

PCL11-A,B

PCL11 EXERCISER V-02
CZPLAAO

AH-E260A-MC-1
COPYRIGHT © 1978
FICHE 1 OF 1

JUN 1978
digital
MADE IN USA

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

PROGRAM CODE: AC-E259A-MC

PROGRAM NAME: CZPLAA0 PCL11 EXERCISER V-02

DATE CREATED: 21-SEP-76

UPDATED: 13-MAR-78

MAINTAINER: SPECIAL SYSTEMS, KANATA

AUTHOR: DAVID G. WIENS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1978, BY DIGITAL EQUIPMENT OF CANADA, LIMITED.
PARALLEL COMMUNICATIONS LINK (PCL11) EXERCISER

	TABLE OF CONTENTS	PAGE
	-----	----
1	GENERAL	1-1
1.1	GENERAL DESCRIPTION	1-1
1.2	DIFFERENCES IN EXERCISER V-02	1-2
2	EXERCISER TABLES	2-1
2.1	STATUS TABLE	2-1
2.2	SUMMARY TABLE	2-2
2.3	ERRORS TABLES	2-3

2.4	CLEARING OF THE TABLES	2-4
3	EXERCISER COMMANDS	3-1
3.1	CONTROL CHARACTERS	3-1
3.1.1	CONTROL-C	3-1
3.1.2	CONTROL-O	3-1
3.1.3	CONTROL-U	3-1
3.1.4	CONTROL-S	3-1
3.1.5	CONTROL-Q	3-2
3.1.6	CARRIAGE RETURN	3-2
3.1.7	LINE FEED	3-2
3.1.8	RUBOUT	3-2
3.2	COMMANDS	3-3
3.2.1	SILO	3-3
3.2.2	MASTER	3-4
3.2.3	SECONDARY	3-4
3.2.4	RIB	3-5
3.2.5	RANGE	3-5
3.2.6	ADD	3-6
3.2.7	DELETE	3-6
3.2.8	CLEAR	3-7
3.2.9	INITIALIZE	3-7
3.2.10	STATUS	3-8
3.2.11	SUMMARY	3-8
3.2.12	ERRORS	3-8
3.2.13	ASSIGN	3-9
3.2.14	GO	3-10
3.2.15	CONTINUE	3-10
3.2.16	**SYNTAX ERROR**	3-11
4	GETTING THE EXERCISER STARTED	4-1
4.1	PREPARATION	4-1
4.2	LOADING	4-1
4.3	DEVICE ADDRESSES	4-1
4.4	STARTING ADDRESSES	4-2
4.5	OPERATING PROCEDURES	4-2
4.6	ERRORS	4-3
	TABLE OF CONTENTS (CONTD)	PAGE
	-----	----
5	COMMAND INPUT MODE DESCRIPTION	5-1
5.1	SHORTENED COMMANDS	5-1
5.2	RUBOUT FEATURES	5-1
5.3	ENTERING COMMAND MODE	5-1
5.4	UPPER OR LOWER CASE	5-2
6	EXERCISE MODE DESCRIPTION	6-1
6.1	TRANSMIT EVENT	6-1
6.2	RECEIVE EVENT	6-1
6.3	DATA GENERATION EVENT	6-1
6.4	ADDRESS QUEUE EVENT	6-2
6.5	ERRORS UPDATE EVENT	6-2
6.6	SPECIAL EVENTS	6-2
APPENDIX A		
	EXERCISER OVERALL FLOW	A-1

SEQ 0003

APPENDIX B

EXERCISER COMMANDS

B-1

APPENDIX C

ERROR DESCRIPTIONS

C-1

APPENDIX D

STATISTICAL INFORMATION

D-1

LISTING...

1-1

SEQ 0004

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 UPON ISSUANCE OF THE 'ERRORS' COMMAND.

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

SEQ 0005

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.
- 1-3
- 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.
2-1

SEQ 0007

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 SUCCESSIVE 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.

2-2

SEQ 0008
SEQ 0009

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

SEQ 0010

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

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, 85, AND IMMEDIATELY THEREAFTER THE STATUS WAS REQUESTED, THE STATUS TABLE WOULD INDICATE ALL OF THE SELECTED

SEQ 0011

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 ADDRES-
SES 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.

3-1

SEQ 0012

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 "\$".

"\$" 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 PDF-11, ECHO " C ", AND PRINT
THE COMMAND MODE PROMPT "\$".

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 "S" 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 "S" AND AWAIT A FURTHER COMMAND.
3-2

SEQ 0013

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 "S".

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 "S".

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 "\$" 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.

3-3

SEQ 0014

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 INITIALIZE

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, THEY 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 CN>

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:

L 1

"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:

"THE SILO HAS BEEN PADDED WITH THE ADDRESS '0'."
3-4

SEQ 0015

SEQ 0016

3.2.2 MASTER

MASTER SET OR CLEAR

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 'SET'
AND 'CLEAR'. 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

SECONDARY SET OR CLEAR

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 CUR-
RENTLY 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 EXER-
CISING.

3-5

SEQ 0017

3.2.4 RIB

RIB SET OR CLEAR

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 PRINT-OUT.

3.2.4 RANGE

RANGE 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.

3-6

3.2.6 ADD

ADD 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

SEQ 0018

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 RE-
DUNDANCIES 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 CON-
JUNCTION 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
3-7

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 DIS-
TURB 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 REC-
EIVER ADDRESSES.

EXAMPLE:

CONSIDER THE FOLLOWING STRING OF COMMANDS:

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

THE RESULT WOULD BE THE "ACTIVATION" OF RECEIVER ADDRESSES: ^{B 2}

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.
3-8

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

SEQ 0020

C 2
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'.
3-9

SEQ 0021

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 OP-
ERATOR 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.
 3-10

SEQ 0022

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 TRANS-
 MITTER 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 'SUCCESSSES' 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 TYP-
 ING '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 REC-
 EIVERS 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 PRE-
 VIOUSLY ENTERED, THE EXERCISER WILL NOT BE
 ABLY TO RUN AND WILL APOLOGIZE THUS:

"* SORRY, I HAVE NO RECEIVER ADDRESSES *"

3.2.15 CONTINUE

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"
3-11

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 *"

SEQ 0023

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)

4-1

SEQ 0024

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.

4-2

SEQ 0025

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^{G 2}

- 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).

4-3

SEQ 0026

SEQ 0027

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-2017D-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

S-1

SEQ 0028

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' (SUM-
MARY, 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 DE-
CODER. "RUBOUT" (DELETE) CHARACTER WILL DELETE THE LAST CHARACTER
WHICH WAS TYPED IN AS PART OF A COMMAND WORD OR ARGUMENT. CONTROL-U
CHARACTER WILL REMOVE ALL THAT HAS BEEN TYPED IN SO FAR ON THIS LINE.

ONE OTHER METHOD OF HAVING THE EXERCISER IGNORE EVERYTHING
TYPED IN SO FAR IS TO TYPE "CONTROL-C".

- A) RUBOUT DELETE LAST CHARACTER
- B) CNTL-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 .

5-2

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.

6-1

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

SEQ 0029

SEQ 0030

IS RECEIVED SUCH AS "SUCCESSFUL TRANSFER", "ERROR", OR "REJECT COMP-^{J 2}LETED".

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.

6-2

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 TRANS-

SEQ 0031

MITTER ERRORS, AND ONE FOR RECEIVER ERRORS.

K 2

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

S ----START

I
LOAD & TRY
DEVICE ADDRESSES

I
INITIALIZE

I
RESTART

A ----MAIN LOOP

I
IF NOT
EMPTY

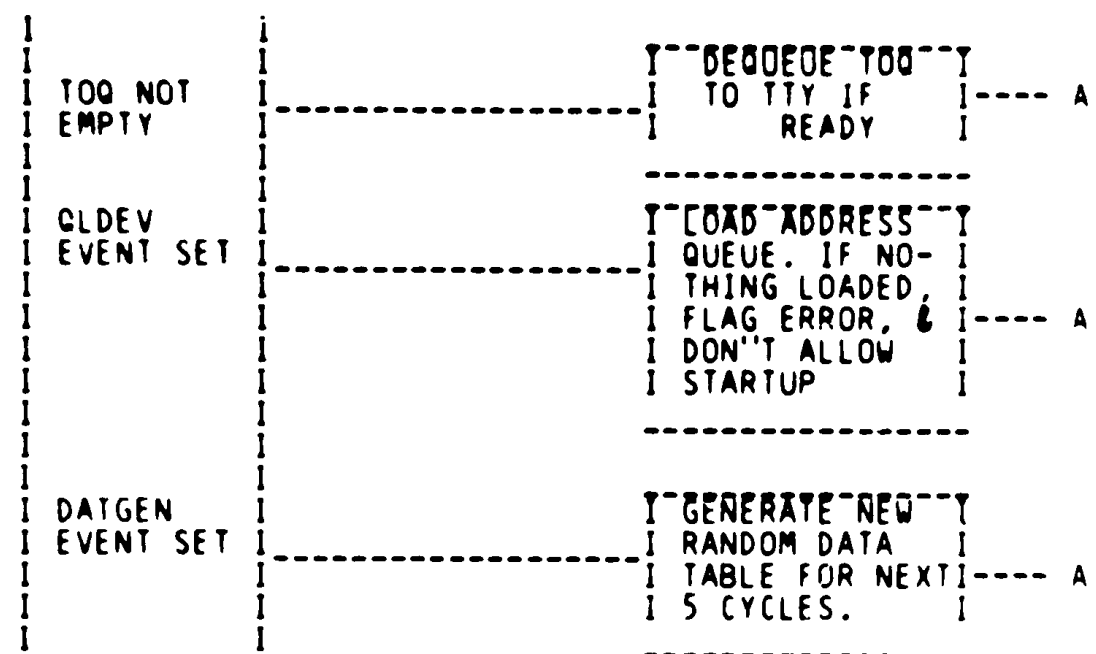
<PROCHR>

PROCESS

CHARACTER

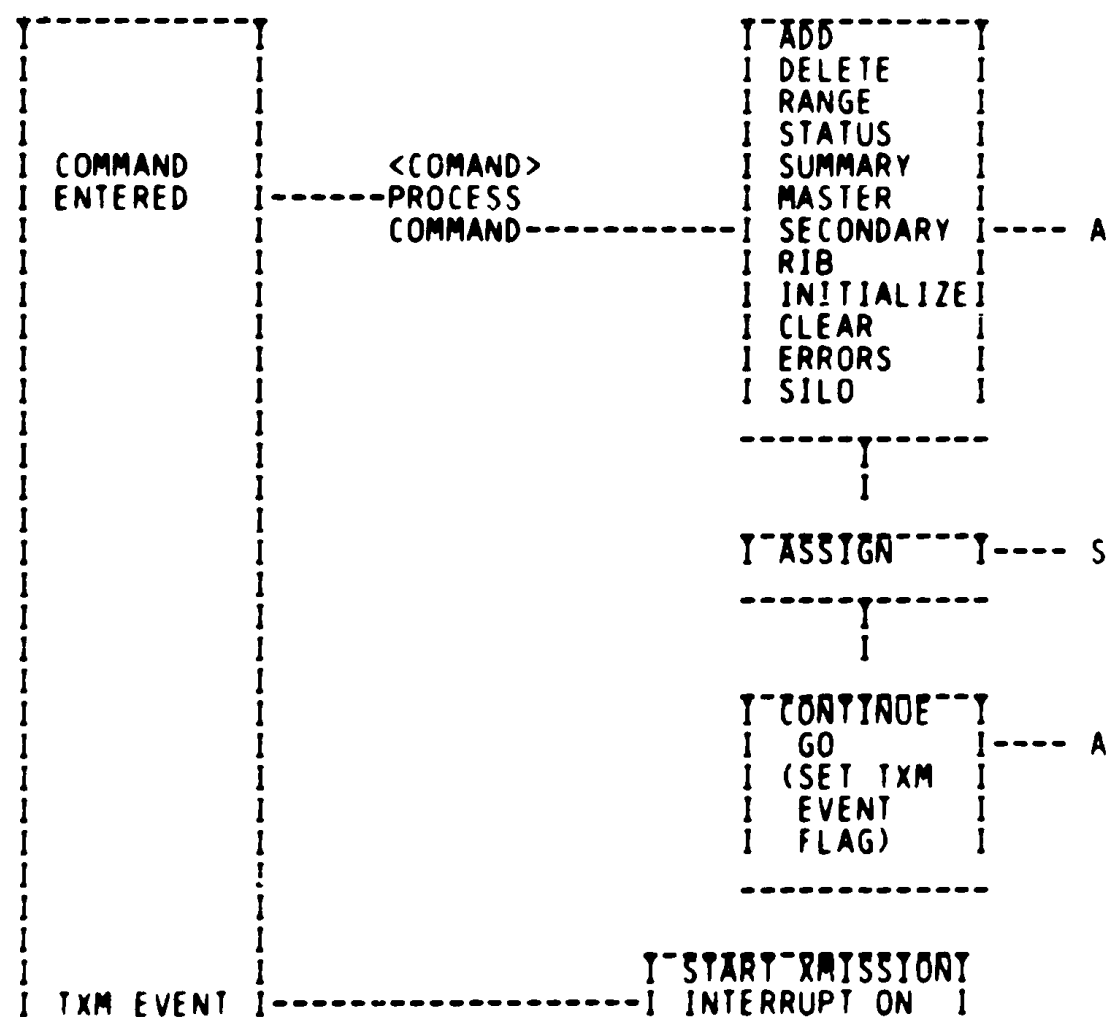
I
PRDEL
I
PRCTLS
I
PRCTLQ
I
PRCR-LF
I
PRCTLU
I
PRCTLO
I

----- A

--27/
/

A-2

SEQ 0034

--27/
/

```

I      SET      I      I UPDATE STATUS I---- A
I      I      I      I DEQUEUE RCVR  I
I      I      I      -----
I      I      I      I-SET-UP-TO-RCV-I
I      I      I      I FLAGS WORD   I---- A
I      I      I      I INTERRUPT ON  I
I      I      I      -----
I      I      I      I-STOP-RMT- I
I      I      I      I PRINT 'MASTER I---- A
I      I      I      I DOWN' MESSAGE I
I      I      I      -----

```

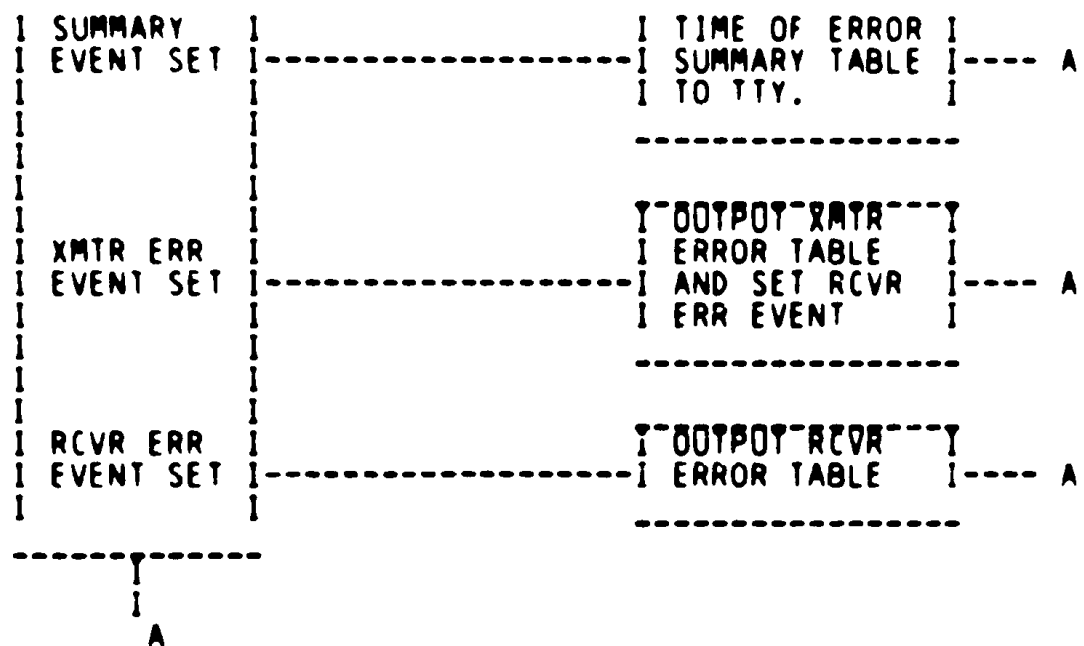
A-3

--37
/

```

I-----I
I  EXTRA SPE- I-----I  PRINT THIS I-----I
I  CIAL EVENT I-----I  UNIT HAS JUST I----- A
I                                     I  BECOME NEW I
I                                     I  MASTER'' I
I                                     I-----I
I                                     I-----I
I                                     I  UPDATE I
I  CLK INTRPT I-----I  CLOCK INTERRUPT I  THE TIME I----- A
I                                     I  HANDLER-----I
I                                     I-----I
I                                     I  ERROR-----I
I                                     I  BUSY I
I  XMTR I-----I  TRANSMITTER I  NOW MAST I
I  INTERRUPT I-----I  INTERRUPT-----I  SUC XFR I----- A
I                                     I  HANDLER I  SOF REJ I
I                                     I-----I
I                                     I-----I
I                                     I  ERROR-----I
I  RCVR I-----I  RECEIVER I  DATA RDY I
I  INTERRUPT I-----I  INTERRUPT-----I  SUC XFR I----- A
I                                     I  HANDLER I  REJ COMP I
I                                     I-----I
I                                     I-----I
I                                     I  OUTPUT ONE-----I
I  STATUS I-----I  LINE-AT-A- I
I  EVENT SET I-----I  TIME OF STATUS I----- A
I                                     I  BRUN STATE I
I                                     I-----I
I                                     I-----I
I                                     I  OUTPUT ONE-----I
I                                     I  LINE-AT-A- I

```



B-1

SEQ 0036

SEQ 0037

EXERCISER COMMANDS

CONTROL CHARACTERS

CHARACTER	ECHOES	EFFECT
-----	-----	-----
CNTRL-C	C	ENTER COMMAND INPUT MODE
CNTRL-J	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

B-2

SEQ 0038

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 C-1	PRINT SUMMARY TABLE

ERROR DESCRIPTIONS

SEQ 0039

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

STATISTICAL INFORMATION

SEQ 0040

STARTING ADDRESS 200
RESTARTING ADDRESS 204
SWITCH OPTIONS NONE
PROGRAM SIZE APPROX 7K
MEMORY OCCUPIED
LOW BOUNDARY 00000
HIGH BOUNDARY 34406

