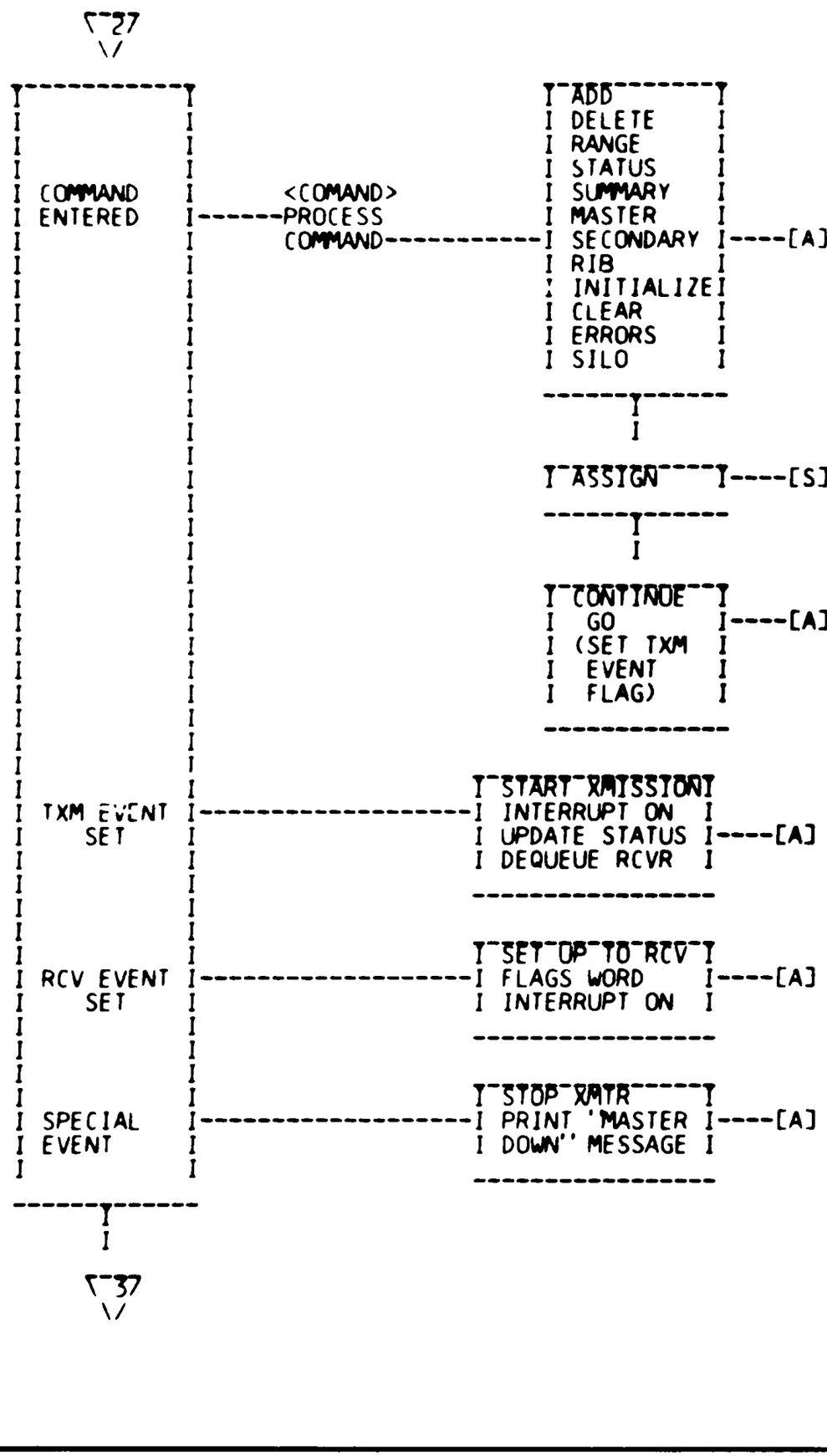
APPENDIX A, B, C, D

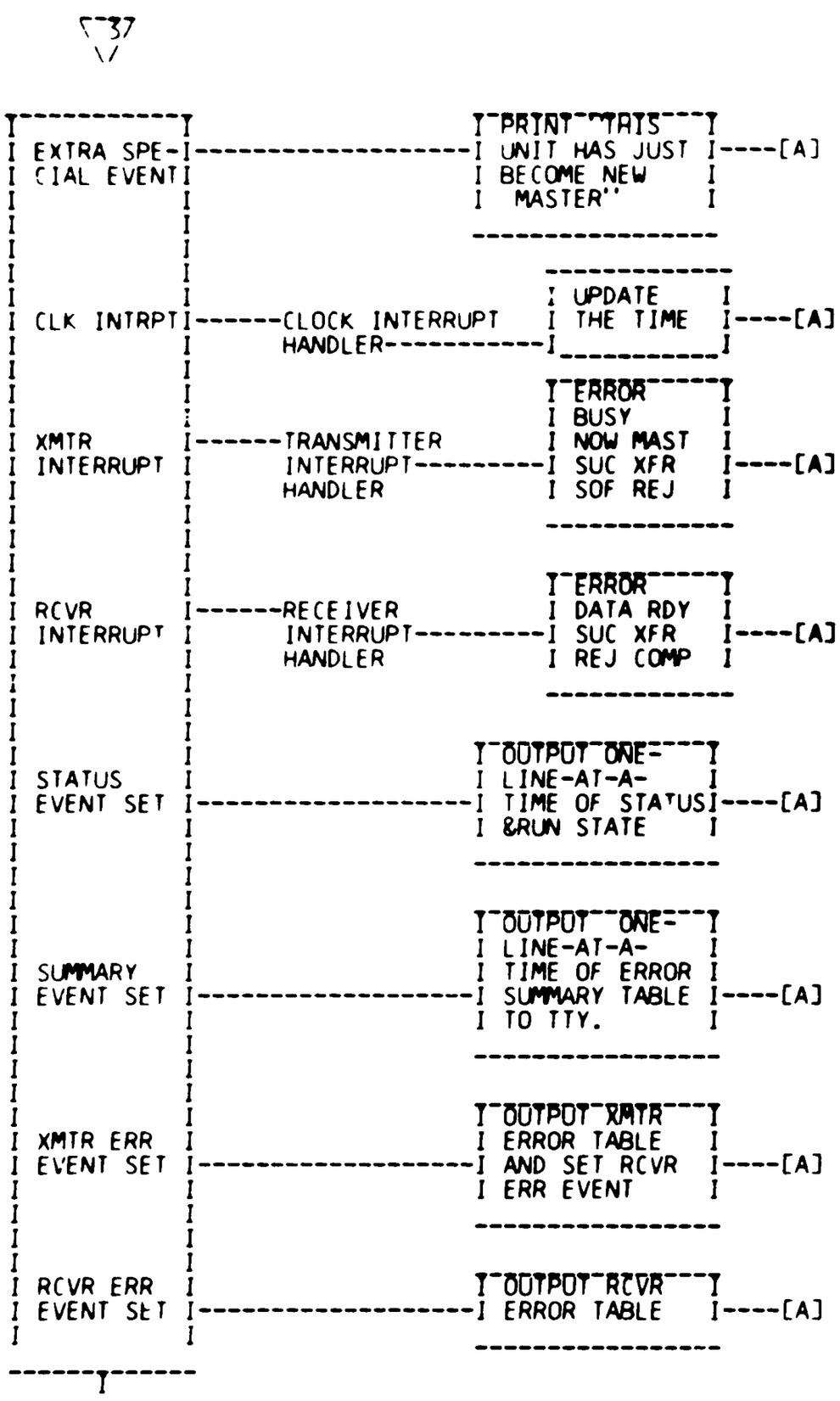
A-1

SEQ 0033

OVERALL FLOW




27
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]
TXM 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


37
EXTRA SPE-CIAL 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 0036

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

C-1

SEQ 0039

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 0040

STATISTICAL INFORMATION

STARTING ADDRESS	200
RESTARTING ADDRESS	204
SWITCH OPTIONS	NONE
PROGRAM SIZE	APPROX 7K
MEMORY OCCUPIED	
LOW BOUNDARY	00000
HIGH BOUNDARY	34474

LOCATIONS TO CHANGE FOR DIFFERENT DEVICE ADDRESSES

DEVICE	CHANGE LOCATION
KEYBOARD STATUS	16136
KEYBOARD DATA	16140
TTY STATUS	16142
TTY DATA	16144
KEYBOARD VECTOR	16146
LINE CLK STATUS	16150
XMTR PRIORITY	2334
RCVR PRIORITY	2346
KBD PRIORITY	2360

USEFUL LOCATIONS	ADDRESS
XMTR DATA BUFFER	20666
RCVR DATA BUFFER	22726
DATA SEED IN TRANSMITTER	17726
RANDOM MULTIPLIER	17746
RANDOM INCREMENT	17750

```
                        .TITLE  CZPLABO PCL11 EXERCISER V02
                        .IDENT  /0003/          ;DRCMAC.MAC  6-JAN-76


                    ; 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 'EYWORD
                    ;       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 SPECIFYI"G THE ADDRESS OF THE
                ; BEGINNING OF THE ASCII STRING TO BE CONVERTED.  THE STRING
                ; MLST 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-BIT.
             .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
```

```
281                                     JSR     R5,REGRES
282                                     .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                 AQSIZE  =       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.
```

```
340
341                             ;  DEVICE BIT DEFINITIONS.
342             100000          ERR     =       100000
343             004000          BUSY    =       4000
344             000200          DONE    =       200
345             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
                000000                  .       =       0
                000200                  .REPT   128.
                                        .WORD   .+2,0                   ;TRAP CATCHERS

                                        .ENDR

                000200                  .       =       200
000200  000200  000167  001574          JMP     PCLEX                   ;PROGRAM STARTS AT 200
000204  000204  012706  002000          MOV     #STKTOP,SP              ;AND RESTARTS AT 204
000210  000210  000167  002354          JMP     PCREST

                002000                  .       =       2000
                002000                  RELZRO  =       .
                002000                  STKTOP  =       .

002000  002000                          PCLEX:                          ;****START HERE*****
002000  002000  012706  002000          MOV     #STKTOP,SP              ;SET STACK POINTER TO TOP
002004  002004  016700  014156          MOV     TXDEV,R0                ;PREPARE TO GENERATE TXM ADDRS
002010  002010  010067  014062          MOV     R0,TCR                  ;GENERATE TCR ADDRESS
002014  002014  062700  000002          ADD     #2,R0
002020  002020  010067  014054          MOV     R0,TSR                  ;GENERATE TSR ADDRESS
002024  002024  062700  000002          ADD     #2,R0
002030  002030  010067  014046          MOV     R0,TSDB                 ;GENERATE TSDB ADDRESS
002034  002034  062700  000002          ADD     #2,R0
002040  002040  010067  014040          MOV     R0,TSBC                 ;GENERATE TSBC ADDRESS
002044  002044  062700  000002          ADD     #2,R0
002050  002050  010067  014032          MOV     R0,TSBA                 ;GENERATE TSBA ADDRESS
002054  002054  062700  000002          ADD     #2,R0
002060  002060  010067  014024          MOV     R0,TMMR                 ;GENERATE TMMR ADDRESS
002064  002064  005200                  INC     R0
002066  002066  010067  014020          MOV     R0,TMMRH                ;AND TMMR HIGH BYTE
002072  002072  005200                  INC     R0
002074  002074  010067  014014          MOV     R0,TSCRC                ;GENERATE TSCRC ADDRESS
002100  002100  016767  014064  014024  MOV     TXVEC,TXMVEC            ;GENERATE TXMVEC ADDRESS

002106  002106  016700  014060          MOV     RCDEV,R0                ;PREPARE TO GENERATE RCVR ADDRS
002112  002112  010067  014000          MOV     R0,RCR                  ;GENERATE RCR ADDRESS
002116  002116  062700  000002          ADD     #2,R0
002122  002122  010067  013772          MOV     R0,RSR                  ;GENERATE RSR ADDRESS
002126  002126  062700  000002          ADD     #2,R0
002132  002132  010067  013764          MOV     R0,RDDB                 ;GENERATE RDDB ADDRESS
002136  002136  062700  000002          ADD     #2,R0
002142  002142  010067  013756          MOV     R0,RDBC                 ;GENERATE RDBC ADDRESS
002146  002146  062700  000002          ADD     #2,R0
002152  002152  010067  013750          MOV     R0,RDBA                 ;GENERATE RDBA ADDRESS
002156  002156  062700  000004          ADD     #4,R0
002162  002162  010067  013742          MOV     R0,RDCRC                ;GENERATE RDCRC ADDRESS
002166  002166  016767  014002  013740  MOV     RCVEC,RCVVEC            ;GENERATE RCVVEC ADDR
```

```
 419  002174                          PCRST:                          ;*****RESTART HERE ******
 420  002174  012706  002000                  MOV     #STKTOP,SP      ;RESET STACK POINTER
 421  002200  012737  003340  000004          MOV     #ERTRAP,@#4     ;SET UP VECTOR FOR ADDRESS ERROR
 422  002206  012737  000340  000006          MOV     #340,@#6
 423  002214  105067  015420                  CLRB    REQINP          ;CLR INPUT REQUEST
 424  002220  105067  015415                  CLRB    CMDENT          ;CLR COMMAND ENTERED FLAG
 425  002224  005067  015450                  CLR     PCLGO           ;CLR XMTR GO FLAG
 426  002230  012767  000204  015524          MOV     #204,RSHOLD     ;SAVE RESTART ADDRESS
 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)  002236  005067  014276                  CLR     TIQ
(1)  002242  016767  014276  014300          MOV     TIQ+QTOP,TIQ+QFRONT
(1)  002250  016767  014270  014274          MOV     TIQ+QTOP,TIQ+QBACK
(1)  002256  005067  014342                  CLR     TOQ
(1)  002262  016767  014342  014344          MOV     TOQ+QTOP,TOQ+QFRONT
(1)  002270  016767  014334  014340          MOV     TOQ+QTOP,TOQ+QBACK
 434  002276  012737  000340  177776          MOV     #340,@#PS       ;DISABLE INTERRUPTS
 435  002304  012777  007600  013620          MOV     #XMTINT,@TXMVEC ;;;SET UP XMTR INTR VECTOR
 436  002312  012777  011124  013614          MOV     #RCVINT,@RCVVEC ;;;SET UP RCVR INTR VECTOR
 437  002320  012777  015642  013620          MOV     #TTIINT,@TTVECT ;;;SET UP TTY INTR VECTOR
 438  002326  016700  013600                  MOV     TXMVEC,R0
 439  002332  012760  000240  000002          MOV     #TXPRIO*32.,2(R0)       ;;;SET TXM PRIORITY
 440  002340  016700  013570                  MOV     RCVVEC,R0
 441  002344  012760  000240  000002          MOV     #RCPRIO*32.,2(R0)       ;;;SET RCVR PRIORITY
 442  002352  016700  013570                  MOV     TTVECT,R0
 443  002356  012760  000200  000002          MOV     #TTPRIO*32.,2(R0)       ;;;SET TTY PRIORITY
 444  002364                                  CALL    PNCRLF
 445  002400                                  CALL    PNCRLF
 446  002414                                  CALL    PNTLIN,<PCLEXM>         ;;;PRINT TITLE MESSAGE
 447  002432                                  CALL    PRINIT                  ;;;INITIALIZE TRANSMITTER
 448  002446  012700  031023          PRST:   MOV     #RSADD,R0               ;;;BUFFER ADDR FOR OCTPNT IN R1
 449  002452  016701  015304                  MOV     RSHOLD,R1               ;;;DATA FOR OCTPNT IN R0
 450  002456  012702  177777                  MOV     #-1,R2                  ;;;DON'T COMPRESS BLANKS
 451  002462  004767  030214                  JSR     PC,OCTJSP               ;;;COMPUTE RESTART ADDRESS
 452  002466                                  CALL    PNTLIN,<RSTMSG>         ;;;PRINT RESTART ADDRESS
 453  002504  012737  002550  000004          MOV     #$TR,@#4                ;;;SET UP TO TEST FOR CLOCK
 454  002512  005067  015132                  CLR     KWFLG                   ;;;CLEAR KW11 FLAG
 455  002516  005777  013426                  TST     @LCS                    ;;;ANY CLOCK?
 456  002522  012767  177777  015120          MOV     #-1,KWFLG               ;;;YES, SET KW11 FLAG
 457  002530  012737  012230  000100          MOV     #CLKINT,@#100           ;;;SET UP CLK VECTOR
 458  002536  012737  000340  000102          MOV     #340,@#102
 459  002544  000167  000020                  JMP     PCREST                  ;;;AND CONTINUE
 460  002550  022626                  $TR:    CMP     (SP)+,(SP)+            ;;;NO CLOCK, CLEAR STACK
 461  002552  005067  015072                  CLR     KWFLG                   ;;;CLR KW11 FLAG
 462  002556  012737  000102  000100          MOV     #102,@#100              ;;;SET UP TO TRAP HALT
 463  002564  005037  000102                  CLR     @#102
 464  002570  005037  177776          PCREST: CLR     @#PS                    ;;;ALLOW INTERRUPTS
 465  002574  012737  002702  000004          MOV     #TRAP4,@#4              ;CHANGE TRAP VECTOR FOR ERROR
 466  002602  052767  100000  015072          BIS     #B15,RCVEV              ;SET RCVR EVENT FLAG
 467  002610  052777  000100  013320          BIS     #B06,@TTRCSR            ;SET TTY KBD INTR ENAB
 468  002616  005067  015134                  CLR     ESCFLG                  ;CLEAR CNTRL-C FLAG
```

```
  469  002622  005067  015030                CLR     QLDEV                   ;CLR QUEUE LOAD EVENT FLAG
  470  002626  005067  015106                CLR     RJCTF                   ;CLEAR REJECT FLAG
  471  002632  005067  015104                CLR     TRNKF                   ;CLEAR TRUNCATE FLAG
  472  002636  005067  015042                CLR     SPCEV                   ;CLEAR MST DWN EVENT
  473  002642  005067  015040                CLR     XSPCEV                  ;CLR NOW MST EVENT
  474  002646                        RTRYA:  CALL    PNCRLF                  ;ENQUEUE CR, & LF
  475  002662                                CALL    PRESC                   ;ENTER COMMAND MODE!
  476  002676  000167  000060                JMP     PCLOOP                  ;GO TO MAIN LOOP
  477
  478
  479  002702  011667  014756        TRAP4:  MOV     (SP),SUMSV              ;SAVE TRAP ADDRESS
  480  002706  162767  000002  014750        SUB     #2,SUMSV                ;ALIGN IT -2
  481  002714  012700  031140                MOV     #TRP4AD,R0              ;SHOW OCT CONV RTN RIGHT ADDRESS
  482  002720  016701  014740                MOV     SUMSV,R1                ; FOR ASCII CHARS.
  483  002724  004767  027752                JSR     PC,OCTJSP
  484  002730                                CALL    PNTLIN,<TRPDMG>         ;PRINT TRAP MESSAGE
  485  002746  012706  002000                MOV     #STKTOP,SP
  486  002752  005037  177776                CLR     @#PS                    ;LOWER PRIORITY
  487  002756  000167  177664                JMP     RTRYA
```

```
                                        .SBTTL MAIN LOOP


002762  005767  013552          PCLOOP: TST     TIQ             ;IS TTY INPUT QUEUE EMPTY?
002766  001406                          BEQ     NXT0            ;YES, TEST ANOTHER FLAG
002770                                  CALL    TTINP           ;NO, PROCESS A CHARACTER
003004  105767  014631          NXT0:   TSTB    CMDENT          ;HAS A COMMAND BEEN ENTERED?
003010  100006                          BPL     NXT1            ;NO, TEST ANOTHER FLAG
003012                                  CALL    COMENT          ;YES, PROCESS COMMAND
003026  005767  013572          NXT1:   TST     TOQ             ;IS TTY OUTPUT QUEUE EMPTY?
003032  001406                          BEQ     NXT2            ;YES, TEST ANOTHER FLAG
003034                                  CALL    TTOUT           ;NO,OUTPUT A CHAR IF DEV RDY
003050  005767  014602          NXT2:   TST     QLDEV           ;IS ADDR QUEUE EMPTY?
003054  100006                          BPL     NXT3            ;NO, TEST ANOTHER FLAG
003056                                  CALL    ADQLD           ;YES, LOAD ADDR QUEUE
003072  005767  014562          NXT3:   TST     DATGEV          ;IS DATA GEN FLAG SET?
003076  100006                          BPL     NXT4            ;NO, TEST ANOTHER FLAG
003100                                  CALL    DATGEN          ;YES GENERATE NEW RANDOM DATA
003114  005767  014554          NXT4:   TST     TXMEV           ;IS XMTR EVENT FLAG SET?
003120  100006                          BPL     NXT5            ;NO,    ST ANOTHER FLAG
003122                                  CALL    TXMIT           ;YES, ENTER XMIT MODULE
003136  005767  014540          NXT5:   TST     RCVEV           ;IS RCVR EVENT FLAG SET?
003142  100006                          BPL     NXT6            ;NO, TEST ANOTHER FLAG
003144                                  CALL    RECV            ;YES, ENTER RECVR MODULE
003160  005767  014520          NXT6:   TST     SPCEV           ;IS SPECIAL EVENT FLAG SET?
003164  100006                          BPL     NXT7            ;NO, TEST ANOTHER FLAG
003166                                  CALL    SPEC            ;YES, HANDLE SPECIAL EVENT
003202  005767  014460          NXT7:   TST     STSEV           ;IS STATUS EVENT FLAG SET?
003206  100006                          BPL     NXT8            ;NO, TEST ANOTHER FLAG
003210                                  CALL    STATUS          ;YES,OUTPUT STATUS
003224  005767  014432          NXT8:   TST     SUMEV           ;IS SUMMARY EVENT FLAG SET?
003230  100006                          BPL     NXT9            ;NO, TEST ANOTHER FLAG
003232                                  CALL    SUMRY           ;YES, OUTPUT ERROR SUMMARY
003246  005767  014434          NXT9:   TST     XSPCEV          ;IS EXTRA-SPECIAL EVENT SET?
003252  100006                          BPL     NXT10
003254                                  CALL    XSPEC           ;YES, HANDLE NOW MASTER.
003270  005767  014356          NXT10:  TST     TEREV           ;IS XMTR ERROR EVENT SET?
003274  100006                          BPL     NXT11           ;NO, TEST ANOTHER FLAG
003276                                  CALL    TEROS           ;YES OUTPUT XMTR ERROR TABLE
003312  005767  014336          NXT11:  TST     REREV           ;IS RCVR ERROR EVENT SET?
003316  100006                          BPL     NXT12           ;NO,
003320                                  CALL    REROS           ;YES, OUTPUT RCVR ERROR TABLE
003334  000167  177422          NXT12:  JMP     PCLOOP          ;STAY IN MAIN LOOP

                                ;TRAP TO 4 HANDLER


003340                          ERTRAP: REGSAV                  ;;;SAVE R0...R5
003344                                  CALL    PNTLIN,<ERTMSG> ;;;PRINT TRAP MESSAGE
003362  005067  014314                  CLR     RCVEV           ;;;CLEAR RCVER EVENT FLAG
003366  005067  014306                  CLR     PCLGO           ;;;CLR PCL GO FLAG
003372  052777  000100  012536          BIS     #B06,@TTRCSR    ;;;SET TTY KBD INTR ENAB
003400  005037  177776                  CLR     @#PS            ;;;DROP CP PRIORITY
003404                                  REGRES
003410  012706  002000                  MOV     #STKTOP,SP      ;FIX STACK
003414  000167  177226                  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.)
003420                  PROC    CPSILO
(1)
(1)  003420             CPSILO:                         ;**ENTRY POINT**
003420  012701  017776          MOV     #ADSILO,R1      ;SET R1 TO POINT TO SILO BUFFER
003424  016602  000002          MOV     2(SP),R2
003430  005302                  DEC     R2              ;GET # OF ARGS. INTO R2
003432  001002                  BNE     11$             ;O.K IF THERE ARE SOME
003434  000167  000446          JMP     CLEAV           ;OTHERWISE, EXIT
003440  010267  014254  11$:    MOV     R2,PADFLG       ;SAVE COUNT FOR LATER USE.
003444  020227  000062          CMP     R2,#50.         ;ARE THERE MORE THAN 50 OBJECTS?
003450  101402                  BLOS    1$              ;NO, CONTINUE
003452  000167  000422          JMP     SYNRTM          ;YES, ERROR
003456  010267  014276  1$:     MOV     R2,OBJCNT       ;SAVE OBJECT COUNT
003462  010200                  MOV     R2,R0           ;TURN LOOK-UP AROUND
003464  005300                  DEC     R0
003466                          MULT    6,R0,R3
003476  060600                  ADD     SP,R0
003500  062700  000004          ADD     #4,R0
003504  016067  000004  014176  MOV     4(R0),FIRST     ;SAVE FIRST ITEM
003512  012767  000000  014172  MOV     #0,NEXT         ;SAVE CURRENT ITEM
003520  021027  000002  CPMLP:  CMP     (R0),#2         ;IS IT CLASS 2?
003524  001402                  BEQ     2$              ;YES, O.K.
003526  000167  000346          JMP     SYNRTM          ;NO, ERROR
003532  026067  000004  014152  2$:     CMP     4(R0),NEXT      ;IS THIS ITEM SAME AS LAST?
003540  001004                  BNE     3$              ;NO, O.K.
003542  112721  000000          MOVB    #0,(R1)+        ;YES, INSERT A PAD VALUE
003546  005267  014206          INC     OBJCNT          ;KEEP OBJECT COUNT UP TO DATE
003552  116011  000004  3$:     MOVB    4(R0),(R1)      ;PUT OBJECT INTO BUFFER
003556  001002                  BNE     4$              ;ERROR IF OBJECT IS 0
003560  000167  000374          JMP     SYNRTM
003564  122711  000037  4$:     CMPB    #37,(R1)        ;ERROR IF IT WAS > 37
003570  103002                  BHIS    5$
003572  000167  000302          JMP     SYNRTM
003576  112167  014110  5$:     MOVB    (R1)+,NEXT      ;SAVE REAL ITEM
003602  062700  177772          ADD     #-6,R0          ;SET UP TO GET NEXT OBJECT
003606  005302                  DEC     R2              ;ARE WE DONE LOADING BUFF?
003610  001343                  BNE     CPMLP           ;NO, KEEP GOING
```

```
  600  003612  026767  014074  014070          CMP     NEXT,FIRST              ;IS LAST OBJECT = FIRST?
  601  003620  001004                          BNE     6$                      ;NO, O.K.
  602  003622  112721  000000                  MOVB    #0,(R1)+                ;YES, INSERT A PAD VALUE
  603  003626  005267  014126                  INC     OBJCNT                  ;AND KEEP OBJECT COUNT UP TO DATE
  604  003632  026727  014122  000062  6$:     CMP     OBJCNT,#50.             ;OBJECT COUNT GOTTEN TOO BIG?
  605  003640  101402                          BLOS    7$                      ;NO, O.K.
  606  003642  000167  000232                  JMP     SYNRTM                  ;YES, ERROR
  607  003646  152777  000060  012236  7$:     BISB    #B05+B04,@TMMRH         ;PREPARE TO LOAD ADDR SILO
  608  003654  012700  000062                  MOV     #50.,R0                 ;HOLD SILO SIZE
  609  003660  026767  014074  014032          CMP     OBJCNT,PADFLG           ;IS OBJECT COUNT DIFFERENT THAN START?
  610  003666  001002                          BNE     SILLD                   ;YES, LEAVE SOMETHING IN PADFLG (SET)
  611  003670  005067  014024          8$:     CLR     PADFLG                  ;NO, CLEAR PAD FLAG
  612  003674  016702  014060          SILLD:  MOV     OBJCNT,R2               ;GET NO. OF OBJECTS
  613  003700  012701  017776                  MOV     #ADSILO,R1              ;GET  OBJECT BUFFER
  614  003704  112177  012200          SLOLP:  MOVB    (R1)+,@TMMR             ;GET AN OBJECT INTO SILO
  615  003710  005302                          DEC     R2                      ;LOADED ALL OBJECTS?
  616  003712  001374                          BNE     SLOLP                   ;NO, CONTINUE
  617  003714  166700  014040                  SUB     OBJCNT,R0               ;YES, IS THAT ALL THAT'LL FIT?
  618  003720  020067  014034                  CMP     R0,OBJCNT
  619  003724  002363                          BGE     SILLD                   ;IF NOT LOAD THEM AGAIN
  620  003726  132777  000001  012156          BITB    #1,@TMMRH               ;SEE IF I AM MASTER
  621  003734  001037                          BNE     CPSLV                   ;IF SO TURN ON SILO AND EXIT
  622  003736                                  CALL    PNTLIN,<MSTMG1>         ;PRINT 'THIS UNIT IS NOT MASTER
  623  003754                                  CALL    PNTLIN,<MSTMG2>         ; BUT HAS BEEN MADE SECONDARY
  624  003772                                  CALL    PNTLIN,<MSTMG3>         ; THE SILO YOU HAVE JUST FILLED
  625  004010                                  CALL    PNTLIN,<MSTMG4>         ; WILL BE USED IF YOU CLEAR THE
  626  004026  152777  000002  012056          BISB    #B01,@TMMRH             ; THE CURRENT MASTER''.
  627  004034  005767  013660          CPSLV:  TST     PADFLG                  ;HAS SILO BEEN PADDED?
  628  004040  001407                          BEQ     9$                      ;NO, CARRY ON.
  629  004042                                  CALL    PNTLIN,<MSTMG5>         ;YES, TELL OPERATOR.
  630  004060  012767  177777  013670  9$:     MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>''.
  631  004066  142777  000020  012016          BICB    #B04,@TMMRH             ;CLR AUTO ADDR
  632  004074  000241                          CLC                             ;CLEAR ''C'' FOR NON ERROR RETURN
  633  004076                          CPSEX:  RETURN
  634
  635  004100  000261                  SYNRTM: SEC                             ;SET ''C'' BIT FOR SYNTAX ERROR
  636  004102  000167  177770                  JMP     CPSEX                   ;AND RETURN
  637
  638
  639  004106  152777  000060  011776  CLEAV:  BISB    #B05+B04,@TMMRH         ;RESET AUTO ADDR & CLR SILO
  640  004114  012767  177777  013634          MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>''.
  641  004122  000241                          CLC                             ;CLEAR ''C'' IN CASE IT WAS SET
  642  004124  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

004130                                  PROC    CPMAST
(1)
(1)  004130                     CPMAST:                                 ;**ENTRY POINT**
004130  022766  000002  000002          CMP     #2,2(SP)                ;GET # OF ARGUMENTS
004136  001402                          BEQ     CPMOK                   ;OKAY IF 2
004140  000261                          SEC
004142  000423                          BR      CPMRET                  ;OTHERWISE, SYNTAX ERROR
004144  010603                  CPMOK:  MOV     SP,R3                   ;GET 2ND WORD OF COMMAND
004146  062703  000004                  ADD     #4,R3
004152                                  KEYWD   R3,#SCTBL               ;PROCESS IT
004164  103446                          BCS     KWDERT                  ;SYNTAX ERROR IF ''C'' SET
004166  117702  011720                  MOVB    @TMMRH,R2               ;GET OLD TMMR
004172  142702  000001                  BICB    #1,R2                   ;REMOVE OLD STATE OF MASTER
004176  052602                          BIS     (SP)+,R2                ;SET NEW STATE OF MASTER
004200  110277  011706                  MOVB    R2,@TMMRH               ;LOAD NEW TMMR
004204  012767  177777  013544          MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>''.
004212                          CPMRET: RETURN                          ;EXIT

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

004214                                  PROC    CPSEC
(1)
(1)  004214                     CPSEC:                                  ;**ENTRY POINT**
004214  022766  000002  000002          CMP     #2,2(SP)                ;ARE THERE 2 ARGUMENTS?
004222  001402                          BEQ     CPSOK                   ;IF YES, PROCEED
004224  000261                          SEC
004226  000424                          BR      CPSRET                  ;OTHERWISE, SYNTAX ERROR
004230  010603                  CPSOK:  MOV     SP,R3                   ;GET 2ND WORD OF COMMAND
004232  062703  000004                  ADD     #4,R3
004236                                  KEYWD   R3,#SCTBL               ;PROCESS IT
004250  103414                          BCS     KWDERT                  ;SYNTAX ERROR IF ''C'' SET
004252  117702  011634                  MOVB    @TMMRH,R2               ;GET OLD TMMR
004256  142702  000002                  BICB    #2,R2                   ;REMOVE OLD STATE OF SECONDARY
004262  006316                          ASL     (SP)
004264  052602                          BIS     (SP)+,R2                ;SET NEW STATE OF SECONDARY
004266  110277  011620                  MOVB    R2,@TMMRH               ;LOAD NEW TMMR
004272  012767  177777  013456          MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>''.
004300                          CPSRET: RETURN                          ;EXIT


004302  032600                  KWDERT: BIT     (SP)+,R0                ;POP BAD WORD OFF STACK
004304  000775                          BR      CPSRET                  ;EXIT


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

004306                                  PROC    CPRIB
(1)
```

```
   (1)  004306                          CPRIB:                                  ;**ENTRY POINT**
 695    004306  022766  000002  000002          CMP     #2,2(SP)                ;2 ARGUMENTS?
 696    004314  001402                          BEQ     CPROK                   ;YES, O.K.
 697    004316  000261                          SEC
 698    004320  000426                          BR      CPRRET                  ;OTHERWISE, ERROR RETURN
 699    004322  010603                  CPROK:  MOV     SP,R3                   ;GET 2ND WORD OF COMMAND
 700    004324  062703  000004                  ADD     #4,R3
 701    004330                                  KEYWD   R3,#SCTBL               ;PROCESS IT
 702    004342  103757                          BCS     KWDERT                  ;IF ERROR, SHOW IT.
 703    004344  016702  003030                  MOV     TXMST,R2                ;GET OLD STATE OF TCR
 704    004350  042702  100000                  BIC     #B15,R2                 ;CLEAR IMAGE OF RIB
 705    004354  000241                          CLC
 706    004356  006016                          ROR     (SP)                    ;GET NEW STATE OF RIB
 707    004360  006016                          ROR     (SP)
 708    004362  052602                          BIS     (SP)+,R2                ;SET IT IN IMAGE
 709    004364  010267  003010                  MOV     R2,TXMST                        ;LOAD NEW STATE OF TCR
 710    004370  012767  177777  013360          MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>"
 711    004376                  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.


004400                  PROC    CPRANG
(1)
(1)  004400     CPRANG: MOV     2(SP),R0        ;**ENTRY POINT**
004400  016600  000002          MOV     2(SP),R0        ;GET # OF OBJECTS
004404  022700  000003          CMP     #3,R0           ;3 OBJECTS?
004410  001047                  BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004412  016600  000004          MOV     4(SP),R0        ;GET CLASS OF FIRST OBJECT
004416  022700  000002          CMP     #2,R0           ;IS IT CLASS 2?
004422  001042                  BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004424  016601  000010          MOV     10(SP),R1       ;GET "TO" ARGUMENT INTO R1
004430  016600  000012          MOV     12(SP),R0       ;GET CLASS OF SECOND OBJECT
004434  022700  000002          CMP     #2,R0           ;IS IT CLASS 2
004440  001033                  BNE     SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004442  016602  000016          MOV     16(SP),R2       ;GET "FROM" ARGUMENT INTO R2
004446  020102                  CMP     R1,R2           ;IS "FROM" LOWER THAN "TO"?
004450  101427                  BLOS    SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
004452  022701  000037          CMP     #37,R1          ;IS "TO">37?
004456  103424                  BLO     SYNRTN          ;IF SO, INDICATE SYNTAX ERROR
004460  010200                  MOV     R2,R0           ;FIND "FROM" ENTRY
004462  001422                  BEQ     SYNRTN          ;IF "FROM"=0, SYNTAX ERROR
004464  006300                  ASL     R0
004466  006300                  ASL     R0
004470  010003                  MOV     R0,R3
004472  006300                  ASL     R0
004474  060300                  ADD     R3,R0
004476  062700  020064          ADD     #RADB0,R0       ;R0 CONTAINS "FROM" ADDRESS
004502  012710  177777  CPRFIL: MOV     #-1,(R0)        ;SET RCV ADDR ACTIVE FLAG.
004506  062700  000014          ADD     #14,R0          ;UPDATE TABLE POINTER
004512  005202                  INC     R2              ;INCREMENT "FROM" ADDRESS
004514  020201                  CMP     R2,R1           ;  UNTIL EQUAL TO "TO" ADDRESS
004516  003771                  BLE     CPRFIL          ;EXIT WHEN COMPLETE.
004520  012767  177777  013230  MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW "PCL>"
004526          CPRRTN: RETURN
                ;
004530  000261  SYNRTN: SEC                     ;SET "C" BIT TO INDICATE SYNTAX ERROR
004532  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.

004536                  PROC    CPADD
(1)
(1)  004536     CPADD:                          ;**ENTRY POINT**
004536  016602  000002          MOV     2(SP),R2        ;GET # OF ARGUMENTS INTO R2
004542  005302                  DEC     R2              ;IGNORE KEYWORD OBJECT
004544  010600                  MOV     SP,R0           ;GENERATE ADDRESS OF ARGUMENTS
004546  062700  000004          ADD     #4,R0           ;POINT R0 AT 1ST OBJECT CLASS
004552  021027  000002  CPALP:  CMP     (R0),#2         ;IS CLASS = 2?
004556  001027                  BNE     SYNTRA          ;NO, INDICATE SYNTAX ERROR
004560  016001  000004          MOV     4(R0),R1        ;GET OBJECT INTO R1
004564  001424                  BEQ     SYNTRA          ;IF IT'S 0, SYNTAX ERROR
004566  022701  000037          CMP     #37,R1
004572  103421                  BLO     SYNTRA          ;IF IT'S >37, SYNTAX ERROR
004574  006301                  ASL     R1              ;FIND THE TABLE ELEMENT
004576  006301                  ASL     R1
004600  010103                  MOV     R1,R3
004602  006301                  ASL     R1
004604  060301                  ADD     R3,R1
004606  062701  020064          ADD     #RADB0,R1
004612  012711  177777          MOV     #-1,(R1)        ;SET ACTIVE FLAG FOR THIS TABLE ENTRY
004616  062700  000006          ADD     #6,R0           ;SET UP FOR NEXT ARGUMENT
004622  005302                  DEC     R2              ;ARE WE DONE?
004624  001352                  BNE     CPALP           ;NO, CONTINUE.
004626  012767  177777  013122  MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>'
004634  CPARTN:  RETURN
004636  000261  SYNTRA:  SEC                     ;SET "C" BIT FOR SYNTAX ERROR
004640  000167  177770          JMP     CPARTN

                ;DELETE RECEIVER ADDRESSES AND ASSOCIATED "ATTEMPTS" AND "SUCCESSES".

004644                  PROC    CPDEL
(1)
(1)  004644     CPDEL:                          ;**ENTRY POINT**
004644  016602  000002          MOV     2(SP),R2        ;GET # OF ARGUMENTS INTO R2
004650  005302                  DEC     R2              ;IGNORE KEYWORD OBJECT
004652  010600                  MOV     SP,R0           ;GENERATE ADDRESS OF ARGUMENTS
004654  062700  000004          ADD     #4,R0           ;POINT R0 AT 1ST OBJECTS CLASS
004660  021027  000002  CPDLP:  CMP     (R0),#2         ;IS CLASS =2
004664  001034                  BNE     SYNTRD          ;NO,INDICATE SYNTAX ERROR
004666  016001  000004          MOV     4(R0),R1        ;GET OBJECT INTO R1
004672  001431                  BEQ     SYNTRD          ;IF IT'S 0, SYNTAX ERROR
004674  022701  000037          CMP     #37,R1
004700  103426                  BLO     SYNTRD          ;IF IT'S >37, SYNTAX ERROR
004702  006301                  ASL     R1              ;FIND TABLE ELEMENT
004704  006301                  ASL     R1
004706  010103                  MOV     R1,R3
004710  006301                  ASL     R1
004712  060301                  ADD     R3,R1
```

```
  810  004714  062701  020064                ADD     #RADB0,R1
  811  004720  005011                       CLR     (R1)                    ;CLEAR ACTIVE FLAG AT TABLE LOCATION
  812  004722  062701  000004                ADD     #4,R1                   ;AND OTHER ENTRIES...
  813  004726  005021                       CLR     (R1)+
  814  004730  005021                       CLR     (R1)+
  815  004732  005021                       CLR     (R1)+
  816  004734  005011                       CLR     (R1)
  817  004736  062700  000006                ADD     #6,R0                   ;SET UP FOR NEXT ARGUMENT
  818  004742  005302                       DEC     R2                      ;ARE WE DONE?
  819  004744  001345                       BNE     CPDLP                   ;NO, CONTINUE
  820  004746  012767  177777  013002        MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE NEW 'PCL>'
  821  004754                       CPDRTN: RETURN
  822  004756  000261                SYNTRD: SEC                     ;SET "C" BIT FOR SYNTAX ERROR
  823  004760  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
004764          PROC    CPCLR
(1)
(1)     004764  CPCLR:                                  ;**ENTRY POINT**

004764  016602  000002          MOV     2(SP),R2        ;GET NUMBER OF OBJECTS INTO R2
004770  022702  000001          CMP     #1,R2           ;ARE THERE ANY ARGUMENTS?
004774  001031                  BNE     SYNTRC          ;IF SO, INDICATE SYNTAX ERROR
004776  012767  177777  012752  MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>''
005004          PROC    PRCLR
(1)
(1)     005004  PRCLR:                                  ;**ENTRY POINT**
005004  012701  000037          MOV     #37,R1          ;SAVE COUNT OF TABLE ELEMENTS
005010  012700  020100          MOV     #RADB,R0        ;SET UP TO CLR ACTIVE FLAGS
005014  005010          CPCLRC: CLR     (R0)            ;CLEAR ALL ELEMENTS OF ENTRY
005016  062700  000004          ADD     #4,R0
005022  005020                  CLR     (R0)+
005024  005020                  CLR     (R0)+
005026  005020                  CLR     (R0)+
005030  005020                  CLR     (R0)+
005032  005301                  DEC     R1              ;DONE?
005034  001367                  BNE     CPCLRC          ;IF NOT, CONTINUE
005036  005067  011362          CLR     AOQ             ;INITIALIZE ADDR QUEUE TO EMPTY
005042  016767  011362  011364  MOV     AOQ+QTOP,AOQ+QFRONT
005050  016767  011354  011360  MOV     AOQ+QTOP,AOQ+QBACK
005056          CPCLRT: RETURN                          ;AND LEAVE
005060  000261          SYNTRC: SEC                     ;SET ''C'' BIT FOR SYNTAX ERROR
005062  000167  177770          JMP     CPCLRT



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

005066          PROC    CPSTAT
(1)
(1)     005066  CPSTAT:                                 ;**ENTRY POINT**
005066  022766  000001  000002  CMP     #1,2(SP)        ;ARE THERE ANY ARGUMENTS?
005074  001011                  BNE     SYNTRS          ;IF SO, INDICATE SYNTAX ERROR
005076  052767  100000  012562  BIS     #B15,STSEV      ;SET STATUS EVENT FLAG
005104  005067  012560          CLR     STPNTR          ;0 STPNTR INDICATES HEADER FIRST
005110  042777  000100  010760  BIC     #B06,@TCR       ;CLR TXM INTERRUPT ENABLE
005116          CPSRTN: RETURN
005120  000261          SYNTRS: SEC                     ;SET ''C'' BIT FOR SYNTAX ERROR
005122  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
                                        ;
                                        ;
                                        ;
005126                                          PROC    CPCNT
(1)
(1)  005126                             CPCNT:                          ;**ENTRY POINT**
005126  022766  000001  000002                  CMP     #1,2(SP)        ;ARE THERE ANY ARGUMENTS?
005134  001027                                  BNE     SYNTCN          ;IF SO, INDICATE SYNTAX ERROR
005136  012767  100000  012534                  MOV     #B15,PCLGO      ;SET PCL GO FLAG
005144                                          CALL    PNCRLF          ;PRINT CR & LF
005160                                          CALL    PNTLIN,<EXCNT>  ;PRINT 'EXERCISER CONTINUING''
005176  005767  012446                          TST     KWFLG           ;GOT A CLOCK?
005202  001403                                  BEQ     CPCNRT          ;NO
005204  012777  000100  010736                  MOV     #B06,@LCS       ;RE-ENABLE CLOCK.
005212                                  CPCNRT: RETURN
005214  000261                          SYNTCN: SEC                     ;SET ''C'' BIT FOR SYNTAX ERROR
005216  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
                        ;

005222                          PROC    CPINIT
(1)
(1)  005222             CPINIT:                         ;**ENTRY POINT**
005222  022766  000001  000002          CMP     #1,2(SP)        ;ARE THERE ANY ARGUMENTS?
005230  001114                          BNE     SYNTIN          ;IF SO, INDICATE SYNTAX ERROR
005232  012767  177777  012516          MOV     #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW 'PCL>''
005240                          PROC    PRINIT
(1)
(1)  005240             PRINIT:                         ;**ENTRY POINT**
005240                          BDINIT  XMTR            ;CLEAR XMITTER.
005246  105067  012371          CLRB    SECNDS          ;CLEAR SECONDS (TIME)
005252  105067  012366          CLRB    MINUTS          ;CLEAR MINUTES
005256  105067  012360          CLRB    TICKS           ;CLEAR TICKS
005262  005067  012360          CLR     HOURS           ;CLEAR HOURS
005266  042767  100000  002104  BIC     #B15,TXMST      ;CLEAR RIB IN COMMAND WORD FOR XMTR
005274                          CALL    PRCLR           ;CLR STATUS TABLE & ADDR QUEUE
005310                          PROC    PARCLR
(1)
(1)  005310             PARCLR:                         ;**ENTRY POINT**
005310  012700  024726          MOV     #ERTBL,R0       ;CLEAR ERROR SUMMARY TABLE
005314  022710  177777  ERTBCL: CMP     #-1,(R0)        ;IS ERR NUM = -1?
005320  001406                  BEQ     ERTCD           ;IF SO, DONE CLEARING
005322  005060  000004          CLR     4(R0)           ;OTHERWISE, CLR OCURRENCES
005326  062700  000006          ADD     #6,R0           ;STEP TO NEXT ENTRY
005332  000167  177756          JMP     ERTBCL          ;AND CONTINUE
005336  012700  001350  ERTCD:  MOV     #37*30,R0       ;ALSO CLEAR ERROR DATA
005342  012701  025150          MOV     #TERTBL,R1      ;FROM DETAILED ERR TABLE
005346  012721  000000  1$:     MOV     #0,(R1)+
005352  005300                  DEC     R0
005354  001374                  BNE     1$
005356  012700  000037          MOV     #37,R0          ;CLEAR ATTEMPS & SUCCESSES
005362  012701  020100          MOV     #RADB,R1        ; FROM STATUS TABLE
005366  062701  000004  2$:     ADD     #4,R1
005372  005021                  CLR     (R1)+
005374  005021                  CLR     (R1)+
005376  005021                  CLR     (R1)+
005400  005021                  CLR     (R1)+
005402  005300                  DEC     R0
005404  001370                  BNE     2$
005406  016767  012312  012312  MOV     ORIGSD,DTSEED   ;RESTORE ORIGINAL DATA SEED
005414  016767  012310  012310  MOV     MLSD,MSGLSD     ;AND MSG LENGTH SEED
005422  005067  012246          CLR     TXMEV           ;CLR XMIT EVENT FLAG
005426  005067  012230          CLR     SUMEV           ;CLR SUMMARY EVENT FLAG
005432  005067  012230          CLR     STSEV           ;CLR STATUS EVENT FLAG
005436  005067  012210          CLR     TEREV
```

```
                                        CLR     REREV                   ;CLR XMTR & RCVR ERROR EVENTS
                                        CLR     RCTXPS                  ;CLEAR DATA PASS NO.
                                        BIS     #B15,DATGEV             ;SET DATA GEN FLAG
                                CPINRT: RETURN
                                SYNTIN: SEC                             ;SET "C" BIT FOR SYNTAX ERROR
                                        JMP     CPINRT


                                ; SET THE SUMMARY EVENT FLAG SO THAT THE ERROR SUMMARY TABLE WILL BE PRINTED


                                        PROC    CPSUM
(1)
                                CPSUM:                                  ;**ENTRY POINT**
                                        CMP     #1,2(SP)                ;ARE THERE ANY ARGUMENTS?
                                        BNE     SYNTSM                  ;IF SO, INDICATE SYNTAX ERROR
                                        MOV     #B15,SUMEV              ;SET SUMMARY EVENT FLAG
                                        CLR     SUMPNT                  ;0 SUMPNT INDICATES HEADER FIRST
                                CPSURT: RETURN
                                SYNTSM: SEC                             ;SET "C" BIT FOR SYNTAX ERROR
                                        JMP     CPSURT


                                ; START THE EXERCISER
                                ;  CLEAR "ATTEMPTS" AND "SUCCESSES" IN STATUS TABLE
                                ;  CLEAR ERROR TABLES
                                ;  CAUSE RCVR ADDRESS QUEUE TO BE LOADED
                                ;  SET PCLGO
                                ;  IF THE RCVR ADDRESS QUEUE IS EMPTY AFTER BEING LOADED,
                                ;    INDICATE THIS BY PRINTING THE FOLLOWING MESSAGE:
                                ;
                                ;       "**SORRY, I HAVE NO RECEIVER ADDRESSES.**"
                                ;
                                ;  AND RETURN TO COMMAND INPUT MODE.


                                        PROC    CPGO
(1)
                                CPGO:                                   ;**ENTRY POINT**
                                        CMP     #1,2(SP)                ;ARE THERE ANY ARGUMENTS?
                                        BNE     SYNTX                   ;IF SO, INDICATE SYNTAX ERROR
                                        CLR     RCTXPS                  ;CLEAR PASS NO.
                                        CLR     AOQ                     ;INIT ADDR QUEUE TO EMPTY
                                        CLRB    SECNDS                  ;CLEAR SECONDS REG.
                                        CLRB    MINUTS                  ;CLEAR MINUTES REG.
                                        CLRB    TICKS                   ;CLEAR TICKS REG.
                                        CLR     HOURS                   ;CLEAR HOURS REG.
                                        MOV     AOQ+QTOP,AOQ+QFRONT
                                        MOV     AOQ+QTOP,AOQ+QBACK
                                        BIS     #B15,QLDEV              ;SET ADDR QUEUE LOAD EVENT
                                        MOV     #-1,PCLGO               ;SET PCL TO GO
                                        CALL    PARCLR                  ;CLEAR STATUS & ERRORS (NOT RCVRS)
```

```
  996  005626                                  CALL    PNCRLF                  ;PRINT CR, LF
  997  005642                                  CALL    PNTLIN,<EXRST>          ;PRINT 'EXERCISER STARTED'.
  998  005660  005767  011764                 TST     KWFLG                   ;GOT A CLOCK?
  999  005664  001403                         BEQ     STRTRT                  ;NO.
 1000  005666  012777  000100  010254         MOV     #B06,@LCS               ;ENABLE CLOCK INTR.
 1001  005674                          STRTRT: RETURN                          ;EXIT
 1002
 1003  005676  000261                  SYNTX:  SEC                             ;SET 'C'TO INDICATE SYNTAX ERROR
 1004  005700  000167  177770                 JMP     STRTRT                  ;AND EXIT
 1005
 1006
```

```
                                        .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.

005704                                  PROC    CPASS
 (1)
 (1)  005704                    CPASS:                                  ;**ENTRY POINT**
005704  010600                          MOV     SP,R0                   ;COPY LIST POINTER
005706  016001  000002                  MOV     2(R0),R1                ;GET NO. OF ARGUMENTS
005712  020127  000005                  CMP     R1,#5                   ;4 ARGUMENTS?
005716  101402                          BLOS    1$
005720  000167  000372                  JMP     SYNXR                   ;ERROR TOO MANY ARGUMENTS
005724  001030                  1$:     BNE     PART1                   ;LESS THAN 4, PARTIAL ASSIGNMENT
005726  016002  000004                  MOV     4(R0),R2                ;GET CLASS OF OBJECT
005732  022702  000002                  CMP     #2,R2                   ;IF IT'S A NUMBER, O.K.
005736  001402                          BEQ     2$
005740  000167  000352                  JMP     SYNXR                   ;ERROR WRONG CLASS
005744  062700  000006          2$:     ADD     #6,R0                   ;NOW GET ACTUAL NUMBER
005750  016002  000002                  MOV     2(R0),R2
005754  032702  000003                  BIT     #3,R2                   ;IS IT ON VECTOR BOUNDARY?
005760  001402                          BEQ     3$                      ;YES
005762  000167  000330                  JMP     SYNXR                   ;ERROR NOT ON VECTOR BOUNDARY
005766  020227  000774          3$:     CMP     R2,#774                 ;IS IT WITHIN VECTOR AREA?
005772  101402                          BLOS    4$                      ;YES
005774  000167  000316                  JMP     SYNXR                   ;ERROR, OUTSIDE VECTOR AREA
006000  010267  010170          4$:     MOV     R2,RCVEC                ;SAVE AS RCVR VECTOR
006004  005301                          DEC     R1
006006  020127  000004          PART1:  CMP     R1,#4                   ;ARE THERE ONLY 3 ARGS?
006012  001030                          BNE     PART2                   ;NO, MUST BE FEWER
006014  016002  000004                  MOV     4(R0),R2                ;GET CLASS OF OBJECT
006020  022702  000002                  CMP     #2,R2                   ;IF IT'S A NUMBER, O.K.
006024  001402                          BEQ     1$
006026  000167  000264                  JMP     SYNXR                   ;ERROR, OBJECT NOT A NUMBER
006032  062700  000006          1$:     ADD     #6,R0                   ;NOW, GET THE NUMBER
006036  016002  000002                  MOV     2(R0),R2
006042  032702  000017                  BIT     #17,R2                  ;CHECK IF ITS A VALID ADDR
006046  001402                          BEQ     2$                      ;IT IS. O.K.
006050  000167  000242                  JMP     SYNXR                   ;ERROR INVALID ADDRESS
006054  020227  164000          2$:     CMP     R2,#164000              ;IS IT IN THE ADDRESS FIELD?
006060  103002                          BHIS    3$                      ;YES, O.K.
006062  000167  000230                  JMP     SYNXR                   ;ERROR. ADDR = OUTSIDE DEVICE AREA
006066  010267  010100          3$:     MOV     R2,RCDEV                ;SAVE AS RCVR ADDRESS
006072  005301                          DEC     R1
006074  020127  000003          PART2:  CMP     R1,#3                   ;ARE THERE ONLY 2 ARGS?
006100  001030                          BNE     PART3                   ;NO, MUST BE ONLY 1.
```

```
1062  006102  016002  000004            MOV     4(R0),R2        ;GET CLASS OF OBJECT
1063  006106  022702  000002            CMP     #2,R2
1064  006112  001402                     BEQ     1$              ;IF IT'S A NUMBER, O.K.
1065  006114  000167  000176            JMP     SYNXR           ;ERROR, WRONG CLASS
1066  006120  062700  000006    1$:     ADD     #6,R0           ;NOW GET THE NUMBER
1067  006124  016002  000002            MOV     2(R0),R2
1068  006130  032702  000003            BIT     #3,R2           ;IS IT ON VECTOR BOUNDARY?
1069  006134  001402                     BEQ     2$              ;YES
1070  006136  000167  000154            JMP     SYNXR           ;ERROR. NOT ON VECTOR BOUNDARY
1071  006142  020227  000774    2$:     CMP     R2,#774         ;IS IT WITHIN VECTOR AREA?
1072  006146  101402                     BLOS    3$              ;YES
1073  006150  000167  000142            JMP     SYNXR           ;ERROR. OUTSIDE VECTOR AREA
1074  006154  010267  010010    3$:     MOV     R2,TXVEC        ;SAVE AS XMTR VECTOR
1075  006160  005301                     DEC     R1
1076  006162  020127  000002    PART3:  CMP     R1,#2           ;ONLY 1 ARGUMENT?
1077  006166  001027                     BNE     PART4           ;NO, MUST BE NONE.
1078  006170  016002  000004            MOV     4(R0),R2        ;GET CLASS OF OBJECT
1079  006174  022702  000002            CMP     #2,R2
1080  006200  001402                     BEQ     1$              ;IF IT'S A NUMBER, O.K.
1081  006202  000167  000110            JMP     SYNXR           ;ERROR. WRONG CLASS
1082  006206  062700  000006    1$:     ADD     #6,R0           ;NOW GET THE NUMBER
1083  006212  016002  000002            MOV     2(R0),R2
1084  006216  032702  000017            BIT     #17,R2          ;CHECK FOR VALID ADDRESS
1085  006222  001402                     BEQ     2$              ;O.K.
1086  006224  000167  000066            JMP     SYNXR
1087  006230  020227  164000    2$:     CMP     R2,#164000      ;IS IT IN ADDRESS FIELD?
1088  006234  103002                     BHIS    3$              ;YES
1089  006236  000167  000054            JMP     SYNXR           ;ERROR, OUTSIDE ADDRESS FIELD
1090  006242  010267  007720    3$:     MOV     R2,TXDEV        ;SAVE AS XMTR ADDRESS
1091  006246  020127  000001    PART4:  CMP     R1,#1           ;NO ARGUMENTS?
1092  006252  001014                     BNE     ASEXT           ;WE'RE DONE!
1093  006254  012767  164200  007704    MOV     #164200,TXDEV   ;NO ARGUMENTS, LOAD DEFAULTS.
1094  006262  012767  000170  007700    MOV     #170,TXVEC
1095  006270  012767  164220  007674    MOV     #164220,RCDEV
1096  006276  012767  000174  007670    MOV     #174,RCVEC
1097  006304  012737  000340  177776  ASEXT:  MOV     #340,@#PS       ;RAISE PROC. PRIORITY
1098  006312  000167  173462            JMP     PCLEX           ;GO START EXERCISER OVER.