LOCATIONS TO CHANGE FOR
DIFFERENT DEVICE ADDRESSES

DEVICE CHANGE LOCATION

KEYBOARD STATUS 16056
KEYBOARD DATA 16060
TTY STATUS 16062
TTY DATA 16064
KEYBOARD VECTOR 16066

LINE CLK STATUS	16070
XMTR PRIORITY	2334
RCVR PRIORITY	2346
KBD PRIORITY	2360

USEFUL LOCATIONS	ADDRESS
------------------	---------

XMTR DATA BUFFER	20600
RCVR DATA BUFFER	22640
DATA SEED IN TRANSMITTER	17640
RANDOM MULTIPLIER	17660

RANDOM INCREMENT	17662
------------------	-------

CZPLAAD PCL11 EXERCISER V-02	MACY11	30A(1052)	28-APR-78	13:50
PCLEXR.P11	17-MAR-78	11:27	TABLE OF CONTENTS	

SEQ 0041

284	DEFINITIONS AND DEVICE INFO
367	PCL11 EXERCISER MAIN PROCEDURE
489	MAIN LOOP
546	COMMAND PROCESSORS:
547	COMMAND PROC. FOR SILO (LOAD)
644	COMMAND PROC. FOR MASTER, SECONDARY AND R.I.B.
713	COMMAND PROC. FOR RANGE
758	COMMAND PROC. FOR ADD AND DELETE
825	COMMAND PROC. FOR CLEAR, STATUS, AND CONTINUE
894	COMMAND PROC. FOR INIT, SUMMARY, AND GO
1008	COMMAND PROC. FOR "ASSIGN"
1103	COMMAND DECODER AND PROCESSOR
1199	RECEIVER ADDRESS QUEUE LOADER ROUTINE
1243	DATA GENERATION (RANDOM) ROUTINE
1288	MULTIPLY ROUTINE FOR DATA GENERATION
1322	TRANSMIT MODULE
1534	RECEIVER MODULE
1716	STATUS MODULE
1809	TRANSMITTER ERRORS MODULE
1864	RECEIVER ERRORS MODULE
1915	SUMMARY MODULE
1971	ERROR UPDATE ROUTINES
1972	TRANSMITTER ERRORS
2010	RECEIVER ERRORS
2049	UTILITY ROUTINES
2050	PROCESS AN INPUT CHARACTER FROM THE TTY
2072	TTY INPUT CHARACTER PROCESSING ROUTINES
2204	TTY OUTPUT HANDLERS
2222	TTY INPUT INTERRUPT PROCESSORS
2233	MESSAGE PRINT ROUTINE
2262	DATA AREAS
2466	KEYWORD TABLE
2527	SOME MORE ASCII STORAGE:
2584	AUXILIARY ROUTINES
2585	CHARACTER PROCESSOR
2628	BINARY TO ASCII CONVERSION
2731	GENERAL BINARY TO ASCII CONVERSION
2780	DOUBLE PRECISION BINARY TO ASCII
2867	DOUBLE PRECISION DIVIDE ROUTINE

2903 INTEGER DIVIDE MAGNITUDE NUMBERS
 2948 QUEUE HANDLING ROUTINES
 3071 COMMAND PROCESSOR INITIATING ROUTINE
 3116 KEYWORD PROCESSING ROUTINE
 3205 REGISTER SAVE & RESTORE ROUTINES
 3232 LEXICAL SCAN ROUTINE
 CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 1
 PCLEXR.P11 17-MAR-78 11:27

SEQ 0042

.TITLE CZPLAAO PCL11 EXERCISER V-02
 .IDENT /0003/ ;DRCMAC.MAC 6-1AN-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	ROUTINE ADDRESS
0	WORD	ROUTINE ADDRESS
2	BYTE	MINIMUM LENGTH OF KEYWORD
3	BYTE	FULL LENGTH OF KEYWORD
4	STRING	KEYWORD

; ENTRIES ARE STORED CONSECUTIVELY IN THE TABLE. EACH ENTRY
 ; MUST BEGIN ON A WORD BOUNDARY. THE KEYWORD OF
 ; THE PREVIOUS ENTRY MAY HAVE TO HAVE A BYTE (CONTAINING
 ; ANYTHING) APPENDED TO IT TO ACCOMPLISH THIS.
 ; ENTRIES MUST BE ARRANGED IN ALPHABETICAL
 ; ORDER (MORE SPECIFICALLY, IN ASCII COLLATING SEQUENCE).
 ; THE TABLE IS ENDED BY AN ENTRY WITH A FULL LENGTH
 ; OF 0.

```

; .MACRO KEYWD OBJ, KWTabL
; .IF NB OBJ
;   MOV OBJ, -(SP)
; .IF B KWTabL
;   .ERROR ;CANNOT SPECIFY OBJ & NOT TABLE ADDRESS.
;   MOV #1, -(SP) ;IF RUN, THIS WILL CAUSE TRAP.
; .ENDC
; .ENDC
; .IF NB KWTabL
;   MOV KWTabL, -(SP)
; .ENDC

```

55 JSR PC,KEYWD F 3
56
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 1-1
PCLEXR.P11 17-MAR-78 11:27

SEQ 0043

57 .ENDM ;KEYWD
58
59 ; -----
60
61 ; LXSCAN MACRO
62
63 ; THIS MACRO CALLS THE ROUTINE LXSCAN TO CONVERT THE CHARACTER
64 ; STRING SPECIFIED BY STR INTO ITS CONSTITUENT SYNTACTIC OBJECTS.
65 ; STR IS AN ASSEMBLER EXPRESSION SPECIFYING THE ADDRESS OF THE
66 ; BEGINNING OF THE ASCII STRING TO BE CONVERTED. THE STRING
67 ; MUST BE ENDED BY A CARRIAGE RETURN CODE.
68 ;
69 .MACRO LXSCAN STR
70 MOV STR,R1
71 JSR R1,LXSCAN
72
73 .ENDM ;LXSCAN
74
75 ; -----
76
77 ; PROC MACRO
78
79 ; THIS MACRO DEFINES THE ENTRY POINT FOR A PROCEDURE. THE PARAMS
80 ; SPECIFIED WILL BE GIVEN THEIR CORRESPONDING OFFSETS RELATIVE TO R5.
81 ; THUS, A PARAMETER PP CAN BE REFERENCED IN THE PROCEDURE BY THE
82 ; ASSEMBLER EXPRESSION '@PP(R5)'. THE PROCEDURE CAN BE CALLED
83 ; USING THE CALL MACRO.
84
85 .MACRO PROC PNAME,PARAMS
86 ZX1 = 0
87
88 .IRP ZX2 <PARAMS>
89 ZX1 = ZX1+2
90 .IRP ZX3 ZX1
91 .LIST
92 ZX2 = ZX3
93 .NLIST
94 .ENDM
95 .ENDM ;ZX2
96
97 .LIST
98
99 PNAME: ;**ENTRY POINT**
100 .NLIST
101 .ENDM ;PROC
102
103 ; -----
104
105 ; RETURN MACRO
106
107 ; THIS MACRO RETURNS FROM A PROCEDURE. IF ANSWR IS SPECIFIED IT
108 ; WILL BE LOADED INTO R0 BEFORE RETURNING.
109
110 .MACRO RETURN ANSWR
111 .IF NB ANSWR
112 MOV ANSWR,R0

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 1-2
PCLEXR.P11 17-MAR-78 11:27

SEQ 0044

.ENDC

```

RIS    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

```

165
166
167
168
CZPLA0 PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 1-3

SEQ 0045

```

      .IF NZ ZX2      ;IF NOT MOST SIG 1-BIT
      .IF NB WORK     ;IF WORK SPECIFIED
      MOV DST,WORK
      .IFF            ;ELSE IF WORK BLANK
      .ENDC
      .MEXIT
      .ENDC
      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,0SP
      .ENDC
      .IFF            ;END .IF NB WORK
      .IF NB WORK     ;ELSE IF MOST SIG 1-BIT
      ADD WORK,DST    ;IF WORK SPECIFIED
      .IFF            ;ELSE IF WORK BLANK
      ADD (SP)+,DST
      .ENDC
      .MEXIT
      .ENDC
      .ENDC
      .IFF            ;END .IF NZ ZX2
      ZX2 = ZX2/2      ;END .IF Z ZX1
      .ENDC            ;ELSE IF BIT 0 = 0
      ASL DST          ;SHIFT CONSTANT RIGHT 1 POSITION.
      .ENDM            ;END .IF NZ ZX2&1
      .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

```

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224

;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>
N
= N+1
.LIST

;UPDATE ENTRIES FOR ERROR P

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

.NLIST

.ENDM

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

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

.NLIST

.ENDM

;REGISTER SAVE MACRO

.MACRO REGSAV
JSR R5,REGSAV
.ENDM

;REGISTER RESTORE MACRO

.MACRO REGRES

JSR R5,REGRES

282
CZPLAAO PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

J 3
.ENDM
MACY11 30A(1052) 28-APR-78 13:50 PAGE 2
DEFINITIONS AND DEVICE INFO

SEQ 0048

.SBTTL DEFINITIONS AND DEVICE INFO

.IDENT '02'

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

; VARIABLE SYMBOL DEFINITIONS.

; DEVICE DEFAULT INFORMATION.

TTDEV = 177560
TTVCTR = 60
TTPRIO = 4
PCLTXM = 164200
PCLRCV = 164220
RCVECT = 174
TXVECT = 170
TXPRIO = 5
RCPRIO = 5

;ADDR OF RCSR FOR TTY.
;INPUT VECTOR ADDR FOR TTY.
;PRIORITY LEVEL FOR TTY.

; QUEUE SIZES.

TISIZE = 20.
TOSIZE = 256.
AOSIZE = 32.

;TIQ SIZE.
;TOQ SIZE.
;ADR QUEUE SIZE

QLEMS = 0
QSIZE = 2
QTOP = 4
QBOT = 6
QFRONT = 10
QBACK = 12

;#ELEMENTS PRESENTLY IN QUEUE.
;#ELEMENTS IN QUEUE SPACE.
;ADDR OF 1ST WORD OF QUEUE SPACE
;=QTOP+(QSIZE*2)
;ADDR OF FRONT ELEMENT OF QUEUE.
;ADDR OF BACK ELEMENT OF QUEUE.

; REGISTER DEFINITIONS.

R0 = X0
R1 = X1
R2 = X2
R3 = X3
R4 = X4
R5 = X5
SP = X6
PC = X7
PS = 177776
SR = 177570

;SWITCH REGISTER.

;SPECIAL CHARACTER DEFINITIONS:

LF. = 12
CR. = 15
CTL.O = 17
CTL.U = 25
CTL.C = 3
CTL.Q = 21
CTL.S = 23
RUBOUT = 177

;CR OR LF TO END LINE.
;..
; O TO THROW AWAY TTY OUTPUT.
; U TO DELETE LINE.
;CNTRL-C TO START INPUT.(C)
;CNTRL-Q TO RESUME PRINTOUT
;CNTRL-S TO SUSPEND PRINTOUT
;RUB OUT TO DELETE CHARACTER.

CZPLAAO PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 2-1
DEFINITIONS AND DEVICE INFO

SEQ 0049

340
 341
 342 100000
 343 004000
 344 000200
 345 000100
 CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 3
 PCLEXR.P11 17-MAR-78 11:27

; DEVICE BIT DEFINITIONS.
 ERR = 100000
 BUSY = 4000
 DONE = 200
 INTENB = 100
 DEFINITIONS AND DEVICE INFO

SEQ 0050

347
 348
 349 100000
 350 040000
 351 020000
 352 010000
 353 004000
 354 002000
 355 001000
 356 000400
 357 000200
 358 000100
 359 000040
 360 000020
 361 000010
 362 000004
 363 000002
 364 000001
 365
 CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 4
 PCLEXR.P11 17-MAR-78 11:27

;BIT DEFINITIONS:
 B15 = 100000
 B14 = 40000
 B13 = 20000
 B12 = 10000
 B11 = 4000
 B10 = 2000
 B09 = 1000
 B08 = 400
 B07 = 200
 B06 = 100
 B05 = 40
 B04 = 20
 B03 = 10
 B02 = 4
 B01 = 2
 B00 = 1

SEQ 0051

367 .SBTTL PCL11 EXERCISER MAIN PROCEDURE
 368
 369 .ENABL ABS
 370 000000 = 0
 371 000200
 372 .REPT 128.
 373 .WORD .+2,0 ;TRAP CATCHERS
 374
 375 .ENDR
 376 = 200
 377 000200 000167 001574 JMP PCLEX ;PROGRAM STARTS AT 200
 378 000204 012706 002000 MOV #STKTOP,SP ;AND RESTARTS AT 204
 379 000210 000167 002354 JMP PCREST
 380
 381 002000 = 2000
 382 002000 RELZRO =
 383 002000 STKTOP =
 384
 385 PCLEX: ;*****START HERE*****
 386 002000 012706 002000 MOV #STKTOP,SP ;SET STACK POINTER TO TOP
 387 002004 016700 014070 MOV TXDEV,R0 ;PREPARE TO GENERATE TXM ADDR
 388 002010 010067 014002 MOV R0,TCR ;GENERATE TCR ADDRESS
 389 002014 062700 000002 ADD #2,R0
 390 002020 010067 013774 MOV R0,TSR ;GENERATE TSR ADDRESS
 391 002024 062700 000002 ADD #2,R0
 392 002030 010067 013766 MOV R0,TSDB ;GENERATE TSDB ADDRESS
 393 002034 062700 000002 ADD #2,R0
 394 002040 010067 013760 MOV R0,TSBC ;GENERATE TSBC ADDRESS
 395 002044 062700 000002 ADD #2,R0
 396 002050 010067 013752 MOV R0,TSBA ;GENERATE TSBA ADDRESS
 397 002054 062700 000002 ADD #2,R0
 398 002060 010067 013744 MOV R0,TMMR ;GENERATE TMMR ADDRESS

```

L 3
399 002064 005200 INC R0
400 002066 010067 013740 MOV R0,TMMRH ;AND TMMR HIGH BYTE
401 002072 005200 INC R0
402 002074 010067 013734 MOV R0,TSCRC ;GENERATE TSCRC ADDRESS
403 002100 016767 013776 013744 MOV TXVEC,TXMVEC ;GENERATE TXMVEC ADDRESS
404
405 002106 016700 013772 MOV RCDEV,R0 ;PREPARE TO GENERATE RCVR ADDRS
406 002112 010067 013720 MOV R0,RCR ;GENERATE RCR ADDRESS
407 002116 062700 000002 ADD #2,R0
408 002122 010067 013712 MOV R0,RSR ;GENERATE RSR ADDRESS
409 002126 062700 000002 ADD #2,R0
410 002132 010067 013704 MOV R0,RDDB ;GENERATE RDDB ADDRESS
411 002136 062700 000002 ADD #2,R0
412 002142 010067 013676 MOV R0,RDBC ;GENERATE RDBC ADDRESS
413 002146 062700 000002 ADD #2,R0
414 002152 010067 013670 MOV R0,RDBA ;GENERATE RDBA ADDRESS
415 002156 062700 000004 ADD #4,R0
416 002162 010067 013662 MOV R0,RDCRC ;GENERATE RDCRC ADDRESS
417 002166 016767 013714 013660 MOV RCVEC,RCVVEC ;GENERATE RCVVEC ADDR
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 5
PCLEXR.P11 17-MAR-78 11:27 PCL11 EXERCISER MAIN PROCEDURE SEQ 0052

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 015332 CLRB REQINP ;CLR INPUT REQUEST
424 002220 105067 015327 CLRB CMDENT ;CLR COMMAND ENTERED FLAG
425 002224 005067 015362 CLR PCLGO ;CLR XMTR GO FLAG
426 002230 012767 000204 015436 MOV #204,RSHOLD ;SAVE RESTART ADDRESS
427 .JRP LC <TI,TO> ;INITIALIZE IO QUEUES TO EMPTY
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 014210 CLR TIQ
(1) 002242 016767 014210 014212 MOV TIQ+QTOP,TIQ+QFRONT
(1) 002250 016767 014202 014206 MOV TIQ+QTOP,TIQ+QBACK
(1) 002256 005067 014254 CLR TOQ
(1) 002262 016767 014254 014256 MOV TOQ+QTOP,TOQ+QFRONT
(1) 002270 016767 014246 014252 MOV TOQ+QTOP,TOQ+QBACK
434 002276 012737 000340 177776 MOV #340,@#PS ;DISABLE INTERRUPTS
435 002304 012777 007600 013540 MOV #XMTINT,@TXMVEC ;:::SET UP XMTR INTR VECTOR
436 002312 012777 011124 013534 MOV #RCVINT,@RCVVEC ;:::SET UP RCVR INTR VECTOR
437 002320 012777 015562 013540 MOV #TTINT,@TIVECT ;:::SET UP TTY INTR VECTOR
438 002326 016700 013520 MOV TXMVEC,R0
439 002332 012760 000240 000002 MOV #TXPRIO*32.,2(R0) ;:::SET TXM PRIORITY
440 002340 016700 013510 MOV RCVVEC,R0
441 002344 012760 000240 000002 MOV #RCPRIO*32.,2(R0) ;:::SET RCVR PRIORITY
442 002352 016700 013510 MOV TIVECT,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 030736 PRST: MOV #RSADD,R0 ;:::BUFFER ADDR FOR OCTPNT IN R1
449 002452 016701 015216 MOV RSHOLD,R1 ;:::DATA FOR OCTPNT IN R0
450 002456 012702 177777 MOV #-1,R2 ;:::DON'T COMPRESS BLANKS
451 002462 004767 030126 JSR PC,OCTJSP ;:::COMPUTE RESTART ADDRESS
452 002466 CALL PNTLIN,<RSTMSG> ;:::PRINT RESTART ADDRESS
453 002504 012737 002550 000004 MOV #STR,@#4 ;:::SET UP TO TEST FOR CLOCK

```

```

M 3
454 002512 005067 015044 CLR KWFLG ;;;CLEAR KW11 FLAG
455 002516 005777 013346 TST @LCS ;;;ANY CLOCK?
456 002522 012767 177777 015032 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 STR- CMP (SP)+,(SP)+ ;;;NO CLOCK, CLEAR STACK
461 002552 005067 015004 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 015004 BIS #B15,RCVEV ;SET RCVR EVENT FLAG
467 002610 052777 000100 013240 BIS #B06,@TTRCSR ;SET TTY KBD INTR ENAB
468 002616 005067 015046 CLR ESCFLG ;CLEAR CNTRL-C FLAG

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 5-1
PCLEXR.P11 17-MAR-78 11:27 PCL11 EXERCISER MAIN PROCEDURE
SEQ 0053

469 002622 005067 014742 CLR OLDEV ;CLR QUEUE LOAD EVENT FLAG
470 002626 005067 015020 CLR RJCTF ;CLEAR REJECT FLAG
471 002632 005067 015016 CLR TRNKF ;CLEAR TRUNCATE FLAG
472 002636 005067 014754 CLR SPCEV ;CLEAR MST DWN EVENT
473 002642 005067 014752 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 014670 TRAP4: MOV (SP),SUMSV ;SAVE TRAP ADDRESS
480 002706 162767 000002 014662 SUB #2,SUMSV ;ALIGN IT -2
481 002714 012700 031053 MOV #TRP4AD,R0 ;SHOW OCT CONV RTN RIGHT ADDRESS
482 002720 016701 014652 MOV SUMSV,R1 ;FOR ASCII CHARS.
483 002724 004767 027664 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

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 6
PCLEXR.P11 17-MAR-78 11:27 MAIN LOOP
SEQ 0054

489 .SBTTL MAIN LOOP
490
491
492 002762 005767 013464 PCLOOP: TST TIQ ;IS TTY INPUT QUEUE EMPTY?
493 002766 001406 BEQ NXT0 ;YES, TEST ANOTHER FLAG
494 002770 CALL TTINP ;NO, PROCESS A CHARACTER
495 003004 105767 014543 NXT0: TSTB CMDENT ;HAS A COMMAND BEEN ENTERED?
496 003010 100006 BPL NXT1 ;NO, TEST ANOTHER FLAG
497 003012 CALL COMENT ;YES, PROCESS COMMAND
498 003026 005767 013504 NXT1: TST TOQ ;IS TTY OUTPUT QUEUE EMPTY?
499 003032 001406 BEQ NXT2 ;YES, TEST ANOTHER FLAG
500 003034 CALL TTOUT ;NO,OUTPUT A CHAR IF DEV RDY
501 003050 005767 014514 NXT2: TST OLDEV ;IS ADDR QUEUE EMPTY?
502 003054 100006 BPL NXT3 ;NO, TEST ANOTHER FLAG
503 003056 CALL ADQLD ;YES, LOAD ADDR QUEUE
504 003072 005767 014474 NXT3: TST DATGEV ;IS DATA GEN FLAG SET?
505 003076 100006 BPL NXT4 ;NO, TEST ANOTHER FLAG
506 003100 CALL DATGEN ;YES,GENERATE NEW RANDOM DATA
507 003114 005767 014466 NXT4: TST TXMEV ;IS XMTR EVENT FLAG SET?
508 003120 100006 BPL NXT5 ;NO, TEST ANOTHER FLAG
509 003122 CALL TXMIT ;YES, ENTER XMIT MODULE
510 003136 005767 014452 NXT5: TST RCVEV ;IS RCVR EVENT FLAG SET?
511 003142 100006 BPL NXT6 ;NO, TEST ANOTHER FLAG

```

512	003144				CALL	RCV		:YES, ENTER RCVR MODIILE
513	003160	005767	014432	NXT6:	TST	SPCEV		:IS SPECIAL EVENT FLAG SET?
514	003164	100006			BPL	NXT7		:NO, TEST ANOTHER FLAG
515	003166				CALL	SPEC		:YES, HANDLE SPECIAL EVENT
516	003202	005767	014372	NXT7:	TST	STSEV		:IS STATUS EVENT FLAG SET?
517	003206	100006			BPL	NXT8		:NO, TEST ANOTHER FLAG
518	003210				CALL	STATUS		:YES, OUTPUT STATUS
519	003224	005767	014344	NXT8:	TST	SUMEV		:IS SUMMARY EVENT FLAG SET?
520	003230	100006			BPL	NXT9		:NO, TEST ANOTHER FLAG
521	003232				CALL	SUMRY		:YES, OUTPUT ERROR SUMMARY
522	003246	005767	014346	NXT9:	TST	XSPCEV		:IS EXTRA-SPECIAL EVENT SET?
523	003252	100006			BPL	NXT10		
524	003254				CALL	XSPEC		:YES, HANDLE NOW MASTER.
525	003270	005767	014270	NXT10:	TST	TEREV		:IS XMTR ERROR EVENT SET?
526	003274	100006			BPL	NXT11		:NO, TEST ANOTHER FLAG
527	003276				CALL	TEROS		:YES, OUTPUT XMTR ERROR TABLE
528	003312	005767	014250	NXT11:	TST	REREV		:IS RCVR ERROR EVENT SET?
529	003316	100006			BPL	NXT12		:NO,
530	003320				CALL	REROS		:YES, OUTPUT RCVR ERROR TABLE
531	003334	000167	177422	NXT12:	JMP	PCLOOP		:STAY IN MAIN LOOP

;TRAP TO 4 HANDLER

536	003340			ERTRAP:	REGSAV			:::SAVE R0...R5
537	003344				CALL	PNTLIN,<ERTMSG>		:::PRINT TRAP MESSAGE
538	003362	005067	014226		CLR	RCVEV		:::CLEAR RCVR EVENT FLAG
539	003366	005067	014220		CLR	PCLGO		:::CLR PCL GO FLAG
540	003372	052777	000100	012456	BIS	#B06,@TTRCSR		:::SET TTY KBD INTR ENAB
541	003400	005037	177776		CLR	@NPS		:::DROP CP PRIORITY
542	003404				REGRES			
543	003410	012706	002000		MOV	#STKTOP,SP		:FIX STACK
544	003414	000167	177226		JMP	RTRYA		:ENTER COMMAND MODE.

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 7
PCLEXR.P11 17-MAR-78 11:27 COMMAND PROCESSORS:

SEQ 0055

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

: LOAD OR CLEAR THE TRANSMITTER ADDRESS (HARDWARE) SILO.
: IF THERE ARE NO ARGUMENTS, CLEAR THE ADDRESS SILO AND SET
: AUTO ADDRESS. IF THERE ARE ARGUMENTS, LOAD THE TOTAL NUMBER
: OF ARGUMENTS (INCLUDING PAD VALUES) INTO THE SILO AS MANY TIMES
: AS THE TOTAL WILL GO INTO 50. LOCATIONS.
:
: IF ANY 2 SEQUENTIAL ARGUMENTS ARE THE SAME, SEPARATE THEM WITH
: A PAD VALUE OF '0'. IF THE FIRST ARGUMENT IS THE SAME AS THE LAST
: ARGUMENT, INSERT A PAD VALUE OF '0' AFTER THE LAST ARGUMENT.
:
: ARGUMENTS HIGHER THAN 37 (OCTAL) WILL NOT BE ACCEPTED
: ARGUMENTS LOWER THAN 1 WILL NOT BE ACCEPTED.
:
: ARGUMENTS MUST BE NUMERIC. DECIMAL ARGUMENTS MUST BE FOLLOWED BY
: A DECIMAL POINT. (18.)
: PROC CPSILO

566	003420	012701	017710	CPSILO:	MOV	#ADSILO,R1	:::ENTRY POINT::
567	003424	016602	000002		MOV	2(SP),R2	:SET R1 TO POINT TO SILO BUFFER
568	003430	005302			DEC	R2	:GET # OF ARGS. INTO R2
569	003432	001002			BNE	11\$	:O.K IF THERE ARE SOME
570	003434	000167	000446		JMP	CLEAV	:OTHERWISE, EXIT.

571	003440	010267	014166		118:	MOV	R2,PADFLG		:SAVE COUNT FOR LATER USE.
572	003444	020227	000062			CMP	R2,#50.		:ARE THERE MORE THAN 50 OBJECTS?
573	003450	101402				BLOS	1\$		:NO, CONTINUE
574	003452	000167	000422			JMP	SYNRTM		:YES, ERROR
575	003456	010267	014210		1\$:	MOV	R2,OBJCNT		:SAVE OBJECT COUNT
576	003462	010200				MOV	R2,R0		:TURN LOOK-UP AROUND
577	003464	005300				DEC	R0		
578	003466					MULT	6,P0,R3		
579	003476	060600				ADD	SP,R0		
580	003500	062700	000004			ADD	#4,R0		
581	003504	016067	000004	014110		MOV	4(R0),FIRST		:SAVE FIRST ITEM
582	003512	012767	000000	014104		MOV	#0,NEXT		:SAVE CURRENT ITEM
583	003520	021027	000002		CPMLP:	CMP	(R0),#2		:IS IT CLASS 2?
584	003524	001402				BEQ	2\$		:YES, O.K.
585	003526	000167	000346			JMP	SYNRTM		:NO, ERROR
586	003532	026067	000004	014064	2\$:	CMP	4(R0),NEXT		:IS THIS ITEM SAME AS LAST?
587	003540	001004				BNE	3\$		:NO, O.K.
588	003542	112721	000000			MOVB	#0,(R1)+		:YES, INSERT A PAD VALUE
589	003546	005267	014120			INC	OBJCNT		:KEEP OBJECT COUNT UP TO DATE
590	003552	116011	000004		3\$:	MOVB	4(R0),(R1)		:PUT OBJECT INTO BUFFER
591	003556	001002				BNE	4\$		:ERROR IF OBJECT IS 0
592	003560	000167	000314			JMP	SYNRTM		
593	003564	122711	000037		4\$:	CMPB	#37,(R1)		:ERROR IF IT WAS > 37
594	003570	103002				BHIS	5\$		
595	003572	000167	000302			JMP	SYNRTM		
596	003576	112167	014022		5\$:	MOVB	(R1)+,NEXT		:SAVE REAL ITEM
597	003602	062700	177772			ADD	#-6,R0		:SET UP TO GET NEXT OBJECT
598	003606	005302				DEC	R2		:ARE WE DONE LOADING BUFF?
599	003610	001343				BNE	CPMLP		:NO, KEEP GOING

CZPLAA0	PCL11	EXERCISER	V-02	MACY11	30A(1052)	28-APR-78	13:50	PAGE 7-1
PCLEXR.P11		17-MAR-78	11:27		COMMAND PROC. FOR SILO (LOAD)			

600	003612	026767	014006	014002		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	014040			INC	OBJCNT		:AND KEEP OBJECT COUNT UP TO DATE
604	003632	026727	014034	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	012156	7\$:	BISB	#B05+B04,@TMMRH		:PREPARE TO LOAD ADDR SILO
608	003654	012700	000062			MOV	#50.,R0		:HOLD SILO SIZE
609	003660	026767	014006	013744		CMP	OBJCNT,PADFLG		:IS OBJECT COUNT DIFFERENT THAN START?
610	003666	001002				BNE	SILLD		:YES, LEAVE SOMETHING IN PADFLG (SET)
611	003670	005067	013736		8\$:	CLR	PADFLG		:NO, CLEAR PAD FLAG
612	003674	016702	013772		SILLD:	MOV	OBJCNT,R2		:GET NO. OF OBJECTS
613	003700	012701	017710			MOV	#ADSILO,R1		:GET OBJECT BUFFER
614	003704	112177	012120		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	013752			SUB	OBJCNT,R0		:YES, IS THAT ALL THAT'LL FIT?
618	003720	020067	013746			CMP	R0,OBJCNT		
619	003724	022363				BGE	SILLD		:IF NOT LOAD THEM AGAIN
620	003726	132777	000001	012076		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	011776		BISB	#B01,@TMMRH		: THE CURRENT MASTER'.
627	004034	005767	013572		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	013602	9\$:	MOV	#-1,ESCFLG		:FLAG COMENT TO ISSUE NEW '\$'.

SEQ 0056

C
P

631	004066	142777	000020	011736		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	011716	CLEAV:	BISB	#B05+B04,@TMMRH		:RESET AUTO ADDR & CLR SILO
640	004114	012767	177777	013546		MOV	#-1,ESCFLG		:FLAG COMENT TO ISSUE NEW 'S'.
641	004122	000241				CLC			:CLEAR 'C' IN CASE IT WAS SET
642	004124	000167	177746			JMP	CPSEX		:RETURN

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 8
PCLEXR.P11 17-MAR-78 11:27 COMMAND PROC. FOR MASTER, SECONDARY AND R.I.B.

SEQ 0057

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

645

646 :PROCESSOR FOR 'MASTER SET (OR CLEAR)' COMMAND

647 : SET MASTER, OR CLEAR MASTER

648

649 004130 PROC CPMAS

(1)

(1) 004130 CPMAS:

650 004130 022766 000002 000002 CMP #2,2(SP) ;**ENTRY POINT**

651 004136 001402 BEQ CPMOK ;GET # OF ARGUMENTS

652 004140 000261 SEC ;OKAY IF 2

653 004142 000423 BR CPMRET ;OTHERWISE, SYNTAX ERROR

654 004144 010603 CPMOK: MOV SP,R3 ;GET 2ND WORD OF COMMAND

655 004146 062703 000004 ADD #4,R3

656 004152 KEYWD R3,#SCTBL ;PROCESS IT

657 004164 103446 BCS KWDERT ;SYNTAX ERROR IF 'C' SET

658 004166 117702 011640 MOVB @TMMRH,R2 ;GET OLD TMMR

659 004172 142702 000001 BICB #1,R2 ;REMOVE OLD STATE OF MASTER

660 004176 052602 BIS (SP)+,R2 ;SET NEW STATE OF MASTER

661 004200 110277 011626 MOVB R2,@TMMRH ;LOAD NEW TMMR

662 004204 012767 177777 013456 MOV #-1,ESCFLG ;FLAG COMENT TO ISSUE NEW 'S'.

663 004212 CPMRET: RETURN ;EXIT

664

665

666 :PROCESSOR FOR 'SECONDARY SET (OR CLEAR)' COMMAND

667 : SET SECONDARY, OR CLEAR SECONDARY

668

669 004214 PROC CPSEC

(1)

(1) 004214 CPSEC:

670 004214 022766 000002 000002 CMP #2,2(SP) ;**ENTRY POINT**

671 004222 001402 BEQ CPSOK ;ARE THERE 2 ARGUMENTS?

672 004224 000261 SEC ;IF YES, PROCEED

673 004226 000424 BR CPSRET ;OTHERWISE, SYNTAX ERROR

674 004230 010603 CPSOK: MOV SP,R3 ;GET 2ND WORD OF COMMAND

675 004232 062703 000004 ADD #4,R3

676 004236 KEYWD R3,#SCTBL ;PROCESS IT

677 004250 103414 BCS KWDERT ;SYNTAX ERROR IF 'C' SET

678 004252 117702 011554 MOVB @TMMRH,R2 ;GET OLD TMMR

679 004256 142702 000002 BICB #2,R2 ;REMOVE OLD STATE OF SECONDARY

680 004262 006316 ASL (SP)

681 004264 052602 BIS (SP)+,R2 ;SET NEW STATE OF SECONDARY

682 004266 110277 011540 MOVB R2,@TMMRH ;LOAD NEW TMMR

683 004272 012767 177777 013370 MOV #-1,ESCFLG ;FLAG COMENT TO ISSUE NEW 'S'.

684 004300 CPSRET: RETURN ;EXIT

685

686

687 004302 032600 KWDERT: BIT (SP)+,R0 ;POP BAD WORD OFF STACK

```

688 004304 000775          BR      CPSRET      D 4          ;EXIT
689
690
691          ;PROCESSOR FOR 'RIB SET' OR 'RIB CLEAR'
692          ; SET 'RIB' BIT IN XMTR COMMAND WORD LOCATION 'TXMST', OR CLEAR IT
693
694 004306          PROC      CPRIB
(1)
CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 8-1
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR MASTER, SECONDARY AND R.I.B.          SEQ 0058

(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          RGR      (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 013272          MOV      #-1,ESCF LG          ;FLAG COMMENT TO ISSUE NEW 'S'
711 004376          CPRRET: RETURN          ;BACK TO CALLER

CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 9
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR RANGE          SEQ 0059

713          .SBTTL  COMMAND PROC. FOR RANGE
714
715
716          ; SET RANGE OF RECEIVER ADDRESSES THAT THIS TRANSMITTER SHOULD TALK TO.
717          ;
718          ; ACCEPT A LOWER AND UPPER RECEIVER ADDRESS (IN THAT ORDER).
719          ; GENERATE A LIST (SET ACTIVE FLAGS) OF RCVR ADDRESSES FROM LOW ADDRESS
720          ; TO HIGH ADDRESS INCLUSIVE.
721
722
723 004400          PROC      CPRANG
(1)
(1) 004400          CPRANG:          ;**ENTRY POINT**
724 004400 016600 000002          MOV      2(SP),R0          ;GET # OF OBJECTS
725 004404 022700 000003          CMP      #3,R0          ;3 OBJECTS?
726 004410 001047          BNE      SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
727 004412 016600 000004          MOV      4(SP),R0          ;GET CLASS OF FIRST OBJECT
728 004416 022700 000002          CMP      #2,R0          ;IS IT CLASS 2?
729 004422 001042          BNE      SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
730 004424 016601 000010          MOV      10(SP),R1          ;GET 'TO' ARGUMENT INTO R1
731 004430 016600 000012          MOV      12(SP),R0          ;GET CLASS OF SECOND OBJECT
732 004434 022700 000002          CMP      #2,R0          ;IS IT CLASS 2?
733 004440 001033          BNE      SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
734 004442 016602 000016          MOV      16(SP),R2          ;GET 'FROM' ARGUMENT INTO R2
735 004446 020102          CMP      R1,R2          ;IS 'FROM' LOWER THAN 'TO'?
736 004450 101427          BLOS      SYNRTN          ;IF NOT, INDICATE SYNTAX ERROR
737 004452 022701 000037          CMP      #37,R1          ;IS 'TO' > 37?
738 004456 103424          BLO      SYNRTN          ;IF SO, INDICATE SYNTAX ERROR
739 004460 010200          MOV      R2,R0          ;FIND 'FROM' ENTRY
740 004462 001422          BEQ      SYNRTN          ;IF 'FROM'=0, SYNTAX ERROR
741 004464 006300          ASI      R0

```

```

E 4
742 004466 006300 ASL R0
743 004470 010003 MOV R0,R3
744 004472 006300 ASL R0
745 004474 060300 ADD R3,R0
746 004476 062700 017776 ADD #RADB0,R0 ;R0 CONTAINS 'FROM' ADDRESS
747 004502 012710 177777 (CPFIL: MOV #-1,(R0) ;SET RCV ADDR ACTIVE FLAG.
748 004506 062700 000014 ADD #14,R0 ;UPDATE TABLE POINTER
749 004512 005202 INC R2 ;INCREMENT 'FROM' ADDRESS
750 004514 020201 CMP R2,R1 ;UNTIL EQUAL TO 'TO' ADDRESS
751 004516 003771 BLE CPFIL ;EXIT WHEN COMPLETE.
752 004520 012767 177777 013142 MOV #-1,ESCFLG ;FLAG COMENT TO ISSUE NEW '$'
753 004526 (CPRTN: RETURN
754
755 004530 000261 SYNRTN: SEC ;SET 'C' BIT TO INDICATE SYNTAX ERROR
756 004532 000167 177770 JMP CPRTN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 10
PCLEXR.P11 17-MAR-78 11:27 COMMAND PROC. FOR ADD AND DELETE SEQ 0060

758 .SBTTL COMMAND PROC. FOR ADD AND DELETE
759
760 ; ADD A RECEIVER ADDRESS TO THE LIST (SET ACTIVE FLAG) OF RCVR ADDRESSES
761 ;
762 ; ACCEPT ARGUMENTS (AT LEAST 1) CHECK THAT THEY FALL IN A RANGE FROM
763 ; 1 TO 37; THEN SET THE CORRESPONDING ACTIVE FLAGS IN THE STAT
764 ; TABLE.
765
766 004536 PROC CPADD
(1)
(1) 004536 CPADD: ;**ENTRY POINT**
767 004536 016602 000002 MOV 2(SP),R2 ;GET # OF ARGUMENTS INTO R2
768 004542 005302 DEC R2 ;IGNORE KEYWORD OBJECT
769 004544 010600 MOV SP,R0 ;GENERATE ADDRESS OF ARGUMENTS
770 004546 062700 000004 ADD #4,R0 ;POINT R0 AT 1ST OBJECT CLASS
771 004552 021027 000002 CPALP: CMP (R0),#2 ;IS CLASS = 2?
772 004556 001027 BNE SYNTRA ;NO, INDICATE SYNTAX ERROR
773 004560 016001 000004 MOV 4(R0),R1 ;GET OBJECT INTO R1
774 004564 001424 BEQ SYNTRA ;IF IT'S 0, SYNTAX ERROR
775 004566 022701 000037 CMP #37,R1 ;IF IT'S >37, SYNTAX ERROR
776 004572 103421 BLO SYNTRA ;FIND THE TABLE ELEMENT
777 004574 006301 ASL R1
778 004576 006301 ASL R1
779 004600 010103 MOV R1,R3
780 004602 006301 ASL R1
781 004604 060301 ADD R3,R1
782 004606 062701 017776 ADD #RADB0,R1
783 004612 012711 177777 MOV #-1,(R1) ;SET ACTIVE FLAG FOR THIS TABLE ENTRY
784 004616 062700 000006 ADD #6,R0 ;SET UP FOR NEXT ARGUMENT
785 004622 005302 DEC R2 ;ARE WE DONE?
786 004624 001352 BNE CPALP ;NO, CONTINUE.
787 004626 012767 177777 013034 MOV #-1,ESCFLG ;FLAG COMENT TO ISSUE NEW '$'
788 004634 (PARTN: RETURN
789 004636 000261 SYNTRA: SEC ;SET 'C' BIT FOR SYNTAX ERROR
790 004640 000167 177770 JMP CPARTN
791
792 ;DELETE RECEIVER ADDRESSES AND ASSOCIATED 'ATTEMPTS' AND 'SUCCESSSES'.
793
794 004644 PROC CPDEL
(1)
(1) 004644 CPDEL: ;**ENTRY POINT**
795 004644 016602 000002 MOV 2(SP),R2 ;GET # OF ARGUMENTS INTO R2
796 004650 005302 DEC R2 ;IGNORE KEYWORD OBJECT
797 004652 010600 MOV SP,R0 ;GENERATE ADDRESS OF ARGUMENTS
798 004654 062700 000004 ADD #4,R0 ;POINT R0 AT 1ST OBJECTS CLASS