1099
1100  006316  000261                  SYNXR:  SEC                     ;SET ''C'' TO FLAG ERROR
1101  006320                                  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)

006322                                  PROC    COMENT
 (1)
 (1)  006322                    COMENT:                                 ;**ENTRY POINT**
006322  105067  011313                  CLRB    CMDENT                  ;CLEAR COMMAND ENTERED FLAG
006326  005067  011424                  CLR     ESCFLG                  ;CLR COMMAND INPUT MODE FLAG
006332                                  CALL    COMAND,<CMDBUF,CMDTBL>  ;CONVERT COMMAND
006352  103015                          BCC     CMDRTN                  ;IF C = 0, EXIT
006354                                  CALL    PNTLIN,<SYNTAX>         ;PRINT ''SYNTAX ERROR'' IF C SET
006372                                  CALL    PRESC                   ;GIVE ANOTHER 'PCL>''
006406  005767  011344          CMDRTN: TST     ESCFLG                  ;SHOULD WE FAKE CNTRL-C?
006412  001406                          BEQ     CMDRET                  ;NO, EXIT
006414                                  CALL    PRESC                   ;YES, GIVE A 'PCL>''
006430                          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''

006432                                  PROC    SPEC
 (1)
 (1)  006432                    SPEC:                                   ;**ENTRY POINT**
006432                                  CALL    PNCRLF                  ;ENQUEUE CR & LF
006446                                  CALL    PNTLIN,<MDNER>          ;ENQUEUE MASTER DOWN MESSAGE
006464                                  CALL    PNCRLF                  ;ANOTHER CR & LF
006500  005067  011200                  CLR     SPCEV                   ;CLR SPECIAL EVENT FLAG
006504  005067  011170          SPERTN: CLR     PCLGO                   ;DON'T ALLOW TXM EVENT
006510  005067  011152                  CLR     STSEV                   ;DISCONTINUE STATUS EVENT
006514  005067  011142                  CLR     SUMEV                   ;DISCONTINUE SUM EVENT
006520                                  CALL    PRESC                   ;FAKE CNTRL-C KEY
006534                                  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 **''

006536                                          PROC    XSPEC
 (1)
 (1)  006536                            XSPEC:                                  ;**ENTRY POINT**
006536                                          CALL    PNCRLF                  ;ENQUEUE CR & LF
006552                                          CALL    PNTLIN,<MSCHNG>         ;ENQUEUE NEW MASTER MESSAGE
006570                                          CALL    PNCRLF                  ;ANOTHER CR & LF
006604  005067  011076                          CLR     XSPCEV                  ;CLR XSPEC EVENT FLAG
006610                                          RETURN                          ;RETURN TO MAIN _OOP


                                        ;PROCESSOR FOR 'DETAILED' ERROR TABLE PRINTOUT COMMAND 'ERRORS''

                                        ;THIS  ROUTINE WILL INITIALIZE THE TEMPORARY LOCATIONS REQUIRED
                                        ;BY THE XMTR AND RCVR ERROR EVENTS. IT WILL THEN QUICKLY CHECK THE
                                        ;RCVR AND XMTR ERROR TABLES FOR ANY ENTRIES. IF NONE EXIST, THE MESSAGE
                                        ;       ''**NO ERRORS TO REPORT YET **
                                        ;WILL BE PRINTED ON THE CONSOLE AND THAT IS ALL.
                                        ; IF ANY ENTRIES WERE FOUND, THE XMTR ERROR EVENT FLAG 'TEREV'' WILL BE SET
                                        ; WHICH WILL CAUSE THE XMTR ERRORS AND THE RCVR ERRORS (IF ANY) TO
                                        ; BE PRINTED.


006612                                          PROC    CPERR
 (1)
 (1)  006612                            CPERR:                                  ;**ENTRY POINT**
006612  022766  000001  000002                  CMP     #1,2(SP)                ;ANY ARGUMENTS?
006620  001044                                  BNE     SYNRR                   ;ERROR IF THERE ARE.
006622  012700  025150                          MOV     #TERTBL,R0              ;POINT AT XMTR TABLE
006626  012701  001350                          MOV     #37*30,R1               ;SET TABLE LENGTH COUNTER
006632  005720                         1$:     TST     (R0)+                   ;ANY ENTRIES?
006634  001016                                  BNE     SERFL                   ;YES, CALL ERROR MODULE
006636  005301                                  DEC     R1                      ;NO, CHECK WHOLE TABLE
006640  001374                                  BNE     1$
006642                                          CALL    PNTLIN,<NOERMG>         ;PRINT 'NO ERRORS''
006660  012767  177777  011070                  MOV     #-1,ESCFLG              ;FLAG COMENT TO ISSUE 'PCL>''
006666  000167  000036                          JMP     CPERTN                  ;AND RETURN
006672  012767  100000  010752         SERFL:  MOV     #B15,TEREV              ;SET XMTR EVENT FLAG
006700  012767  000001  011062                  MOV     #1,ERR0                 ;INITIATE ERROR #
006706  012767  000001  011056                  MOV     #1,ERR1                 ;INITIATE RCVR ADDR
006714  012767  025150  011052                  MOV     #TERTBL,ERR2            ;INITIATE TABLE POINTER
006722  052767  100000  010766                  BIS     #B15,HDRFLG             ;SET HEADER FLAG
006730                                 CPERTN:  RETURN

006732  000261                         SYNRR:  SEC                             ;SET ''C'' BIT TO INDICATE SYNTAX ERR
006734  000167  177770                          JMP     CPERTN
```

```
1199                                    .SBTTL RECEIVER ADDRESS QUEUE LOADER ROUTINE
1200
1201                                    ; LOAD THE RECEIVER ADDRESS SOFTWARE QUEUE
1202                                    ;
1203                                    ; THE TRANSMITTER EVENT WILL PICK RCVR ADDRESSES FROM THE QUEUE
1204                                    ; UNTIL IT IS EMPTY.
1205                                    ;
1206                                    ; IF THE QUEUE IS LOADED BUT STILL REMAINS EMPTY, AND PCLGO IS SET,
1207                                    ; THIS MEANS THE OPERATOR IS TRYING TO RUN BUT HASN'T ENTERED
1208                                    ; ANY RECEIVER ADDRESSES (USING 'RANGE'' OR ''ADD'' ETC.)
1209                                    ;
1210
1211
1212  006740                                PROC    ADQLD
 (1)
 (1)  006740                            ADQLD:                          ;**ENTRY POINT**
1213  006740                                REGSAV                      ;SAVE R0...R5
1214  006744  005067  010706               CLR     QLDEV               ;CLR QUEUE LOAD EVENT FLAG
1215  006750  012702  000037               MOV     #37,R2              ;POSITION COUNTER
1216  006754  012700  020100               MOV     #RADB,R0            ;R0 POINTS AT TABLE
1217  006760  022710  177777       QCHACT: CMP     #-1,(R0)            ;IS ACTIVE FLAG SET IN TABLE?
1218  006764  001002                       BNE     QLOOK               ;NO, LOOK FOR NEXT ONE
1219  006766  004767  000036               JSR     PC,QLOAD            ;YES, LOAD ADDR INTO QUEUE
1220  006772  062700  000014       QLOOK:  ADD     #14,R0
1221  006776  005302                       DEC     R2                  ;ALL ENTRIES CHECKED?
1222  007000  001367                       BNE     QCHACT
1223  007002  005767  007416               TST     ADQ                 ;IS THERE ANYTHING IN QUEUE?
1224  007006  001424                       BEQ     QMTRN               ;NO,THIS IS NOT A VALID PASS
1225  007010  005267  010722               INC     RCTXPS              ;YES, INCR PASS NO.
1226  007014  012767  100000  010652 QRTN: MOV     #B15,TXMEV          ;SET XMTR EVENT FLAG
1227  007022                                RECRES                      ;RESTORE R0...R5
1228  007026                                RETURN
1229
1230
1231  007030  116067  000002  021231 QLOAD: MOVB    2(R0),BKELEM+1      ;GET RCV ADDR INTO HIGH BYTE
1232  007036                                CALL    ENQ,<BKELEM,ADQ>    ;PUT IT IN ADDR QUEUE
1233  007056  000207                       RTS     PC
1234
1235
1236  007060  005767  010614       QMTRN: TST     PCLGO               ;ARE WE RUNNING?
1237  007064  001753                       BEQ     QRTN                ;NO, LEAVE
1238  007066                                CALL    PRCTLO              ;STOP PRINTING ANYTHING ELSE
1239  007102                                CALL    PNTLIN,<MTQMSG>     ;TELL OPERATOR NO RCVRS
1240  007120                                CALL    PRESC               ;FAKE CNTRL-C
1241  007134  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 > 600, 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.


007136                  PROC    DATGEN
(1)
(1)  007136     DATGEN:                                 ;**ENTRY POINT**
007136  005067 010574          CLR     RCTXPS          ;CLEAR DATA PASS FLAG
007142  005067 010572          CLR     RJCTF           ;CLEAR REJECT SOFTWARE FLAG
007146  005067 010570          CLR     TRNKF           ;CLR TRUNCATE SOFTWARE FLAG
007152  042767 177000 010552    BIC     #177000,MSGLSD  ;LENGTH NOT TO EXCEED 1000
007160  001003                 BNE     DTCNT           ;IF NON-0, OKAY, CONTINUE.
007162  012767 000001 010542    MOV     #1,MSGLSD       ;IF 0, MAKE IT AT LEAST 1
007170  026727 010536 000600 DTCNT: CMP  MSGLSD,#600    ;IS LENGTH LESS THAN 600?
007176  002403                 BLT     DTGNC
007200  012767 177777 010534    MOV     #-1,TRNKF       ;NO,SET TRUNCATE SOFTWARE FLAG
007206  022767 170000 010512 DTGNC: CMP  #170000,DTSEED ;IS DATA SEED -10000 OR MORE?
007214  101005                 BHI     DTGNCO
007216  012767 177777 010514    MOV     #-1,RJCTF       ;YES SET REJECT SOFTWARE FLAG
007224  005067 010512          CLR     TRNKF           ;AND CLR TRUNCATE SFTWR FLAG
007230  012700 020666       DTGNCO: MOV  #DATBUF,R0     ;POINT R0 TO TOP OF BUFFER
007234  016705 010472          MOV     MSGLSD,R5       ;R5 IS MESSAGE LENGTH
007240  006105                 ROL     R5              ;DOUBLE IT
007242  005405                 NEG     R5              ;NEGATE IT
007244  010567 010474          MOV     R5,TXML         ;SAVE IT FOR TRANSMIT MODULE
007250  016705 010456          MOV     MSGLSD,R5       ;R5 IS MSG LENGTH AGAIN
007254  016720 010446       DTCLP: MOV  DTSEED,(R0)+    ;PUT WORD INTO BUFFER
007260                       MULP    RANM,DTSEED     ;GENERATE NEW RANDOM NUMBER
007302  066767 010442 010416    ADD     INK,DTSEED
007310  005305                 DEC     R5
007312  003360                 BGT     DTCLP           ;IF R5 = 0, FINISHED
007314  016767 010406 010410 DTCDON: MOV  DTSEED,MSGLSD ;GET NEW LENGTH
007322  012767 100000 010344    MOV     #B15,TXMEV      ;SET XMIT EVENT FLAG
007330  005067 010324          CLR     DATGEV          ;CLEAR DAT GEN EVENT FLAG
007334                       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

007336  012601          MLI:    MOV     (SP)+,R1        ;GET MULTIPLICAND
007340  011603                  MOV     (SP),R3         ;GET MULTIPLIER
007342  010446                  MOV     R4,-(SP)        ;SAVE R4
007344  012704  000020          MOV     #16.,R4         ;SET UP FOR 16 BIT MULTIPLY
007350  005002                  CLR     R2
007352  006002          MUL:    ROR     R2              ;SHIFT PRODUCT
007354  006003                  ROR     R3
007356  103001                  BCC     CYCL            ;CHECK IF MULTIPLIER BIT IS 0
007360  060102                  ADD     R1,R2           ;ADD IN MULTIPLICAND
007362  005304          CYCL:   DEC     R4              ;COUNT LOOP
007364  003372                  BGT     MUL
007366  012604                  MOV     (SP)+,R4        ;RESTORE R4
007370  006202                  ASR     R2              ;ONE LAST SHIFT
007372  006003                  ROR     R3              ;PRODUCT IS IN R3
007374  010316                  MOV     R3,(SP)         ;STORE RESULT ON STACK
007376  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.


007400  060101          TXMST:  .WORD   060101                  ;TRANSMITTER COMMAND WORD

007402                  TXMIT:  PROC    TXMIT
(1)
(1)  007402             TXMIT:                                  ;**ENTRY POINT**
007402  005767  010272          TST     PCLGO                   ;CAN PCL GO?
007406  001455                  BEQ     TXRTN                   ;IF NOT, RETURN
007410  005067  010260          CLR     TXMEV                   ;CLR XMTR EVENT FLAG
007414                          BDINIT  XMTR                    ;CLR XMTR HARDWARE
007422  005767  006776          TST     AOQ                     ;ANY ELEMENTS IN ADDR QUEUE?
007426  001446                  BEQ     TXOUT                   ;IF NOT, FILL IT UP AGAIN
007430  016777  010310  006446  MOV     TXML,@TSBC              ;LOAD BYTE COUNT
007436  012777  020666  006442  MOV     #DATBUF,@TSBA           ;LOAD BUS ADDRESS
007444                          CALL    DEQ,<FRELEM,AOQ>        ;DEQUFUE RECEIVER ADDRESS
007464  116767  020575  010260  MOVB    FRELEM+1,CURAD          ;GET ADDRESS TABLE OFFSET
007472  016700  010254          MOV     CURAD,R0                ;FIND ADDRESS TABLE ENTRY
007476  006300                  ASL     R0                      ;       IN ORDER TO UPDATE
007500  006300                  ASL     R0                      ;       ''ATTEMPTS''
007502  010005                  MOV     R0,R5
007504  006300                  ASL     R0
007506  060500                  ADD     R5,R0
007510  062700  020064          ADD     #RADB0,R0
007514  062760  000001  000006  ADD     #1,6(R0)                ;UPDATE DOUBLE WORD
007522  005560  000004          ADC     4(R0)
007526  016777  020532  006342  MOV     FRELEM,@TCR             ;GIVE RECEIVER ADDR TO XMTR
007534  056777  177640  006334  BIS     TXMST,@TCR              ;START TRANSMISSION
007542                  TXRTN:  RETURN
007544  012767  100000  010104  TXOUT:  MOV     #B15,QLDEV      ;SET QUEUE LOAD EVENT FLAG
007552  026727  010160  000004  CMP     RCTXPS,#4               ;IS THIS THE 5TH PASS?
007560  001402                  BEQ     TXDGEN                  ;IF SO, SET DATA GEN EVENT FLAG
007562  000167  177754          JMP     TXRTN                   ;RETURN
007566  012767  100000  010064  TXDGEN: MOV     #B15,DATGEV     ;SET DAT GEN EVENT FLAG
007574  000167  177742          JMP     TXRTN                   ;RETURN
```

```
                                ;;;TRANSMITTER INTERRUPT HANDLER
                                ;;; SORT THE CAUSES OF TRANSMITTER INTERRUPTS
                                ;;; AND GO TO THE PROPER HANDLING ROUTINE.


            100000              TXER    -       100000
            000020              TXBSY   =       20
            000200              TXSUC   -       200
            000040              TXSOR   =       40
            000004              NOWMST  =       4

007600                          XMTINT: REGSAV                          ;;;SAVE R0...R5
007604  042777  000100  006264          BIC     #B06,@TCR               ;;;CLR TXM INTR ENABLE
007612  012737  000200  177776          MOV     #200,@#PS               ;;;ALLOW RCVR INTR BUT NOT TTY
007620  032777  100000  006252          BIT     #TXER,@TSR              ;;;IS THERE A HARDWARE ERROR?
007626  001402                          BEQ     XMTIS                   ;;;NO, CHECK SUC TXF
007630  000167  000114                  JMP     ERRINT                  ;;;HANDLE ERROR
007634  032777  000200  006236  XMTIS:  BIT     #TXSUC,@TSR             ;;;WAS IT A SUC TRANSFER?
007642  001402                          BEQ     XMTSR                   ;;;NO, CHECK SOFTWARE REJECT
007644  000167  000610                  JMP     SUCINT                  ;;;HANDLE SUC TXF
007650  032777  000040  006222  XMTSR:  BIT     #TXSOR,@TSR             ;;;WAS MESSAGE REJECTED?
007656  001402                          BEQ     XMTPNM                  ;;;NO,CHECK NOW MASTER
007660  000167  000772                  JMP     SORINT                  ;;;HANDLE REJECT
007664  132777  000004  006220  XMTPNM: BITB    #NOWMST,@TMMRH          ;;;IS THIS UNIT NOW MASTER?
007672  001402                          BEQ     XMTIB                   ;;;NO, WAS RECVR BUSY?
007674  000167  001074                  JMP     NMINT                   ;;;YES, HANDLE NOW MASTER INT
007700  032777  000020  006172  XMTIB:  BIT     #TXBSY,@TSR             ;;;WAS RECEIVER BUSY
007706  001402                          BEQ     XMTPRB                  ;;;NO, INTERRUPT WAS ERRONEOUS
007710  000167  000514                  JMP     BSYINT                  ;;;HANDLE BUSY
007714  012767  100000  007752  XMTPRB: MOV     #B15,TXMEV              ;;;SET XMTR EVENT FLAG
007722                                  ERROT   \N              ;;;ERROR:ERRONEOUS INTERRUPT FROM XMTR
 (1)
 (1)                                                    ;**** XMTR ERROR 1 *****
 (1)
007742                                  REGRES                          ;;;RESTORE R0...R5
007746  000002                          RTI                             ;;;RETURN
```

```
                                        ;XMTR ERROR INTERRUPT ROUTINE
                                        ;
                                        ; DETERMINE AND REPORT THE CAUSE OF THE TRANSMITTER ERROR.
                                        ;
007750  032777  040000  006122  ERRINT: BIT     #B14,@TSR              ;IS ERROR NON EXISTANT LOCATION?
007756  001412                         BEQ     ERRA
007760                               ERROT   \N              ;ERROR:NON-EXISTANT LOC ERROR IN XMTR
                                        ;
                                        ;**** XMTR ERROR 2 *****
                                        ;
010000  000167  000402                 JMP     ERRX
010004  032777  020000  006066  ERRA:   BIT     #B13,@TSR              ;IS ERROR MEM OFL?
010012  001412                         BEQ     ERRB
010014                               ERROT   \N              ;ERROR:MEM OFL ERROR IN XMTR
                                        ;
                                        ;**** XMTR ERROR 3 *****
                                        ;
010034  000167  000346                 JMP     ERRX
010040  032777  010000  006032  ERRB:   BIT     #B12,@TSR              ;IS ERROR TXM ERROR?
010046  001472                         BEQ     ERRC
010050  017767  006024  007676         MOV     @TSR,RSPC              ;SAVE RESPONSE CODES
010056  042767  177760  007670         BIC     #-20,RSPC              ;REMOVE GARBAGE
010064  022767  000015  007662         CMP     #15,RSPC               ;RSP CODES 11&01?
010072  001012                         BNE     ERRB1
010074                               ERROT   \N              ;ERROR:TXM ERROR. RCVR ACCEPTED A NULL
                                        ;
                                        ;**** XMTR ERROR 4 *****
                                        ;
010114  000167  000266                 JMP     ERRX
010120  042767  000003  007626  ERRB1:  BIC     #3,RSPC                ;DISCARD RSP A'S
010126  001012                         BNE     ERRB2                  ;RSB B - 00?
010130                               ERROT   \N              ;ERROR:TXM ERROR. RCVR HAS GONE OFF LINE
                                        ;
                                        ;**** XMTR ERROR 5 *****
                                        ;
010150  000167  000232                 JMP     ERRX
010154  022767  000010  007572  ERRB2:  CMP     #10,RSPC               ;RSP B = 10?
010162  001012                         BNE     ERRB3
010164                               ERROT   \N              ;ERROR:TXM ERROR. WORD OR CRC REJECTED
                                        ;
                                        ;**** XMTR ERROR 6 *****
                                        ;
010204  000167  000176                 JMP     ERRX
010210                       ERRB3:  ERROT   \N              ;ERROR:MISCELLANEOUS TXM ERROR
                                        ;
                                        ;**** XMTR ERROR 7 *****
                                        ;
010230  000167  000152                 JMP     ERRX
010234  032777  004000  005636  ERRC:   BIT     #B11,@TSR              ;MASTER DOWN?
010242  001417                         BEQ     ERRD
010244                               ERROT   \N              ;ERROR: M A S T E R  D O W N
                                        ;
                                        ;**** XMTR ERROR 10 *****
                                        ;
```

```
 1439  010264  005067  007410                CLR     PCLGO                   ;STOP TRANSMITTER
 1440  010270  012767  100000  007406        MOV     #B15,SPCEV              ;SET SPECIAL EVENT FLAG
 1441  010276  000167  000104                JMP     ERRX
 1442  010302  032777  002000  005570  ERRD: BIT     #B10,@TSR               ;TIMEOUT ERROR?
 1443  010310  001422                        BEQ     ERRE
 1444  010312  032777  100000  005556        BIT     #B15,@TCR               ;IS 'R.I.B.' SET?
 1445  010320  001404                        BEQ     1$                      ;NO MUST BE WEIRD ERROR.
 1446  010322  032777  000020  005550        BIT     #TXBSY,@TSR             ;YES, IS TDM BUS BUSY?
 1447  010330  001010                        BNE     2$                      ;YES, RECEIVER WASN'T THERE
 1448  010332                          1$:   ERROT   \N              ;ERROR: ERRONEOUS TIMEOUT IN TRANSMITTER
  (1)
  (1)                                                                   ;**** XMTR ERROR 11 *****
  (1)
 1449  010352  000167  000030          2$:   JMP     ERRX
 1450  010356  032777  001000  005514  ERRE: BIT     #B09,@TSR               ;TXM SILO OVERRUN?
 1451  010364  001410                        BEQ     ERRX
 1452  010366                                ERROT   \N              ;ERROR:SILO OVERRUN ERROR IN XMTR
  (1)
  (1)                                                                   ;**** XMTR ERROR 12 *****
  (1)
 1453  010406                          ERRX: BDINIT  XMTR                    ;CLEAR ALL XMTR HARDWARE
 1454  010414  012767  100000  007252        MOV     #B15,TXMEV              ;SET XMTR EVENT FLAG
 1455  010422                                REGRES                          ;RESTORE R0...R5
 1456  010426  000002                        RTI                             ;RETURN
```

```
                                        ;BUSY INTERRUPT ROUTINE
                                        ;
010430  042777  000020  005442  BSYINT: BIC     #TXBSY,@TSR             ;CLEAR TDM BUS BUSY
010436  012767  100000  007230          MOV     #B15,TXMEV              ;SET TXM EVENT FLAG
010444                                  BDINIT  XMTR                    ;CLEAR HARDWARE
010452                                  REGRES                          ;RESTORE R0...R5
010456  000002                          RTI                             ;RETURN
                                        ;
                                        ;SUCCESSFUL TRANSFER INTERRUPT ROUTINE
                                        ;
010460  042777  000200  005412  SUCINT: BIC     #TXSUC,@TSR             ;CLEAR SUC TXF
010466  005767  007250                  TST     TRNKF                   ;IS SFTWR TRUNCATE FLAG SET?
010472  001047                          BNE     SUCIA                   ;YES, GO CHECK SORE
010474  032777  000040  005376  SUCINY: BIT     #TXSOR,@TSR             ;IS SORE SET?
010502  001414                          BEQ     SUCINP                  ;NO, CONTINUE
010504                                  ERROT   \N              ;ERROR:MESSAGE TRUNCATED UNEXPECTEDLY
 (1)                                                            ;**** XMTR ERROR 13 *****
 (1)
 (1)
010524  000421                          BR      SHBRT                   ;DON'T INC SUC CONNS IF ERROR
010526  042777  000040  005344          BIC     #TXSOR,@TSR             ;CLR SORE
010534  016700  007212          SUCINP: MOV     CURAD,R0                ;PREPARE TO UPDATE STATUS TABLE
010540  006300                          ASL     R0                      ;GET TABLE ENTRY FOR CURRENT
010542  006300                          ASL     R0                      ;       RECEIVER ADDRESS IN
010544  010005                          MOV     R0,R5                   ;
010546  006300                          ASL     R0                      ;       ORDER TO UPDATE
010550  060500                          ADD     R5,R0                   ;
010552  062700  020064                  ADD     #RADB0,R0               ;       ''SUCCESSFUL'' ELEMENT
010556  062760  000001  000012          ADD     #1,12(R0)               ;UPDATE DOUBLE WORD
010564  005560  000010                  ADC     10(R0)
010570  012767  100000  007076  SHBRT: MOV     #B15,TXMEV              ;SET TRANSMIT EVENT FLAG
010576                                  BDINIT  XMTR                    ;CLEART HARDWARE
010604                                  REGRES                          ;RESTORE R0...R5
010610  000002                          RTI                             ;RETURN
                                        ;
010612  032777  000040  005260  SUCIA:  BIT     #TXSOR,@TSR             ;IS SORE SET? (SHOULD BE SET)
010620  001011                          BNE     SUCCS
010622                                  ERROT   \N              ;ERROR:MSG FAILED TO BE TRUNCATED
 (1)                                                            ;**** XMTR ERROR 14 *****
 (1)
 (1)
010642  000752                          BR      SHBRT                   ;DON'T INC SUC CONNS IF ERROR
010644  042777  000040  005226  SUCCS:  BIC     #TXSOR,@TSR             ;CLR SORE
010652  000*67  177616                  JMP     SUCINY                  ;PROCESS SUC TXF AS NORMAL
```

```
1498                                    ;SOFTWARE REJECT INTERRUPT ROUTINE
1499
1500  010656  005767  007056          SORINT: TST     RJCTF                   ;IS SFTWR REJECT FLAG SET?
1501  010662  001432                          BEQ     SORIP                   ;IF NOT, SOMETHING'S WRONG
1502  010664  016700  007062                  MOV     CURAD,R0                ;PREPARE TO UPDATE STATUS TABLE
1503  010670  006300                          ASL     R0                      ;GET TABLE ENTRY FOR CURRENT
1504  010672  006300                          ASL     R0                      ;       RECEIVER ADDRESS IN
1505  010674  010003                          MOV     R0,R3
1506  010676  006300                          ASL     R0                      ;       ORDER TO UPDATE
1507  010700  060300                          ADD     R3,R0
1508  010702  062700  020064                  ADD     #RADB0,R0               ;       ''SUCCESSFUL'' ELEMENT
1509  010706  062760  000001  000012          ADD     #1,12(R0)               ;UPDATE ENTRY
1510  010714  005560  000010                  ADC     10(R0)
1511  010720  042777  000040  005152  SORCNT: BIC     #TXSOR,@TSR             ;CLR SORE
1512  010726  012767  100000  006740          MOV     #B15,TXMEV              ;SET XMIT EVENT FLAG
1513  010734                                  BDINIT  XMTR                    ;CLEAR HARDWARE
1514  010742                                  REGRES                          ;RESTORE R0...R5
1515  010746  000002                          RTI                             ;RETURN
1516
1517  010750                          SORIP:  ERROT   \N              ;ERROR:ERRONEOUS REJECT BY RECEIVER
  (1)
  (1)                                                                   ;***** XMTR ERROR 15 *****
  (1)
1518  010770  000167  177724                  JMP     SORCNT                  ;RETURN
1519
1520
1521
1522                                    ;NOW MASTER INTERRUPT ROUTINE
1523                                    ;SET XSPEC EVENT FLAG (XSPEV)
1524                                    ;AND CLR NOW MST
1525
1526  010774  142777  000004  005110  NMINT:  BICB    #NOWMST,@TMMRH          ;CLEAR NOW MASTER
1527  011002  012767  100000  006676          MOV     #B15,XSPCEV             ;SET EXTRA-SPECIAL EVENT
1528  011010  052777  000100  005060          BIS     #B06,@TCR               ;RE-SET INTERRUPT ENABLE
1529  011016                                  REGRES                          ;RESTORE R0...R5
1530  011022  000002                          RTI                             ;RETURN
1531
1532
```

```
                                        .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

011024                                          .PROC   RECV
 (1)
 (1)  011024                            RECV:
1546  011024  005067  006652                    CLR     RCVEV           ;**ENTRY POINT**
1547  011030  004767  000040                    JSR     PC,DBFCLR       ;CLR RCVR EVENT FLAG
1548  011034                                    BDINIT  RCVR            ;CLEAR RCVR DATA BUFFER FIRST
1549  011042  012777  022726  005056           MOV     #RCBUF,@RDBA    ;CLR HARDWARE (REGARDLESS)
1550  011050  012777  176400  005046           MOV     #-1400,@RDBC    ;LOAD RCVR BUS ADDRESS
1551  011056  012777  020000  005032           MOV     #RCWD,@RCR      ;LOAD BYTE COUNT FOR 600 WORDS
1552  011064  052777  000100  005024           BIS     #B06,@RCR       ;SET RCV WORD IN RCVR
1553  011072                                    RETURN                  ;SET RCVR INTERRUPT ENABLE
1554
1555
1556
1557  011074                            DBFCLR: REGSAV                  ;SAVE R0...R5
1558  011100  012700  022726                    MOV     #RCBUF,R0       ;R0 POINTS AT BUFFER AREA
1559  011104  012701  000700                    MOV     #700,R1         ;R1 HOLDS BUFFER SIZE
1560  011110  005020                    DBCLP:  CLR     (R0)+           ;CLEAR BUFF LOC
1561  011112  005301                            DEC     R1              ;DONE ?