```

```

799 004660 021027 000002      CPDLP:  CMP      (R0),#2      ;IS CLASS =2
800 004664 001034      BNE      SYNRD      ;NO, INDICATE SYNTAX ERROR
801 004666 016001 000004      MOV      4(R0),R1      ;GET OBJECT INTO R1
802 004672 001431      BEQ      SYNRD      ;IF IT'S 0, SYNTAX ERROR
803 004674 022701 000037      CMP      #37,R1
804 004700 103426      BLO      SYNRD      ;IF IT'S >37, SYNTAX ERROR
805 004702 006301      ASL      R1      ;FIND TABLE ELEMENT
806 004704 006301      ASL      R1
807 004706 010103      MOV      R1,R3
808 004710 006301      ASL      R1
809 004712 060301      ADD      R3,R1

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 10-1
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR ADD AND DELETE

810 004714 062701 017776      ADD      #RADBO,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 012714      MOV      #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW '$'
821 004754      LPDRTN: RETURN
822 004756 000261      SYNRD: SEC      ;SET 'C' BIT FOR SYNTAX ERROR
823 004760 000167 177770      JMP      CPDRTN

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 11
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR CLEAR, STATUS, AND CONTINUE

825      .SBTTL  COMMAND PROC. FOR CLEAR, STATUS, AND CONTINUE
826
827
828
829      ; CLEAR THE ENTIRE LIST OF RECEIVER ADDRESSES (CLEAR ACTIVE FLAGS)
830      ;
831      ; TO USE THIS ROUTINE BY A CALL MACRO, ENTER AT PRCLR AS FOLLOWS :
832      ;
833      ;      CALL      PRCLR
834      ;      PROC      CPCLR      ;CLEAP STATUS TABLE
835      ;
836      ;**ENTRY POINT**
837      MOV      2(SP),R2      ;GET NUMBER OF OBJECTS INTO R2
838      CMP      #1,R2      ;ARE THERE ANY ARGUMENTS?
839      BNE      SYNRD      ;IF SO, INDICATE SYNTAX ERROR
840      MOV      #-1,ESCFLG      ;FLAG COMENT TO ISSUE NEW '$'
841      PROC      PRCLR
842      ;**ENTRY POINT**
843      MOV      #37,R1      ;SAVE COUNT OF TABLE ELEMENTS
844      MOV      #RADB,R0      ;SET UP TO CLR ACTIVE FLAGS
845      CPCLRC:  CLR      (R0)      ;CLEAR ALL ELEMENTS OF ENTRY
846      ADD      #4,R0
847      CLR      (R0)+
848      CLR      (R0)+
849      CLR      (R0)+
850      CLR      (R0)+
851      DEC      R1      ;DONE?
852      BNE      CPCLRC      ;IF NOT, CONTINUE
853      CLR      AQQ
854      MOV      AQQ+QTOP,AQQ+QFRONT      ;INITIALIZE ADDR QUEUE TO EMPTY

```

SEQ 0061

SEQ 0062

```

853 005050 016767 011266 011272      MOV      A00+0TOP,A00+0BACK
854 005056                                CPCLRT: RETURN
855 005060 000261                        SYNTRC: SEC      ;AND LEAVE
856 005062 000167 177770                JMP      CPCLRT    ;SET 'C' BIT FOR SYNTAX ERROR
857
858
859
860      ; SET THE STATUS EVENT FLAG SO THAT THE STATUS TABLE WILL BE PRINTED.
861
862 005066      PROC      CPSTAT
(1)
(1) 005066      CPSTAT:      ;**ENTRY POINT**
863 005066 022766 000001 000002      CMP      #1,2(SP)      ;ARE THERE ANY ARGUMENTS?
864 005074 001011                        BNE      SYNTRS      ;IF SO, INDICATE SYNTAX ERROR
865 005076 052767 100000 012474      BIS      #B15,STSEV    ;SET STATUS EVENT FLAG
866 005104 005067 012472                        CLR      STPNTR    ;0 STPNTR INDICATES HEADER FIRST
867 005110 042777 000100 010700      BIC      #B06,STCLR    ;CLR TXM INTERRUPT ENABLE
868 005116      CPSRTN: RETURN
869 005120 000261      SYNTRS: SEC      ;SET 'C' BIT FOR SYNTAX ERROR
870 005122 000167 177770                JMP      CPSRTN
871
872
873
874      ; CONTINUE EXERCISING. DO NOT AFFECT THE STATUS TABLE
CZPLAA0 PCL11 EXERCISER V-02  MACY11 30A(1052) 28-APR-78 13:50 PAGE 11-1
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR CLEAR, STATUS, AND CONTINUE
SEQ 0063

875      ; DO NOT AFFECT THE ERROR TABLE
876      ; DO NOT AFFECT THE RCVR ADDRESS QUEUE
877      ; SIMPLY SET PCLGO
878
879
880
881 005126      PROC      CPCNT
(1)
(1) 005126      CPCNT:      ;**ENTRY POINT**
882 005126 022766 000001 000002      CMP      #1,2(SP)      ;ARE THERE ANY ARGUMENTS?
883 005134 001027                        BNE      SYNTRN      ;IF SO, INDICATE SYNTAX ERROR
884 005136 012767 100000 012446      MOV      #B15,PCLGO    ;SET PCL GO FLAG
885 005144                        CALL      PNCRLF      ;PRINT CR & LF
886 005160                        CALL      PNTLIN,<EXCNT>    ;PRINT 'EXERCISER CONTINUING'
887 005176 005767 012360                        TST      KWFLG      ;GOT A CLOCK?
888 005202 001403                        BEQ      CPCNRT      ;NO
889 005204 012777 000100 010656      MOV      #B06,ALCS      ;RE-ENABLE CLOCK.
890 005212      CPCNRT: RETURN
891 005214 000261      SYNTRN: SEC      ;SET 'C' BIT FOR SYNTAX ERROR
892 005216 000167 177770                JMP      CPCNRT
CZPLAA0 PCL11 EXERCISER V-02  MACY11 30A(1052) 28-APR-78 13:50 PAGE 12
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR INIT, SUMMARY, AND GO
SEQ 0064

894      .SBTTL  COMMAND PROC. FOR INIT, SUMMARY, AND GO
895
896      ;TO USE CALL MACRO, ENTER AT PRINTIT AND PARCLR
897      ; CLEAR STATUS TABLE, SUMMARY TABLE, ERROR TABLE
898      ; CLEAR 'RIB' IN XMTR COMMAND WORD 'TXMST'
899      ; CLEAR CLOCK DATA
900      ; CLEAR TRANSMITTER HARDWARE
901      ; INITIALIZE DATA PATTERN
902
903
904
905 005222      PROC      CPINIT
(1)

```

```

(1) 005222
906 005222 022766 000001 000002
907 005230 001114
908 005232 012767 177777 012430
909 005240
(1)
(1) 005240
910 005240
911 005246 105067 012303
912 005252 105067 012300
913 005256 105067 012272
914 005262 005067 012272
915 005266 042767 100000 002104
916 005274
917 005310
(1)
(1) 005310
918 005310 012700 024640
919 005314 022710 177777
920 005320 001406
921 005322 005060 000004
922 005326 062700 000006
923 005332 000167 177756
924 005336 012700 001350
925 005342 012701 025062
926 005346 012721 000000
927 005352 005300
928 005354 001374
929 005356 012700 000037
930 005362 012701 020012
931 005366 062701 000004
932 005372 005021
933 005374 005021
934 005376 005021
935 005400 005021
936 005402 005300
937 005404 001370
938 005406 016767 012224 012224
939 005414 016767 012222 012222
940 005422 005067 012160
941 005426 005067 012142
942 005432 005067 012142
943 005436 005067 012122
CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 12-1
PCLEXR.P11 17-MAR-78 11:27 COMMAND PROC. FOR INIT, SUMMARY, AND GO
944 005442 005067 012120
945 005446 005067 012176
946 005452 052767 100000 012112
947 005460
948 005462 000261
949 005464 000167 177770
950
951
952
953
954
955
956 005470
(1)
(1) 005470
957 005470 022766 000001 000002
958 005476 001006

```

M 4

CPINIT:

CMP #1,2(SP)
 BNE SYNTIN
 MOV #1,ESCF LG
 PROC PRINT

;**ENTRY POINT**
 ;ARE THERE ANY ARGUMENTS?
 ;IF SO, INDICATE SYNTAX ERROR
 ;FLAG COMENT TO ISSUE NEW 'S'

PRINIT:

BDINIT XMTR
 CLRB SECNDS
 CLRB MINUTS
 CLRB TICKS
 CLR HOURS
 BIC #B15,TXMST
 CALL PRCLR
 PROC PARCLR

;**ENTRY POINT**
 ;CLEAR XMITTER.
 ;CLEAR SECONDS (TIME)
 ;CLEAR MINUTES
 ;CLEAR TICKS
 ;CLEAR HOURS
 ;CLEAR RIB IN COMMAND WORD FOR XMTR
 ;CLR STATUS TABLE & ADDR QUEUE

PARCLR:

MOV #ERTBL,R0
 ERTBCL: CMP #-1,(R0)
 BEQ ERTCD
 CLR 4(R0)
 ADD #6,R0
 JMP ERTBCL
 ERTCD: MOV #37*30,R0
 MOV #TERTBL,R1
 1\$: MOV #0,(R1)+
 DEC R0
 BNE 1\$
 MOV #37,R0
 MOV #RADB,R1
 2\$: ADD #4,R1
 CLR (R1)+
 CLR (R1)+
 CLR (R1)+
 CLR (R1)+
 DEC R0
 BNE 2\$
 MOV ORIGSD,DTSEED
 MOV MLSD,MSGLSD
 CLR TXMEV
 CLR SUMEV
 CLR STSEV
 CLR TEREV

;**ENTRY POINT**
 ;CLEAR ERROR SUMMARY TABLE
 ;IS ERR NUM = -1?
 ;IF SO, DONE CLEARING
 ;OTHERWISE, CLR OCCURENCES
 ;STEP TO NEXT ENTRY
 ;AND CONTINUE
 ;ALSO CLEAR ERROR DATA
 ;FROM DETAILED ERR TABLE

;CLEAR ATTEMPS & SUCCESSES
 ;FROM STATUS TABLE

;RESTORE ORIGINAL DATA SEED
 ;AND MSG LENGTH SEED
 ;CLR XMIT EVENT FLAG
 ;CLR SUMMARY EVENT FLAG
 ;CLR STATUS EVENT FLAG

CPINRT: RETURN
 SYNTIN: SEC
 JMP CPINRT

;CLR XMTR & RCVR ERROR EVENTS
 ;CLEAR DATA PASS NO.
 ;SET DATA GEN FLAG
 ;SET 'C' BIT FOR SYNTAX ERROR

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

PROC CPSUM

CPSUM:

CMP #1,2(SP)
 BNE SYNTSM

;**ENTRY POINT**
 ;ARE THERE ANY ARGUMENTS?
 ;IF SO, INDICATE SYNTAX ERROR

C P

C P

SEQ 0065

```

959 005500 012767 100000 012066      MOV      #B15,SUMEV      ;SET SUMMARY EVENT FLAG
960 005506 005067 012072      CLR      SUMPNT      ;0 SUMPNT INDICATES HEADER FIRST
961 005512      CPSURT: RETURN
962 005514 000261      SYNTSM: SEC      ;SET 'C' BIT FOR SYNTAX ERROR
963 005516 000167 177770      JMP      CPSURT
964
965
966
967
968      ; START THE EXERCISER
969      ; CLEAR 'ATTEMPTS' AND 'SUCCESSSES' IN STATUS TABLE
970      ; CLEAR ERROR TABLES
971      ; CAUSE RCVR ADDRESS QUEUE TO BE LOADED
972      ; SET PCLGO
973      ; IF THE RCVR ADDRESS QUEUE IS EMPTY AFTER BEING LOADED,
974      ; INDICATE THIS BY PRINTING THE FOLLOWING MESSAGE:
975      ;
976      ;     '**SORRY, I HAVE NO RECEIVER ADDRESSES.**'
977      ;
978      ; AND RETURN TO COMMAND INPUT MODE.
979
980
981
982 005522      PROC      CPGO
(1)
(1) 005522      CPGO:
983 005522 022766 000000 000002      CMP      #1,2(SP)      ;**ENTRY POINT**
984 005530 001062      BNE      SYNTAX      ;ARE THERE ANY ARGUMENTS?
985 005532 005067 012112      CLR      RCTXPS      ;IF SO, INDICATE SYNTAX ERROR
986 005536 005067 010574      CLR      A0Q      ;CLEAR PASS NO.
987 005542 105067 012007      CLRB     SECNDS      ;INIT ADDR QUEUE TO EMPTY
988 005546 105067 012004      CLRB     MINUTS      ;CLEAR SECONDS REG.
989 005552 105067 011776      CLRB     TICKS      ;CLEAR MINUTES REG.
990 005556 005067 011776      CLR      HOURS      ;CLEAR TICKS REG.
991 005562 016767 010554 010556      MOV      A0Q+QTOP,A0Q+QFRONT      ;CLEAR HOURS REG.
992 005570 016767 010546 010552      MOV      A0Q+QTOP,A0Q+QBACK
993 005576 052767 100000 011764      BIS      #B15,QLDEV      ;SET ADDR QUEUE LOAD EVENT
994 005604 012767 177777 012000      MOV      #-1,PCLGO      ;SET PC TO GO
995 005612      CALL     PARCLR      ;CLEAR STATUS & ERRORS (NOT RCVRs)
CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 12-2
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR INIT, SUMMARY, AND GO      SEQ 0066

996 005626      CALL     PNCRLF      ;PRINT CR, LF
997 005642      CALL     PNTLIN,<EXRST>      ;PRINT 'EXERCISER STARTED'.
998 005660 005767 011676      TST      KWFLG      ;GOT A CLOCK?
999 005664 001403      BEQ      STRTRT      ;NO.
1000 005666 012777 000100 010174      MOV      #B06,QLCS      ;ENABLE CLOCK INTR.
1001 005674      STRTRT: RETURN      ;EXIT
1002
1003 005676 000261      SYNTAX: SEC      ;SET 'C' TO INDICATE SYNTAX ERROR
1004 005700 000167 177770      JMP      STRTRT      ;AND EXIT
1005
1006
CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 13
PCLEXR.P11 17-MAR-78 11:27      COMMAND PROC. FOR 'ASSIGN'      SEQ 0067

1008      .SBTTL  COMMAND PROC. FOR 'ASSIGN'
1009
1010      ; ASSIGN XMTR ADDR & VECTOR AND RCVR ADDR & VECTOR.
1011      ; THE ASSIGN COMMAND MAY HAVE 0, 1, 2, 3 OR 4 ARGUMENTS:
1012      ;
1013      ;     0 = ASSIGN STANDARD ADDRESSES & VECTORS TO RCVR & XMTR
1014      ;     1 = ARGUMENT IS ASSUMED TO BE XMTR ADDRESS

```

J 4

```

:      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.

```

1015									
1016									
1017									
1018									
1019									
1020									
1021									
1022									
1023	005704			PRCC	CPASS				
(1)									
(1)	005704								
1024	005704	010600		CPASS:					**ENTRY POINT**
1025	005706	016001	000002		MOV	SP,R0			:COPY LIST POINTER
1026	005712	020127	000005		MOV	2(R0),R1			:GET NO. OF ARGUMENTS
1027	005716	101402			CMP	R1,#5			:4 ARGUMENTS?
1028	005720	000167	000372		BLOS	1\$			
1029	005724	001030		1\$:	JMP	SYN XR			:ERROR TOO MANY ARGUMENTS
1030	005726	016002	000004		BNE	PART1			:LESS THAN 4, PARTIAL ASSIGNMENT
1031	005732	022702	000002		MOV	4(R0),R2			:GET CLASS OF OBJECT
1032	005736	001402			CMP	#2,R2			:IF IT'S A NUMBER, O.K.
1033	005740	000167	000352		BEQ	2\$			
1034	005744	062700	000006		JMP	SYN XR			:ERROR WRONG CLASS
1035	005750	016002	000002	2\$:	ADD	#6,R0			:NOW GET ACTUAL NUMBER
1036	005754	032702	000003		MOV	2(R0),R2			
1037	005760	001402			BIT	#3,R2			:IS IT ON VECTOR BOUNDARY?
1038	005764	000167	000330		BEQ	3\$			:YES
1039	005766	020227	000774		JMP	SYN XR			:ERROR NOT ON VECTOR BOUNDARY
1040	005772	101402		3\$:	CMP	R2,#774			:IS IT WITHIN VECTOR AREA?
1041	005774	000167	000316		BLOS	4\$			:YES
1042	006000	010267	010102		JMP	SYN XR			:ERROR, OUTSIDE VECTOR AREA
1043	006004	005301		4\$:	MOV	R2,RCVEC			:SAVE AS RCVR VECTOR
1044	006006	020127	000004		DEC	R1			
1045	006012	001030		PART1:	CMP	R1,#4			:ARE THERE ONLY 3 ARGS?
1046	006014	016002	000004		BNE	PART2			:NO, MUST BE FEWER
1047	006020	022702	000002		MOV	4(R0),R2			:GET CLASS OF OBJECT
1048	006024	001402			CMP	#2,R2			:IF IT'S A NUMBER, O.K.
1049	006026	000167	000264		BEQ	1\$			
1050	006032	062700	000006	1\$:	JMP	SYN XR			:ERROR, OBJECT NOT A NUMBER
1051	006036	016002	000002		ADD	#6,R0			:NOW, GET THE NUMBER
1052	006042	032702	000017		MOV	2(R0),R2			
1053	006046	001402			BIT	#17,R2			:CHECK IF ITS A VALID ADDR
1054	006050	000167	000242		BEQ	2\$			:IT IS. O.K.
1055	006054	020227	164000	2\$:	JMP	SYN XR			:ERROR INVALID ADDRESS
1056	006060	103002			CMP	R2,#164000			:IS IT IN THE ADDRESS FIELD?
1057	006062	000167	000230		BHIS	3\$			:YES, O.K.
1058	006066	010267	010012	3\$:	JMP	SYN XR			:ERROR. ADDR = OUTSIDE DEVICE AREA
1059	006072	005301			MOV	R2,RCDEV			:SAVE AS RCVR ADDRESS
1060	006074	020127	000003		DEC	R1			
1061	006100	001030		PART2:	CMP	R1,#3			:ARE THERE ONLY 2 ARGS?
					BNE	PART3			:NO, MUST BE ONLY 1.

CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 13-1
 PCLEXR.P11 17-MAR-78 11:27 COMMAND PROC. FOR "ASSIGN"

SEQ 0068

1064	006112	001402			BEQ	1\$		:IF IT'S A NUMBER, O.K.
1065	006114	000167	000176		JMP	SYNXX		: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	SYNXX		:ERROR, NOT ON VECTOR BOUNDARY
1071	006142	020227	000774	2\$:	CMP	R2,#774		:IS IT WITHIN VECTOR AREA?
1072	006146	101402			BLCS	3\$		:YES
1073	006150	000167	000142		JMP	SYNXX		:ERROR, OUTSIDE VECTOR AREA
1074	006154	010267	007722	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	SYNXX		: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	SYNXX		
1087	006230	020227	164000	2\$:	CMP	R2,#164000		:IS IT IN ADDRESS FIELD?
1088	006234	103002			BHIS	3\$		:YES
1089	006236	000167	000054		JMP	SYNXX		:ERROR, OUTSIDE ADDRESS FIELD
1090	006242	010267	007632	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		MOV	#164200, TXDEV		:NO ARGUMENTS, LOAD DEFAULTS.
1094	006262	012767	000170		MOV	#170, TXVEC		
1095	006270	012767	164220		MOV	#164220, RCDEV		
1096	006276	012767	000174		MOV	#174, RCVEC		
1097	006304	012737	000340	ASEXT:	MOV	#340, @NPS		:RAISE PROC. PRIORITY
1098	006312	000167	173462		JMP	PCLEX		:GO START EXERCISER OVER.
1099								
1100	006316	000261		SYNXX:	SEC			:SET 'C' TO FLAG ERROR
1101	006320				RETURN			:RETURN

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 14
PCLEXR.P11 17-MAR-78 11:27 COMMAND DECODER AND PROCESSOR

SEQ 0069

1103					.SBTTL COMMAND DECODER AND PROCESSOR		
1104							
1105					:CONVERT COMMAND TO IT'S PROCESSING ROUTINE ADDRESS		
1106					:IF C BIT IS SET, INDICATE SYNTAX ERROR		
1107					:RETURN TO COMMAND INPUT MODE		
1108					:IF ESCFLG <> 0, CALL PRESC (CNTRL-C)		
1109							
1110	006322				PROC	COMENT	
(1)							
(1)	006322			COMENT:			:**ENTRY POINT**
1111	006322	105067	011225		CLRB	CMDENT	:CLEAR COMMAND ENTERED FLAG
1112	006326	005067	011336		CLR	ESCFLG	:CLR COMMAND INPUT MODE FLAG
1113	006332				CALL	COMAND,<CMDBUF,CMDTBL>	:CONVERT COMMAND
1114	006352	103015			BCC	CMDRTN	:IF C = 0, EXIT
1115	006354				CALL	PNTLIN,<SYNTAX>	:PRINT 'SYNTAX ERROR' IF C SET
1116	006372				CALL	PRESC	:GIVE ANOTHER '\$'
1117	006406	005767	011256	CMDRTN:	TST	ESCFLG	:SHOULD WE FAKE CNTRL-C?
1118	006412	001406			BEQ	CMDRET	:NO, EXIT
1119	006414				CALL	PRESC	:YES, GIVE A '\$'
1120	006430			CMDRET:	RETURN		:RETURN TO MAIN LOOP
1121							
1122							

```

1123      ;PROCESSOR FOR SPECIAL EVENT
1124      ;INDICATE 'NO MASTER' AND CLEAR EVENTS
1125      ;THIS EVENT IS CALLED BY THE MAIN LOOP AFTER
1126      ;THE OCCURRENCE OF A MASTER DOWN INTERRUPT
1127      ;THE RESULT OF THIS EVENT IS THAT THE PROGRAM WILL
1128      ;PRINT MASTER DOWN IN THE FOLLOWING FASION:
1129      ;
1130      ; * * * * * MASTER DOWN * * * * *
1131      ;
1132      ;THEN CLEAR ACTIVE EVENT FLAGS BY ENTERING COMMAND MODE.
1133      ;TO RECOVER FROM THIS STATE AND RESUME EXERCISING:
1134      ; (A) SET MASTER ON ANY ONE PCL11
1135      ; (B) TYPE THE COMMAND 'CONTINUE'
1136
1137 006432      PROC      SPEC
1138      (1)
1139      (1) 006432      SPEC:      ;**ENTRY POINT**
1140      006432      CALL      PNCRLF      ;ENQUEUE CR & LF
1141      006446      CALL      PNTLIN,<MDNER> ;ENQUEUE MASTER DOWN MESSAGE
1142      006500 005067 011112      CALL      PNCRLF      ;ANOTHER CR & LF
1143      006504 005067 011102      CLR      SPCEV      ;CLR SPECIAL EVENT FLAG
1144      006510 005067 011064      SPERTN: CLR      PCLGO      ;DON'T ALLOW TXM EVENT
1145      006514 005067 011054      CLR      STSEV      ;DISCONTINUE STATUS EVENT
1146      006520      CLR      SUMEV      ;DISCONTINUE SUM EVENT
1147      006534      CALL      PRESC      ;FAKE CNTRL-C KEY
1148      RETURN
1149
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 15
PCLEXR.P11 17-MAR-78 11:27      COMMAND DECODER AND PROCESSOR
SEQ 0070

1148      ;PROCESSOR FOR 'EXTRA' SPECIAL EVENT
1149      ;TO INDICATE WHEN THIS UNIT HAS JUST BECOME
1150      ;MASTER AS A RESULT OF PREVIOUSLY HAVING
1151      ;SECONDARY SET AND THE CURRENT BUS MASTER
1152      ;HAVING JUST DROPPED MASTER.
1153      ;WHEN 'NOW MASTER' INTERRUPT OCCURS, THE
1154      ;CONSOLE DEVICE WILL PRINT
1155      ; '*** THIS UNIT HAS BECOME NEW MASTER ***'
1156
1157 006536      PROC      XSPEC
1158      (1)
1159      (1) 006536      XSPEC:      ;**ENTRY POINT**
1160      006552      CALL      PNCRLF      ;ENQUEUE CR & LF
1161      006570      CALL      PNTLIN,<MSCHNG> ;ENQUEUE NEW MASTER MESSAGE
1162      006604 005067 011010      CALL      PNCRLF      ;ANOTHER CR & LF
1163      006610      CLR      XSPCEV      ;CLR XSPEC EVENT FLAG
1164      RETURN      ;RETURN TO MAIN LOOP
1165
1166      ;PROCESSOR FOR 'DETAILED' ERROR TABLE PRINTOUT COMMAND 'ERRORS'
1167
1168      ;THIS ROUTINE WILL INITIALIZE THE TEMPORARY LOCATIONS REQUIRED
1169      ;BY THE XMTR AND RCVR ERROR EVENTS. IT WILL THEN QUICKLY CHECK THE
1170      ;RCVR AND XMTR ERROR TABLES FOR ANY ENTRIES. IF NONE EXIST, THE MESSAGE
1171      ; '***NO ERRORS TO REPORT YET ***'
1172      ;WILL BE PRINTED ON THE CONSOLE AND THAT IS ALL.
1173      ; IF ANY ENTRIES WERE FOUND, THE XMTR ERROR EVENT FLAG 'TEREV' WILL BE SET
1174      ; WHICH WILL CAUSE THE XMTR ERRORS AND THE RCVR ERRORS (IF ANY) TO
1175      ; BE PRINTED.
1176
1177 006612      PROC      CPERR
1178      (1)
1179      (1) 006612      CPERR:      ;**ENTRY POINT**

```

```

1178 006612 022766 000001 000002      CMP      #1,2(SP)      ;ANY ARGUMENTS?
1179 006620 001044      BNE      SYNRR      ;ERROR IF THERE ARE.
1180 006622 012700 025062      MOV      #TERTBL,R0      ;POINT AT XMTR TABLE
1181 006626 012701 001350      MOV      #37*30,R1      ;SET TABLE LENGTH COUNTER
1182 006632 005720      1$: TST      (R0)+      ;ANY ENTRIES?
1183 006634 001016      BNE      SERFL      ;YES, CALL ERROR MODULE
1184 006636 005301      DEC      R1      ;NO, CHECK WHOLE TABLE
1185 006640 001374      BNE      1$
1186 006642      CALL      PNTLIN,<NOERMG>      ;PRINT 'NO ERRORS'
1187 006660 012767 177777 011002      MOV      #-1,ESCFLG      ;FLAG COMENT TO ISSUE '$'
1188 006666 000167 000036      JMP      CPERTN      ;AND RETURN
1189 006672 012767 100000 010664 SERFL: MOV      #B15,TEREV      ;SET XMTR EVENT FLAG
1190 006700 012767 000001 010774      MOV      #1,ERR0      ;INITIATE ERROR #
1191 006706 012767 000001 010770      MOV      #1,ERR1      ;INITIATE RCVR ADDR
1192 006714 012767 025062 010764      MOV      #TERTBL,ERR2      ;INITIATE TABLE POINTER
1193 006722 052767 100000 010700      BIS      #B15,HDRFLG      ;SET HEADER FLAG
1194 006730      CPERTN: RETURN
1195
1196 006732 000261      SYNRR: SEC      ;SET 'C' BIT TO INDICATE SYNTAX ERR
1197 006734 000167 177770      JMP      CPERTN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 16
PCLEXR.P11 17-MAR-78 11:27      RECEIVER ADDRESS QUEUE LOADER ROUTINE

```

SEQ 0071

```

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
1213 (1)
1214 (1) 006740      ADQLD:      ;**ENTRY POINT**
1215 006740      REGSAV      ;SAVE R0...R5
1216 006744 005067 010620      CLR      QLDEV      ;CLR QUEUE LOAD EVENT FLAG
1217 006750 012702 000037      MOV      #37,R2      ;POSITION COUNTER
1218 006754 012700 020012      MOV      #RADB,R0      ;R0 POINTS AT TABLE
1219 006760 022710 177777      QCHACT: CMP      #-1,(R0)      ;IS ACTIVE FLAG SET IN TABLE?
1220 006766 004767 000036      BNE      QLOOK      ;NO, LOOK FOR NEXT ONE
1221 006772 062700 000014      JSR      PC,QLOAD      ;YES, LOAD ADDR INTO QUEUE
1222 006776 005302      QLOOK: ADD      #14,R0
1223 007000 001367      DEC      R2      ;ALL ENTRIES CHECKED?
1224 007002 005767 007330      BNE      QCHACT
1225 007006 001424      TST      AQQ      ;IS THERE ANYTHING IN QUEUE?
1226 007010 005267 010634      BEQ      QMTRN      ;NO, THIS IS NOT A VALID PASS
1227 007014 012767 100000 010564 QRTN: INC      RCTXPS      ;YES, INCR PASS NO.
1228 007022      MOV      #B15,IXMEV      ;SET XMTR EVENT FLAG
1229 007026      REGRES      ;RESTORE R0...R5
1230      RETURN
1231 007030 116067 000002 021143 QLOAD: MOV      2(R0),BKELEM+1      ;GET RCVR ADDR INTO HIGH BYTE
1232 007036      CALL      ENQ,<BKELEM,AQQ>      ;PUT IT IN ADDR QUEUE
1233 007056 000207      RTS      PC
1234
1235
1236 007060 005767 010526      QMTRN: TST      PCLGO      ;ARE WE RUNNING?

```



```

1294
1295 007336 012601      MLI:  MOV  (SP)+,R1      ;GET MULTIPLICAND
1296 007340 011603      MOV  (SP),R3      ;GET MULTIPLIER
1297 007342 010446      MOV  R4,-(SP)      ;SAVE R4
1298 007344 012704 000020 MOV  #16.,R4      ;SET UP FOR 16 BIT MULTIPLY
1299 007350 005002
1300 007352 006002      MUL:  CLR  R2      ;SHIFT PRODUCT
1301 007354 006003      ROR  R2
1302 007356 103001      ROR  R3
1303 007360 060102      BCC  CYCL      ;CHECK IF MULTIPLIER BIT IS 0
1304 007362 005304      ADD  R1,R2      ;ADD IN MULTIPLICAND
1305 007364 003372      CYCL: DEC  R4      ;COUNT LOOP
1306 007366 012604      BGT  MUL      ;RESTORE R4
1307 007370 006202      MOV  (SP)+,R4      ;ONE LAST SHIFT
1308 007372 006003      ASR  R2      ;PRODUCT IS IN R3
1309 007374 010316      ROR  R3      ;STORE RESULT ON STACK
1310 007376 000134      MOV  R3,(SP)
1311      JMP  @ (R4)+      ;RETURN
1312
1313
1314
1315
1316
1317
1318
1319
1320

```

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

```

CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 19
PCLEXR.P11 17-MAR-78 11:27 TRANSMIT MODULE

SEQ 0074

```

1322      .SBTTL TRANSMIT MODULE
1323
1324
1325
1326      ; CONTROL TRANSMISSION OF RANDOM DATA TO DE-QUEUED RECEIVER ADDRESSES
1327
1328      ; IF 'PCLGO' IS SET, ('GO' OR 'CONTINUE' SET IT), START TRANSMITTING
1329      ; THE DATA CONTAINED IN THE BUFFER 'DATBUF'.
1330      ; THE TARGET RECEIVER ADDRESS IS DEQUEUED FROM THE ADDRESS QUEUE.
1331      ; THE MESSAGE LENGTH WAS DETERMINED DURING THE GENERATION OF THE DATA.
1332
1333      ; NOTE THAT BIT 15 OF 'TXMST' MAY BE 1 OR 0 DEPENDING UPON THE USE
1334      ; OF THE 'RIB SET' OR 'RIB CLEAR' COMMANDS.
1335
1336
1337 007400 060101      TXMST: .WORD 060101      ;TRANSMITTER COMMAND WORD
1338
1339 007402      PROC  TXMIT
1340      (1)
1341      (1) 007402      TXMIT:      ;**ENTRY POINT**
1342 007402 005767 010204      TST  PCLGO      ;CAN PCL GO?
1343 007406 001455      BEQ  TXRTN      ;IF NOT, RETURN
1344 007410 005067 010172      CLR  TXMEV      ;CLR XMTR EVENT FLAG
1345 007414      BDINIT XMTR      ;CLR XMTR HARDWARE
1346 007422 005767 006710      TST  A00      ;ANY ELEMENTS IN ADDR QUEUE?
1347 007426 001446      BEQ  TXOUT      ;IF NOT, FILL IT UP AGAIN
1348 007430 016777 010222 006366      MOV  TXML,@TSBC      ;LOAD BYTE COUNT
1349 007436 012777 020600 006362      MOV  #DATBUF,@TSBA      ;LOAD BUS ADDRESS
1350 007444      CALL  DEQ,<FRELEM,A00>      ;DEQUEUE RECEIVER ADDRESS
1351 007464 116767 020507 010172      MOVB FRELEM+1,CURAD      ;GET ADDRESS TABLE OFFSET
1352 007472 016700 010166      MOV  CURAD,R0      ;FIND ADDRESS TABLE ENTRY
1353 007476 006300      ASL  R0      ;IN ORDER TO UPDATE
1354 007500 006300      ASL  R0      ; 'ATTEMPTS'

```

```

C 5
1353 007502 010005 MOV R0,R5
1354 007504 006300 ASL R0
1355 007506 060500 ADD R5,R0
1356 007510 062700 017776 ADD #RADBO,R0
1357 007514 062760 000001 000006 ADD #1,6(R0) ;UPDATE DOUBLE WORD
1358 007522 005560 000004 ADC 4(R0)
1359 007526 016777 020444 006262 MOV FRELEM,@TCR ;GIVE RECEIVER ADDR TO XMTR
1360 007534 056777 177640 006254 BIS TXMST,@TCR ;START TRANSMISSION
1361 007542 TXRTN: RETURN
1362
1363 007544 012767 100000 010016 TXOUT: MOV #B15,QLDEV ;SET QUEUE LOAD EVENT FLAG
1364 007552 026727 010072 000004 CMP RCTXP,#4 ;IS THIS THE 5TH PASS?
1365 007560 001402 BEQ TXDGEN ;IF SO, SET DATA GEN EVENT FLAG
1366 007562 000167 177754 JMP TXRTN ;RETURN
1367 007566 012767 100000 007776 TXDGEN: MOV #B15,DAIGEV ;SET DAT GEN EVENT FLAG
1368 007574 000167 177742 JMP TXRTN ;RETURN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 20
PCLEXR.P11 17-MAR-78 11:27 TRANSMIT MODULE
SEQ 0075

1370 ;;;TRANSMITTER INTERRUPT HANDLER
1371 ;;; SORT THE CAUSES OF TRANSMITTER INTERRUPTS
1372 ;;; AND GO TO THE PROPER HANDLING ROUTINE.
1373
1374
1375 100000 TXER = 100000
1376 000020 TXBSY = 20
1377 000200 TXSUC = 200
1378 000040 TXSOR = 40
1379 000004 NOWMST = 4
1380
1381 007600 XMTINT: REGSAV ;;;SAVE R0...R5
1382 007604 042777 000100 006204 BIC #B06,@TCR ;;;CLR TXM INTR ENABLE
1383 007612 012737 000200 177776 MOV #200,@NPS ;;;ALLOW RCVR INTR BUT NOT TRY
1384 007620 032777 100000 006172 BIT #TXER,@TSR ;;;IS THERE A HARDWARE ERROR?
1385 007626 001402 BEQ XMTIS ;;;NO, CHECK SUC TXF
1386 007630 000167 000114 JMP ERRINT ;;;HANDLE ERROR
1387 007634 032777 000200 006156 XMTIS: BIT #TXSUC,@TSR ;;;WAS IT A SUC TRANSFER?
1388 007642 001402 BEQ XMTSR ;;;NO, CHECK SOFTWARE REJECT
1389 007644 000167 000610 JMP SUCINT ;;;HANDLE SUC TXF
1390 007650 032777 000040 006142 XMTSR: BIT #TXSOR,@TSR ;;;WAS MESSAGE REJECTED?
1391 007656 001402 BEQ XMTPNM ;;;NO,CHECK NOW MASTER
1392 007660 000167 000772 JMP SORINT ;;;HANDLE REJECT
1393 007664 132777 000004 006140 XMTPNM: BITB #NOWMST,@TMMRH ;;;IS THIS UNIT NOW MASTER?
1394 007672 001402 BEQ XMTIB ;;;NO, WAS RECVR BUSY?
1395 007674 000167 001074 JMP NMINT ;;;YES, HANDLE NOW MASTER INT
1396 007700 032777 000020 006112 XMTIB: BIT #TXBSY,@TSR ;;;WAS RECEIVER BUSY
1397 007706 001402 BEQ XMTPRB ;;;NO, INTERRUPT WAS FRONEOUS
1398 007710 000167 000514 JMP BSYINT ;;;HANDLE BUSY
1399 007714 012767 100000 007664 XMTPRB: MOV #B15,IXMEV ;;;SET XMTR EVENT FLAG
1400 007722 ERROT N ;;;ERROR:ERRONEOUS INTERRUPT FROM XMTR
(1)
(1)
(1)
1401 007742 REGRES ;;;RESTORE R0...R5
1402 007746 000002 RTI ;;;RETURN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 21
PCLEXR.P11 17-MAR-78 11:27 TRANSMIT MODULE
SEQ 0076

1404 ;XMTR ERROR INTERRUPT ROUTINE
1405
1406
1407 ; DETERMINE AND REPORT THE CAUSE OF THE TRANSMITTER ERROR.
1408

```

```

1409
1410 007750 032777 040000 006042 ERRINT: BIT #B14,@TSR ;IS ERROR NON EXISTANT LOCATION?
1411 007756 001412 BEQ ERRA ;ERROR:NON-EXISTANT LOC ERROR IN XMTR
1412 007760 ERROT N ;**** XMTR ERROR 2 ****
(1)
(1)
(1)
1413 010000 000167 000402 006006 ERRA: JMP ERRX ;IS ERROR MEM OFL?
1414 010004 032777 020000 BEQ #B13,@TSR ;ERROR:MEM OFL ERROR IN XMTR
1415 010012 001412 ERROT ERRA N ;**** XMTR ERROR 3 ****
(1)
(1)
(1)
1417 010034 000167 000346 005752 ERRA: JMP ERRX ;IS ERROR TXM ERROR?
1418 010040 032777 010000 BEQ #B12,@TSR ;SAVE RESPONSE CODES
1419 010046 001472 BEQ ERRC ;REMOVE GARBAGE
1420 010050 017767 005744 007610 MOV @TSR,RSPC ;RSP CODES 11&01?
1421 010056 042767 177760 007602 BIC #-20,RSPC ;ERROR:TXM ERROR. RCVR ACCEPTED A NULL
1422 010064 022767 000015 007574 CMP #15,RSPC ;**** XMTR ERROR 4 ****
1423 010072 001012 BNE ERRA1 ;DISCARD RSP A'S
1424 010074 ERROT N ;RSB B = 00?
(1) ;ERROR:TXM ERROR. RCVR HAS GONE OFF LINE
(1) ;**** XMTR ERROR 5 ****
(1)
1425 010114 000167 000266 007540 ERRA1: JMP ERRX ;RSP B = 10?
1426 010120 042767 000003 BIC #3,RSPC ;ERROR:TXM ERROR. WORD OR CRC REJECTED
1427 010126 001012 BNE ERRA2 ;**** XMTR ERROR 6 ****
1428 010130 ERROT N ;RSP B = 10?
(1) ;ERROR:TXM ERROR. WORD OR CRC REJECTED
(1) ;**** XMTR ERROR 7 ****
(1)
1429 010150 000167 000232 007504 ERRA2: JMP ERRX ;MASTER DOWN?
1430 010154 022767 000010 CMP #10,RSPC ;ERROR: M A S T E R D O W N .
1431 010162 001012 BNE ERRA3 ;**** XMTR ERROR 10 ****
1432 010164 ERROT N ;STOP TRANSMITTER
(1) ;SET SPECIAL EVENT FLAG
(1)
1433 010204 000167 000176 005556 ERRC: JMP ERRX ;TIMEOUT ERROR?
1434 010210 ERRA3: ERROT N ;IS 'R.I.B.' SET?
(1) ;NO MUST BE WEIRD ERROR.
(1) ;YES, IS TDM BUS BUSY?
(1) ;YES, RECEIVER WASN'T THERE.
1435 010230 000167 000152 005556 ERRC: JMP ERRX ;STOP TRANSMITTER
1436 010234 032777 004000 BEQ #B11,@TSR ;SET SPECIAL EVENT FLAG
1437 010242 001417 BEQ ERRA4 ;TIMEOUT ERROR?
1438 010244 ERROT N ;IS 'R.I.B.' SET?
(1) ;NO MUST BE WEIRD ERROR.
(1) ;YES, IS TDM BUS BUSY?
(1) ;YES, RECEIVER WASN'T THERE.
CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 21-1
PCI EXR.P11 17-MAR-78 11:27 TRANSMIT MODULE
1439 010264 005067 007322 CLR PCLGO ;STOP TRANSMITTER
1440 010270 012767 100000 007320 MOV #B15,SPCEV ;SET SPECIAL EVENT FLAG
1441 010276 000167 000104 JMP ERRX ;TIMEOUT ERROR?
1442 010302 032777 002000 005510 ERRA4: BIT #B10,@TSR ;IS 'R.I.B.' SET?
1443 010310 001422 BEQ ERRE ;NO MUST BE WEIRD ERROR.
1444 010312 032777 100000 005476 BIT #B15,@TSR ;YES, IS TDM BUS BUSY?
1445 010320 001404 BEQ 1$ ;YES, RECEIVER WASN'T THERE.
1446 010322 032777 000020 005470 BIT #TXBSY,@TSR
1447 010330 001010 BNE 2$

```

SEQ 0077