1562  011114  003375                            BGT     DBCLP           ;NO,LOOP
1563  011116                                    REGRES                  ;YES, RESTORE R0...R5
1564  011122  000207                            RTS     PC              ;AND RETURN
```

```
                                        ;;;RECEIVER INTERRUPT HANDLER.

                100000                  RCER    =       100000
                000400                  RCDOR   =       400
                000200                  RCSUC   =       200
                000040                  RCRCM   =       40

011124                          RCVINT: REGSAV                          ;;;SAVE R0...R5
011130  032777  100000  004762          BIT     #RCER,@RSR              ;;;IS THERE A HARDWARE ERROR?
011136  001402                          BEQ     RCVINA                  ;;;NO,CHECK DATA OUTPUT READY
011140  000167  000100                  JMP     RERINT                  ;;;HANDLE ERROR INTERRUPT
011144  032777  000400  004746  RCVINA: BIT     #RCDOR,@RSR             ;;;IS DATA OUTPUT READY?
011152  001402                          BEQ     RCVINB                  ;;;NO, CHECK SUC TXF
011154  000167  000472                  JMP     RDOINT                  ;;;HANDLE DATA OUTPUT RDY INTERRUPT
011160  032777  000200  004732  RCVINB: BIT     #RCSUC,@RSR             ;;;SUCCESSFUL TRANSFER SET?
011166  001402                          BEQ     RCVINC                  ;;;NO, CHECK RECECT COMPLETED
011170  000167  000536                  JMP     RSUINT                  ;;;HANDLE SUC TXF INTERRUPT
011174  032777  000040  004716  RCVINC: BIT     #RCRCM,@RSR             ;;;REJECT COMPLETED INTERRUPT?
011202  001402                          BEQ     RCVIND                  ;;;NO,ERRONEOUS INTERRUPT
011204  000167  000776                  JMP     RRJINT                  ;;;HANDLE REJ-COM INTERRUPT
011210                          RCVIND: ERROR   \N              ;;;ERROR:UNKNOWN RECEIVER INTERRUPT OCCURRED
 (1)
 (1)                                                                    ;**** RCVR ERROR 16 *****
 (1)
011230  012767  100000  006444          MOV     #B15,RCVEV              ;;;SET RCVR EVENT FLAG
011236                                  REGRES                          ;;;RESTORE R0...R5
011242  000002                          RTI                             ;RETURN
```

```
1591                                    ;RCVR ERROR INTERRUPT ROUTINE
1592
1593  011244  032777  040000  004646  RERINT: BIT     #B14,@RSR               ;IS ERROR NON EXST LOC ERR?
1594  011252  001412                          BEQ     RERRA
1595  011254                                  ERROR   \N              ;ERROR:NON EXST LOC ERR IN RCVR
 (1)
 (1)                                                                    ;**** RCVR ERROR 17 *****
 (1)
1596  011274  000167  000332                  JMP     RERRX
1597  011300  032777  020000  004612  RERRA:  BIT     #B13,@RSR               ;IS ERROR MEM OFLO?
1598  011306  001412                          BEQ     RERRB
1599  011310                                  ERROR   \N              ;ERROR:MEM OFLO ERROR IN RCVR
 (1)
 (1)                                                                    ;**** RCVR ERROR 20 *****
 (1)
1600  011330  000167  000276                  JMP     RERRX
1601  011334  032777  010000  004556  RERRB:  BIT     #B12,@RSR               ;ANY TRANSMISSION ERRORS?
1602  011342  001457                          BEQ     RERRC
1603  011344  017767  004550  006402          MOV     @RSR,RSPC               ;GET RESPONSE CODES
1604  011352  042767  177760  006374          BIC     #-20,RSPC               ;REMOVE GARBAGE
1605  011360  032767  000003  006366          BIT     #3,RSPC                 ;LOOK AT TXM RSP CODES
1606  011366  001012                          BNE     RERRB1
1607  011370                                  ERROR   \N              ;ERROR:TXM WENT OFF LINE
 (1)
 (1)                                                                    ;**** RCVR ERROR 21 *****
 (1)
1608  011410  000167  000216                  JMP     RERRX
1609  011414  042767  000003  006332  RERRB1: BIC     #3,RSPC                 ;REMOVE TXM RSP CODES NOW
1610  011422  022767  000010  006324          CMP     #10,RSPC                ;WAS THERE A CRC ERROR?
1611  011430  001012                          BNE     RERRB2
1612  011432                                  ERROR   \N              ;ERROR:RCVR CRC ERROR OCCURRED
 (1)
 (1)                                                                    ;**** RCVR ERROR 22 *****
 (1)
1613  011452  000167  000154                  JMP     RERRX
1614  011456                          RERRB2: ERROR   \N              ;ERROR:FIRST WORD RECVD NOT VALID
 (1)
 (1)                                                                    ;**** RCVR ERROR 23 *****
 (1)
1615  011476  000167  000130                  JMP     RERRX
1616  011502  032777  004000  004410  RERRC:  BIT     #B11,@RSR               ;WAS ERROR A PARITY ERROR?
1617  011510  001412                          BEQ     RERRD
1618  011512                                  ERROR   \N              ;ERROR:RCVR DETECTED INVALID PARITY
 (1)
 (1)                                                                    ;**** RCVR ERROR 24 *****
 (1)
1619  011532  000167  000074                  JMP     RERRX
1620  011536  032777  002000  004354  RERRD:  BIT     #B10,@RSR               ;DID RCVR TIME OUT?
1621  011544  001412                          BEQ     RERRE
1622  011546                                  ERROR   \N              ;ERROR:RCVR TIMEOUT ERROR OCCURRED
 (1)
 (1)                                                                    ;**** RCVR ERROR 25 *****
 (1)
1623  011566  000167  000040                  JMP     RERRX
1624  011572  032777  001000  004320  RERRE:  BIT     #B09,@RSR               ;WAS IT BYTE COUNT OVERFLOW?
1625  011600  001414                          BEQ     RERRX
```

```
 1626  011602  052777  100000  004306          BIS     #B15,@RCR               ;IMMEDIATLY SET REJECT IN RCVR
 1627  011610  042777  040000  004300          BIC     #B14,@RCR               ;CLEAR RCVR NPR BIT
 1628  011616  042777  001000  004274          BIC     #B09,@RSR               ;CLR BYTE COUNT OVERFLOW
 1629  011624                          RERTRN: REGRES                          ;RESTORE R0...R5
 1630  011630  000002                          RTI                             ;RETURN
 1631  011632                          RERRX:  BDINIT  RCVR                    ;CLEAR RECVR
 1632  011640  012767  100000  006034          MOV     #B15,RCVEV              ;SET RCV EVENT FLAG
 1633  011646  000167  177752                  JMP     RERTRN
 1634
```

```
                                        ;RCVR DATA OUTPUT RDY INTERRUPT ROUTINE
                                        
011652  042777  000100  004236  RDOINT: BIC     #B06,@RCR               ;CLEAR INTR ENAB
011660  017767  004236  006046          MOV     @RDDB,RCSEED            ;GET FLAG-WORD FOR DATA SEED
011666  022767  170000  006040          CMP     #170000,RCSEED          ;LOOK AT DATA SEED
011674  101011                          BHI     RDOINC                  ;IF MORE -VE THAN -10000
011676  052777  100000  004212          BIS     #B15,@RCR               ;       SET REJECT IN RCVR
011704                          RDORTN: REGRES                          ;RESTORE R0...R5
011710  052777  000100  004200          BIS     #B06,@RCR               ;RE-SET INTR ENAB
011716  000002                          RTI                             ;RETURN
011720  052777  040001  004170  RDOINC: BIS     #B14+B00,@RCR           ;START RECEIVER GOING
011726  000167  177752                  JMP     RDORTN
                                        
                                        ;RCVR SUC TXF INTERRUPT ROUTINE
                                        
011732  032777  000040  004160  RSUINT: BIT     #RCRCM,@RSR             ;WAS RECOM SET?
011740  001403                          BEQ     RSUCNT                  ;NO, CONTINUE
011742  042777  000040  004150          BIC     #RCRCM,@RSR             ;YES, CLR IT AND CONTINUE
011750  016701  005760          RSUCNT: MOV     RCSEED,R1               ;GET DATA SEED
011754  042701  177000                  BIC     #177000,R1              ;MASK MSG LENGTH
011760  001002                          BNE     RSNTO                   ;IF NOT 0, OKAY
011762  012701  000001                  MOV     #1,R1                   ;IF 0, MAKE IT 1
011766  162701  000001          RSNTO:  SUB     #1,R1                   ;ACCOUNT FOR FLAG WORD
011772  020127  000600                  CMP     R1,#600                 ;IS IT GREATER THAN 600?
011776  003402                          BLE     RSFLSZ                  ;NO, LEAVE IT ALONE
012000  012701  000600                  MOV     #600,R1                 ;YES, SET IT TO LIMIT (600)
012004  012700  001400          RSFLSZ: MOV     #1400,R0
012010  067700  004110                  ADD     @RDBC,R0                ;DETERMINE # OF BYTES RECEIVED
012014  001451                          BEQ     RSULV                   ;NOTHING TO CHECK, EXIT
012016  006200                          ASR     R0                      ;R0 = NO. OF WORDS RECEIVED
012020  020001                          CMP     R0,R1                   ;DID WE RECV ALL THE WORDS?
012022  001412                          BEQ     RSUTST                  ;YES, CARRY ON
012024  103456                          BLO     ERNE                    ;NOT ENOUGH !
012026                                  ERROR   \N              ;ERROR:RCVR GOT TOO MANY WORDS..
 (1)
 (1)                                                                    ;**** RCVR ERROR 26 *****
 (1)
012046  000434                          BR      RSULV                   ;EXIT IF ERROR
012050  012705  022726          RSUTST: MOV     #RCBUF,R5               ;R5 POINTS AT RECVD DATA
012054                          RSUGEN: MULP    RANM,RCSEED
012076  066767  005646  005630          ADD     RANK,RCSEED             ;GENERATE CHECK WORD
012104  026725  005624                  CMP     RCSEED,(R5)+            ;COMPARE IT WITH RECV'D WORD
012110  001411                          BEQ     RSUCHK
012112                                  ERROR   \N              ;ERROR:DATA WORD RECV'D WAS BAD
 (1)
 (1)                                                                    ;**** RCVR ERROR 27 *****
 (1)
012132  000402                          BR      RSULV                   ;EXIT IF ERROR
012134  005300                  RSUCHK: DEC     R0                      ;CHECKED ALL WORDS?
012136  001346                          BNE     RSUGEN                  ;NO,CONTINUE
012140                          RSULV:  BDINIT  RCVR                    ;CLR HARDWARE
012146                                  REGRES                          ;RESTORE R0...R5
012152  012767  100000  005522          MOV     #B15,RCVEV              ;SET RCVR EVENT FLAG
012160  000002                          RTI                             ;RETURN
                                        
012162                          ERNE:   ERROR   \N              ;ERROR: RCVR GOT TOO FEW WORDS ..
```

```
 (1)
 (1)                                                        ;**** RCVR ERROR 30 *****
 (1)
1686  012202  000167  177732          JMP     RSULV        ;EXIT IF ERROR
1687
1688                                ;RCVR REJECT INTERRUPT ROUTINE
1689
1690  012206                        RRJINT: BDINIT  RCVR     ;CLEAR HARDWARE
1691  012214  012767  100000  005460        MOV     #B15,RCVEV  ;SET RCVR EVENT FLAG
1692  012222                                REGRES           ;RESTORE R0...R5
1693  012226  000002                        RTI              ;RETURN
1694
1695                                ;KW11 CLOCK INTERRUPT ROUTINE
1696                                ;UPDATE TICKS. IF = 60. UPDATE SECONDS.
1697                                ;IF SECS = 60, UPDATE MINUTS. IF MINUTS - 60
1698                                ;UPDATE HOURS.
1699
1700  012230  042777  000200  003712  CLKINT: BIC     #B07,@LCS   ;JUST IN CASE IT MATTERS.
1701  012236  105267  005400                INCB    TICKS       ;UPDATE TICKS
1702  012242  126727  005374  000074        CMPB    TICKS,#60.  ;GET TO 60 YET?
1703  012250  103424                        BLO     CLKEXT      ;NO
1704  012252  105067  005364                CLRB    TICKS       ;YES, CLR TICKS
1705  012256  105267  005361                INCB    SECNDS      ;& UPDATE SECONDS
1706  012262  126727  005355  000074        CMPB    SECNDS,#60. ;GET 60 SECS?
1707  012270  103414                        BLO     CLKEXT      ;NO
1708  012272  105067  005345                CLRB    SECNDS      ;YES, CLR SECNDS
1709  012276  105267  005342                INCB    MINUTS      ;& UPDATE MINUTES
1710  012302  126727  005336  000074        CMPB    MINUTS,#60. ;GET 60 MINUTES?
1711  012310  103404                        BLO     CLKEXT      ;NO
1712  012312  105067  005326                CLRB    MINUTS      ;YES, CLEAR MINUTS
1713  012316  005267  005324                INC     HOURS       ;& UPDATE HOURS
1714  012322  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
                                        ;


012324                                          PROC    STATUS

012324                                  STATUS:                                 ;**ENTRY POINT**
012324  005767  004274                          TST     TOQ                     ;IS TTY OUT QUEUE EMPTY?
012330  001402                                  BEQ     60$
012332  000167  000540                          JMP     STRTN                   ;NO, RETURN TO WHEREVER
012336  005767  005326                  60$:    TST     STPNTR                  ;IS OUTPUT POINTER AT TABLE?
012342  001504                                  BEQ     STHDR                   ;NO, GO PRINT HEADER
012344  026727  005320  020664          STCHK:  CMP     STPNTR,#RADEND          ;IS POINTER AT END OF TABLE?
012352  002402                                  BLT     59$                     ;YES, DONE; EXIT
012354  000167  000476                          JMP     STARXT
012360  005777  005304                  59$:    TST     @STPNTR                 ;CHECK RCV ADDR ACTIVE FLAG
012364  001002                                  BNE     58$
012366  000167  000452                          JMP     STBADV                  ;ADVANCE TABLE IF FLAG CLR
012372  012700  032054                  58$:    MOV     #STLIN1,R0              ;OUTPUT POINTER FOR OCTJSP
012376  062767  000002  005264                  ADD     #2,STPNTR
012404  017701  005260                          MOV     @STPNTR,R1              ;OCTJSP DUMPS DATA IN STPNTR
012410  012702  177777                          MOV     #-1,R2                  ;DON'T COMPRESS BLANKS
012414  004767  020262                          JSR     PC,OCTJSP
012420  062767  000002  005242                  ADD     #2,STPNTR
012426  012700  032076                          MOV     #STLIN2,R0              ;OUTPUT POINTER FOR CDDMG
012432  016701  005232                          MOV     STPNTR,R1               ;OUTPUT WORD FOR CDDMG
012436  012702  177777                          MOV     #-1,R2                  ;DON'T COMPRESS BLANKS
012442  004767  020434                          JSR     PC,CDDMG
012446  020027  032123                  STYHR:  CMP     R0,#STLIN2+25           ;BLANK REST OF FIELD OUT
012452  103003                                  BHIS    STYON                   ;BY INSERTING SPACES
012454  112720  000040                          MOVB    #' ,(R0)+
012460  000772                                  BR      STYHR
012462  062767  000004  005200          STYON:  ADD     #4,STPNTR
012470  012700  032124                          MOV     #STLIN3,R0              ;OUTPUT POINTER FOR CDDMG
012474  016701  005170                          MOV     STPNTR,R1               ;OUTPUT WORD FOR CDDMG
012500  012702  177777                          MOV     #-1,R2                  ;DON'T COMPRESS BLANKS
012504  004767  020372                          JSR     PC,CDDMG
012510  020027  032140                  STYTHR: CMP     R0,#STLIN3+14           ;BLANK REST OF FIELD OUT
012514  103003                                  BHIS    STYONT                  ;BY INSERTING SPACES
012516  112720  000040                          MOVB    #' ,(R0)+
012522  000772                                  BR      STYTHR
012524                                  STYONT: CALL    PNTLIN,<STLIN>          ;ENQUEUE STATUS LINE FOR OUTPUT
012542  062767  000004  005120                  ADD     #4,STPNTR               ;UPDATE POINTER FOR NEXT LINE
012550  000167  000322                          JMP     STRTN                   ;RETURN

012554  132777  000001  003330          STHDR:  BITB    #1,@TMMRH               ;IS MASTER SET HERE?
012562  001407                                  BEQ     1$                      ;NO.
012564                                          CALL    PNTLIN,<THUMST>         ;YES, TELL OPERATOR
```

```
 1770  012602  132777  000002  003302  1$:     BITB    #2,@TMMRH               ;WELL, IS SECONDARY SET?
 1771  012610  001407                          BEQ     2$                      ;NO.
 1772  012612                                  CALL    PNTLIN,<THUSCN>         ;YES, TELL OPERATOR
 1773  012630  032767  100000  174542  2$:     BIT     #B15,TXMST              ;IS "RIB" SET?
 1774  012636  001410                          BEQ     3$                      ;NO, TELL OPERATOR
 1775  012640                                  CALL    PNTLIN,<RBSTMG>         ;YES, TELL OPERATOR
 1776  012656  000407                          BR      4$
 1777  012660                          3$:     CALL    PNTLIN,<RBCLMG>
 1778  012676  005767  004746          4$:     TST     KWFLG                   ;GOT A CLOCK?
 1779  012702  001444                          BEQ     5$                      ;NO, FORGET ABOUT TIME.
 1780  012704  005002                          CLR     R2                      ;SUPPRESS LEADING ZEROS
 1781  012706  012700  031741                  MOV     #TMLIN1,R0              ;OUTPUT POINTER FOR DECPNT
 1782  012712  016701  004730                  MOV     HOURS,R1                ;OUTPUT DATA FOR DECPNT
 1783  012716  004767  020002                  JSR     PC,DECPNT               ;ENQUEUE "HOURS"
 1784  012722  005002                          CLR     R2                      ;SUPPRESS LEADING ZEROS
 1785  012724  112720  000072                  MOVB    #':,(R0)+               ;INSERT COLON
 1786  012730  116701  004710                  MOVB    MINUTS,R1               ;GET "MINUTES" FOR DECPNT
 1787  012734  004767  017764                  JSR     PC,DECPNT               ;ENQUEUE "MINUTES"
 1788  012740  005002                          CLR     R2                      ;SUPPRESS LEADING ZEROS
 1789  012742  112720  000072                  MOVB    #':,(R0)+               ;INSERT ANOTHER COLON
 1790  012746  116701  004671                  MOVB    SECNDS,R1               ;GET "SECONDS" FOR DECPNT
 1791  012752  004767  017746                  JSR     PC,DECPNT               ;ENQUEUE "SECONDS"
 1792  012756  005002                          CLR     R2                      ;SUPPRESS LEADING ZEROS
 1793  012760  112720  000072                  MOVB    #':,(R0)+               ;INSERT ANOTHER COLON
 1794  012764  116701  004652                  MOVB    TICKS,R1                ;GET "TICKS" FOR DECPNT
 1795  012770  004767  017730                  JSR     PC,DECPNT               ;ENQUEUE "TICKS"
 1796  012774  105010                          CLRB    (R0)                    ;ENQUEUE "0" PRINT TERMINATOR.
 1797  012776                                  CALL    PNTLIN,<ELPSTM>         ;PRINT "ELAPSED TIME"
 1798  013014                          5$:     CALL    PNTLIN,<STITLE>         ;ENQUEUE STATUS HEADER
 1799  013032  012767  020100  004630          MOV     #RADB,STPNTR            ;SET POINTER TO TOP OF STAT LIST
 1800  013040  000167  000032                  JMP     STRTN                   ;RETURN
 1801
 1802  013044  062767  000014  004616  STBADV: ADD     #14,STPNTR              ;SET POINTER TO NEXT ACTIVE FLAG
 1803  013052  000167  000020                  JMP     STRTN
 1804
 1805  013056  005067  004604          STARXT: CLR     STSEV                   ;CLR STATUS EVENT FLAG
 1806  013062                                  CALL    PRESC                   ;FAKE CNTRL-C KEY
 1807  013076                          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).


1819  013100                    PROC    TEROS
 (1)
 (1)  013100              TEROS:                                  ;**ENTRY POINT**
1820  013100  005767  003520         TST     TOQ                 ;IS THE TTY OUTPUT QUEUE EMPTY?
1821  013104  001402         BEQ     60$                         ;YES, CONTINUE
1822  013106  000167  000324         JMP     TERTN               ;NO, EXIT
1823  013112  005767  004600   60$:  TST     HDRFLG              ;HEADER PRINTED YET?
1824  013116  001422         BEQ     1$                          ;YES, SKIP IT
1825  013120  005067  004572         CLR     HDRFLG              ;NO, PRINT IT NOW.
1826  013124                   CALL    PNTLIN,<TERHDR>
1827  013142                   CALL    PNTLIN,<TRHLIN>
1828  013160  000167  000252         JMP     TERTN               ;AND RETURN
1829  013164  026727  004600  000016  1$:   CMP     ERR0,#16      ;IS THIS ERROR # 16?
1830  013172  001042         BNE     3$                          ;NOT YET.
1831  013174  005067  004452         CLR     TEREV               ;YES, CLEAR XMTR ERROR EVENT
1832  013200  052767  100000  004446         BIS     #B15,REREV  ;SET RCVR ERROR EVENT
1833  013206  052767  100000  004502         BIS     #B15,HDRFLG ;SET HEADER FLAG
1834  013214  012767  000001  004550         MOV     #1,ERR1     ;LOAD INITIAL XMTR #
1835  013222  012767  026616  004544         MOV     #RERTBL,ERR2 ;POINT RCVR ERR EVENT AT ITS TABLE
1836  013230  005767  004460         TST     PRINTD              ;DID WE PRINT ANYTHING?
1837  013234  001007         BNE     2$                          ;YES, O.K.
1838  013236                   CALL    PNTLIN,<NONMG>              ;NO, PRINT ''(NONE)''
1839  013254              2$:  CALL    PNCRLF                      ;NEW LINE
1840  013270  005067  004420         CLR     PRINTD
1841  013274  000167  000136         JMP     TERTN               ;RETURN
1842  013300  005777  004470   3$:   TST     @ERR2               ;ANY ERRORS?
1843  013304  001434         BEQ     4$                          ;NO.
1844  013306  012700  032362         MOV     #TRRNO,R0           ;READY TO PRINT ERROR NO.
1845  013312  016701  004452         MOV     ERR0,R1
1846  013316  004767  017360         JSR     PC,OCTJSP           ;ERROR NO. IS IN OCTAL
1847  013322  012700  032377         MOV     #TRRCN,R0           ;READY TO PRINT RCVR NO.
1848  013326  016701  004440         MOV     ERR1,R1
1849  013332  004767  017344         JSR     PC,OCTJSP           ;RCVR NO. IS ALSO IN OCTAL
1850  013336  012700  032416         MOV     #TRERC,R0           ;NOW PRINT ERROR COUNT
1851  013342  017701  004426         MOV     @ERR2,R1
1852  013346  004767  017344         JSR     PC,DECJSP           ;ERROR COUNT IS IN DECIMAL
1853  013352                   CALL    PNTLIN,<TRELIN>             ;ENQUEUE TABLE OUTPUT LINE
1854  013370  052767  100000  004316         BIS     #B15,PRINTD ;SET 'PRINTED FLAG'
1855  013376  062767  000002  004370  4$:   ADD     #2,ERR2     ;UPDATE TABLE POINTER
1856  013404  026727  004362  000037         CMP     ERR1,#37    ;ALL RCVR'S DONE?
1857  013412  001404         BEQ     5$                          ;YES.
1858  013414  005267  004352         INC     ERR1                ;NO, DO NEXT
1859  013420  000167  000012         JMP     TERTN
1860  013424  012767  000001  004340  5$:   MOV     #1,ERR1     ;NEXT ERROR NO.
1861  013432  005267  004332         INC     ERR0
1862  013436              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)".

013440                          PROC    REROS
 (1)
 (1) 013440                     REROS:                          ;**ENTRY POINT**
013440  005767  003160                  TST     TOQ             ;IS TTY OUTPUT QUEUE EMPTY?
013444  001402                  BEQ     60$             ;YES, CONTINUE
013446  000167  000310          JMP     RERTN           ;NO, RETURN
013452  005767  004240          60$:    TST     HDRFLG          ;IS HEADER PRINTED YET?
013456  001422                  BEQ     1$              ;YES, SKIP IT
013460  005067  004232          CLR     HDRFLG          ;NO, CLEAR FLAG
013464                          CALL    PNTLIN,<RERHDR> ;AND PRINT HEADER
013502                          CALL    PNTLIN,<RRHLIN>
013520  000167  000236          JMP     RERTN           ;AND RETURN
013524  026727  004240  000031  1$:     CMP     ERR0,#31        ;DONE LAST ERROR?
013532  001034                  BNE     3$              ;NOT YET
013534  005067  004114          CLR     REREV           ;YES, CLEAR RCVR ERROR EVENT
013540  005767  004150          TST     PRINTD          ;DID WE PRINT ANYTHING?
013544  001007                  BNE     2$              ;YES, O.K.
013546                          CALL    PNTLIN,<NONMG>  ;NO, PRINT "(NONE)"
013564                  2$:     CALL    PNCRLF
013600  005067  004110          CLR     PRINTD
013604                          CALL    PRESC           ;WRAP IT UP!
013620  000167  000136          JMP     RERTN
013624  005777  004144          3$:     TST     @ERR2           ;ANY ERRORS?
013630  001434                  BEQ     4$              ;NO.
013632  012700  032517          MOV     #RCRNO,R0       ;READY TO PRINT ERROR NO.
013636  016701  004126          MOV     ERR0,R1
013642  004767  017034          JSR     PC,OCTJSP       ;ERROR NO.S ARE IN OCTAL
013646  012700  032534          MOV     #RCTRN,R0       ;READY TO PRINT XMTR NO.
013652  016701  004114          MOV     ERR1,R1
013656  004767  017020          JSR     PC,OCTJSP       ;XMTR NO.S ARE IN OCTAL
013662  012700  032553          MOV     #RCERC,R0       ;NOW PRINT ERROR COUNT
013666  017701  004102          MOV     @ERR2,R1
013672  004767  017020          JSR     PC,DECJSP       ;ERROR COUNTS ARE IN DECIMAL
013676                          CALL    PNTLIN,<RCELIN> ;ENQUEUE TABLE OUTPUT LINE
013714  052767  100000  003772  BIS     #B15,PRINTD     ;SET "PRINTED" FLAG
013722  062767  000002  004044  4$:     ADD     #2,ERR2         ;UPDATE TABLE POINTER
013730  026727  004036  000037  CMP     ERR1,#37        ;ALL XMTRS DONE?
013736  001404                  BEQ     5$
013740  005267  004026          INC     ERR1            ;NO, DO NEXT
013744  000167  000012          JMP     RERTN
013750  012767  000001  004014  5$:     MOV     #1,ERR1         ;YES, NEXT ERROR
013756  005267  004006          INC     ERR0
013762                  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''.
                ;


013764                  PROC    SUMRY
 (1)
 (1) 013764     SUMRY:
1927 013764 005767 002634       TST     TOQ             ;**ENTRY POINT**
1928 013770 001126              BNE     SMRTN           ;IS OUTPUT QUEUE EMPTY?
1929 013772 005767 003674       TST     SUMPNT          ;NO, RETURN
1930 013776 001472              BEQ     SUMHDR          ;IS OUTPUT POINTER AT TABLE?
1931 014000 027727 003666 177777 SMCHK: CMP     @SUMPNT,#-1     ;NO, GO PRINT HEADER
1932 014006 001507              BEQ     SMARXT          ;IS ERROR # = -1 ?
1933 014010 016704 003656       MOV     SUMPNT,R4       ;YES,DONE; EXIT
1934 014014 005764 000004       TST     4(R4)           ;CHECK TOTAL COUNT OF THIS ERR
1935 014020 001475              BEQ     SMADV           ;IF 0, TRY NEXT ERROR
1936 014022 012700 032574       MOV     #SMLIN1,R0      ;R0 IS OUTPUT POINTER FOR OCTJSP
1937 014026 011401              MOV     (R4),R1         ;R1 IS DATA BUFF FOR OCTJSP
1938 014030 012702 177777       MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1939 014034 004767 016642       JSR     PC,OCTJSP
1940 014040 012700 032616       MOV     #SMLIN2,R0
1941 014044 016401 000002       MOV     2(R4),R1
1942 014050 012702 177777       MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1943 014054 004767 016622       JSR     PC,OCTJSP
1944 014060 012700 032642       MOV     #SMLIN3,R0
1945 014064 016401 000004       MOV     4(R4),R1
1946 014070 020127 177770       CMP     R1,#177770      ;REACHED MAX COUNT?
1947 014074 101415              BLOS    SMPN            ;NO,OK
1948 014076 112720 000115       MOVB    #'M,(R0)+       ;YES, PRINT 'MXCNT'' IN
1949 014102 112720 000130       MOVB    #'X,(R0)+       ;  NOTE THAT 'MXCNT''
1950 014106 112720 000103       MOVB    #'C,(R0)+       ;  ALSO HAS ONLY 5
1951 014112 112720 000116       MOVB    #'N,(R0)+       ;  ASCII CHARACTERS.
1952 014116 112720 000124       MOVB    #'T,(R0)+
1953 014122 000404              BR      SMSPN
1954 014124 012702 177777       MOV     #-1,R2          ;DON'T COMPRESS BLANKS
1955 014130 004767 016562       SMPN:   JSR     PC,DECJSP
1956 014134             SMSPN:  CALL    PNTLIN,<SMLIN>  ;ENQUEUE SUM LINE FOR OUTPUT
1957 014152 062767 000006 003512        ADD     #6,SUMPNT       ;UPDATE POINTER FOR NEXT LINE
1958 014160 000167 000062       JMP     SMRTN           ;RETURN
1959
1960 014164             SUMHDR: CALL    PNTLIN,<SMTTLE> ;ENQUEUE SUMMARY HEADER
1961 014202 012767 024726 003462        MOV     #ERTBL,SUMPNT   ;SET POINTER TO TOP OF SUM LIST
1962 014210 000167 000032       JMP     SMRTN           ;RETURN
1963
1964 014214 062767 000006 003450 SMADV: ADD     #6,SUMPNT       ;SET POINTER TO NEXT ENTRY
1965 014222 000167 177552       JMP     SMCHK
1966
1967 014226 005067 003430       SMARXT: CLR     SUMEV           ;CLEAR SUMMARY EVENT FLAG
1968 014232             CALL    PRESC           ;FAKE CNTRL-C KEY
```

```
1969 014246                  SMRTN:  RETURN                          ;RETURN
```

```
                                        .SBTTL ERROR UPDATE ROUTINES
                                        .SBTTL  TRANSM.TTER ERRORS

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

                                        PROC    ERRMOD,<ERRNUM,ERRADR>
 (3)            000002          ERRNUM  =       2
 (3)            000004          ERRADR  =       4
 (1)
 (1)  014250                    ERRMOD:                                 ;**ENTRY POINT**
1977  014250  010046                          MOV     R0,-(SP)          ;SAVE R0 ON STACK
1978  014252  010146                          MOV     R1,-(SP)          ; AND R1
1979  014254  010246                          MOV     R2,-(SP)          ; AND R2
1980  014256  010346                          MOV     R3,-(SP)          ;AND R3
1981  014260  017767  001612  003476          MOV     @TCR,TEMP         ;SAVE ADDRESS OF CONNECTED RCVR
1982  014266  042767  160377  003470          BIC     #160377,TEMP
1983  014274  000367  003464                  SWAB    TEMP              ;GET IT IN THE RIGHT BYTE
1984  014300  016500  000002                  MOV     ERRNUM(R5),R0     ;GET ERROR NUMBER
1985  014304  010067  003456                  MOV     R0,TEMP1          ;SAVE ERROR NUMBER
1986  014310  006300                          ASL     R0                ;FIND TABLE ENTRY
1987  014312  010001                          MOV     R0,R1             ;       IN ORDER TO UPDATE
1988  014314  006300                          ASL     R0                ;       ADDRESS FIELD
1989  014316  060100                          ADD     R1,R0
1990  014320  062700  024720                  ADD     #ERTBL0,R0
1991  014324  016560  000004  000002          MOV     ERRADR(R5),2(R0)  ;ENTER ERROR ADDRESS
1992  014332  026027  000004  177771          CMP     4(R0),#177771     ;AT MAX COUNT YET?
1993  014340  103401                          BLO     ERMIN             ;NO, INCREMENT COUNT
1994  014342  000444                          BR      ERMIS             ;YES SKIP UPDATE.
1995  014344  005260  000004          ERMIN:  INC     4(R0)             ;UPDATE OCCURRENCES
1996  014350  005367  003412                  DEC     TEMP1             ; (ERR # - 1)
1997  014354                                  MULT    37,TEMP1,R3       ; (E-1)X37
1998  014420                                  MULT    2,TEMP1,R3        ; [(E-1)X37]X2
1999  014424  005367  003334                  DEC     TEMP              ; RCVR ADDR - 1
2000  014430                                  MULT    2,TEMP,R3         ; (R-1)X2
2001  014434  066767  003326  003322          ADD     TEMP1,TEMP        ; [(E-1)X37]X2 + (R-1)X2
2002  014442  062767  025150  003314          ADD     #TERTBL,TEMP      ;TEMP = #TERTBL + <[(E-1)X37]X2 + (R-1)X2>
2003  014450  005277  003310                  INC     @TEMP             ;UPDATE TABLE ENTRY FOR THIS ERROR
2004  014454  012603                  ERMIS:  MOV     (SP)+,R3          ;RESTORE R3
2005  014456  012602                          MOV     (SP)+,R2          ;RESTORE R2
2006  014460  012601                          MOV     (SP)+,R1          ;RESTORE R1
2007  014462  012600                          MOV     (SP)+,R0          ;RESTORE R0
2008  014464                                  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**
014466  010046                          MOV     R0,-(SP)                ;SAVE R0 ON THE STACK
014470  010146                          MOV     R1,-(SP)                ;AND R1
014472  010246                          MOV     R2,-(SP)                ;AND R2
014474  010346                          MOV     R3,-(SP)                ;AND R3
014476  017767  001414  003260          MOV     @RCR,TEMP               ;GET ADDR OF CONNECTED XMTR
014504  042767  160377  003252          BIC     #160377,TEMP
014512  000367  003246                  SWAB    TEMP                    ; INTO RIGHT BYTE
014516  016500  000002                  MOV     ERRNUM(R5),R0           ;GET ERROR NUMBER
014522  010067  003240                  MOV     R0,TEMP1                ;AND SAVE IT
014526  006300                          ASL     R0                      ; FIND 'SUM' TABLE ENTRY
014530  010001                          MOV     R0,R1                   ; IN ORDER TO UPDATE
014532  006300                          ASL     R0                      ; ADDRESS FIELD
014534  060100                          ADD     R1,R0                   ; AND # OF OCCURRENCES
014536  062700  024720                  ADD     #ERTBL0,R0
014542  016560  000004  000002          MOV     ERRADR(R5),2(R0)        ;ENTER ERROR ADDRESS
014550  026027  000004  177771          CMP     4(R0),#177771           ;AT MAX COUNT YET?
014556  103401                          BLO     ERMINR                  ;NO, UPDATE ERROR COUNT
014560  000447                          BR      ERMISR                  ;YES, SKIP UPDATES
014562  005260  000004          ERMINR: INC     4(R0)                   ;UPDATE OCCURRENCES
014566  162767  000015  003172          SUB     #15,TEMP1               ;ALIGN ERRORS TABLE FOR RCVR
014574  005367  003166                  DEC     TEMP1                   ; ERR # - 1 (E-1)
014600                                  MULT    37,TEMP1,R3             ; (E-1)X37
014644                                  MULT    2,TEMP1,R3              ; [(E-1)X37]X2
014650  005367  003110                  DEC     TEMP                    ; XMTR ADDR - 1  (T-1)
014654                                  MULT    2,TEMP,R3               ; (T-1)X2
014660  066767  003102  003076          ADD     TEMP1,TEMP              ; [(E-1)X37]X2 + (T-1)X2
014666  062767  026616  003070          ADD     #RERTBL,TEMP            ; TEMP = #RERTBL+[(E-1)X37]X2 + (T-1)X2
014674  005277  003064                  INC     @TEMP                   ;UPDATE RCVR ERROR COUNT
014700  012603                  ERMISR: MOV     (SP)+,R3                ;RESTORE R3
014702  012602                          MOV     (SP)+,R2                ;RESTORE R2
014704  012601                          MOV     (SP)+,R1                ;RESTORE R1
014706  012600                          MOV     (SP)+,R0                ;RESTORE R0
014710                                  RETURN
```

```
2049                                    .SBTTL UTILITY ROUTINES
2050                                    .SBTTL  PROCESS AN INPUT CHARACTER FROM THE TTY
2051
2052  014712                            PROC    TTINP
 (1)
 (1)  014712                    TTINP:                                  ;**ENTRY POINT**
2053
2054  014712                            CALL    DEQ,<FRELEM,TIQ>        ;GET TTY INPUT CHARACTER.
2055  014732  142767  000200  013324    BICB    #200,FRELEM             ;CLEAR OFF PARITY BIT.
2056  014740  116700  013320            MOVB    FRELEM,R0
2057  014744  120027  000141            CMPB    R0,#141                 ;CONVERT LOWER CASE INPUT
2058  014750  002405                    BLT     TTINH                   ;  TO REGULAR (UPPER CASE)
2059  014752  120027  000172            CMPB    R0,#172                 ;  LETTERS INSTEAD.
2060  014756  003002                    BGT     TTINH
2061  014760  042700  000040            BIC     #B05,R0                 ;CONVERT CHARACTER
2062  014764  010067  013274    TTINH:  MOV     R0,FRELEM               ; IN 'FRELEM'
2063  014770  012701  016034            MOV     #CMCHTB,R1              ; TO ITS PROCESSING
2064  014774  004767  015654            JSR     PC,PROCHR               ; ROUTINE.
2065  015000  001407                    BEQ     CMMISC                  ;IF MISC CHAR, CALL MISC PROC.
2066  015002                            CALL    @R1,<FRELEM>            ;CALL PROCESSING ROUTINE.
2067  015016  000407                    BR      TTRET                   ;RETURN.
2068
2069  015020                    CMMISC: CALL    PRMISC,<FRELEM>         ;PROCESS MISCELLANEOUS CHARACTER
2070  015036                    TTRET:  RETURN                          ;RETURN.
```

```
                        .SBTTL  TTY INPUT CHARACTER PROCESSING ROUTINES

                        ;
                        ;  PROCESSOR FOR CARRIAGE RETURN
                        ;  THIS CHARACTER SIGNIFIES THE END OF A COMMAND LINE.
                        ;
015040                  PROC    PRCR
 (1)
 (1)  015040            PRCR:                                   ;**ENTRY POINT**
2079  015040  105767  002574          TSTB    REQINP            ;IF INPUT NOT BEING TYPED,
2080  015044  002015                  BGE     CRRET             ; IGNORE CHARACTER.
2081  015046  112777  000015  001326  MOVB    #CR.,@CBUFPT      ;PUT CR IN BUFFER.
2082  015054  105367  002561          DECB    CMDENT            ;SET COMMAND ENTERED FLAG.
2083  015060  105067  002554          CLRB    REQINP            ;CLEAR INPUT REQUESTED FLAG.
2084  015064                          CALL    PNCRLF            ;ECHO CR & LF.
2085  015100                  CRRET:  RETURN                    ;RETURN.
2086
2087                        ;
2088                        ;  PROCESSOR FOR CONTROL-O CHARACTER.
2089                        ;  ALL CHARACTERS PRESENTLY IN THE TTY OUTPUT QUEUE ARE THROWN AWAY.
2090                        ;
2091  015102                PROC    PRCTLO
 (1)
 (1)  015102            PRCTLO:                                 ;**ENTRY POINT**
2092  015102  005067  001516          CLR     TOQ               ;THROW AWAY ALL CHARACTERS
2093  015106  016767  001524  001520  MOV     TOQ+QBACK,TOQ+QFRONT  ; IN TTY OUTPUT QUEUE.
2094  015114                          CALL    PNTLIN,<CTLOMG>   ;ECHO ''^O''.
2095  015132                          RETURN                    ;RETURN.
2096
2097
2098
2099                        ;
2100                        ;  PROCESSOR FOR CONTROL S.
2101                        ;  TEMPORARILY SUSPEND  PRINTOUT ON CONSOLE DEVICE
2102                        ;
2103                        ;  PRINTOUT WILL BE RESUMED UPON RECEIPT OF A CNTRL-Q
2104                        ;  OR ANY OTHER KEYBOARD INTERRUPT.
2105
2106  015134                PROC    PRCTLS
 (1)
 (1)  015134            PRCTLS:                                 ;**ENTRY POINT**
2107  015134  052767  100000  002560  BIS     #B15,THLTFL       ;SET TTY HALT FLAG
2108  015142                          RETURN
2109
2110
2111                        ;
2112                        ;  PROCESSOR FOR CONTROL Q.
2113                        ;  RESUME PRINTOUT ON CONSOLE IF TTOUT QUEUE HAS NOT BEEN
2114                        ;  CLEARED BY SOME OTHER MEANS.
2115
2116  015144                PROC    PRCTLQ
 (1)
 (1)  015144            PRCTLQ:                                 ;**ENTRY POINT**
2117  015144  042767  100000  002550  BIC     #B15,THLTFL       ;CLEAR TTY HALT FLAG
2118  015152                          RETURN
2119
```

```
2120
2121                                    ;
2122                                    ;PROCESSOR FOR CONTROL-C.
2123                                    ;ECHO ''^C'' AND GO TO PRESC ROUTINE.
2124
2125  015154                                    PROC    PRCTLC
 (1)
 (1)  015154                            PRCTLC:                                 ;**ENTRY POINT**
2126  015154                                    CALL    PNTLIN,<CTLCMG>         ;ECHO ''^C''
2127  015172  000167  000016                    JMP     PRESC                   ;GO TO PRESC ROUTINE.
2128
2129
2130                                    ;
2131                                    ;  PROCESSOR FOR CONTROL-U.
2132                                    ;  INPUT COMMAND BEING TYPED IS DISCARDED.
2133
2134  015176                                    PROC    PRCTLU
 (1)
 (1)  015176                            PRCTLU:                                 ;**ENTRY POINT**
2135  015176                                    CALL    PNTLIN,<CTLUMG>         ;ECHO ''^U''.
2136                                                                            ;DROP INTO PRESC
2137
2138
2139
2140                                    ;
2141                                    ;  PROCESSOR FOR CNTRL-C KEY.
2142                                    ;  THIS KEY IS USED TO REQUEST INPUT.  ALL OTHER EVENTS WILL CEASE
2143                                    ;  AFTER THE CURRENT LINE.  A 'PCL>'' WILL BE ECHOED ON THE TTY.  THE
2144                                    ;  OPERATOR IS NOW FREE TO TYPE IN A COMMAND.
2145                                    ;
2146
2147  015214                                    PROC    PRESC
 (1)
 (1)  015214                            PRESC:                                  ;**ENTRY POINT**
2148  015214  005767  002430                    TST     KWFLG                   ;GOT A CLOCK?
2149  015220  001402                            BEQ     14$                     ;NO.
2150  015222  005077  000722                    CLR     @LCS                    ;STOP THE CLOCK
2151  015226  105367  002406            14$:    DECB    REQINP                  ;SET INPUT REQUESTED FLAG.
2152  015232  005067  002430                    CLR     STSEV                   ;CLR STATUS EVENT
2153  015236  005067  002420                    CLR     SUMEV                   ;CLR SUMMARY EVENT
2154  015242  005067  002404                    CLR     TEREV                   ;CLR XMTR ERROR EVENT
2155  015246  005067  002402                    CLR     REREV                   ;AND RCVR ERROR EVENT
2156  015252  005067  002422                    CLR     PCLGO                   ;STOP XMTR
2157  015256                                    CALL    ENQ,<A.P,TOQ>           ;ECHO 'PCL>''.
2158  015276                                    CALL    ENQ,<A.C,TOQ>
2159  015316                                    CALL    ENQ,<A.L,TOQ>
2160  015336                                    CALL    ENQ,<A.PR,TOQ>
2161  015356  012767  016176  001016            MOV     #CMDBUF,CBUFPT          ;INITIALIZE COMMAND BUF PTR.
2162  015364  105067  002251                    CLRB    CMDENT                  ;CLEAR COMMAND ENTERED FLAG.
2163  015370                                    RETURN                          ;RETURN.
2164
2165
2166                                    ;
2167                                    ;  PROCESS RUB OUT KEY.
2168                                    ;  LAST CHARACTER IN COMMAND BUFFER IS DELETED.  A ''\'' IS ECHOED.
2169                                    ;
```

```
                                        PROC    PRDEL
                                PRDEL:                                  ;**ENTRY POINT**
015372  105767  002242                        TSTB    REQINP          ;IF INPUT NOT REQUESTED,
015376  002034                                BGE     PDRET           ; THEN IGNORE RUBOUT CHAR.
015400  026727  000776  016176                CMP     CBUFPT,#CMDBUF  ;IF AT BEGINNING OF BUFFER,
015406  003016                                BGT     1$              ; THEN ISSUE NEW 'PCL>'
015410                                        CALL    PNCRLF
015424                                        CALL    PRESC
015440  000167  000024                        JMP     PDRET
015444  005367  000732                1$:     DEC     CBUFPT          ;DECREMENT BUFFER POINTER.
015450                                        CALL    ENQ,<A.BKSL,TOQ>        ;ECHO A '\'.
015470                                PDRET:  RETURN                  ;RETURN.


                                ;
                                ;  PROCESSOR FOR MISCELLANEOUS CHARACTERS.
                                ;  THE CHARACTER IS APPENDED TO THE COMMAND BUFFER.
                                ;


015472                                  PROC    PRMISC,<CHAR>
        000002                  CHAR    =       2
                                PRMISC:                                 ;**ENTRY POINT**
015472  105767  002142                        TSTB    REQINP          ;IF INPUT NOT REQUESTED,
015476  002034                                BGE     PMRET           ; THEN JUST IGNORE CHARACTER.
015500  117500  000002                        MOVB    @CHAR(R5),R0
015504  120027  000040                        CMPB    R0,#' '         ;IF CHARACTER
015510  002403                                BLT     PM176           ; IS NONPRINTING
015512  120027  000140                        CMPB    R0,#140         ; THEN CHANGE
015516  002402                                BLT     PM187           ; IT TO
015520  012700  000077                PM176:  MOV     #'?,R0          ; ASCII '?'.
015524  026727  000652  016401        PM187:  CMP     CBUFPT,#CBFEND-1        ;IF WE ARE AT END OF BUFFER,
015532  001416                                BEQ     PMRET           ; THEN JUST RETURN.
015534  110077  000642                        MOVB    R0,@CBUFPT      ;PUT CHARACTER INTO BUFFER
015540  005267  000636                        INC     CBUFPT          ;UPDATE BUFFER POINTER.
015544  110067  012516                        MOVB    R0,BKELEM
015550                                        CALL    ENQ,<BKELEM,TOQ>        ;ECHO CHARACTER.
015570                                PMRET:  RETURN                  ;RETURN.
```

```
                                        .SBTTL  TTY OUTPUT HANDLERS


                                ;
                                ;  OUTPUT A CHARACTER TO THE TELETYPE IF IT IS READY.
                                ;
015572                                  PROC    TTOUT
(1)
(1)  015572                     TTOUT:                          ;**ENTRY POINT**
015572  105777  000344                  TSTB    @TTXCSR         ;IF DEVICE IS NOT READY,
015576  002020                          BGE     TORET           ; THEN JUST RETURN.
015600  005767  002116                  TST     THLTFL          ;IS TTY HALTED (CNTRL-S)?
015604  001402                          BEQ     1$              ;NO, O.K. TO PRINT
015606  000167  000026                  JMP     TORET           ;YES, DON'T DO ANYTHING.
015612                          1$:     CALL    DEQ,<FRELEM,TOQ>        ;GET NEXT CHAR TO TYPE.
015632  116777  012426  000304          MOVB    FRELEM,@TTXBUF  ;OUTPUT IT.
015640                          TORET:  RETURN                  ;RETURN.


                                        .SBTTL  TTY INPUT INTERRUPT PROCESSORS

015642                          TTIINT:                         ;;;**INTERRUPT ENTRY POINT**
015642  005067  002054                  CLR     THLTFL          ;;;CLEAR TTY HALT FLAG ON INPUT
015646                                  REGSAV                  ;;;SAVE R0 - R5.
015652  117767  000262  012414          MOVB    @TTRBUF,INTTMP  ;;;GET INPUT CHARACTER.
015660                                  CALL    ENQ,<INTTMP,TIQ>        ;;;PUT IT IN TTY INPUT QUEUE.
015700                                  REGRES                  ;;;RESTORE R0 - R5.
015704  000002                          RTI                     ;;;RETURN.
```

```
                                        .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.
                                ;

                                        PROC    PNTLIN,<MESSAG>
  (3)           000002          MESSAG  -       2
  (1)
  (1)                           PNTLIN:                                 ;**ENTRY POINT**
2245  015706  016546  000002              MOV     MESSAG(R5),-(SP)        ;GET ADDRESS OF MESSAGE.
2246  015712  117667  000000  012346  PLODEC: MOVB    @(SP),BKELEM            ;GET A CHARACTER.
2247  015720  001412                      BEQ     PLCRLF                  ;IF NULL, APPEND CR & LF.
2248  015722                              CALL    ENQ,<BKELEM,TOQ>        ;ENQUEUE CHAR FOR TTY OUTPUT.
2249  015742  005216                      INC     @SP                     ;POINT TO NEXT CHARACTER.
2250  015744  000762                      BR      PLODEC                  ;PROCESS IT.
2251
2252  015746  005026                  PLCRLF: CLR     (SP)+                   ;POP ADDR FROM STACK.
2253  015750  000400                      BR      PL219                   ;APPEND CR & LF.
2254
2255
2256                                  ;
2257                                  ;  ENQUEUE A CR & LF IN TTY OUTPUT QUEUE.
2258                                  ;
2259  015752                              PROC    PNCRLF
  (1)
  (1)  015752                     PNCRLF:                                 ;**ENTRY POINT**
2260  015752                      PL219:  CALL    ENQ,<A.CR,TOQ>          ;ENQUEUE A CR.
2261  015772                              CALL    ENQ,<A.LF,TOQ>          ;ENQUEUE A LF.
2262  016012                              CALL    ENQ,<ZERO,TOQ>          ;ENQUEUE A NULL FILL CHAR.
2263  016032                              RETURN                          ;RETURN.
```

```
                                        .SBTTL DATA AREAS

                                ;  TABLE ASSOCIATING SPECIAL TTY INPUT CHARACTERS WITH THEIR
                                ;  PROCESSING ROUTINES.
016034                          CMCHTB:
016034  000003  015154                .WORD   CTL.C,PRCTLC
016040  000012  015040          A.LF:   .WORD   LF.,PRCR
016044  000015  015040          A.CR:   .WORD   CR.,PRCR
016050  000017  015102                .WORD   CTL.O,PRCTLO
016054  000021  015144                .WORD   CTL.Q,PRCTLQ
016060  000023  015134                .WORD   CTL.S,PRCTLS
016064  000025  015176                .WORD   CTL.U,PRCTLU
016070  000177  015372                .WORD   RUBOUT,PRDEL
016074  177777                        .WORD   -1


                                ;DEVICE ADDRESS TABLES:

016076  164200                TCR:    .WORD   PCLTXM
016100  164202                TSR:    .WORD   PCLTXM+2
016102  164204                TSDB:   .WORD   PCLTXM+4
016104  164206                TSBC:   .WORD   PCLTXM+6
016106  164210                TSBA:   .WORD   PCLTXM+10
016110  164212                TMMR:   .WORD   PCLTXM+12
016112  164213                TMMRH:  .WORD   PCLTXM+13
016114  164214                TSCRC:  .WORD   PCLTXM+14
016116  164220                RCR:    .WORD   PCLRCV
016120  164222                RSR:    .WORD   PCLRCV+2
016122  164224                RDDB:   .WORD   PCLRCV+4
016124  164226                RDBC:   .WORD   PCLRCV+6
016126  164230                RDBA:   .WORD   PCLRCV+10
016130  164234                RDCRC:  .WORD   PCLRCV+14

016132  000170                TXMVEC: .WORD   170
016134  000174                RCVVEC: .WORD   174

016136  177560                TTRCSR: .WORD   TTDEV
016140  177562                TTRBUF: .WORD   TTDEV+2
016142  177564                TTXCSR: .WORD   TTDEV+4
016144  177566                TTXBUF: .WORD   TTDEV+6
016146  000060                TTVECT: .WORD   TTVCTR                  ;VECTOR ADDRESS.
016150  177546                LCS:    .WORD   177546                  ;KW11-L LINE CLOCK ADDR


                                ;  CHARACTER CONSTANTS.  THESE ARE DEFINED AS WORDS SO THAT THEY MAY
                                ;  BE ENQUEUED.
016152  000134                A.BKSL: .WORD   '\                      ;ASCII '\'
016154  000000                ZERO:   .WORD   0                       ;ASCII NULL
016156  000120                A.P:    .WORD   'P                      ;ASCII 'P'
016160  000103                A.C:    .WORD   'C                      ;ASCII 'C'
016162  000114                A.L:    .WORD   'L                      ;ASCII 'L'
016164  000076                A.PR:   .WORD   '>                      ;ASCII '>' (PROMPT)

                                        .EVEN


                                ;DEVICE ADDRESS AND VECTOR VARIABLES
```

```
                                        ;CHANGE THESE LOCATIONS TO MODIFY ALL DEVICE ADDRESSES AND VECTORS
                                        ;FOR PCL11.
                                        
016166  164200                  TXDEV:  .WORD   164200                  ;DEFAULT IS 164200
016170  000170                  TXVEC:  .WORD   170                     ;DEFAULT IS 170
016172  164220                  RCDEV:  .WORD   164220                  ;DEFAULT IS 164220
016174  000174                  RCVEC:  .WORD   174                     ;DEFAULT IS 174
                                        
                                        ;  BUFFER & MESSAGE AREAS.
016176  000204                  CMDBUF: .BLKB   132.                    ;TTY INPUT COMMAND BUFFER.
        016402                  CBFEND  =       .
016402  033333                  CBUFPT: .WORD   33333                   ;CMDBUF BUFFER POINTER.
                                        
016404  047536     000          CTLOMG: .ASCIZ  '^O'                    ;FOR ECHOING CONTROL CHARACTERS.
016407     136  000125          CTLUMG: .ASCIZ  '^U'                    ; ''
016412  041536     000          CTLCMG: .ASCIZ  '^C'                    ; ''
                                        
016415     040  051105  047522  RPMSG:  .ASCIZ  ' ERROR'
016422  000122
                                        ;  QUEUE DEFINITIONS.
                                        .IRP    LC      <AO,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
(1) 016424  000000                  AOQ:    .WORD   0                       ;QELEMS
(1) 016426  000040                          .WORD   AOSIZE                  ;QSIZE
(1) 016430  016440  016540                  .WORD   AOAREA,AOEND            ;QTOP & QBOT
(1) 016434  033333  033333                  .WORD   33333,33333             ;QFRONT & QBACK
(1) 016440  000040-                 AOAREA: .BLKW   AOSIZE                  ;AOQ AREA
(1)         016540                  AOEND   =       .
(1) 016540  000000                  TIQ:    .WORD   0                       ;QELEMS
(1) 016542  000024                          .WORD   TISIZE                  ;QSIZE
(1) 016544  016554  016624                  .WORD   TIAREA,TIEND            ;QTOP & QBOT
(1) 016550  033333  033333                  .WORD   33333,33333             ;QFRONT & QBACK
(1) 016554  000024                  TIAREA: .BLKW   TISIZE                  ;TIQ AREA
(1)         016624                  TIEND   -       .
(1) 016624  000000                  TOQ:    .WORD   0                       ;QELEMS
(1) 016626  000400                          .WORD   TOSIZE                  ;QSIZE
(1) 016630  016640  017640                  .WORD   TOAREA,TOEND            ;QTOP & QBOT
(1) 016634  033333  033333                  .WORD   33333,33333             ;QFRONT & QBACK
(1) 016640  000400                  TOAREA: .BLKW   TOSIZE                  ;TOQ AREA
(1)         017640                  TOEND   =       .
                                        
                                        ;  FLAG VARIABLES.
                                        ;         < 0 -=> TRUE
                                        ;         >-0 -=> FALSE
                                        ;
017640     333                  REQINP: .BYTE   333                     ;INPUT REQUEST IS BEING TYPED.
017641     333                  CMDENT: .BYTE   333                     ;COMMAND HAS BEEN ENTERED.
```

```
2358  017642  000          TICKS:  .BYTE   0
2359  017643  000          SECNDS: .BYTE   0
2360  017644  000          MINUTS: .BYTE   0
2361          017646               .EVEN
2362  017646  000000       HOURS:  .WORD   0
2363  017650  000000       KWFLG:  .WORD   0
2364  017652  000000       TEREV:  .WORD   0
2365  017654  000000       REREV:  .WORD   0
2366  017656  000000       QLDEV:  .WORD   0
2367  017660  000000       DATGEV: .WORD   0
2368  017662  000000       SUMEV:  .WORD   0
2369  017664  000000       SUMSV:  .WORD   0
2370  017666  000000       STSEV:  .WORD   0
2371  017670  000000       STPNTR: .WORD   0
2372  017672  000000       SUMPNT: .WORD   0
2373  017674  000000       TXMEV:  .WORD   0
2374  017676  000000       TXMSV:  .WORD   0
2375  017700  000000       PCLGO:  .WORD   0
2376  017702  000000       RCVEV:  .WORD   0
2377  017704  000000       SPCEV:  .WORD   0
2378  017706  000000       XSPCEV: .WORD   0
2379  017710  000000       FIRST:  .WORD   0
2380  017712  000000       NEXT:   .WORD   0
2381  017714  000000       PRINTD: .WORD   0
2382  017716  000000       HDRFLG: .WORD   0
2383  017720  000000       PADFLG: .WORD   0
2384  017722  000000       THLTFL: .WORD   0
2385
2386                               .EVEN
2387
2388                       ;  DATA VARIABLES.