```

1448 010332          18:  ERROT  N      E 5      ;ERROR: ERRONEOUS TIMEOUT IN TRANSMITTER
(1)
(1)
(1)
1449 010352 000167 000030          28:  JMP      ERRX
1450 010356 032777 001000 005434 ERRE:  BIT      #B09,@TSR      ;TXM SILO OVERRUN?
1451 010364 001410          BEQ      ERRX
1452 010366          ERROT  N      ;ERROR:SILO OVERRUN ERROR IN XMTR
(1)
(1)
(1)
1453 010406          ERRX:  BDINIT  XMTR      ;CLEAR ALL XMTR HARDWARE
1454 010414 012767 100000 007164      MOV      #B15,TXMEV      ;SET XMTR EVENT FLAG
1455 010422          REGRES      ;RESTORE R0...R5
1456 010426 000002          RTI      ;RETURN
CZPLAA0 PCL11 EXERCISER V-02  MACY11 30A(1052) 28-APR-78 13:50 PAGE 22
PCLEXR.P11 17-MAR-78 11:27      TRANSMIT MODULE
SEQ 0078

1458          ;BUSY INTERRUPT ROUTINE
1459
1460 010430 042777 000020 005362 BSYINT: BIC      #TXBSY,@TSR      ;CLEAR TDM BUS BUSY
1461 010436 012767 100000 007142      MOV      #B15,TXMEV      ;SET TXM EVENT FLAG
1462 010444          BDINIT  XMTR      ;CLEAR HARDWARE
1463 010452          REGRES      ;RESTORE R0...R5
1464 010456 000002          RTI      ;RETURN
1465
1466          ;SUCCESSFUL TRANSFER INTERRUPT ROUTINE
1467
1468
1469 010460 042777 000200 005332 SUCINT: BIC      #TXSUC,@TSR      ;CLEAR SUC TXF
1470 010466 005767 007162          TST      TRNKF      ;IS SFTWR TRUNCATE FLAG SET?
1471 010472 001047          BNE      SUCIA      ;YES, GO CHECK SORE
1472 010474 032777 000040 005316 SUCINY: BIT      #TXSOR,@TSR      ;IS SORE SET?
1473 010502 001414          BEQ      SUCINP      ;NO, CONTINUE
1474 010504          ERROT  N      ;ERROR:MESSAGE TRUNCATED UNEXPECTEDLY
(1)
(1)
(1)
1475 010524 000421          BR      SHBRT      ;DON'T INC SUC CONNS IF ERROR
1476 010526 042777 000040 005264 SUCINP: BIC      #TXSOR,@TSR      ;CLR SORE
1477 010534 016700 007124          MOV      CURAD,R0      ;PREPARE TO UPDATE STATUS TABLE
1478 010540 006300          ASL      R0      ;GET TABLE ENTRY FOR CURRENT
1479 010542 006300          ASL      R0      ;RECEIVER ADDRESS IN
1480 010544 010005          MOV      R0,R5
1481 010546 006300          ASL      R0
1482 010550 060500          ADD      R5,R0
1483 010552 062700 017776          ADD      #RADB0,R0
1484 010556 062760 000001 000012          ADD      #1,12(R0)
1485 010564 005560 000010          ADC      10(R0)
1486 010570 012767 100000 007010 SHBRT:  MOV      #B15,TXMEV      ;SET TRANSMIT EVENT FLAG
1487 010576          BDINIT  XMTR      ;CLEAR HARDWARE
1488 010604          REGRES      ;RESTORE R0...R5
1489 010610 000002          RTI      ;RETURN
1490
1491 010612 032777 000040 005200 SUCIA:  BIT      #TXSOR,@TSR      ;IS SORE SET? (SHOULD BE SET)
1492 010620 001011          BNE      SUCCS
1493 010622          ERROT  N      ;ERROR:MSG FAILED TO BE TRUNCATED
(1)
(1)
(1)
1494 010642 000752          BR      SHBRT      ;DON'T INC SUC CONNS IF ERROR
1495 010644 042777 000040 005146 SUCCS:  BIC      #TXSOR,@TSR      ;CLR SORE
1496 010652 000167 177616          JMP      SUCINY      ;PROCESS SUC TXF AS NORMAL

```

```

1498 ;SOFTWARE REJECT INTERRUPT ROUTINE
1499
1500 010656 005767 006770 SORINT: TST RJCTF ;IS SFTWR REJECT FLAG SET?
1501 010662 001432 BEQ SORIP ;IF NOT, SOMETHING'S WRONG
1502 010664 016700 006774 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 017776 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 005072 SORCNT: BIC #TXSOR,@TSR ;CLR SORE
1512 010726 012767 100000 006652 MOV #B15, TXMEV ;SET XMIT EVENT FLAG
1513 010734 BDINIT XMTX ;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 ;NOW MASTER INTERRUPT ROUTINE
1521 ;SET XSPEC EVENT FLAG (XSPEV)
1522 ;AND CLR NOW MST
1523
1524
1525
1526 010774 142777 000004 005030 NMINT: BICB #NOWMST,@TMMRH ;CLEAR NOW MASTER
1527 011002 012767 100000 006610 MOV #B15,XSPCEV ;SET EXTRA-SPECIAL EVENT
1528 011010 052777 000100 005000 BIS #B06,@TCR ;RE-SET INTERRUPT ENABLE
1529 011016 REGRES ;RESTORE R0...R5
1530 011022 000002 RTI ;RETURN
1531
1532

```

```

1534 .SBTTL RECEIVER MODULE
1535
1536 ;CONTROL RECEPTION OF DATA.
1537 ; DATA IS STORED IN A BUFFER "RCBUF"
1538 ; FOR LATER CHECKING.
1539 ; "RCBUF" BUFFER IS CLEARED BEFORE RECEPTION.
1540
1541 ;ACTIVE IF RCVEV IS -VE
1542
1543 020000 RCWD - B13
1544
1545 011024 PROC RECV
(1)
(1)
1546 011024 005067 006564 RECV: 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 022640 004776 MOV #RCBUF,@RDBA ;CLR HARDWARE (REGARDLESS)
1550 011050 012777 176400 004766 MOV #-1400,@RDBC ;LOAD RCVR BUS ADDRESS
;LOAD BYTE COUNT FOR 600 WORDS

```

```

1551 011056 012777 020000 004752      MOV    #RCWD,RCR      6 5      :SET RCV WORD IN RCVR
1552 011064 052777 000100 004744      BIS    #B06,RCR      :SET RCVR INTERRUPT ENABLE
1553 011072      RETURN
1554
1555
1556
1557 011074      DBFCLR: REGSAV      :SAVE R0...R5
1558 011100 012700 022640      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
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 25
PCLEXR.P11 17-MAR-78 11:27      RECEIVER MODULE
SEQ 0081

1566      :;;RECEIVER INTERRUPT HANDLER.
1567
1568      100000      RCER    =    100000
1569      000400      RCDOR  =    400
1570      000200      RCSUC  =    200
1571      000040      RCRCM  =    40
1572
1573 011124      RCVINT: REGSAV      :;;SAVE R0...R5
1574 011130 032777 100000 004702      BIT    #RCER,RCR      :;;IS THERE A HARDWARE ERROR?
1575 011136 001402      BEQ    RCVINA      :;;NO,CHECK DATA OUTPUT READY
1576 011140 000167 000100      JMP    RERINT      :;;HANDLE ERROR INTERRUPT
1577 011144 032777 000400 004666      RCVINA: BIT    #RCDOR,RCR      :;;IS DATA OUTPUT READY?
1578 011152 001402      BEQ    RCVINB      :;;NO, CHECK SUC TXF
1579 011154 000167 000472      JMP    RDOINT      :;;HANDLE DATA OUTPUT RDY INTERRUPT
1580 011160 032777 000200 004652      RCVINB: BIT    #RCSUC,RCR      :;;SUCCESSFUL TRANSFER SET?
1581 011166 001402      BEQ    RCVINC      :;;NO, CHECK REECT COMPLETED
1582 011170 000167 000536      JMP    RSUINT      :;;HANDLE SUC TXF INTERRUPT
1583 011174 032777 000040 004636      RCVINC: BIT    #RCRCM,RCR      :;;REJECT COMPLETED INTERRUPT?
1584 011202 001402      BEQ    RCVIND      :;;NO,ERRONEOUS INTERRUPT
1585 011204 000167 000776      JMP    RRJINT      :;;HANDLE REJ-COM INTERRUPT
1586 011210      RCVIND: ERROR    N      :;;ERROR:UNKNOWN RECEIVER INTERRUPT OCCURRED
(1)
(1)
(1)
1587 011230 012767 100000 006356      MOV    #B15,RCVEV      :;;SET RCVR EVENT FLAG
1588 011236      REGRES      :;;RESTORE R0...R5
1589 011242 000002      RTI      :RETURN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 26
PCLEXR.P11 17-MAR-78 11:27      RECEIVER MODULE
SEQ 0082

1591      :RCVR ERROR INTERRUPT ROUTINE
1592
1593 011244 032777 040000 004566      RERINT: BIT    #B14,RCR      :IS ERROR NON EXST LOC ERR?
1594 011252 001412      BEQ    RERRA      :
1595 011254      ERROR    N      :ERROR:NON EXST LOC ERR IN RCVR
(1)
(1)
(1)
1596 011274 000167 000332      RERRA: JMP    RERRX      :IS ERROR MEM OFLO?
1597 011300 032777 020000 004532      BIT    #B13,RCR      :
1598 011306 001412      BEQ    RERRB      :
1599 011310      ERROR    N      :ERROR:MEM OFLO ERROR IN RCVR
(1)
(1)
(1)
1600 011330 000167 000276      JMP    RERRX      :;;RCVR ERROR 20 *****

```

Line	Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

1644	011710	052777	000100	004120		BIS	#B06,RCR		:RE-SET INTR ENAB
1645	011716	000002				RTI			:RETURN
1646	011720	052777	040001	004110	RDOINC:	BIS	#B14+B00,RCR		:START RECEIVER GOING
1647	011726	000167	177752			JMP	RDORTN		
1648									
1649									
1650									
1651	011732	032777	000040	004100	RSUINT:	BIT	#RCRCM,RSR		:WAS RECOM SET?
1652	011740	001403				BEG	RSUCNT		:NO, CONTINUE
1653	011742	042777	000040	004070		BIC	#RCRCM,RSR		:YES, CLR IT AND CONTINUE
1654	011750	016701	005672		RSUCNT:	MOV	RCSEED,R1		:GET DATA SEED
1655	011754	042701	177000			BIC	#177000,R1		:MASK MSG LENGTH
1656	011760	001002				BNE	RSNT0		:IF NOT 0, OKAY
1657	011762	012701	000001			MOV	#1,R1		:IF 0, MAKE IT 1
1658	011766	162701	000001		RSNT0:	SUB	#1,R1		:ACCOUNT FOR FLAG WORD
1659	011772	020127	000600			CMP	R1,#600		:IS IT GREATER THAN 600?
1660	011776	003402				BLE	RSFLSZ		:NO, LEAVE IT ALONE
1661	012000	012701	000600			MOV	#600,R1		:YES, SET IT TO LIMIT (600)
1662	012004	012700	001400		RSFLSZ:	MOV	#1400,R0		
1663	012010	067700	004030			ADD	RDDBC,R0		:DETERMINE # OF BYTES RECEIVED
1664	012014	001451				BEQ	RSULV		:NOTHING TO CHECK, EXIT
1665	012016	006200				ASR	R0		:R0 = NO. OF WORDS RECEIVED
1666	012020	020001				CMP	R0,R1		:DID WE RECV ALL THE WORDS?
1667	012022	001412				BEQ	RSUTST		:YES, CARRY ON
1668	012024	103456				BLO	ERNE		:NOT ENOUGH !
1669	012026					ERROR	N		:ERROR:RCVR GOT TOO MANY WORDS.!
(1)									
(1)									:**** RCVR ERROR 26 ****
(1)									
1670	012046	000434				BR	RSULV		:EXIT IF ERROR
1671	012050	012705	022640		RSUTST:	MOV	#RCBUF,R5		:R5 POINTS AT RECD DATA
1672	012054				RSUGEN:	MULP	RANM,RCSEED		
1673	012076	066767	005560	005542		ADD	RANK,RCSEED		:GENERATE CHECK WORD
1674	012104	026725	005536			CMP	RCSEED,(R5)+		:COMPARE IT WITH RECV'D WORD
1675	012110	001411				BEQ	RSUCHK		
1676	012112					ERROR	N		:ERROR:DATA WORD RECV'D WAS BAD
(1)									
(1)									:**** RCVR ERROR 27 ****
(1)									
1677	012132	000402				BR	RSULV		:EXIT IF ERROR
1678	012134	005300			RSUCHK:	DEC	R0		:CHECKED ALL WORDS?
1679	012136	001346				BNE	RSUGEN		:NO,CONTINUE
1680	012140				RSULV:	BDINIT	RCVR		:CLR HARDWARE
1681	012146					REGRES			:RESTORE R0...R5
1682	012152	012767	100000	005434		MOV	#B15,RCVEV		:SET RCVR EVENT FLAG
1683	012160	000002				RTI			:RETURN
1684									
1685	012162				ERNE:	ERROR	N		:ERROR: RCVR GOT TOO FEW WORDS !.
CZPLAAO PCL11	EXERCISER V-02	MACY11	30A(1052)	28-APR-78	13:50	PAGE	27-1		
PCLEXR.P11	17-MAR-78 11:27		RECEIVER MODULE						
(1)									
(1)									:**** RCVR ERROR 30 ****
(1)									
1686	012202	000167	177732			JMP	RSULV		:EXIT IF ERROR
1687									
1688									
1689									
1690	012206				RRJINT:	BDINIT	RCVR		:CLEAR HARDWARE
1691	012214	012767	100000	005372		MOV	#B15,RCVEV		:SET RCVR EVENT FLAG
1692	012222					REGRES			:RESTORE R0...R5
1693	012226	000002				RTI			:RETURN
1694									

SEQ 0085

J 5

```

;KW11 CLOCK INTERRUPT ROUTINE
;UPDATE TICKS. IF = 60. UPDATE SECONDS.
;IF SECS = 60. UPDATE MINUTS. IF MINUTS = 60
;UPDATE HOURS.

```

```

CLKINT: BIC      #807,ALCS      ;JUST IN CASE IT MATTERS.
        INCB     TICKS         ;UPDATE TICKS
        CMPB     TICKS,#60.     ;GET TO 60 YET?
        BLC      CLKEXT        ;NO
        CLRB     TICKS         ;YES, CLR TICKS
        INCB     SECONDS        ;& UPDATE SECONDS
        CMPB     SECONDS,#60.   ;GET 60 SECS?
        BLC      CLKEXT        ;NO
        CLRB     SECONDS        ;YES, CLR SECONDS
        INCB     MINUTS         ;& UPDATE MINUTES
        CMPB     MINUTS,#60.    ;GET 60 MINUTES?
        BLC      CLKEXT        ;NO
        CLRB     MINUTS         ;YES, CLEAR MINUTS
        INC      HOURS          ;& UPDATE HOURS
        CLKEXT: RTI            ;RETURN

```

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 28
 PCLEXR.P11 17-MAR-78 11:27 STATUS MODULE

SEQ 0086

.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

```

	PROC	STATUS	
STATUS:	TST	TOQ	;**ENTRY POINT**
	BEQ	60\$	;IS TTY OUT QUEUE EMPTY?
	JMP	STRN	;NO, RETURN TO WHEREVER
60\$:	TST	STPNTR	;IS OUTPUT POINTER AT TABLE?
	BEQ	STHDR	;NO, GO PRINT HEADER
STCHK:	CMP	STPNTR,#RADEND	;IS POINTER AT END OF TABLE?
	BLT	59\$	;YES, DONE; EXIT
	JMP	STARXT	
59\$:	TST	@STPNTR	;CHECK RCV ADDR ACTIVE FLAG
	BNE	58\$	
	JMP	STBADV	;ADVANCE TABLE IF FLAG CLR
58\$:	MOV	#STLIN1,R0	;OUTPUT POINTER FOR OCTJSP
	ADD	#2,STPNTR	
	MOV	@STPNTR,R1	;OCTJSP DUMPS DATA IN STPNTR
	MOV	#-1,R2	;DON'T COMPRESS BLANKS
	JSR	PC,OCTJSP	
	ADD	#2,STPNTR	
	MOV	#STLIN2,R0	;OUTPUT POINTER FOR CDDMG
	MOV	STPNTR,R1	;OUTPUT WORD FOR CDDMG
	MOV	#-1,R2	;DON'T COMPRESS BLANKS
	JSR	PC,CDDMG	
STYHR:	CMP	R0,#STLIN2+25	;BLANK REST OF FIELD OUT
	BHIS	STYON	;BY INSERTING SPACES
	MOVB	#',(R0)+	
	BR	STYHR	

```

1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728 012324
(1)
(1) 012324
1729 012324 005767 004206
1730 012330 001402
1731 012332 000167 000540
1732 012336 005767 005240
1733 012342 001504
1734 012344 026727 005232 020576
1735 012352 002402
1736 012354 000167 000476
1737 012360 005777 005216
1738 012364 001002
1739 012366 000167 000452
1740 012372 012700 031767
1741 012376 062767 000002 005176
1742 012404 017701 005172
1743 012410 012702 177777
1744 012414 004767 020174
1745 012420 062767 000002 005154
1746 012426 012700 032011
1747 012432 016701 005144
1748 012436 012702 177777
1749 012442 004767 020346
1750 012446 020027 032036
1751 012452 103003
1752 012454 112720 000040
1753 012460 000772

```

Line	Address	Offset	Value	Label	Operation	Comments
1754	012462	062767	000004	005112	STYON: ADD #4,STPNTR	
1755	012470	012700	032037		MOV #STLIN3,R0	;OUTPUT POINTER FOR CDDMG
1756	012474	016701	005102		MOV STPNTR,R1	;OUTPUT WORD FOR CDDMG
1757	012500	012702	177777		MOV #-1,R2	;DON'T COMPRESS BLANKS
1758	012504	004767	020304		JSR PC,CDDMG	
1759	012510	020027	032053	STYTHR: CMP R0,#STLIN3+14		
1760	012514	103003			BHIS STYONT	;BLANK REST OF FIELD OUT
1761	012516	112720	000040		MOVB #'',(R0)+	;BY INSERTING SPACES
1762	012522	000772			BR STYTHR	
1763	012524			STYONT: CALL PNTLIN,<STLIN>		
1764	012542	062767	000004	005032	ADD #4,STPNTR	;ENQUEUE STATUS LINE FOR OUTPUT
1765	012550	000167	000322		JMP STRN	;UPDATE POINTER FOR NEXT LINE
1766						;RETURN
1767	012554	132777	000001	003250	STHDR: BITB #1,@TMMRH	;IS MASTER SET HERE?
1768	012562	001407			BEQ 1\$	;NO.
1769	012564				CALL PNTLIN,<THUMST>	;YES, TELL OPERATOR
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 28-1						
PCLEXR.P11 17-MAR-78 11:27 STATUS MODULE						
1770	012602	132777	000002	003222	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	004660	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	031654		MOV #TMLIN1,R0	;OUTPUT POINTER FOR DECPNT
1782	012712	016701	004642		MOV HOURS,R1	;OUTPUT DATA FOR DECPNT
1783	012716	004767	017714		JSR PC,DECPNT	;ENQUEUE 'HOURS'
1784	012722	005002			CLR R2	;SUPPRESS LEADING ZEROS
1785	012724	112720	000072		MOVB #'',(R0)+	;INSERT COLON
1786	012730	116701	004622		MOVB MINUTS,R1	;GET 'MINUTES' FOR DECPNT
1787	012734	004767	017676		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	004603		MOVB SECNDS,R1	;GET 'SECONDS' FOR DECPNT
1791	012752	004767	017660		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	004564		MOVB TICKS,R1	;GET 'TICKS' FOR DECPNT
1795	012770	004767	017642		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>		
1799	013032	012767	020012	004542	MOV #RADB,STPNTR	;ENQUEUE STATUS HEADER
1800	013040	000167	000032		JMP STRN	;SET POINTER TO TOP OF STAT LIST
1801						;RETURN
1802	013044	062767	000014	004530	STBADV: ADD #14,STPNTR	;SET POINTER TO NEXT ACTIVE FLAG
1803	013052	000167	000020		JMP STRN	
1804						
1805	013056	005067	004516		STARXT: CLR STSEV	;CLR STATUS EVENT FL.
1806	013062				CALL PRESC	;FAKE ALT MODE KEY
1807	013076				STRN: RETURN	;RETURN
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 29						
PCLEXR.P11 17-MAR-78 11:27 TRANSMITTER ERRORS MODULE						
.SBTTL TRANSMITTER ERRORS MODULE						
;DUMP A TABLE OF TRANSMITTER ERRORS INCLUDING:						

L 5

```

; 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).