2389  017724  133333       ORIGSD: .WORD   133333
2390  017726  000000       DTSEED: .WORD   0
2391  017730  000333       MLSD:   .WORD   333
2392  017732  000000       MSGLSD: .WORD   0
2393  017734  000000       RCSEED: .WORD   0
2394  017736  000000       RCTXPS: .WORD   0
2395  017740  000000       RJCTF:  .WORD   0
2396  017742  000000       TRNKF:  .WORD   0
2397  017744  000000       TXML:   .WORD   0
2398  017746  037565       RANM:   .WORD   37565
2399  017750  012247       RANK:   .WORD   12247
2400  017752  000000       CURAD:  .WORD   0
2401  017754  000000       RSPC:   .WORD   0
2402  017756  000000       ESCFLG: .WORD   0
2403  017760  000000       OBJCNT: .WORD   0
2404  017762  000000       RSHOLD: .WORD   0
2405  017764  000000       TEMP:   .WORD   0
2406  017766  000000       TEMP1:  .WORD   0
2407  017770  000000       ERR0:   .WORD   0
2408  017772  000000       ERR1:   .WORD   0
2409  017774  000000       ERR2:   .WORD   0
2410
2411
2412                       ;ADDRESS SILO DATA BUFFER AREA
2413
```

```
2414  017776  000102                  ADSILO: .BLKB   66.                     ;66. BYTE AREA FOR SILO DATA
2415
2416                                  ;RECEIVER ADDRESS TABLE
2417
2418          000001                  X       =       1
2419  020100                          RADB:
2420          000037                          .REPT   31.
2421                                          .WORD   0                       ;ACTIVITY FLAG
2422                                          .WORD   X                       ;RECEIVER ADDRESS
2423                                          .WORD   0,0                     ;ATTEMPTS ENTRY
2424                                          .WORD   0,0                     ;SUCCESSES ENTRY
2425
2426                                  X       -       X+1
2427                                          .ENDR
2428  020664  020664                  RADEND: .WORD   .
2429          020064                  RADB0=  RADB-14
2430
2431                                  ;TRANSMITTER DATA BUFFER:
2432
2433  020666  001020                  DATBUF: .BLKW   1020
2434
2435
2436                                  ;RECEIVER DATA BUFFER:
2437
2438  022726  001000                  RCBUF:  .BLKW   1000
2439
2440                                  ;EXERCISER ERROR TABLE
2441
2442          000001                  Y       =       1                       ;INITIAL ERROR NUMBER
2443  024726                          ERTBL:
2444          000030                          .REPT   N-1
2445                                          .WORD   Y                       ;ERROR NUMBER
2446                                          .WORD   0                       ;ERROR ADDRESS
2447                                          .WORD   0                       ;NO. OF OCCURRENCES SINCE INIT
2448
2449                                  Y       -       Y+1
2450                                          .ENDR
2451  025146  177777                          .WORD   -1                      ;LAST ERROR # IS -1
2452
2453          024720                  ERTBL0  =       ERTBL-6
2454
2455
2456                                  ;       DETAILED ERROR TABLES FOR RCVR AND XMTR ERRORS:
2457
2458  025150  000623                  TERTBL: .BLKW   37*15                    ;RESERVE SPACE FOR XMTR ERRORS
2459
2460
2461  026616  000623                  RERTBL: .BLKW   37*15                    ;RESERVE SPACE FOR RCVR ERRORS
2462
2463
2464
2465
2466  030264  033333                  FRELEM: .WORD   33333                   ;STORAGE FOR DEQUEUED ELEMENT.
2467  030266  033333                  BKELEM: .WORD   33333                   ;STORAGE FOR ENQUEUED ELEMENT.
2468  030270  033333                  TCBFPT: .WORD   33333                   ;CMDBUF POINTER USED DURING SCAN
2469  030272  033333                  TCBIN:  .WORD   33333                   ;BINARY VALUE OF INPUT PARAMETER
```

```
                                INTTMP: .WORD   33333                   ;TEMP STORAGE FOR INTERRUPT PROC

                                        .SBTTL  KEYWORD TABLE

                                ;KEYWORD TABLE ASSOCIATING A COMMAND WITH ITS PROCESSING ROUTINE

                                CMDTBL: .WORD   CPADD
                                        .BYTE   2,3
                                        .ASCII  /ADD /
                                        .WORD   CPASS
                                        .BYTE   2,6
                                        .ASCII  /ASSIGN/
                                        .WORD   CPCLR
                                        .BYTE   2,5
                                        .ASCII  /CLEAR /
                                        .WORD   CPCNT
                                        .BYTE   2,8.
                                        .ASCII  /CONTINUE/
                                        .WORD   CPDEL
                                        .BYTE   1,6
                                        .ASCII  /DELETE/
                                        .WORD   CPERR
                                        .BYTE   1,6
                                        .ASCII  /ERRORS/
                                        .WORD   CPGO
                                        .BYTE   1,2
                                        .ASCII  /GO/
                                        .WORD   CPINIT
                                        .BYTE   1,10.
                                        .ASCII  /INITIALIZE/
                                        .WORD   CPMAST
                                        .BYTE   1,6
                                        .ASCII  /MASTER/
                                        .WORD   CPRANG
                                        .BYTE   2,5
                                        .ASCII  /RANGE /
                                        .WORD   CPRIB
                                        .BYTE   2,3
                                        .ASCII  /RIB /
                                        .WORD   CPSEC
                                        .BYTE   2,9.
                                        .ASCII  /SECONDARY /
                                        .WORD   CPSILO
                                        .BYTE   2,4
                                        .ASCII  /SILO/
                                        .WORD   CPSTAT
                                        .BYTE   2,6
                                        .ASCII  /STATUS/
                                        .WORD   CPSUM
                                        .BYTE   2,7
                                        .ASCII  /SUMMARY /
                                        .WORD   0,0
```

```
 2522
 2523
 2524  030532  000000                    SCTBL:  .WORD   0
 2525  030534     001      005                  .BYTE   1,5
 2526  030536  046103  040505  020122           .ASCII  /CLEAR /
 2527  030544  000001                           .WORD   1
 2528  030546     001      003                  .BYTE   1,3
 2529  030550  042523  020124                   .ASCII  /SET /
 2530  030554  000000  000000                   .WORD   0,0
 2531
 2532
 2533                                           .SBTTL  SOME MORE ASCII STORAGE:
 2534
 2535  030560  020040  020040  020052  MDNER:   .ASCIZ  /    * * * * * *  MASTER DOWN  * * * * * */
       030566  020052  020052  020052
       030574  020052  020052  046440
       030602  051501  042524  020122
       030610  047504  047127  020040
       030616  020052  020052  020052
       030624  020052  020052  000052
 2536  030632  020040  025052  052040  MSCHNG  .ASCIZ  /  ** THIS UNIT HAS BECOME 'NEW MASTER' **/
       C30640  044510  020123  047125
       030646  052111  044040  051501
       030654  041040  041505  046517
       030662  020105  047042  053505
       030670  046440  051501  042524
       030676  021122  025040  000052
 2537  030704  020040  025052  051452  SYNTAX: .ASCIZ  /  ***SYNTAX ERROR***/
       030712  047131  040524  020130
       030720  051105  047522  025122
       030726  025052     000
 2538  030731     120  046103  030461  PCLEXM: .ASCIZ  /PCL11 EXERCISER V02  CZPLAB0  (SEP-78)/
       030736  042440  042530  041522
       030744  051511  051105  053040
       030752  031060  020040  055103
       030760  046120  041101  020060
       030766  024040  042523  026520
       030774  034067  000051
 2539  031000  042522  052123  051101  RSTMSG: .ASCII  /RESTART AT ADDRESS /
       031006  020124  052101  040440
       031014  042104  042522  051523
       031022     040
 2540  031023     132  055132  055132  RSADD:  .ASCIZ  /ZZZZZZ/
       031030  000132
 2541  031032  025052  051440  051117  MTQMSG: .ASCIZ  /** SORRY, I HAVE NO RECEIVER ADDRESSES. **/
       031040  054522  020054  020111
       031046  040510  042526  047040
       031054  020117  042522  042503
       031062  053111  051105  040440
       031070  042104  042522  051523
       031076  051505  020041  025052
       031104     000
 2542  031105     124  040522  050120  TRPDMG: .ASCII  /TRAPPED TO 4 FROM LOCATION /
       031112  042105  052040  020117
       031120  020064  051106  046517
       031126  046040  041517  052101
```

```
                031134  047511  020116
 2543           031140  047116  047116  020116  TRP4AD: .ASCIZ  /NNNNN  ./
                031146  020440     000
 2544           031151     104  053105  041511  ERTMSG: .ASCIZ  /DEVICE ADDRESS ERROR. USE ''ASSIGN'' COMMAND./
                031156  020105  042101  051104
                031164  051505  020123  051105
                031172  047522  027122  052440
                031200  042523  021040  051501
                031206  044523  047107  020042
                031214  047503  046515  047101
                031222  027104     000
 2545           031225     105  042530  041522  EXRST:  .ASCIZ  /EXERCISER STARTED/
                031232  051511  051105  051440
                031240  040524  052122  042105
                031246     000
 2546           031247     105  042530  041522  EXCNT:  .ASCIZ  /EXERCISER CONTINUING/
                031254  051511  051105  041440
                031262  047117  044524  052516
                031270  047111  000107
 2547           031274  025052  052052  044510  MSTMG1: .ASCIZ  /***THIS UNIT IS NOT MASTER***/
                031302  020123  047125  052111
                031310  044440  020123  047516
                031316  020124  040515  052123
                031324  051105  025052  000052
 2548           031332  052502  020124  040510  MSTMG2: .ASCIZ  /BUT HAS NOW BEEN MADE SECONDARY./
                031340  020123  047516  020127
                031346  042502  047105  046440
                031354  042101  020105  042523
                031362  047503  042116  051101
                031370  027131     000
 2549           031373     124  042510  051440  MSTMG3: .ASCIZ  /THE SILO YOU HAVE JUST LOADED WILL BE/
                031400  046111  020117  047531
                031406  020125  040510  042526
                031414  045040  051525  020124
                031422  047514  042101  042105
                031430  053440  046111  020114
                031436  042502     000
 2550           031441     125  042523  020104  MSTMG4: .ASCIZ  /USED IF YOU CLEAR THE CURRENT MASTER./
                031446  043111  054440  052517
                031454  041440  042514  051101
                031462  052040  042510  041440
                031470  051125  042522  052116
                031476  046440  051501  042524
                031504  027122     000
 2551           031507     124  042510  051440  MSTMG5: .ASCIZ  /THE SILO HAS BEEN PADDED WITH ADDRESS '0'/
                031514  046111  020117  040510
                031522  020123  042502  047105
                031530  050040  042101  042504
                031536  020104  044527  044124
                031544  040440  042104  042522
                031552  051523  021040  021060
                031560     000
 2552           031561     124  044510  020123  THUMST: .ASCIZ  /THIS UNIT IS -MASTER-/
                031566  047125  052111  044440
                031574  020123  046455  051501
                031602  042524  026522     000
```

```
 2553  031607     124  044510  020123  THUSCN: .ASCIZ  /THIS UNIT IS -SECONDARY-/
       031614  047125  052111  044440
       031622  020123  051455  041505
       031630  047117  040504  054522
       031636  000055
 2554  031640  051042  041111  020042  RBSTMG: .ASCIZ  /'RIB' IS -SET-/
       031646  051511  026440  042523
       031654  026524     000
 2555  031657     042  044522  021102  RBCLMG: .ASCIZ  /'RIB' IS -CLEAR-/
       031664  044440  020123  041455
       031672  042514  051101  000055
 2556  031700  046105  050101  042523  ELPSTM: .ASCII  /ELAPSED TIME (HRS:MIN:SEC:TIK).../
       031706  020104  044524  042515
       031714  024040  051110  035123
       031722  044515  035116  042523
       031730  035103  044524  024513
       031736  027056     056
 2557  031741     060  030072  030072  TMLIN1: .ASCIZ  /0:0:0:0          /
       031746  030072  020040  020040
       031754  020040  020040  000040
 2558  031762  041522  051126  040440  STITLE: .ASCIZ  /RCVR ADDRESS  CONNECTION ATTEMPTS  SUCCESSFUL CONNECTIONS/
       031770  042104  042522  051523
       031776  020040  047503  047116
       032004  041505  044524  047117
       032012  040440  052124  046505
       032020  052120  020123  051440
       032026  041525  042503  051523
       032034  052506  020114  047503
       032042  047116  041505  044524
       032050  047117  000123
 2559
 2560  032054                  STLIN:
 2561  032054  047116  047116  020040  STLIN1: .ASCII  /NNNN              /
       032062  020040  020040  020040
       032070  020040  020040  020040
 2562  032076  047116  047116  020040  STLIN2: .ASCII  /NNNN                  /
       032104  020040  020040  020040
       032112  020040  020040  020040
       032120  020040  020040
 2563  032124  047116  047116  020040  STLIN3: .ASCIZ  /NNNN          /
       032132  020040  020040  020040
       032140  000040
 2564
 2565
 2566  032142  051105  047522  020122  SMTTLE: .ASCIZ  /ERROR NUMBER    ERROR ADDRESS    NO. OF OCCURRENCES/
       032150  052516  041115  051105
       032156  020040  020040  051105
       032164  047522  020122  042101
       032172  051104  051505  020123
       032200  020040  047040  027117
       032206  047440  020106  041517
       032214  052503  051122  047105
       032222  042503  000123
 2567
 2568  032226  020040  025052  047040  NOERMG: .ASCIZ  /  ** NO ERRORS TO REPORT YET **/
       032234  020117  051105  047522
```

```
                032242  051522  052040  020117
                032250  042522  047520  052122
                032256  054440  052105  025040
                032264  000052
 2569           032266  051124  047101  046523  TERHDR: .ASCIZ  /TRANSMITTER ERRORS:/
                032274  052111  042524  020122
                032302  051105  047522  051522
                032310  000072
 2570           032312  051105  047522  020122  TRHLIN: .ASCIZ  /ERROR NO.    CONCTD RCVR    ERROR COUNT/
                032320  047516  020056  020040
                032326  041440  047117  052103
                032334  020104  041522  051126
                032342  020040  020040  051105
                032350  047522  020122  047503
                032356  047125  000124
 2571           032362                          TRELIN:
 2572           032362  047116  047116  020116  TRRNO:  .ASCII  /NNNNN        /
                032370  020040  020040  020040
                032376     040
 2573           032377     116  047116  047116  TRRCN:  .ASCII  /NNNNN          /
                032404  020040  020040  020040
                032412  020040  020040
 2574           032416  047116  047116  020040  TRERC:  .ASCIZ  /NNNN   /
                032424  000040
 2575           032426  042522  042503  053111  RERHDR: .ASCIZ  /RECEIVER ERRORS:/
                032434  051105  042440  051122
                032442  051117  035123     000
 2576           032447     105  051122  051117  RRHLIN: .ASCIZ  /ERROR NO.    CONCTD XMTR    ERROR COUNT/
                032454  047040  027117  020040
                032462  020040  047503  041516
                032470  042124  054040  052115
                032476  020122  020040  042440
                032504  051122  051117  041440
                032512  052517  052116     000
 2577           032517                          RCELIN:
 2578           032517     116  047116  047116  RCRNO:  .ASCII  ./NNNNN        /
                032524  020040  020040  020040
                032532  020040
 2579           032534  047116  047116  020116  RCTRN:  .ASCII  /NNNNN          /
                032542  020040  020040  020040
                032550  020040     040
 2580           032553     116  047116  020116  RCERC:  .ASCIZ  /NNNN   /
                032560  020040     000
 2581           032563     040  024040  047516  NONMG:  .ASCIZ  /  (NONE)/
                032570  042516  000051
 2582           032574                          SMLIN:
 2583           032574  047116  047116  020040  SMLIN1: .ASCII  /NNNN              /
                032602  020040  020040  020040
                032610  020040  020040  020040
 2584           032616  047116  047116  020040  SMLIN2: .ASCII  /NNNN                /
                032624  020040  020040  020040
                032632  020040  020040  020040
                032640  020040
 2585           032642  047116  047116  020040  SMLIN3: .ASCIZ  /NNNN    /
                032650  020040     000
 2586
```

```
2587
```

```
2589            032654                          .EVEN
2590                                            .SBTTL AUXILIARY ROUTINES
2591                                            .SBTTL  CHARACTER PROCESSOR
2592
2593                                    ;  CONVERT A CHARACTER TO ITS PROCESSING ROUTINE ADDRESS BASED
2594                                    ;  UPON A TABLE OF ENTRIES IN THE FOLLOWING FORM:
2595                                    ;                          I--------------I--------------I
2596                                    ;                          I   (UNUSED)   I  CHARACTER   I
2597                                    ;                          I--------------I--------------I
2598                                    ;                          I   PROCESSING ROUTINE ADDR   I
2599                                    ;                          I--------------I--------------I
2600
2601                                    ;  THE TABLE MUST BE ARRANGED IN ASCENDING ORDER OF CHARACTER VALUES.
2602
2603                                    ;  THE TABLE ENDS WITH A DUMMY ENTRY FOR CHARACTER FF (HEXADECIMAL).
2604
2605                                    ;  ON ENTRY:    R0 - CHARACTER
2606                                    ;               R1 = TABLE ADDRESS
2607                                    ;
2608                                    ;  CALL:        JSR     PC,PROCHR
2609                                    ;
2610                                    ;  ON RETURN:   R0 - CHARACTER
2611                                    ;               R1 - PROCESSING ROUTINE ADDRESS, IF ANY
2612                                    ;               Z  = 1 -=> CHARACTER NOT IN TABLE
2613
2614  032654                            PROCHR:                                 ;**ENTRY POINT**
2615  032654  121100                    PCLOOK: CMPB    @R1,R0                  ;COMPARE TABLE CHAR
2616                                                                            ; WITH ARG. CHAR.
2617  032656  001404                            BEQ     PCCALL                  ;IF SAME, RETURN PROC.
2618                                                                            ; ROUTINE ADDR.
2619  032660  101006                            BHI     PCQUIT                  ;IF >, THEN ARG. CHAR
2620                                                                            ; NOT IN TABLE.
2621  032662  062701  000004                    ADD     #4,R1                   ;IF <, POINT TO NEXT
2622                                                                            ; TABLE ENTRY.
2623  032666  000772                            BR      PCLOOK                  ;TRY AGAIN.
2624
2625  032670  016101  000002            PCCALL: MOV     2(R1),R1                ;R1 = PROCESSING
2626                                                                            ; ROUTINE ADDRESS.
2627  032674  000207                            RTS     PC                      ;RETURN: Z BIT IS
2628                                                                            ; OFF.
2629
2630  032676  005001                    PCQUIT: CLR     R1                      ;R1 = 0
2631  032700  000207                            RTS     PC                      ;RETURN: Z BIT IS ON.
2632
```

```
                        .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 SPECIFYIN  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 COMPRESS 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
```

```
2677                            ;INPUTS:
2678                            ;
2679                            ;       R0 ADDRESS TO STORE FIRST BYTE IN OUTPUT STRING
2680                            ;       R1 NUMBER TO BE CONVERTED
2681                            ;       R2=ZERO COMPRESSION INDICATOR
2682                            ;               IF R2 EQ 0, THEN SUPPRESS ZEROS
2683                            ;               IF R2 NE 0, THEN DO NOT SUPPRESS ZEROS.
2684                            ;
2685                            ;IF CBTA IS CALLED, THEN R2 MUST CONTAIN THE FOLLOWING INFORMATION
2686                            ;
2687                            ;       LOW BYTE-CONVERSION RADIX (2-10.)
2688                            ;       BIT 8=MAGNITUDE/SIGNED CONVERSION (1 SIGNED)
2689                            ;
2690                            ;       BIT 9 =ZERO COMPRESS FLAG (0-COMPRESS LEADING ZEROS)
2691                            ;
2692                            ;       BIT 10=BLANK FILL FLAG (1=REPLACE LEADING ZEROS WITH BLANKS
2693                            ;               IF ZERO COMPRESS DISABLED, 0-ZERO FILL).
2694                            ;
2695                            ;       BITS 11-15=FIELD WIDTH (1-32)
2696                            ;
2697                            ;
2698                            ;OUTPUTS:
2699                            ;
2700                            ;       R0-ADDRESS OF NEXT BYTE AFTER LAST DIGIT STORED.
2701                            ;
2702                            ;       IF THE CONVERTED DIGIT EXCEEDS 9, THE RESULT IS BIASED TO FALL
2703                            ;       IN THE RANGE A - Z
2704                            ;
2705
2706                            ;CONVERT 6 DIGIT OCTAL TO ASCII MAGNITUDE
2707
2708  032702                    OCTJSP:
2709  032702                            STCVT   8.,6.,MAGN,NOCOMP,BLANKS        ;PUSH CONVERSION PARAMETERS
2710  032706  000411                    BR      SETCN                   ;CONVERT TO ASCII
2711
2712                            ;CONVERT 6 DIGIT OCTAL TO ASCII (ZERO COMPR)
2713
2714  032710                    OCTPNT:
2715  032710                            STCVT   8.,6.,MAGN,COMPRES,NOBLKANK
2716  032714  000406                    BR      SETCN
2717
2718                            ;CONVERT 5 DIGIT DECIMAL TO ASCII MAGNITUDE
2719
2720  032716                    DECJSP:
2721  032716                            STCVT   10.,5.,MAGN,NOCOMP,BLANKS
2722  032722  000403                    BR      SETCN
2723
2724                            ;CONVERT 5 DIGIT DECIMAL TO ASCII (ZERO COMPR)
2725
2726  032724                    DECPNT:
2727  032724                            STCVT   10.,5.,MAGN,COMPRES,NOBLANK
2728  032730  000400                    BR      SETCN
2729
2730  032732                    SETCN:
2731  032732  005702                    TST     R2                      ;SUPPRESS ZEROS?
2732  032734  001002                    BNE     20$                     ;IF NE, NO
```

```
 2733  032736  042716  001000            BIC     #1*'000,(SP)            ;ENABLE ZERO SUPPRESS
 2734  032742                      20$:
 2735  032742  012602                    MOV     (SP)+,R2                ;SET CONTROL WORD
```

```
2737
2738
2739  032744                          CBTA:
2740  032744  004567  000406                JSR     R5,SAVRG                ;SAVE THE NON-VOLATILE REGISTERS
2741  032750  110205                        MOVB    R2,R5                   ;COPY RADIX BYTE
2742  032752  000302                        SWAB    R2                      ;POSITION REMAINING TO LOW BYTE
2743  032754  106202                        ASRB    R2                      ;SHIFT OFF MAG. FLAG
2744  032756  103005                        BCC     10$                     ;UNSIGNED IF C IS CLR
2745  032760  005701                        TST     R1                      ;POSITIVE VALUE?
2746  032762  100003                        BPL     10$                     ;IF PL, YES
2747  032764  005401                        NEG     R1                      ;MAKE VALUE POSITIVE
2748  032766  112720  000055                MOVB    #'-,(R0)+               ;INSERT A MINUS SIGN
2749  032772                          10$:
2750  032772  010004                        MOV     R0,R4                   ;COPY STRING POINTER
2751  032774  000241                        CLC                             ;CLEAR CARRY
2752  032776  106002                        RORB    R2                      ;SHIFT OFF SUPPR FLAG
2753  033000  006002                        ROR     R2                      ;TRANSFER TO R2
2754  033002  006003                        ROR     R3                      ;GET BLANK/ZERO PAD FLAG
2755  033004  105003                        CLRB    R3                      ;CLEAR COUNT BYTE
2756  033006  150203                        BISB    R2,R3                   ;TRANSFER COUNT BYTE
2757  033010  105002                        CLRB    R2                      ;CLEAR FILL BYTE
2758  033012  152702  000060                BISB    #'0,R2                  ;SET FILL BYTE
2759  033016  010100                        MOV     R1,R0                   ;DIVIDEND TO R0
2760  033020                          1$:
2761  033020  010501                        MOV     R5,R1                   ;SET CONVERSION RADIX
2762  033022  004767  000272                JSR     PC,DIV                  ;DIVIDE EM UP
2763  033026  020127  000011                CMP     R1,#9.                  ;RESULT EXCEED NUMERICS?
2764  033032  101402                        BLOS    15$                     ;IF LOS, NO
2765  033034  062701  000007                ADD     #7,R1                   ;BIAS TO FALL IN ALPHA
2766  033040                          15$:
2767  033040  060201                        ADD     R2,R1                   ;ADD CHARACTER BIAS
2768  033042  010146                        MOV     R1,-(SP)                ;SAVE CHARACTER
2769  033044  105303                        DECB    R3                      ;DECREMENT CHARACTER COUNT
2770  033046  003412                        BLE     3$                      ;IF LE NO DIGITS LEFT
2771  033050  005700                        TST     R0                      ;ZERO QUOTIENT
2772  033052  001006                        BNE     2$                      ;IF NE, YES, GO AGAIN
2773  033054  005702                        TST     R2                      ;SUPPRESS ZEROS
2774  033056  100006                        BPL     3$                      ;IF PL, YES, ALL DONE
2775  033060  005703                        TST     R3                      ;SUBSTITUTE BLANKS?
2776  033062  100002                        BPL     2$                      ;IF PL, NO
2777  033064  042702  000020                BIC     #20,R2                  ;CONVERT FILL TO BLANK
2778  033070                          2$:
2779  033070  004767  177724                JSR     PC,1$                   ;DIVIDE AGAIN
2780  033074                          3$:
2781  033074  112624                        MOVB    (SP)+,(R4)+             ;STORE A DIGIT
2782  033076  010400                        MOV     R4,R0                   ;STORE TERMINAL ADDRESS
2783  033100                                RETURN
2784
```

```
                        .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

033102                  CDDMG:
033102  004567  000250          JSR     R5,SAVRG                ;SAVE THE NON-VOLATILE REGISTERS
033106  010003                  MOV     R0,R3                   ;COPY THE STRING POINTER
033110  012704  023420          MOV     #10000.,R4              ;SET DIVISOR
033114                          CBTAS   10.,0,NOSIGN,BLANKS     ;SET CONVERSION PARAMETERS
033130  022104                  CMP     (R1)+,R4                ;TEST FOR OVERFLOW
033132  103042                  BHIS    40$                     ;IF HIS, OVERFLOW
033134  011102                  MOV     (R1),R2                 ;GET LOW PART OF NUMBER
033136  014101                  MOV     -(R1),R1                ;GET HIGH PART OF NUMBER
033140  010400                  MOV     R4,R0                   ;COPY DIVISOR
033142  004767  000110          JSR     PC,DDIV                 ;DO DOUBLE PREC. DIVIDE
033146  010046                  MOV     R0,-(SP)                ;SAVE REMAINDER
033150  010201                  MOV     R2,R1                   ;COPY QUOTIENT
```

```
2842  033152  001011                    BNE     11$                     ;IF NE, SOMETHING TO PRINT
2843  033154  012702  000005            MOV     #5,R2                   ;OTHERWISE, FILL FIELD WITH BLANKS
2844  033160  112723  000040    21$:    MOVB    #' ,(R3)+
2845  033164  005302                    DEC     R2
2846  033166  001374                    BNE     21$
2847  033170  052705  003000            BIS     #3000,R5                ;DISABLE BLANK SUPPRESSION
2848  033174  000411                    BR      20$
2849  033176  012702  024000    11$:    MOV     #5*4000,R2              ;SET FIELD WIDTH
2850  033202  004767  000020            JSR     PC,30$                  ;OUTPUT HIGH ORDER DIGITS
2851  033206  052705  001000            BIS     #1*1000,R5              ;DISABLE ZERO COMPRESS
2852  033212  042705  002000            BIC     #1*2000,R5              ;DISABLE BLANKS
2853  033216  010003            31$:    MOV     R0,R3                   ;SET STRING POINTER
2854  033220                    20$:
2855  033220  012601                    MOV     (SP)+,R1                ;GET LOW ORDER VALUE
2856  033222  012702  020000            MOV     #4*4000,R2              ;SET FIELD WIDTH
2857  033226                    30$:
2858  033226  010300                    MOV     R3,R0                   ;GET STRING POINTER
2859  033230  050502                    BIS     R5,R2                   ;INCLUDE RADIX & BLANK SUPPRESS
2860  033232  004767  177506            JSR     PC,CBTA                 ;CONVERT TO ASCII
2861  033236  000406                    BR      60$                     ;EXIT
2862  033240                    40$:
2863  033240  012702  000011            MOV     #9.,R2                  ;GET COUNT
2864  033244                    50$:
2865  033244  112720  000052            MOVB    #'*,(R0)+               ;FILL FIELD WITH ASTERISKS
2866  033250  005302                    DEC     R2
2867  033252  001374                    BNE     50$                     ;   ''SOB     R2,50$''
2868  033254                    60$:
2869  033254                            RETURN
2870
2871
2872
2873                                      .SBTTL  DOUBLE PRECISION DIVIDE ROUTINE
2874                                    ;
2875                                    ; INPUTS:
2876                                    ;
2877                                    ; R2=LOW ORDER OF DIVIDEND
2878                                    ; R1-HIHG ORDER OF DIVIDEND
2879                                    ; R0-DIVISOR (15 BITS UNSIGNED)
2880                                    ;
2881                                    ; OUTPUTS:
2882                                    ;
2883                                    ; R2=LOW ORDER OF QUOTIENT
2884                                    ; R1=HIGH ORDER OF QUOTIENT
2885                                    ; R0=REMAINDER
2886                                    ;
2887
2888  033256                    DDIV:
2889  033256  010346                            MOV     R3,-(SP)                ;SAVF R3
2890  033260  012703  000040                    MOV     #32.,R3                 ;SET ITERATION COUNT IN R3
2891  033264  010046                            MOV     R0,-(SP)                ;STACK DIVISOR
2892  033266  005000                            CLR     R0                      ;SET REMAINDER TO 0
2893  033270                            1$:
2894  033270  006302                            ASL     R2                      ;SHIFT THE ENTIRE DIVIDEND
2895  033272  006101                            ROL     R1                      ;... ONE BIT TO THE LEFT AND...