```

				PROC	TEROS	
1819	013100					
(1)						
(1)	013100					TEROS:
1820	013100	005767	003432	TST	TOQ	;**ENTRY POINT**
1821	013104	001402		BEQ	60\$	:IS THE TTY OUTPUT QUEUE EMPTY?
1822	013106	000167	000324	JMP	TERTN	:YES, CONTINUE
1823	013112	005767	004512	60\$: TST	HDRFLG	:NO, EXIT
1824	013116	001422		BEQ	1\$	:HEADER PRINTED YET?
1825	013120	005067	004504	CLR	HDRFLG	:YES, SKIP IT
1826	013124			CALL	PNTLIN,<TERHDR>	:NO, PRINT IT NOW.
1827	013142			CALL	PNTLIN,<TRHLIN>	
1828	013160	000167	000252	JMP	TERTN	:AND RETURN
1829	013164	026727	004512	000016 1\$: CMP	ERR0,#16	:IS THIS ERROR # 16?
1830	013172	001042		BNE	3\$	:NOT YET.
1831	013174	005067	004364	CLR	TEREV	:YES, CLEAR XMTR ERROR EVENT
1832	013200	052767	100000	004360 BIS	#B15,REREV	:SET RCVR ERROR EVENT
1833	013206	052767	100000	004414 BIS	#B15,HDRFLG	:SET HEADER FLAG
1834	013214	012767	000001	004462 MOV	#1,ERR1	:LOAD INITIAL XMTR #
1835	013222	012767	026530	004456 MOV	#RERTBL,ERR2	:POINT RCVR ERR EVENT AT ITS TABLE
1836	013230	005767	004372	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	004332	CLR	PRINTD	
1841	013274	000167	000136	JMP	TERTN	:RETURN
1842	013300	005777	004402	3\$: TST	@ERR2	:ANY ERRORS?
1843	013304	001434		BEQ	4\$	:NO.
1844	013306	012700	032275	MOV	#TRRNO,R0	:READY TO PRINT ERROR NO.
1845	013312	016701	004364	MOV	ERR0,R1	
1846	013316	004767	017272	JSR	PC,OCTJSP	:ERROR NO. IS IN OCTAL
1847	013322	012700	032312	MOV	#TRRCN,R0	:READY TO PRINT RCVR NO.
1848	013326	016701	004352	MOV	ERR1,R1	
1849	013332	004767	017256	JSR	PC,OCTJSP	:RCVR NO. IS ALSO IN OCTAL
1850	013336	012700	032331	MOV	#TRERC,R0	:NOW PRINT ERROR COUNT
1851	013342	017701	004340	MOV	@ERR2,R1	
1852	013346	004767	017256	JSR	PC,DECJSP	:ERROR COUNT IS IN DECIMAL
1853	013352			CALL	PNTLIN,<TRELIN>	:ENQUEUE TABLE OUTPUT LINE
1854	013370	052767	100000	004230 BIS	#B15,PRINTD	:SET 'PRINTED FLAG'
1855	013376	062767	000002	004302 4\$: ADD	#2,ERR2	:UPDATE TABLE POINTER
1856	013404	026727	004274	000037 CMP	ERR1,#37	:ALL RCVR'S DONE?
1857	013412	001404		BEQ	5\$	:YES.
1858	013414	005267	004264	INC	ERR1	:NO, DO NEXT
1859	013420	000167	000012	JMP	TERTN	
1860	013424	012767	000001	004252 5\$: MOV	#1,ERR1	:NEXT ERROR NO.
1861	013432	005267	004244	INC	ERR0	
1862	013436			TERTN: RETURN		

CZPLAA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 30
PCLEXR.P11 17-MAR-78 11:27 RECEIVER ERRORS MODULE

SEQ 0089

.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

```

M 5
;ERRORS ARE PRINTED, SIMPLY PRINT '(NONE)'.
M 5

```
PROC      REROS
REROS:
TST      TOQ
BEQ      60$
JMP      RERTN
60$:     TST      HDRFLG
BEQ      1$
CLR      HDRFLG
CALL     PNTLIN,<RERHDR>
CALL     PNTLIN,<RRHLIN>
JMP      RERTN
1$:      CMP      ERRO,#31
BNE      3$
CLR      REREV
TST      PRINTD
BNE      2$
CALL     PNTLIN,<NONMG>
2$:      CALL     PNCRLF
CLR      PRINTD
CALL     PRESC
JMP      RERTN
3$:      TST      @ERR2
BEQ      4$
MOV      #RCRNO,R0
MOV      ERRO,R1
JSR      PC,OCTJSP
MOV      #RCTRN,R0
MOV      ERR1,R1
JSR      PC,OCTJSP
MOV      #RCERC,R0
MOV      @ERR2,R1
JSR      PC,DECJSP
CALL     PNTLIN,<RCELIN>
BIS      #B15,PRINTD
4$:      ADD      #2,ERR2
CMP      ERR1,#37
BEQ      5$
INC      ERR1
JMP      RERTN
5$:      MOV      #1,ERR1
INC      ERRO
RERTN:   RETURN
```

```
;**ENTRY POINT**
;IS TTY OUTPUT QUEUE EMPTY?
;YES, CONTINUE
;NO, RETURN
;IS HEADER PRINTED YET?
;YES, SKIP IT
;NO, CLEAR FLAG
;AND PRINT HEADER
;AND RETURN
;DONE LAST ERROR?
;NOT YET
;YES, CLEAR RCVR ERROR EVENT
;DID WE PRINT ANYTHING?
;YES, O.K.
;NO, PRINT '(NONE)'
;WRAP IT UP.
;ANY ERRORS?
;NO.
;READY TO PRINT ERROR NO.
;ERROR NO.S ARE IN OCTAL
;READY TO PRINT XMTR NO.
;XMTR NO.S ARE IN OCTAL
;NOW PRINT ERROR COUNT
;ERROR COUNTS ARE IN DECIMAL
;ENQUEUE TABLE OUTPUT LINE
;SET 'PRINTED' FLAG
;UPDATE TABLE POINTER
;ALL XMTRS DONE?
;NO, DO NEXT
;YES, NEXT ERROR
```

1871
1872
1873 013440
(1)
(1) 013440
1874 013440 005767 003072
1875 013444 001402
1876 013446 000167 000310
1877 013452 005767 004152
1878 013456 001422
1879 013460 005067 004144
1880 013464
1881 013502
1882 013520 000167 000236
1883 013524 026727 004152 000031 1\$:
1884 013532 001034
1885 013534 005067 004026
1886 013540 005767 004062
1887 013544 001007
1888 013546
1889 013564
1890 013600 005067 004022
1891 013604
1892 013620 000167 000136
1893 013624 005777 004056
1894 013630 001434
1895 013632 012700 032432
1896 013636 016701 004040
1897 013642 004767 016746
1898 013646 012700 032447
1899 013652 016701 004026
1900 013656 004767 016732
1901 013662 012700 032466
1902 013666 017701 004014
1903 013672 004767 016732
1904 013676
1905 013714 052767 100000 003704
1906 013722 062767 000002 003756 4\$:
1907 013730 026727 003750 000037
1908 013736 001404
1909 013740 005267 003740
1910 013744 000167 000012
1911 013750 012767 000001 003726 5\$:
1912 013756 005267 003720
1913 013762

CZPLAAO PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27
MACY:1 30A(1052) 28-APR-78 13:50 PAGE 31
SUMMARY MODULE

SEQ 0090

.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
```

1915
1916
1917
1918
1919
1920

N 5

OF EXPANDING TO 65,528. QUANTITIES OF ERRORS IN EXCESS OF
THIS WILL BE LISTED AS 'MXCNT'.

1921							
1922							
1923							
1924							
1925							
1926	013764			PROC	SUMRY		
(1)							
(1)	013764			SUMRY:			***ENT.Y POINT**
1927	013764	005767	002546	TST	TOO		:IS OUTPUT QUEUE EMPTY?
1928	013770	001126		BNE	SMRTN		:NO, RETURN
1929	013772	005767	003606	TST	SUMPNT		:IS OUTPUT POINTER AT TABLE?
1930	013776	001472		BEQ	SUMHDR		:NO, GO PRINT HEADER
1931	014000	027727	003600	177777	SMCHK: CMP	@SUMPNT,#-1	:IS ERROR # = -1 ?
1932	014006	001507		BEQ	SMARXT		:YES,DONE; EXIT
1933	014010	016704	003570	MOV	SUMPNT,R4		
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	032507	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	016554	JSR	PC,OCTJSP		
1940	014040	012700	032531	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	016534	JSR	PC,OCTJSP		
1944	014060	012700	032555	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	016474	SMPN: JSR	PC,DECJSP		
1956	014134			SMSPN: CALL	PNTLIN,<SMLIN>		:ENQUEUE SUM LINE FOR OUTPUT
1957	014152	062767	000006	003424	ADD	#6,SUMPNT	:UPDATE POINTER FOR NEXT LINE
1958	014160	000167	000062	JMP	SMRTN		:RETURN
1959							
1960	014164			SUMHDR: CALL	PNTLIN,<SMTTL>		:ENQUEUE SUMMARY HEADER
1961	014202	012767	024640	003374	MOV	#ERTBL,SUMPNT	:SET POINTER TO TOP OF SUM LIST
1962	014210	000167	000032	JMP	SMRTN		:RETURN
1963							
1964	014214	062767	000006	003362	SMADV: ADD	#6,SUMPNT	:SET POINTER TO NEXT ENTRY
1965	014222	000167	177552	JMP	SMCHK		
1966							
1967	014226	005067	003342	SMARXT: CLR	SUMEV		:CLEAR SUMMARY EVENT FLAG
1968	014232			CALL	PRESC		:FAKE ALT MODE KEY
CZPLAAO PCL11 EXERCISER V-02				MACY11	30A(1052)	28-APR-78 13:50	PAGE 31-1
PCLEXR.P11 17-MAR-78 11:27					SUMMARY MODULE		
1969	014246			SMRTN: RETURN			:RETURN
CZPLAAO PCL11 EXERCISER V-02				MACY11	30A(1052)	28-APR-78 13:50	PAGE 32
PCLEXR.P11 17-MAR-78 11:27					ERROR UPDATE ROUTINES		
1971					.SBTTL ERROR UPDATE ROUTINES		
1972					.SBTTL TRANSMITTER ERRORS		
1973							
1974					:CALLED BY THE MACRO 'CALL ERRMOD,<P,ERRADR>'		
1975							
1976	014250			PROC	ERRMOD,<ERRNUM,ERRADR>		

SEQ 0091

SEQ 0092

```

      (3)          000002          ERRNUM =      2          B 6
      (3)          000004          ERRADR =      4
      (1)
      (1) 014250          ERRMOD:
1977 014250 010046          MOV      R0,-(SP)          ;**ENTRY POINT**
1978 014252 010146          MOV      R1,-(SP)          ;SAVE R0 ON STACK
1979 014254 010246          MOV      R2,-(SP)          ;AND R1
1980 014256 010346          MOV      R3,-(SP)          ;AND R2
1981 014260 017767 001532 003410          MOV      @TCR,TEMP          ;AND R3
1982 014266 042767 160377 003402          BIC      #160377,TEMP          ;SAVE ADDRESS OF CONNECTED RCVR
1983 014274 000367 003376          SWAB      TEMP          ;GET IT IN THE RIGHT BYTE
1984 014300 016500 000002          MOV      ERRNUM(R5),R0          ;GET ERROR NUMBER
1985 014304 010067 003370          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 024632          ADD      #ERTBLO,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 003324          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 003246          DEC      TEMP          ; RCVR ADDR - 1
2000 014430          MULT      2,TEMP,R3          ; (R-1)x2
2001 014434 066767 003240 003234          ADD      TEMP1,TEMP          ; (E-1)x37 x2 + (R-1)x2
2002 014442 062767 025062 003226          ADD      #TERTBL,TEMP          ; TEMP = #TERTBL + < (E-1)x37 x2 + (R-1)x2>
2003 014450 005277 003222          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
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 33
PCLEXR.P11 17-MAR-78 11:27 RECEIVER ERRORS

```

SEQ 0093

```

2010          .SBTTL RECEIVER ERRORS
2011
2012          ;CALLED BY THE MACRO 'CALL      ERRMOD,<P,ERRADR>'
2013
2014 014466          PROC      ERRMOD,<ERRNUM,ERRADR>
      (3)          000002          ERRNUM =      2
      (3)          000004          ERRADR =      4
      (1)
      (1) 014466          ERRMOD:
2015 014466 010046          MOV      R0,-(SP)          ;**ENTRY POINT**
2016 014470 010146          MOV      R1,-(SP)          ;SAVE R0 ON THE STACK
2017 014472 010246          MOV      R2,-(SP)          ;AND R1
2018 014474 010346          MOV      R3,-(SP)          ;AND R2
2019 014476 017767 001334 003172          MOV      @RCR,TEMP          ;AND R3
2020 014504 042767 160377 003164          BIC      #160377,TEMP          ;GET ADDR OF CONNECTED XMTR
2021 014512 000367 003160          SWAB      TEMP          ; INTO RIGHT BYTE
2022 014516 016500 000002          MOV      ERRNUM(R5),R0          ;GET ERROR NUMBER
2023 014522 010067 003152          MOV      R0,TEMP1          ;AND SAVE IT
2024 014526 006300          ASL      R0          ; FIND 'SUM' TABLE ENTRY
2025 014530 010001          MOV      R0,R1          ; IN ORDER TO UPDATE
2026 014532 006300          ASL      R0          ; ADDRESS FIELD
2027 014534 060100          ADD      R1,R0          ; AND # OF OCCURRENCES
2028 014536 062700 024632          ADD      #ERTBLO,R0
2029 014542 016560 000004 000002          MOV      ERRADR(R5),2(R0)          ;ENTER ERROR ADDRESS

```

2030	014550	026027	000004	177771		CMP	4(R0),#177771	: AT MAX COUNT YET?
2031	014556	103401				BLO	ERMINR	: NO, UPDATE ERROR COUNT
2032	014560	000447				BR	ERMISR	: YES, SKIP UPDATES
2033	014562	005260	000004		ERMINR:	INC	4(R0)	: UPDATE OCCURRENCES
2034	014566	162767	000015	003104		SUB	#15,TEMP1	: ALIGN ERRORS TABLE FOR RCVR
2035	014574	005367	003100			DEC	TEMP1	: ERR # - 1 (E-1)
2036	014600					MULT	37,TEMP1,R3	: (E-1)x37
2037	014644					MULT	2,TEMP1,R3	: (E-1)x37 x2
2038	014650	005367	003022			DEC	TEMP	: XMTR ADDR - 1 (T-1)
2039	014654					MULT	2,TEMP,R3	: (T-1)x2
2040	014660	066767	003014	003010		ADD	TEMP1,TEMP	: (E-1)x37 x2 + (T-1)x2
2041	014666	062767	026530	003002		ADD	#RERTBL,TEMP	: TEMP = #RERTBL + (E-1)x37 x2 + (T-1)x2
2042	014674	005277	002776			INC	@TEMP	: UPDATE RCVR ERROR COUNT
2043	014700	012603			ERMISR:	MOV	(SP)+,R3	: RESTORE R3
2044	014702	012602				MOV	(SP)+,R2	: RESTORE R2
2045	014704	012601				MOV	(SP)+,R1	: RESTORE R1
2046	014706	012600				MOV	(SP)+,R0	: RESTORE R0
2047	014710					RETURN		

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 34
PCLEXR.P11 17-MAR-78 11:27 UTILITY ROUTINES

SEQ 0094

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	013236		BICB	#200,FRELEM	: CLEAR OFF PARITY BIT.
2056	014740	116700	013232			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	013206		TTINH:	MOV	R0,FRELEM	: IN "FRELEM"
2063	014770	012701	015754			MOV	#CMCHTB,R1	: TO ITS PROCESSING
2064	014774	004767	015566			JSR	PC,PROCHR	: ROUTINE.
2065	015000	001407				BEQ	CMISC	: IF MISC CHAR, CALL MISC PROC.
2066	015002					CALL	@R1,<FRELEM>	: CALL PROCESSING ROUTINE.
2067	015016	000407				BR	TTRET	: RETURN.
2068								
2069	015020				CMISC:	CALL	PRMISC,<FRELEM>	: PROCESS MISCELLANEOUS CHARACTER
2070	015036				TTRET:	RETURN		: RETURN.