2896  033274  006100                            ROL     R0                      ;... INTO THE REMAINDER
2897  033276  020016                            CMP     R0,(SP)                 ;IS REMAINDER GE DIVISOR?
```

```
2898  033300  103402                  BLO     2$                      ;NO, SKIP TO ITERATION CONTROL
2899  033302  161600                  SUB     (SP),R0                 ;YES, SUB DIVISOR OUT
2900  033304  005202                  INC     R2                      ;AND INCR THE QUOTIENT
2901  033306                  2$:
2902  033306  005303                  DEC     R3                      ;REPEAT AS LONG AS NECESSARY
2903  033310  003367                  BGT     1$
2904  033312  005726                  TST     (SP)+                   ;PURGE DIVISOR FROM STACK
2905  033314  012603                  MOV     (SP)+,R3                ;RESTORE R3
2906  033316                          RETURN
2907
2908
2909                                  .SBTTL  INTEGER DIVIDE MAGNITUDE NUMBERS
2910                  ;
2911                  ;
2912                  ; INPUTS:
2913                  ;
2914                  ; R0=DIVIDEND
2915                  ; R1=DIVISOR
2916                  ;
2917                  ; OUTPUTS:
2918                  ;
2919                  ; QUOTIENT IS RETURNED IN R0
2920                  ; REMAINDER IS RETURNED IN R1
2921                  ;
2922                  ;
2923  033320          DIV:
2924  033320  012746  000020          MOV     #20,-(SP)               ;SET LOOP COUNT
2925  033324  010146                  MOV     R1,-(SP)                ;SAVE DIVISOR FOR SUBTRACTS
2926  033326  005001                  CLR     R1                      ;CLEAR REMAINDER
2927  033330                  30$:
2928  033330  006300                  ASL     R0                      ;DOUBLE LEFT SHIFT.
2929  033332  006101                  ROL     R1
2930  033334  020116                  CMP     R1,(SP)                 ;SUBTRACT OUT DIVISOR
2931  033336  103402                  BLO     40$                     ;IF LO, NO
2932  033340  161601                  SUB     (SP),R1                 ;SUBTRACT OUT DIVISOR
2933  033342  005200                  INC     R0                      ;ADD IN LOW BIT
2934  033344                  40$:
2935  033344  005366  000002          DEC     2(SP)                   ;DECREMENT REPEAT COUNT
2936  033350  003367                  BGT     30$                     ;IF GT, MORE TO GO
2937  033352                  50$:
2938  033352  022626                  CMP     (SP)+,(SP)+             ;CLEAN STACK
2939  033354                          RETURN
2940
2941                  ; SAVE/RESTORE NONVOLATILE REGISTERS
2942
2943
2944  033356  010446          SAVRG:  MOV     R4,-(SP)                ;SAVE R4 &R3
2945  033360  010346                  MOV     R3,-(SP)
2946  033362  010546                  MOV     R5,-(SP)                ;PUT RETURN ADDRESS ON STACK
2947  033364  016605  000006          MOV     6(SP),R5                ;RETRIEVE REAL R5
2948  033370  004736                  JSR     PC,@(SP)+               ;CALL THE CALLER ...
2949  033372  012603                  MOV     (SP)+,R3                ;RESTORE NON VOLATILE REGISTERS
2950  033374  012604                  MOV     (SP)+,R4
2951  033376  012605                  MOV     (SP)+,R5
2952  033400                          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>
 (3)            000002          ITEM    -       2
 (3)            000004          QUEUE           4
 (1)
 (1)  033402                    ENQ:                                    ;**ENTRY POINT**
                                ;
                                ;  APPEND  ITEM  (A WORD) TO THE FIRST-IN-FIRST-OUT LIST  QUEUE .
                                ;
 033402  016504  000004                 MOV     QUEUE(R5),R4            ;GET QUEUE ADDRESS.
 033406  021464  000002                 CMP     @R4,QSIZE(R4)           ;IF QUEUE IS FULL,
 033412  002020                         BGE     NQFULL                  ; SIGNAL TRAGIC ERROR.
 033414  017574  000002  000012        MOV     @ITEM(R5),@QBACK(R4)    ;PUT ITEM AT BACK OF QUEUE.
 033422  062764  000002  000012        ADD     #2,QBACK(R4)            ;UPDATE BACK POINTER.
 033430  026464  000012  000006        CMP     QBACK(R4),QBOT(R4)      ; ''
 033436  103403                         BLO     NQNOWP                  ; ''
 033440  016464  000004  000012        MOV     QTOP(R4),QBACK(R4)      ; ''
 033446  005214                 NQNOWP: INC     @R4                     ;INCREMENT NO. OF ELEMENTS.
 033450  000241                         CLC                             ;INDICATE SUCCESSFUL ENQ.
 033452                         NQRET:  RETURN                          ;RETURN.
 (1)  033452  000207                    RTS     PC
 033454  000261                 NQFULL: SEC                             ;INDICATE UNSUCCESSFUL ENQ.
 033456  000775                         BR      NQRET                   ;IGNORE ITEM & RETURN.


                                        PROC    DEQ,<ITEM,QUEUE>
 (3)            000002          ITEM    -       2
 (3)            000004          QUEUE   =       4
 (1)
 (1)  033460                    DEQ:                                    ;**ENTRY POINT**
                                ;
                                ;  REMOVE A WORD ENTRY FROM THE FIRST-IN-FIRST-OUT LIST  QUEUE  &
                                ;  STORE IT AT  ITEM .
                                ;
 033460  016504  000004                 MOV     QUEUE(R5),R4            ;GET QUEUE ADDRESS.
 033464  005714                         TST     @R4                     ;IF QUEUE IS EMPTY,
 033466  001420                         BEQ     DQEMP                   ; SIGNAL TRAGIC ERROR.
 033470  017475  000010  000002        MOV     @QFRONT(R4),@ITEM(R5)   ;RETRIEVE FRONT ELEMENT.
 033476  062764  000002  000010        ADD     #2,QFRONT(R4)           ;UPDATE FRONT POINTER.
 033504  026464  000010  000006        CMP     QFRONT(R4),QBOT(R4)     ; ''
 033512  103403                         BLO     DQNOWP                  ; ''
 033514  016464  000004  000010        MOV     QTOP(R4),QFRONT(R4)     ; ''
 033522  005314                 DQNOWP: DEC     @R4                     ;DECREMENT NO. OF ELEMENTS.
 033524  000241                         CLC                             ;INDICATE SUCCESSFUL DEQ.
 033526                         DQRET:  RETURN                          ;RETURN.
 (1)  033526  000207                    RTS     PC

 033530  000261                 DQEMP:  SEC                             ;INDICATE UNSUCCESSFUL DEQ.
 033532  012775  177777  000002        MOV     #-1,@ITEM(R5)           ;SET ITEM TO ALL ONES.
```

```
  3000  033540  000772                        BR      DQRET                   ;RETURN.
```

```
3002                            ; SUBROUTINE TO SCAN AN INPUT COMMAND & CALL ITS
3003                            ; PROCESSING ROUTINE.
3004
3005                                    .LIST   MEB
3006
3007
3008                                    .MACRO  SPAN    REG,CHAR,?L
3009
3010                            ;  THIS MACRO SCANS THE STRING OF CHARACTERS STARTING AT
3011                            ;  @REG UNTIL IT FINDS ONE NOT EQUAL TO CHAR.  REG IS
3012                            ;  SET POINTING TO THAT CHARACTER.
3013
3014                            L:      CMPB    (REG)+,CHAR
3015                                    BEQ     L
3016                                    DEC     REG
3017
3018                                    .ENDM
3019
3020                                    .MACRO  BREAK   REG,CHRSET,?HH,?JJ
3021
3022                            ;  THIS MACRO SCANS THE STRING STARTING AT @REG UNTIL
3023                            ;  IT FINDS A CHARACTER THAT IS A MEMBER OF CHRSET.
3024                            ;  REG IS SET POINTING TO THAT CHARACTER.
3025
3026                            ;  EACH MEMBER OF CHRSET IS AN ADDRESSABLE QUANTITY.
3027
3028
3029                            HH:
3030                                    .IRP    LS,<CHRSET>
3031                                    CMPB    @REG,LS
3032                                    BEQ     JJ
3033                                    .ENDM
3034                                    INC     REG
3035                                    BR      HH
3036
3037                            JJ:
3038                                    .ENDM
3039
3040
3041                                    .MACRO  SYNCLS  CHAR,CLASS,?CC,?DD,?EE,?FF,?GG
3042
3043                            ;  THIS MACRO DETERMINES THE SYNTACTIC CLASS OF AN
3044                            ;  OBJECT BEGINNING WITH CHAR.  THE CLASS IS RETURNED
3045                            ;  IN CLASS AS FOLLOWS:
3046                            ;       CLASS   =       0 (WORD) IF CHAR - (A,...,Z,-)
3047                            ;                       2(NUMBER)IF CHAR = (0,...,9)
3048                            ;                       6(END OF LINE) IF CHAR - CARRIAGE RETURN
3049                            ;                       4(CHARACTER STRING) OTHERWISE
3050
3051                                    CMPB    CHAR,#'A
3052                                    BLT     EE
3053                                    CMPB    CHAR,#'Z
3054                                    BGT     DD
3055                            CC:     CLR     CLASS
3056                                    BR      GG
3057
```

```
3058                            DD:     MOV     #4,CLASS
3059                                    BR      GG
3060
3061                            EE:     CMPB    CHAR,#'0
3062                                    BLT     FF
3063                                    CMPB    CHAR,#'9
3064                                    BGT     DD
3065                                    MOV     #2,CLASS
3066                                    BR      GG
3067
3068                            FF:     CMPB    CHAR,#'-
3069                                    BEQ     CC
3070                                    CMPB    CHAR,#15
3071                                    BNE     DD
3072                                    MOV     #6,CLASS
3073                            GG:
3074
3075                                    .ENDM
```

```
                        .SBTTL  COMMAND PROCESSOR INITIATING ROUTINE

                        PROC    COMAND,<INPLIN,KWTABL>
(3)             000002  INPLIN  =       2
(3)             000004  KWTABL  =       4
(1)
(1)  033542             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.
                        ;
033542  010546                 MOV     R5,-(SP)        ;SAVE PAR LIST POINTER.
033544  010667  000076         MOV     SP,CMMARK       ;SAVE STACK POINTER.
033550                  .       LXSCAN  INPLIN(R5)      ;LEXICALLY SCAN  INPLIN .
(1) 033550  016501  000002     MOV     INPLIN(R5),R1
(1) 033554  004167  000310     JSR     R1,LXSCAN
033560  103426                 BCS     CMRET           ;IF ERROR, RETURN WITH C 1.
033562  005716                 TST     @SP             ;HAVE WE ANY OBJECTS ?
033564  003005                 BGT     1$              ;NO, IGNORE BLANK COMMAND.
033566  012767  177777  164162       MOV     #-1,ESCFLG
033574  000167  000036         JMP     CMRET
033600  016700  000042   1$:   MOV     CMMARK,R0       ;YES, DETERMINE ADDRESS
033604  162700  000006         SUB     #6,R0           ; OF FIRST OBJECT.
033610  017705  000032         MOV     @CMMARK,R5      ;RESTORE PAR LIST POINTER.
033614                  .       KEYWD   R0,KWTABL(R5)   ;GET ADDR OF COMMAND PROCESSOR.
(1) 033614  010046             MOV     R0,-(SP)
(1) 033616  016546  000004     MOV     KWTABL(R5),-(SP)
(1) 033622  004767  000024     JSR     PC,KEYWD
033626  103403                 BCS     CMRET           ;IF INVALID COMMAND, MAKE
                                                        ; ERROR RETURN.
033630  012705  033650         MOV     #NULPAR,R5      ;LOAD NULL PAR LIST ADDRESS.
033634  004736                 JSR     PC,@(SP)+       ;PROCESS COMMAND & POP ADDR.
033636  016706  000004   CMRET: MOV     CMMARK,SP       ;RESTORE STACK TO ENTRY STATUS.
033642  030026                 BIT     R0,(SP)+        ;THROW AWAY SAVED R5 (WITHOUT
                                                        ; AFFECTING C BIT).
033644                  .       RETURN                  ;RETURN TO CALLING PROGRAM.
(1) 033644  000207             RTS     PC
                        ; DATA AREAS.
033646  033333         CMMARK: .WORD   33333           ;STORAGE FOR STACK PTR ON ENTRY.
033650  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

033652                          KEYWD:                                  ;**ENTRY POINT**
033652                                  REGSAV                          ;SAVE REGISTERS.
(1) 033652  004567  000160              JSR     R5,REGSAV
033656  162706  000006              SUB     #6,SP                   ;ALLOCATE STACK SPACE
033662  016602  000026              MOV     SRC(SP),R2              ;GET ADDR OF INPUT
                                                                        ; OBJECT
033666  005722                      TST     (R2)+                   ;IF OBJECT NOT A WORD
033670  001051                      BNE     NOTHER                  ; THEN EXIT: NOT FOUND
033672  012205                      MOV     (R2)+,R5                ;GET # CHAR IN OBJECT.
033674  010566  000004              MOV     R5,LENINP(SP)           ;STORE ON STACK.
033700  011204                      MOV     @R2,R4                  ;GET ADDR OF INPUT WORD.
033702  010466  000002              MOV     R4,ADRINP(SP)           ;STORE ON STACK.
033706  016603  000024              MOV     TABLAD(SP),R3           ;GET ADDR. OF KEYWORD
                                                                        ; TABLE.
033712  012316              GTLENS: MOV     (R3)+,@SP               ;SAVE ROUTINE ADDR OF
                                                                        ; 1ST ELEMENT.
033714  112302                      MOVB    (R3)+,R2                ;GET MINIMUM LENGTH
033716  042702  177400              BIC     #177400,R2              ; OF TABLE WORD.
033722  112301                      MOVB    (R3)+,R1                ;GET FULL LENGTH.
033724  042701  177400              BIC     #177400,R1              ;
033730  001431                      BEQ     NOTHER                  ;IF 0, THEN NO MORE
                                                                        ; TABLE TO SEARCH
033732  122423              NXTCH:  CMPB    (R4)+,(R3)+             ;COMPARE INPUT CHAR
                                                                        ; WITH TABLE CHAR.
```

```
3177  033734  002427                  BLT     NOTHER                  ;IF<, THEN NO MATCH
3178                                                                  ; EXISTS.
3179  033736  003005                  BGT     NXTWD                   ;IF>, THEN TRY NEXT
3180                                                                  ; TABLE WORD.
3181  033740  005305                  DEC     R5                      ;IF INPUT STRING
3182                                                                  ; EXHAUSTED,
3183  033742  001413                  BEQ     THFND                   ; WE MAY HAVE FOUND
3184                                                                  ; MATCH.
3185  033744  005301                  DEC     R1                      ;IF MORE CHAR IN TABLE WORD
3186  033746  001371                  BNE     NXTCH                   ; TO TEST, GO & TEST THEM.
3187  033750  005201                  INC     R1
3188  033752  060103          NXTWD:  ADD     R1,R3                   ;GET ADDR OF NEXT
3189  033754  042703  000001        BIC     #1,R3                   ; TABLE ENTRY.
3190  033760  016604  000002        MOV     ADRINP(SP),R4           ;POINT TO BEGINNING
3191                                                                  ; OF INPUT WORD.
3192  033764  016605  000004        MOV     LENINP(SP),R5           ;GET LENGTH OF INPUT WORD.
3193  033770  000750                  BR      GTLENS                  ;GET LENGTHS OF TABLE WORD.
3194
3195  033772  026602  000004  THFND: CMP     LENINP(SP),R2           ;IF LEN(INP.WD) < MIN LEN (TABLE
3196  033776  002406                  BLT     NOTHER                  ; WORD), WORD IS NOT IN TABLE.
3197  034000  011666  000026        MOV     (SP),RESULT(SP)         ;SAVE ROUTINE ADDR. OF
3198                                                                  ; MATCH.
3199  034004  062706  000006        ADD     #6,SP                   ;FREE LOCAL STACK SPACE.
3200  034010  000241                  CLC                             ;CLEAR CARRY BIT.
3201  034012  000405                  BR      KWEXIT
3202
3203                          ;  WORD IS NOT IN TABLE. SET RESULT TO 0 & SET Z BIT ON.
3204  034014  005066  000026  NOTHER: CLR     RESULT(SP)              ;CLEAR RESULT.
3205  034020  062706  000006        ADD     #6,SP                   ;FREE LOCAL STACK SPACE.
3206  034024  000261                  SEC                             ;SET CARRY BIT.
3207  034026                  KWEXIT: REGRES                          ;RESTORE REGISTERS.
 (1)  034026  004567  000020        JSR     R5,REGRES
3208  034032  012616                  MOV     (SP)+,@SP               ;POP AN ARGUMENT.
3209  034034  000207                  RTS     PC                      ;RETURN.
```

```
                        .SBTTL  REGISTER SAVE & RESTORE ROUTINES

                ;  SUBROUTINE TO SAVE R0 - R5 ON STACK
                ;  CALLING SEQUENCE:    JSR     R5,REGSAV
034036          REGSAV:                                 ;**ENTRY POINT**
034036  010446          MOV     R4,-(SP)
034040  010346          MOV     R3,-(SP)
034042  010246          MOV     R2,-(SP)
034044  010146          MOV     R1,-(SP)
034046  010046          MOV     R0,-(SP)
034050  000115          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
034052          REGRES:                                 ;**ENTRY POINT**
034052  030026          BIT     R0,(SP)+                ;THROW AWAY OLD R5 VALUE.
034054  012600          MOV     (SP)+,R0
034056  012601          MOV     (SP)+,R1
034060  012602          MOV     (SP)+,R2
034062  012603          MOV     (SP)+,R3
034064  012604          MOV     (SP)+,R4
034066  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):
                ;       I--------------I
                ;       I       N      I <--- SP
                ;       I--------------I
                ;       I  NTH OBJECT  I
                ;       I--------------I
                ;       :       :      :
                ;       I--------------I
                ;       I  1ST OBJECT  I
                ;       I--------------I
                ;  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.
```

```
3294
3295                                    ;  IF LXSCAN DETECTS AN ERROR CONDITION, IT RETURNS WITH THE CARRY (C)
3296                                    ;  BIT SET; OTHERWISE IT IS CLEAR.  AT THE MOMENT, THE
3297                                    ;  ONLY ERROR CONDITION DETECTED BY LXSCAN IS A STRING WHICH IS MISSING
3298                                    ;  ITS CLOSING DELIMITER.
3299
3300  034070                            LXSCAN:                         ;**ENTRY POINT**
3301  034070  005002                           CLR     R2                ;CLEAR #OBJECTS.
3302  034072  012605                           MOV     (SP)+,R5          ;GET ADDR. OF COMMAND BUFFER.
3303  034074                            NXTEL:  SPAN    R5,<#' >          ;R5 = ADDR (1ST NONBLANK CHAR)
 (1)  034074  122527  000040          64$:    CMPB    (R5)+,#'
 (1)  034100  001775                           BEQ     64$
 (1)  034102  005305                           DEC     R5
3304  034104                                  SYNCLS  @R5,R4            ;DETERMINE SYNTACTIC CLASS.
 (1)  034104  121527  000101                   CMPB    @R5,#'A
 (1)  034110  002410                           BLT     67$
 (1)  034112  121527  000132                   CMPB    @R5,#'Z
 (1)  034116  003002                           BGT     66$
 (1)  034120  005004                    65$:    CLR     R4
 (1)  034122  000424                           BR      69$
 (1)  034124  012704  000004            66$:    MOV     #4,R4
 (1)  034130  000421                           BR      69$
 (1)  034132  121527  000060            67$:    CMPB    @R5,#'0
 (1)  034136  002406                           BLT     68$
 (1)  034140  121527  000071                   CMPB    @R5,#'9
 (1)  034144  003367                           BGT     66$
 (1)  034146  012704  000002                   MOV     #2,R4
 (1)  034152  000410                           BR      69$
 (1)  034154  121527  000055            68$:    CMPB    @R5,#'-
 (1)  034160  001757                           BEQ     65$
 (1)  034162  121527  000015                   CMPB    @R5,#15
 (1)  034166  001356                           BNE     66$
 (1)  034170  012704  000006                   MOV     #6,R4
3305  034174  000174  034200                   JMP     @SJT(R4)          ;PROCESS OBJECT
3306
3307  034200  034210                    SJT:    .WORD   SCWORD            ;WORD PROCESSOR.
3308  034202  034250                            .WORD   SCNO              ;NUMBER PROCESSOR.
3309  034204  034414                            .WORD   SCCHAR            ;CHARACTER STRING PROCESSOR.
3310  034206  034454                            .WORD   SCEOL             ;END OF LINE PROCESSOR.
3311
3312  034210  010503                    SCWORD: MOV     R5,R3             ;R3 = START ADDR OF OBJECT
3313  034212                                    BREAK   R5,<<#' >,<#15>>  ;R5 - ADDR (NEXT BLANK OR CR)
 (2)  034212  121527  000040                   CMPB    @R5,#'
 (2)  034216  001405                           BEQ     65$
 (2)  034220  121527  000015                   CMPB    @R5,#15
 (2)  034224  001402                           BEQ     65$
 (1)  034226  005205                           INC     R5
 (1)  034230  000770                           BR      64$
3314  034232  010346                           MOV     R3,-(SP)          ;PUSH ADDR OF OBJECT ONTO STACK.
3315  034234  160503                           SUB     R5,R3             ;R3 = (-LENGTH OF OBJECT)
3316  034236  005403                    PLAC:   NEG     R3                ;NEGATE TO GET LENGTH.
3317  034240  010346                           MOV     R3,-(SP)          ;PUSH LENGTH ONTO STACK.
3318  034242  010446                           MOV     R4,-(SP)          ;PUSH SYNTACTIC CLASS ONTO
3319                                                                     ; STACK
3320  034244  005202                    LXINCN: INC     R2                ;INCREMENT
3321  034246  000712                           BR      NXTEL             ;GO TO SCAN NEXT ELEMENT
```

```
3322
3323  034250  005000            SCNO:   CLR     R0              ;CLEAR ACCUMULATED NO.
3324  034252  010504                    MOV     R5,R4           ;SAVE POINTER TO 1ST DIGIT.
3325  034254  121527  000060    SNXTDG: CMPB    @R5,#'0         ;IF CHAR <'0'
3326  034260  002415                    BLT     SNTDIG          ; THEN TREAT AS DELIMITER.
3327  034262  121527  000071            CMPB    @R5,#'9         ;IF CHAR >'9'
3328  034266  003012                    BGT     SNTDIG          ; THEN TREAT AS DELIMITER.
3329  034270                            MULT    10.,R0,R3       ;MULTIPLY PREVIOUS DIGITS BY 10.
 (2)  034270  006300                    ASL     R0
 (2)  034272  010003                    MOV     R0,R3
 (2)  034274  006300                    ASL     R0
 (2)  034276  006300                    ASL     R0
 (2)  034300  060300                    ADD     R3,R0
3330  034302  112503                    MOVB    (R5)+,R3
3331  034304  142703  000060            BICB    #60,R3          ;CLEAR TOP BITS OF ASCII CODE.
3332  034310  060300                    ADD     R3,R0           ;ADD DIGIT.
3333  034312  000760                    BR      SNXTDG          ;GET NEXT DIGIT.
3334
3335  034314  121527  000040    SNTDIG: CMPB    @R5,#' '        ;IF DELIMITER = SPACE, TRY
3336  034320  001413                    BEQ     TRYOCT          ; CONVERTING OCTAL NO.
3337  034322  121527  000015            CMPB    @R5,#15         ;IF DELIMITER = CR, TRY
3338  034326  001410                    BEQ     TRYOCT          ; CONVERTING OCTAL NO.
3339  034330  122527  000056            CMPB    (R5)+,#'.       ;IF DELIMITER IS NOT DOT,
3340  034334  001052                    BNE     LXERR           ; THEN SIGNAL LXSCAN ERROR.
3341  034336  010046            PUSHNO: MOV     R0,-(SP)        ;PUT CONVERTED NO. ON STACK.
3342  034340  005046                    CLR     -(SP)           ;SET OBJECT LENGTH TO ZERO.
3343  034342  012746  000002            MOV     #2,-(SP)        ;SET OBJECT CLASS TO 2
3344                                                            ; (NUMBER).
3345  034346  000736                    BR      LXINCN          ;INCREMENT OBJ. COUNT &
3346                                                            ; SCAN NEXT OBJ.
3347
3348  034350  005000            TRYOCT: CLR     R0              ;CLEAR ACCUMULATED NO.
3349  034352  121427  000060    ONXTDG: CMPB    @R4,#'0         ;IF CHAR < '0',
3350  034356  002413                    BLT     ODELIM          ; THEN TREAT AS DELIMITER.
3351  034360  121427  000067            CMPB    @R4,#'7         ;IF CHAR > '7',
3352  034364  003010                    BGT     ODELIM          ; THEN TREAT AS DELIMITER.
3353  034366                            MULT    8.,R0           ;MULTIPLY PREVIOUS DIGITS BY 8.
 (2)  034366  006300                    ASL     R0
 (2)  034370  006300                    ASL     R0
 (2)  034372  006300                    ASL     R0
3354  034374  112403                    MOVB    (R4)+,R3        ;GET CHARACTER.
3355  034376  142703  000060            BICB    #60,R3          ;CLEAR TOP BITS OF ASCII CODE.
3356  034402  060300                    ADD     R3,R0           ;ADD DIGIT.
3357  034404  000762                    BR      ONXTDG          ;GET NEXT DIGIT.
3358
3359  034406  020405            ODELIM: CMP     R4,R5           ;IF NOT AT END OF NO.(DUE TO
3360  034410  001024                    BNE     LXERR           ; "8" OR "9") , LXSCAN ERROR.
3361  034412  000751                    BR      PUSHNO          ;GENERATE OBJECT FOR NUMBER.
3362
3363  034414  112500            SCCHAR: MOVB    (R5)+,R0        ;R0 = DELIMITER OF STRING.
3364  034416  010503                    MOV     R5,R3           ;R3 = ADDR (1ST CHAR OF
3365                                                            ; STRING ITSELF)
3366  034420                            BREAK   R5,<R0,#15>     ;R5 - ADDR (NEXT DELIM OR CR).
 (2)  034420  121500                    CMPB    @R5,R0
 (2)  034422  001405                    BEQ     65$
 (2)  034424  121527  000015            CMPB    @R5,#15
```

```
  (2)  034430  001402                    BEQ     65$
  (1)  034432  005205                    INC     R5
  (1)  034434  000771                    BR      64$
3367   034436  121527  000015            CMPB    @R5,#15         ;WAS CR FOUND BEFORE DELIMITER?
3368   034442  001407                    BEQ     LXERR           ;YES, PROCESS ERROR
3369   034444  010346                    MOV     R3,-(SP)        ;PUSH ADDR. OF STRING ONTO
3370                                                             ; STACK.
3371   034446  160503                    SUB     R5,R3           ;R3 = (-LENGTH OF STRING EXCL.
3372                                                             ; DELIMITER)
3373   034450  005205                    INC     R5              ;R5 = ADDR (CHAR AFTER CLOSING
3374                                                             ; DELIMITER).
3375   034452  000671                    BR      PLAC            ;FINISH OFF PROCESSING STRING.
3376
3377   034454  010246            SCEOL:  MOV     R2,-(SP)        ;PUSH NO. OF OBJECTS ONTO STACK
3378   034456  000241                    CLC                     ;INDICATE NO ERROR .
3379   034460  000111            LXIT:   JMP     @R1             ;RETURN TO CALLING PROGRAMME .
3380
3381                              ;  PROCESS LEXICAL SCAN ERROR .
3382   034462  000261            LXERR:  SEC                     ;INDICATE ERROR OCCURRED .
3383   034464  005302                    DEC     R2              ;CLEAN UP STACK & EXIT .
3384   034466  002774                    BLT     LXIT            ; ''
3385   034470  062706  000006            ADD     #6,SP           ; ''
3386   034474  000772                    BR      LXERR           ; ''
3387
3388
3389           000001                    .END
```

```
ADQLD   006740            503   1212#
ADRINP= 000002           3147#  3164*  3190
ADSILO  017776            566    613   2414#
AQAREA  016440           2350#
AQEND = 016540           2350#
AQQ     016424            851*   852*   853*   986*   991*   992*  1223  1232  1344  1348  2350#
AQSIZE= 000040            308#  2350
ASEXT   006304           1092   1097#
A.BKSL  016152           2180   2311#
A.C     016160           2158   2314#
A.CR    016044           2260   2272#
A.L     016162           2159   2315#
A.LF    016040           2261   2271#
A.P     016156           2157   2313#
A.PR    016164           2160   2316#
BKELEM  030266           1231*  1232   2203*  2204   2246*  2248  2467#
BSYINT  010430           1398   1460#
BUSY  = 004000            343#
B00   = 000001            364#  1646
B01     000002            363#   626    910   1343   1453   1462  1487  1513  1548  1631  1680  1690
B02   = 000004            362#
B03     000010            361#
B04   = 000020            360#   607    631    639
B05   - 000040            359#   607    639   2061
B06   - 000100            358#   467    540    867    889   1000  1382  1528  1552  1638  1644
B07   - 000200            357#  1700
B08   - 000400            356#
B09   - 001000            355#  1450   1624   1628
B10   = 002000            354#  1442   1620
B11   = 004000            353#  1436   1616
B12   = 010000            352#  1418   1601
B13   = 020000            351#  1414   1543   1597
B14   = 040000            350#  1410   1593   1627   1646
B15   - 100000            349#   466    704    865    884   915   946   959   993  1189  1193  1226  1284
                         1363   1367   1399   1440   1444  1454  1461  1486  1512  1527  1587  1626  1632
                         1642   1682   1691   1773   1832  1833  1854  1905  2107  2117
CBFEND= 016402           2199   2331#
CBTA    032744           2739#  2860
CBUFPT  016402           2081*  2161*  2174   2179*  2199  2201*  2202*  2332#
CDDMG   033102           1749   1758   2829#
CHAR  = 000002           2190#  2193
CLEAV   004106            570    639#
CLKEXT  012322           1703   1707   1711   1714#
CLKINT  012230            457   1700#
CMCHTB  016034           2063   2269#
CMDBUF  016176           1113   2161   2174   2330#
CMDENT  017641            424*   495   1111*  2082*  2162*  2357#
CMDRET  006430           1118   1120#
CMDRTN  006406           1114   1117#
CMDTBL  030276           1113   2476#
CMMARK  033646           3098*  3105   3107   3113   3119#
CMMISC  015020           2065   2069#
CMRET   033636           3100   3104   3109   3113#
COMAND  033542           1113   3079#
COMENT  006322            497   1110#
CPADD   004536            766#  2476
```

```
CPALP   004552   771#   786
CPARTN  004634   788#   790
CPASS   005704  1023#  2479
CPCLR   004764   834#  2482
CPCLRC  005014   843#   850
CPCLRT  005056   854#   856
CPCNRT  005212   888    890#   892
CPCNT   005126   881#  2485
CPDEL   004644   794#  2488
CPDLP   004660   799#   819
CPDRTN  004754   821#   823
CPERR   006612  1177#  2491
CPERTN  006730  1188   1194#  1197
CPGO    005522   982#  2494
CPINIT  005222   905#  2497
CPINRT  005460   947#   949
CPMAST  004130   649#  2500
CPMLP   003520   583#   599
CPMOK   004144   651    654#
CPMRET  004212   653    663#
CPRANG  004400   723#  2503
CPRFIL  004502   747#   751
CPRIB   004306   694#  2506
CPROK   004322   696    699#
CPRRET  004376   698    711#
CPRRTN  004526   753#   756
CPSEC   004214   669#  2509
CPSEX   004076   633#   636    642
CPSILO  003420   565#  2512
CPSLV   004034   621    627#
CPSOK   004230   671    674#
CPSRET  004300   673    684#   688
CPSRTN  005116   868#   870
CPSTAT  005066   862#  2515
CPSUM   005470   956#  2518
CPSURT  005512   961#   963
CRRET   015100  2080   2085#
CR.    = 000015   333#  2081   2272
CTLCMG  016412  2126   2336#
CTLOMG  016404  2094   2334#
CTLUMG  016407  2135   2335#
CTL.C  = 000003   336#  2270
CTL.O  = 000017   334#  2273
CTL.Q  = 000021   337#  2274
CTL.S  = 000023   338#  2275
CTL.U  = 000025   335#  2276
CURAD   017752  1349*  1350   1477   1502   2400#
CYCL    007362  1302   1304#
DATBUF  020666  1272   1347   2433#
DATGEN  007136   506   1258#
DATGEV  017660   504    946*  1285*  1367*  2367#
DBCLP   011110  1560#  1562
DBFCLR  011074  1547   1557#
DDIV    033256  2839   2888#
DECJSP  032716  1852   1903   1955   2720#
DECPNT  032724  1783   1787   1791   1795   2726#
```

```
DEQ     033460          1348    2054    2219    2981#
DIV     033320          2732    2923#
DONE  = 000200           344#
DQEMP   033530          2988    2998#
DQNOWP  033522          2992    2994#
DQRET   033526          2996#   3000
DTCDON  007314          1283#
DTCLP   007254          1278#   1282
DTCNT   007170          1263    1265#
DTGNC   007206          1266    1268#
DTGNCO  007230          1269    1272#
DTSEED  017726           938*   1268    1278    1279*   1280*   1283    2390#
ELPSTM  031700          1797    2556#
ENQ     033402          1232    2157    2158    2159    2160    2180    2204    2231    2248    2260    2261    2262    2961#
ERADR = 012162          1400#   1412#   1416#   1424#   1428#   1432#   1434#   1438#   1448#   1452#   1474#   1493#   1517#
                        1586#   1595#   1599#   1607#   1612#   1614#   1618#   1622#   1669#   1676#   1685#
ERMIN   014344          1993    1995#
ERMINR  014562          2031    2033#
ERMIS   014454          1994    2004#
ERMISR  014700          2032    2043#
ERNE    012162          1668    1685#
ERR   = 100000           342#
ERRA    010004          1411    1414#
ERRADR= 000004          1976#   1991    2014#   2029
ERRB    010040          1415    1418#
ERRB1   010120          1423    1426#
ERRB2   010154          1427    1430#
ERRB3   010210          1431    1434#
ERRC    010234          1419    1436#
ERRD    010302          1437    1442#
ERRE    010356          1443    1450#
ERRINT  007750          1386    1410#
ERRMOD  014250          1400    1412    1416    1424    1428    1432    1434    1438    1448    1452    1474    1493    1517
                        1976#
ERRMOR  014466          1586    1595    1599    1607    1612    1614    1618    1622    1669    1676    1685    2014#
ERRNUM= 000002          1976#   1984    2014#   2022
ERRX    010406          1413    1417    1425    1429    1433    1435    1441    1449    1451    1453#
ERR0    017770          1190*   1829    1845    1861*   1883    1896    1912*   2407#
ERR1    017772          1191*   1834*   1848    1856    1858*   1860*   1899    1907    1909*   1911*   2408#
ERR2    017774          1192*   1835*   1842    1851    1855*   1893    1902    1906*   2409#
ERTBCL  005314           919#    923
ERTBL   024726           918    1961    2443#   2453
ERTBLO= 024720          1990    2028    2453#
ERTCD   005336           920     924#
ERTMSG  031151           537    2544#
ERTRAP  003340           421     536#
ESCFLG  017756           468*    630*    640*    662*    683*    710*    752*    787*    820*    839*    908*    1112*   1117
                        1187*   2402#   3103*
EXCNT   031247           886    2546#
EXRST   031225           997    2545#
FIRST   017710           581*    600    2379#
FRELEM  030264          1348    1349    1359    2054    2055*   2056    2062*   2066    2069    2219    2220    2466#
GTLENS  033712          3167#   3193
HDRFLG  017716          1193*   1823    1825*   1833*   1877    1879*   2382#
HOURS   017646           914*    990*   1713*   1782    2362#
INPLIN= 000002          3079#   3099
```

```
INTENB= 000100   345#
INTTMP  030274  2230* 2231  2470#
ITEM  = 000002  2961# 2968  2981# 2989* 2999*
KEYWD   033652   656   676   701  3108  3154#
KWDERT  004302   657   677   687#  702
KWEXIT  034026  3201  3207#
KWFLG   017650   454*  456*  461*  887   998  1778  2148  2363#
KWTABL= 000004  3079# 3108
LCS     016150   455   889* 1000* 1700* 2150* 2306#
LENINP= 000004  3148# 3162* 3192  3195
LF.   = 000012   332#  2271
LXERR   034462  3340  3360  3368  3382#  3386
LXINCN  034244  3320# 3345
LXIT    034460  3379# 3384
LXSCAN  034070  3099  3300#
MDNER   030560  1139  2535#
MESSAG= 000002  2244# 2245
MINUTS  017644   912*  988* 1709* 1710  1712* 1786  2360#
MLI     007336  1279  1295# 1672
MLSD    017730   939  2391#
MSCHNG  030632  1159  2536#
MSGLSD  017732   939* 1262* 1264* 1265  1273  1277  1283* 2392#
MSTMG1  031274   622  2547#
MSTMG2  031332   623  2548#
MSTMG3  031373   624  2549#
MSTMG4  031441   625  2550#
MSTMG5  031507   629  2551#
MTQMSG  031032  1239  2541#
MUL     007352  1300# 1305
N     - 000031   244# 1400# 1412# 1416# 1424# 1428# 1432# 1434# 1438# 1448# 1452# 1474# 1493#
         1517# 1586# 1595# 1599# 1607# 1612# 1614# 1618# 1622# 1669# 1676# 1685# 2444
NEXT    017712   582*  586   596*  600  2380#
NMINT   010774  1395  1526#
NOERMG  032226  1186  2568#
NONMG   032563  1838  1888  2581#
NOTHER  034014  3160  3173  3177  3196  3204#
NOWMST= 000004  1379# 1393  1526
NQFULL  033454  2967  2977#
NQNOWP  033446  2971  2973#
NQRET   033452  2975# 2978
NULPAR  033650  3111  3120#
NXTCH   033732  3175# 3186
NXTEL   034074  3303# 3321
NXTWD   033752  3179  3188#
NXT0    003004   493   495#
NXT1    003026   496   498#
NXT10   003270   523   525#
NXT11   003312   526   528#
NXT12   003334   529   531#
NXT2    003050   499   501#
NXT3    003072   502   504#
NXT4    003114   505   507#
NXT5    003136   508   510#
NXT6    003160   511   513#
NXT7    003202   514   516#
NXT8    003224   517   519#
```

```
NXT9    003246   520    522#
OBJCNT  017760   575*   589*   603*   604    609    612    617    618    2403#
OCTJSP  032702   451    483    1744   1846   1849   1897   1900   1939   1943   2708#
OCTPNT  032710   2714#
ODELIM  034406   3350   3352   3359#
ONXTDG  034352   3349#  3357
ORIGSD  017724   938    2389#
PADFLG  017720   571*   609    611*   627    2383#
PARCLR  005310   917#   995
PART1   006006   1029   1044#
PART2   006074   1045   1060#
PART3   006162   1061   1076#
PART4   006246   1077   1091#
PCCALL  032670   2617   2625#
PCLEX   002000   377    385#   1098
PCLEXM  030731   446    2538#
PCLGO   017700   425*   539*   884*   994*   1142*  1236   1340   1439*  2156*  2375#
PCLOOK  032654   2615#  2623
PCLOOP  002762   476    492#   531
PCLRCV= 164220   299#   2291   2292   2293   2294   2295   2296
PCLTXM= 164200   298#   2283   2284   2285   2286   2287   2288   2289   2290
PCQUIT  032676   2619   2630#
PCREST  002570   379    459    464#
PCRST   002174   419#
PDRET   015470   2173   2178   2181#
PLAC    034236   3316#  3375
PLCRLF  015746   2247   2252#
PLODEC  015712   2246#  2250
PL219   015752   2253   2260#
PMRET   015570   2192   2200   2205#
PM176   015520   2195   2198#
PM187   015524   2197   2199#
PNCRLF  015752   444    445    474    885    996    1138   1140   1158   1160   1839   1889   2084   2176
                 2259#
PNTLIN  015706   446    452    484    537    622    623    624    625    629    886    997    1115   1139
                 1159   1186   1239   1763   1769   1772   1775   1777   1797   1798   1826   1827   1838
                 1853   1880   1881   1888   1904   1956   1960   2094   2126   2135   2244#
PRCLR   005004   840#   916
PRCR    015040   2078#  2271   2272
PRCTLC  015154   2125#  2270
PRCTLO  015102   1238   2091#  2273
PRCTLQ  015144   2116#  2274
PRCTLS  015134   2106#  2275
PRCTLU  015176   2134#  2276
PRDEL   015372   2171#  2277
PRESC   015214   475    1116   1119   1145   1240   1806   1891   1968   2127   2147#  2177
PRINIT  005240   447    909#
PRINTD  017714   1836   1840*  1854*  1886   1890*  1905*  2381#
PRMISC  015472   2069   2190#
PROCHR  032654   2064   2614#
PRST    002446   448#
PS    = 177776   328#   434*   464*   486*   541*   1097*  1383*
PUSHNO  034336   3341#  3361
QBACK = 000012   317#   433*   853*   992*   2093   2968*  2969*  2970   2972*
QBOT  = 000006   315#   2970   2991
QCHACT  006760   1217#  1222
```

```
QELEMS= 000000  312#
QFRONT= 000010  316#  433*  852*  991*  2093*  2989  2990*  2991  2993*
QLDEV   017656  469*  501  993*  1214*  1363*  2366#
QLOAD   007030  1219  1231#
QLOOK   006772  1218  1220#
QMTRN   007060  1224  1236#
QRTN    007014  1226#  1237  1241
QSIZE = 000002  313#  2966
QTOP  = 000004  314#  433  852  853  991  992  2972  2993
QUEUE = 000004  2961#  2965  2981#  2986
RADB    020000  842  930  1216  1799  2419#  2429
RADBO = 020064  746  782  810  1356  1483  1508  2429#
RADEND  020664  1734  2428#
RANK    017750  1280  1673  2399#
RANM    017746  1279  1672  2398#
RBCLMG  031657  1777  2555#
RBSTMG  031640  1775  2554#
RCBUF   022726  1549  1558  1671  2438#
RCDEV   016172  405  1058*  1095*  2326#
RCDOR = 000400  1569#  1577
RCELIN  032517  1904  2577#
RCER  - 100000  1568#  1574
RCERC   032553  1901  2580#
RCPRIO= 000005  303#  441
RCR     016116  406*  1548*  1551*  1552*  1626*  1627*  1631*  1638*  1642*  1644*  1646*  1680*  1690*
                2019  2291#
RCRCM   000040  1571#  1583  1651  1653
RCRNO   032517  1895  2578#
RCSEED  017734  1639*  1640  1654  1672*  1673*  1674  2393#
RCSUC = 000200  1570#  1580
RCTRN   032534  1898  2579#
RCTXPS  017736  945*  985*  1225*  1259*  1364  2394#
RCVEC   016174  417  1042*  1096*  2327#
RCVECT= 000174  300#
RCVEV   017702  466*  510  538*  1546*  1587*  1632*  1682*  1691*  2376#
RCVINA  011144  1575  1577#
RCVINB  011160  1578  1580#
RCVINC  011174  1581  1583#
RCVIND  011210  1584  1586#
RCVINT  011124  436  1573#
RCVVEC  016134  417*  436*  440  2299#
RCWD  = 020000  1543#  1551
RDBA    016126  414*  1549*  2295#
RDBC    016124  412*  1550*  1663  2294#
RDCRC   016130  416*  2296#
RDDB    016122  410*  1639  2293#
RDOINC  011720  1641  1646#
RDOINT  011652  1579  1638#
RDORTN  011704  1643#  1647
RECV    011024  512  1545#
REGRES  034052  542  1227  1401  1455  1463  1488  1514  1529  1563  1588  1629  1643  1681
                1692  2232  3207  3229#
REGSAV  034036  536  1213  1381  1557  1573  2229  3155  3215#
RELZRO= 002000  382#
REQINP  017640  423*  2079  2083*  2151*  2172  2191  2356#
REREV   017654  528  944*  1832*  1885*  2155*  2365#
```

```
RERHDR  032426          1880    2575#
RFRINT  011244          1576    1593#
REROS   013440           530    1873#
RERRA   011300          1594    1597#
RERRB   011334          1598    1601#
RERRB1  011414          1606    1609#
RERRB2  011456          1611    1614#
RERRC   011502          1602    1616#
RERRD   011536          1617    1620#
RERRE   011572          1621    1624#
RERRX   011632          1596    1600    1608    1613    1615    1619    1623    1625    1631#
RERTBL  026616          1835    2041    2461#
RERTN   013762          1876    1882    1892    1910    1913#
RERTRN  011624          1629#   1633
RESULT= 000026          3152#   3197*   3204*
RETURN= 000022          3149#
RJCTF   017740           470*   1260*   1270*   1500    2395#
RPMSG   016415          2338#
RRHLIN  032447          1881    2576
RRJINT  012206          1585    1690#
RSADD   031023           448    2540#
RSFLSZ  012004          1660    1662#
RSHOLD  017762           426*    449    2404#
RSNTO   011766          1656    1658#
RSPC    017754          1420*   1421*   1422    1426*   1430    1603*   1604*   1605    1609*   1610    2401#
RSR     016120           408*   1574    1577    1580    1583    1593    1597    1601    1603    1616    1620    1624    1628*
                        1651    1653*   2292#
RSTMSG  031000           452    2539#
RSUCHK  012134          1675    1678#
RSUCNT  011750          1652    1654#
RSUGEN  012054          1672#   1679
RSUINT  011732          1582    1651#
RSULV   012140          1664    1670    1677    1680#   1686
RSUST   012050          1667    1671#
RTRYA   002646           474#    487     544
RUBOUT= 000177           339#   2277
SAVRG   033356          2740    2830    2944#
SCCHAR  034414          3309    3363#
SCEOL   034454          3310    3377#
SCNO    034250          3308    3323#
SCTBL   030532           656     676     701    2524#
SCWORD  034210          3307    3312#
SECNDS  017643           911*    987*   1705*   1706    1708*   1790    2359#
SERFL   006672          1183    1189#
SETCN   032732          2710    2716    2722    2728    2730#
SHBRT   010570          1475    1486#   1494
SILLD   003674           610     612#    619
SJT     034200          3305    3307#
SLOLP   003704           614#    616
SMADV   014214          1935    1964#
SMARXT  014226          1932    1967#
SMCHK   014000          1931#   1965
SMLIN   032574          1956    2582#
SMLIN1  032574          1936    2583#
SMLIN2  032616          1940    2584#
SMLIN3  032642          1944    2585#
```

```
SMPN    014130            1944*   1955#
SMRTN   014246            1928    1958    1962    1969#
SMSPN   014134            1953    1956#
SMTTLE  032142            1960    2566#
SNTDIG  034314            3326    3328    3335#
SNXTDG  034254            3325#   3333
SORCNT  010720            1511#   1518
SORINT  010656            1392    1500#
SORIP   010750            1501    1517#
SPCEV   017704             472*    513    1141*   1440*   2377#
SPEC    006432             515    1137#
SPERTN  006504            1142#
SR    = 177570             329#
SRC   = 000026            3151#   3157
STARXT  013056            1736    1805#
STATUS  013324             518    1728#
STBADV  013044            1739    1902#
STCHK   012344            1734#
STHDR   012554            1733    1767#
STITLE  031762            1798    2558#
STKTOP= 002000             378     383#    386     420     485     543
STLIN   032054            1763    2560#
STLIN1  032054            1740    2561#
STLIN2  032076            1746    1750    2562#
STLIN3  032124            1755    1759    2563#
STPNTR  017670             866*   1732    1734    1737    1741*   1742    1745*   1747    1754*   1756    1764*   1799*   1802*
                          2371#
STRTN   013076            1731    1765    1800    1803    1807#
STRTRT  005674             999    1001#   1004
STSEV   017666             516     865*    942*   1143*   1805*   2152*   2370#
STYHR   012446            1750#   1753
STYON   012462            1751    1754#
STYONT  012524            1760    1763#
STYTHR  012510            1759#   1762
SUCCS   010644            1492    1495#
SUCIA   010612            1471    1491#
SUCINP  010534            1473    1477#
SUCINT  010460            1389    1469#
SUCINY  010474            1472#   1496
SUMEV   017662             519     941*    959*   1144*   1967*   2153*   2368#
SUMHDR  014164            1930    1960#
SUMPNT  017672             960*   1929    1931    1933    1957*   1961*   1964*   2372#
SUMRY   013764             521    1926#
SUMSV   017664             479*    480*    482    2369#
SYNRR   006732            1179    1196#
SYNRTM  004100             574     585     592     595     606     635#
SYNRTN  004530             726     729     733     736     738     740     755#
SYNTAX  030704            1115    2537#
SYNTCN  005214             883     891#
SYNTIN  005462             907     948#
SYNTRA  004636             772     774     776     789#
SYNTRC  005060             838     855#
SYNTRD  004756             800     802     804     822#
SYNTRS  005120             864     869#
SYNTSM  005514             958     962#
SYNTX   005676             984    1003#
```

```
SYNXR   006316   1028  1033  1038  1041  1049  1054  1057  1065  1070  1073  1081  1086  1089
                 1100#
TABLAD= 000024   3150#  3165
TCBFPT  030270   2468#
TCBIN   030272   2469#
TCR     016076    388*  867*  910*  1343*  1359*  1360*  1382*  1444  1453*  1462*  1487*  1513*  1528*
                 1981  2283#
TEMP    017764   1981*  1982*  1983*  1999*  2000*  2001*  2002*  2003*  2019*  2020*  2021*  2038*  2039*
                 2040*  2041*  2042*  2405#
TEMP1   017766   1985*  1996*  1997*  1998*  2001  2023*  2034*  2035*  2036*  2037*  2040  2406#
TEREV   017652    525  943*  1189*  1831*  2154*  2364#
TERHOR  032266   1826  2569#
TEROS   013100    527  1819#
TERTBL  025150    925  1180  1192  2002  2458#
TERTN   013436   1822  1828  1841  1859  1862#
THEND   033772   3183  3195#
THLTFL  017722   2107*  2117*  2216  2228*  2384#
THUMST  031561   1769  2552#
THUSCN  031607   1772  2553#
TIAREA  016554   2350#
TICKS   017642    913*  989*  1701*  1702  1704*  1794  2358#
TIEND = 016624   2350#
TIQ     016540    433*  492  2054  2231  2350#
TISIZE= 000024    306#  2350
TMLIN1  031741   1781  2557#
TMMR    016110    398*  614*  2288#
TMMRH   016112    400*  607*  620  626*  631*  639*  658  661*  678  682*  1393  1526*  1767
                 1770  2289#
TOAREA  016640   2350#
TOEND - 017640   2350#
TOQ     016624    433*  498  1729  1820  1874  1927  2092*  2093*  2157  2158  2159  2160  2180
                 2204  2219  2248  2260  2261  2262  2350#
TORET   015640   2215  2218  2221#
TOSIZE= 000400    307#  2350
TRAP4   002702    465  479#
TRELIN  032362   1853  2571#
TRERC   032416   1850  2574#
TRHLIN  032312   1827  2570#
TRNKF   017742    471*  1261*  1267*  1271*  1470  2396#
TRPDMG  031105    484  2542#
TRP4AD  031140    481  2543#
TRRCN   032377   1847  2573#
TRRNO   032362   1844  2572#
TRYOCT  034350   3336  3338  3348#
TSBA    016106    396*  1347*  2287#
TSBC    016104    394*  1346*  2286#
TSCRC   016114    402*  2290#
TSDB    016102    392*  2285#
TSR     016100    390*  1384  1387  1390  1396  1410  1414  1418  1420  1436  1442  1446  1450
                 1460*  1469*  1472  1476*  1491  1495*  1511*  2284#
TTDEV - 177560    295#  2301  2302  2303  2304
TTIINT  015642    437  2227#
TTINH   014764   2058  2060  2062#
TTINP   014712    494  2052#
TTOUT   015572    500  2213#
TTPRIO- 000004    297#  443
```

```
TTRBUF  016140            2230    2302#
TTRCSR  016136             467*    540*   2301#
TTRET   015036            2067    2070#
TTVCTR= 000060             296#   2305
TTVECT  016146             437*    442    2305#
TTXBUF  016144            2220*   2304#
TTXCSR  016142            2214    2303#
TXBSY = 000020            1376#   1396    1446    1460
TXDEV   016166             387    1090*   1093*   2324#
TXDGEN  007566            1365    1367#
TXER  = 100000            1375#   1384
TXMEV   017674             507     940*   1226*   1284*   1342*   1399*   1454*   1461*   1486*   1512*   2373#
TXMIT   007402             509    1339#
TXML    017744            1276*   1346    2397#
TXMST   007400             703     709*    915*   1337#   1360    1773
TXMSV   017676            2374#
TXMVEC  016132             403*    435*    438    2298#
TXOUT   007544            1345    1363#
TXPRIO= 000005             302#    439
TXRTN   007542            1341    1361#   1366    1368
TXSOR = 000040            1378#   1390    1472    1476    1491    1495    1511
TXSUC - 000200            1377#   1387    1469
TXVEC   016170             403    1074*   1094*   2325#
TXVECT- 000170             301#
X       000040            2418#   2427#
XMTIB   007700            1394    1396#
XMTINT  007600             435    1381#
XMTIS   007634            1385    1387#
XMTPNM  007664            1391    1393#
XMTPRB  007714            1397    1399#
XMTSR   007650            1388    1390#
XSPCEV  017706             473*    522    1161*   1527*   2378#
XSPEC   006536             524    1157#
Y     - 000031            2442#   2450#
ZERO    016154            2262    2312#
ZX1   - 000000             565#    578#    649#    669#    694#    723#    766#    794#    834#    840#    862#    881#    905#
                           909#    917#    956#    982#   1023#   1110#   1137#   1157#   1177#   1212#   1258#   1339#   1545#
                          1728#   1819#   1873#   1926#   1976#   1997#   1998#   2000#   2014#   2036#   2037#   2039#   2052#
                          2078#   2091#   2106#   2116#   2125#   2134#   2147#   2171#   2190#   2213#   2244#   2259#   2961#
                          2981#   3079#   3329#   3353#
ZX2   - 000000             578#   1997#   1998#   2000#   2036#   2037#   2039#   3329#   3353#
$BLK  = 000000            2709#   2715#   2721#   2727#
$BLKS = 002000            2833#
$SGN  = 000000            2709#   2715#   2721#   2727#
$SGNS = 000000            2833#
$SUP  - 000000            2709#   2715#   2721#   2727#
$TR     002550             453     460#
.       034476             370#    374     376#    381#    382     383    1279    1400    1412    1416    1424    1428    1432
                          1434    1438    1448    1452    1474    1493    1517    1586    1595    1599    1607    1612    1614
                          1618    1622    1669    1672    1676    1685    2330#   2331    2350#   2361#   2414#   2428    2433#
                          2438#   2458#   2461#   2589#   2833
```

```
BDINIT   229#   910  1343  1453  1462  1487  1513  1548  1631  1680  1690
BREAK   3020#  3313  3366
CALL     129#   444   445   446   447   452   474   475   484   494   497   500   503   506   509
          512   515   518   521   524   527   530   537   622   623   624   625   629   885   886
          916   995   996   997  1113  1115  1116  1119  1138  1139  1140  1145  1158  1159  1160
         1186  1232  1238  1239  1240  1348  1400  1412  1416  1424  1428  1432  1434  1438  1448
         1452  1474  1493  1517  1586  1595  1599  1607  1612  1614  1618  1622  1669  1676  1685
         1763  1769  1772  1775  1777  1797  1798  1806  1826  1827  1838  1839  1853  1880  1881
         1888  1889  1891  1904  1956  1960  1968  2054  2066  2069  2084  2094  2126  2135  2157
         2158  2159  2160  2176  2177  2180  2204  2219  2231  2248  2260  2261  2262
CBTAS   2807#  2833
ERROR    260#  1586  1595  1599  1607  1612  1614  1618  1622  1669  1676  1685
ERROT    248#  1400  1412  1416  1424  1428  1432  1434  1438  1448  1452  1474  1493  1517
HEDING   200#
KEYWD     44#   656   676   701  3108
LXSCAN    69#  3099
MULP     217#  1279  1672
MULT     151#   578  1997  1998  2000  2036  2037  2039  3329  3353
PROC      85#   565   649   669   694   723   766   794   834   840   862   881   905   909   917
          956   982  1023  1110  1137  1157  1177  1212  1258  1339  1545  1728  1819  1873  1926
         1976  2014  2052  2078  2091  2106  2116  2125  2134  2147  2171  2190  2213  2244  2259
         2961  2981  3079
REGRES   280#   542  1227  1401  1455  1463  1488  1514  1529  1563  1588  1629  1643  1681  1692
         2232  3207
REGSAV   274#   536  1213  1381  1557  1573  2229  3155
RETURN   110#   633   663   684   711   753   788   821   854   868   890   947   961  1001  1101
         1120  1146  1162  1194  1228  1286  1361  1553  1807  1862  1913  1969  2008  2047  2070
         2085  2095  2108  2118  2163  2181  2205  2221  2263  2783  2869  2906  2939  2952  2975
         2996  3116
SPAN    3008#  3303
STCVT   2661#  2709  2715  2721  2727
SYNCLS  3041#  3304

. ABS.  034476     000

 ERRORS DETECTED:  0

 PCLEXR,PCLEXR/CR/NL:TTM:ME<PCLEXR
 RUN-TIME: 11 15 1 SECONDS
 RUN-TIME RATIO: 137/28 4.8
 CORE USED:  10K  (19 PAGES)
```