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 35
PCLEXR.P11 17-MAR-78 11:27 TTY INPUT CHARACTER PROCESSING ROUTINES

SEQ 0095

2072						.SBTTL	TTY INPUT CHARACTER PROCESSING ROUTINES	
2073								
2074								
2075						:	PROCESSOR FOR CARRIAGE RETURN	
2076						:	THIS CHARACTER SIGNIFIES THE END OF A COMMAND LINE.	
2077						:		
2078	015040					PROC	PRCR	
(1)								
(1)	015040				PRCR:			: **ENTRY POINT**
2079	015040	105767	002506			TSTB	REQINP	: IF INPUT NOT BEING TYPED,
2080	015044	002015				BGE	CRRET	: IGNORE CHARACTER.
2081	015046	112777	000015	001240		MOVB	#CR,@CBUFF PT	: PUT CR IN BUFFER.
2082	015054	105367	002473			DECB	CMDENT	: SET COMMAND ENTERED FLAG.
2083	015060	105067	002466			CLRB	REQINP	: CLEAR INPUT REQUESTED FLAG.
2084	015064					CALL	PNCRLF	: ECHO CR & LF.

```

2085 015100          CRRET: RETURN                      ;RETURN.
2086
2087
2088          :
2089          : PROCESSOR FOR CONTROL-O CHARACTER.
2090          : ALL CHARACTERS PRESENTLY IN THE TTY OUTPUT QUEUE ARE THROWN AWAY.
2091 015102          PROC PRCTLO
2092          :
2093          : PRCTLO:
2094          : CLR TOQ
2095          : MOV TOQ+QBACK,TOQ+QFRONT
2096          : CALL PNTLIN,<CTLQMG>
2097          : RETURN
2098          : **ENTRY POINT**
2099          : THROW AWAY ALL CHARACTERS
2100          : IN TTY OUTPUT QUEUE.
2101          : ECHO " O".
2102          : RETURN.

```

```

2103          :
2104          : PROCESSOR FOR CONTROL S.
2105          : TEMPORARILY SUSPEND PRINTOUT ON CONSOLE DEVICE
2106          :
2107          : PRINTOUT WILL BE RESUMED UPON RECEIPT OF A CNTRL-Q
2108          : OR ANY OTHER KEYBOARD INTERRUPT.
2109          :

```

```

2110          :
2111          : PROC PRCTLS
2112          :
2113          : PRCTLS:
2114          : BIS #B15,THLTFL
2115          : RETURN
2116          : **ENTRY POINT**
2117          : SET TTY HALT FLAG

```

```

2118          :
2119          : PROCESSOR FOR CONTROL Q.
2120          : RESUME PRINTOUT ON CONSOLE IF TTOUT QUEUE HAS NOT BEEN
2121          : CLEARED BY SOME OTHER MEANS.
2122          :

```

```

2123          :
2124          : PROC PRCTLQ
2125          :
2126          : PRCTLQ:
2127          : BIC #B15,THLTFL
2128          : RETURN
2129          : **ENTRY POINT**
2130          : CLEAR TTY HALT FLAG

```

CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 35-1
 PCLEXR.P11 17-MAR-78 11:27 TTY INPUT CHARACTER PROCESSING ROUTINES

SEQ 0096

```

2131          :
2132          : PROCESSOR FOR CONTROL-C.
2133          : ECHO " C" AND GO TO PRESC ROUTINE.
2134          :

```

```

2135          :
2136          : PROC PRCTLC
2137          :
2138          : PRCTLC:
2139          : CALL PNTLIN,<CTLQMG>
2140          : JMP PRESC
2141          : **ENTRY POINT**
2142          : ECHO " C"
2143          : GO TO PRESC ROUTINE.

```

```

2144          :
2145          : PROCESSOR FOR CONTROL-U.
2146          : INPUT COMMAND BEING TYPED IS DISCARDED.
2147          :

```

```

2148          :
2149          : PROC PRCTLU
2150          :
2151          : PRCTLU:
2152          : **ENTRY POINT**

```

```

2135 015176                                CALL    PNTLIN,<CTLUMG>          ;ECHO " U".
2136                                         ;DROP INTO PRESC
2137
2138
2139
2140
2141      :
2142      : PROCESSOR FOR CNTRL-C KEY.
2143      : THIS KEY IS USED TO REQUEST INPUT.  ALL OTHER EVENTS WILL CEASE
2144      : AFTER THE CURRENT LINE.  A '$' WILL BE ECHOED ON THE TTY.  THE
2145      : OPERATOR IS NOW FREE TO TYPE IN A COMMAND.
2146      :
2147 015214                                PROC    PRESC
2148      (1)
2149      (1) 015214                                PRESC:                                ;**ENTRY POINT**
2150 015214 005767 002342                                TST      KWFLG                                ;GOT A CLOCK?
2151 015220 001402                                BEQ      14$                                ;NO.
2152 015222 005077 000642                                CLR      @LCS                                ;STOP THE CLOCK
2153 015226 105367 002320      14$:  DECB     REQINP                                ;SET INPUT REQUESTED FLAG.
2154 015232 005067 002342                                CLR      STSEV                                ;CLR STATUS EVENT
2155 015236 005067 002332                                CLR      SUMEV                                ;CLR SUMMARY EVENT
2156 015242 005067 002316                                CLR      TEREV                                ;CLR XMTR ERROR EVENT
2157 015246 005067 002314                                CLR      REREV                                ;AND RCVR ERROR EVENT
2158 015252 005067 002334                                CLR      PCLGO                                ;STOP XMTR
2159 015256 012767 016110 001010      CALL     ENQ,<A.$,TOQ>          ;ECHO '$'.
2160 015276 105067 002243      MOV      #CMDBUF,CBUFPT          ;INITIALIZE COMMAND BUF PTR.
2161 015304 005067 002243      CLRB     CMDENT                                ;CLEAR COMMAND ENTERED FLAG.
2162                                         ;RETURN.
2163
2164      :
2165      : PROCESS RUB OUT KEY.
2166      : LAST CHARACTER IN COMMAND BUFFER IS DELETED.  A " " IS ECHOED.
2167      :
2168 015312                                PROC    PRDEL
2169      (1)
2170      (1) 015312                                PRDEL:                                ;**ENTRY POINT**
2171 015312 105767 002234                                TSTB     REQINP                                ;IF INPUT NOT REQUESTED,
2172 015316 002034                                BGE      PDRET                                ; THEN IGNORE RUBOUT CHAR.
2173 015320 026727 000770 016110      CMP      CBUFPT,#CMDBUF          ;IF AT BEGINNING OF BUFFER,
2174 015326 003016                                BGT      1$                                ; THEN ISSUE NEW '$'
2175 015330                                CALL     PNCRLF
2176 015334                                CALL     PRESC
2177 015360 000167 000024      JMP      PDRET
2178 015364 005367 000724      1$:  DEC      CBUFPT                                ;DECREMENT BUFFER POINTER.
2179 015370                                CALL     ENQ,<A.BKSL,TOQ>          ;ECHO A " ".
2180 015410      PDRET:  RETURN                                ;RETURN.
2181
2182      :
2183      : PROCESSOR FOR MISCELLANEOUS CHARACTERS.
2184      : THE CHARACTER IS APPENDED TO THE COMMAND BUFFER.
2185      :
2186
2187 015412                                PROC    PRMISC,<CHAR>
2188      (3) 000002                                CHAR    =      2
2189      (1)
2190      (1) 015412                                PRMISC:                                ;**ENTRY POINT**

```

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 35-2
PCLEXR.P11 17-MAR-78 11:27 TTY INPUT CHARACTER PROCESSING ROUTINES

SEQ 0097

```

2188 015412 105767 002134 TSTB REQINP ; IF INPUT NOT REQUESTED,
2189 015416 002034 BGE PMRET ; THEN JUST IGNORE CHARACTER.
2190 015420 117500 000002 MOVB @CHAR(R5),R0
2191 015424 120027 000040 CMPB R0,#' ; IF CHARACTER
2192 015430 002403 BLT PM176 ; IS NONPRINTING
2193 015432 120027 000140 CMPB R0,#140 ; THEN CHANGE
2194 015436 002402 BLT PM187 ; IT TO
2195 015440 012700 000077 MOV #'?',R0 ; ASCII '?'.
2196 015444 026727 000644 016313 PM176: CMP CBUFPT,#CBFEND-1 ; IF WE ARE AT END OF BUFFER,
2197 015452 001416 BEQ PMRET ; THEN JUST RETURN.
2198 015454 110077 000634 MOVB R0,@CBUFPT ; PUT CHARACTER INTO BUFFER
2199 015460 005267 000630 INC CBUFPT ; UPDATE BUFFER POINTER.
2200 015464 110067 012510 MOVB R0,BKELEM
2201 015470 CALL ENQ,<BKELEM,TOQ> ; ECHO CHARACTER.
2202 015510 PMRET: RETURN ; RETURN.
CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 36
PCLEXR.P11 17-MAR-78 11:27 TTY OUTPUT HANDLERS
SEQ 0098

2204 .SBTTL TTY OUTPUT HANDLERS
2205
2206
2207
2208
2209
2210 015512
2211 (1)
2212 (1) 015512
2213 015512 105777 000344 TTYOUT:
2214 015516 002020 TSTB @TTXCSR ; **ENTRY POINT**
2215 015520 005767 002110 BGE TORET ; IF DEVICE IS NOT READY,
2216 015524 001402 TST THLTFL ; THEN JUST RETURN.
2217 015532 000167 000026 BEQ 1$ ; IS TTY HALTED (CNTRL-S)?
2218 015560 116777 012420 000304 JMP TORET ; NO, O.K. TO PRINT
2219
2220
2221
2222
2223
2224 015562
2225 015562 005067 002046 1$: CALL DEQ,<FRELEM,TOQ> ; YES, DON'T DO ANYTHING.
2226 015566 117767 000262 012406 MOVB FRELEM,@TTXBUF ; GET NEXT CHAR TO TYPE.
2227 015572 000002 TORET: RETURN ; OUTPUT IT.
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241 015626
2242 (3) 000002
2243 (1)
2244 (1) 015626
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
28
```

```

2242 015626 016546 000002      6 6
2243 015632 117667 000000 012340 PLODEC: MOV    MSGAG(R5),-(SP)      ;GET ADDRESS OF MESSAGE.
2244 015640 001412                MOV    @ (SP),BKELEM      ;GET A CHARACTER.
2245 015642                BEQ    PLCLRF      ;IF NULL, APPEND CR & LF.
2246 015662 005216                CALL    ENQ,<BKELEM,TOO>      ;ENQUEUE CHAR FOR TTY OUTPUT.
2247 015664 000762                INC    @SP      ;POINT TO NEXT CHARACTER.
2248                BR    PLODEC      ;PROCESS IT.
2249 015666 005026                PLCLRF: CLR    (SP)+      ;POP ADDR FROM STACK.
2250 015670 000400                BR    PL219      ;APPEND CR & LF.
2251
2252
2253                :
2254                : ENQUEUE A CR & LF IN TTY OUTPUT QUEUE.
2255                :
2256 015672                PROC    PNCRLF
(1)
(1) 015672                PNCRLF:
2257 015672                PL219: CALL    ENQ,<A.CR,TOO>      ;**ENTRY POINT**
2258 015712                CALL    ENQ,<A.LF,TOO>      ;ENQUEUE A CR.
2259 015732                CALL    ENQ,<ZERO,TOO>      ;ENQUEUE A LF.
2260 015752                RETURN      ;ENQUEUE A NULL FILL CHAR.
;RETURN.
CZPLAAO PCL11 EXERCISER V-02      MACY11 30A(1052) 28-APR-78 13:50 PAGE 38
PCLEXR.P11 17-MAR-78 11:27      DATA AREAS

```

SEQ 0100

```

2262                .SBTTL DATA AREAS
2263
2264                : TABLE ASSOCIATING SPECIAL TTY INPUT CHARACTERS WITH THEIR
2265                : PROCESSING ROUTINES.
2266 015754                CMCHTB:
2267 015754 000003 015154                .WORD    CTL.C,PRCTL C
2268 015760 000012 015040                A.LF: .WORD    LF.,PRCR
2269 015764 000015 015040                A.CR: .WORD    CR.,PRCR
2270 015770 000017 015102                .WORD    CTL.O,PRCTLO
2271 015774 000021 015144                .WORD    CTL.Q,PRCTLO
2272 016000 000023 015134                .WORD    CTL.S,PRCTLS
2273 016004 000025 015176                .WORD    CTL.U,PRCTLU
2274 016010 000177 015312                .WORD    RUBOUT,PRDEL
2275 016014 177777                .WORD    -1
2276
2277
2278                ;DEVICE ADDRESS TABLES:
2279
2280 016016 164200                TCR: .WORD    PCLTXM
2281 016020 164202                TSR: .WORD    PCLTXM+2
2282 016022 164204                TSDB: .WORD    PCLTXM+4
2283 016024 164206                TSBC: .WORD    PCLTXM+6
2284 016026 164210                TSBA: .WORD    PCLTXM+10
2285 016030 164217                TMMR: .WORD    PCLTXM+12
2286 016032 164213                TMMRH: .WORD    PCLTXM+13
2287 016034 164214                TSCRC: .WORD    PCLTXM+14
2288 016036 164220                RCR: .WORD    PCLRCV
2289 016040 164222                RSR: .WORD    PCLRCV+2
2290 016042 164224                RDDB: .WORD    PCLRCV+4
2291 016044 164226                RDBC: .WORD    PCLRCV+6
2292 016046 164230                RDBA: .WORD    PCLRCV+10
2293 016050 164234                RDCRC: .WORD    PCLRCV+14
2294
2295 016052 000170                TXMVEC: .WORD    170
2296 016054 000174                RCVVEC: .WORD    174
2297
2298 016056 177560                TTRCSR: .WORD    TTDEV
2299 016060 177562                TTRBUF: .WORD    TTDEV+2
2300 016062 177564                TTXCSR: .WORD    TTDEV+4

```

2301 016064 177566
 2302 016066 000060
 2303 016070 177546
 2304
 2305
 2306
 2307
 2308 016072 000044
 2309 016074 000134
 2310 016076 000000
 2311
 2312
 2313
 2314
 2315
 2316
 2317

TTXBUF: .WORD TTDEV+6
 TIVECT: .WORD TIVCTR
 LCS: .WORD 177546

; VECTOR ADDRESS.
 ; KW11-L LINE CLOCK ADDR

; CHARACTER CONSTANTS. THESE ARE DEFINED AS WORDS SO THAT THEY MAY
 ; BE ENQUEUED.

A.S: .WORD 'S' ; ASCII 'S'
 A.BKSL: .WORD ' ' ; ASCII ' '
 ZERO: .WORD 0 ; ASCII NULL

.EVEN

; DEVICE ADDRESS AND VECTOR VARIABLES
 ; CHANGE THESE LOCATIONS TO MODIFY ALL DEVICE ADDRESSES AND VECTORS
 ; FOR PCL11.

CZPLA0 PCL11 EXERCISER V-02
 PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 38-1
 DATA AREAS

SEQ 0101

2318 016100 164200
 2319 016102 000170
 2320 016104 164220
 2321 016106 000174
 2322
 2323

TXDEV: .WORD 164200
 TXVEC: .WORD 170
 RCDEV: .WORD 164220
 RCVEC: .WORD 174

; DEFAULT IS 164200
 ; DEFAULT IS 170
 ; DEFAULT IS 164220
 ; DEFAULT IS 174

2324 016110 000204
 2325 016314 016314
 2326 016314 033333
 2327
 2328 016316 047536 000
 2329 016321 136 000125
 2330 016324 041536 000
 2331

; BUFFER & MESSAGE AREAS.
 CMDBUF: .BLKB 132.
 CBFEND =
 CBUFPT: .WORD 33333

; TTY INPUT COMMAND BUFFER.
 ; CMDBUF BUFFER POINTER.

2332 016327 040 051105 047522
 016334 000122

CTLONG: .ASCIZ " O"
 CTLUMG: .ASCIZ " U"
 CTLCMG: .ASCIZ " C"
 RPMSG: .ASCIZ " ERROR"

; FOR ECHOING CONTROL CHARACTERS.
 ; "
 ; "

2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344

; QUEUE DEFINITIONS.
 .IRP LC <AO, TI, TO>
 .EVEN
 .LIST
 LC'Q: .WORD 0
 .WORD LC'SIZE
 .WORD LC'AREA, LC'END
 .WORD 33333, 33333
 LC'AREA: .BLKW LC'SIZE
 LC'END
 .NI LIST
 .ENDM

; QELEMS
 ; QSIZE
 ; QTOP & QBOT
 ; QFRONT & QBACK
 ; LC'Q AREA

(1) 016336 000000
 (1) 016340 000040
 (1) 016342 016352 016452
 (1) 016346 033333 033333
 (1) 016352 000040
 (1) 016452 016452
 (1) 016452 000000
 (1) 016454 000024
 (1) 016456 016466 016536
 (1) 016462 033333 033333
 (1) 016466 000024
 (1) 016536 016536
 (1) 016536 000000
 (1) 016540 000400
 (1) 016542 016552 017552

AQQ: .WORD 0
 .WORD AOSIZE
 .WORD AOAREA, AOEND
 .WORD 33333, 33333
 AOAREA: .BLKW AOSIZE
 AOEND =
 TIQ: .WORD 0
 .WORD TISIZE
 .WORD TIAREA, TIEND
 .WORD 33333, 33333
 TIAREA: .BLKW TISIZE
 TIEND =
 TOQ: .WORD 0
 .WORD TOSIZE
 .WORD TOAREA, TOEND

; QELEMS
 ; QSIZE
 ; QTOP & QBOT
 ; QFRONT & QBACK
 ; AQQ AREA
 ; QELEMS
 ; QSIZE
 ; QTOP & QBOT
 ; QFRONT & QBACK
 ; TIQ AREA
 ; QELEMS
 ; QSIZE
 ; QTOP & QBOT

(1)	016546	033333	033333						
(1)	016552	000400		TOAREA: .WORD	33333	33333		1	6
(1)		017552		TOEND =					
2345									
2346				: FLAG VARIABLES.					
2347				: < 0 ==> TRUE					
2348				: >=0 ==> FALSE					
2349									
2350	017552	333		REQINP: .BYTE	333				
2351	017553	333		CMDENT: .BYTE	333				
2352	017554	000		TICKS: .BYTE	0				
2353	017555	000		SECNDS: .BYTE	0				
2354	017556	000		MINUTS: .BYTE	0				
CZPLAAO PCL11 EXERCISER V-02									
PCLEXR.P11 17-MAR-78 11:27				MACV11 30A(1052) 28-APR-78 13:50					
				PAGE 38-2					
				DATA AREAS					

:QFRONT & QBACK
:TOO AREA

:INPUT REQUEST IS BEING TYPED.
:COMMAND HAS BEEN ENTERED.

SEQ 0102

2355		017560			
2356	017560	000000		HOURS: .WORD	0
2357	017562	000000		KWFLG: .WORD	0
2358	017564	000000		TEREV: .WORD	0
2359	017566	000000		REREV: .WORD	0
2360	017570	000000		QLDEV: .WORD	0
2361	017572	000000		DATGEV: .WORD	0
2362	017574	000000		SUMEV: .WORD	0
2363	017576	000000		SUMSV: .WORD	0
2364	017600	000000		STSEV: .WORD	0
2365	017602	000000		STPNTR: .WORD	0
2366	017604	000000		SUMPNT: .WORD	0
2367	017606	000000		TXMEV: .WORD	0
2368	017610	000000		TXMSV: .WORD	0
2369	017612	000000		PCLGO: .WORD	0
2370	017614	000000		RCVEV: .WORD	0
2371	017616	000000		SPCEV: .WORD	0
2372	017620	000000		XSPCEV: .WORD	0
2373	017622	000000		FIRST: .WORD	0
2374	017624	000000		NEXT: .WORD	0
2375	017626	000000		PRINTD: .WORD	0
2376	017630	000000		HDRFLG: .WORD	0
2377	017632	000000		PADFLG: .WORD	0
2378	017634	000000		THLTFL: .WORD	0
2379					
2380				.EVEN	
2381					
2382				: DATA VARIABLES.	
2383	017636	133333		ORIGSD: .WORD	133333
2384	017640	000000		DTSEED: .WORD	0
2385	017642	000333		MLSD: .WORD	333
2386	017644	000000		MSGLSO: .WORD	0
2387	017646	000000		RCSEED: .WORD	0
2388	017650	000000		RCTXPS: .WORD	0
2389	017652	000000		RJCTF: .WORD	0
2390	017654	000000		TRNKF: .WORD	0
2391	017656	000000		TXML: .WORD	0
2392	017660	037565		RANM: .WORD	37565
2393	017662	012247		RANK: .WORD	12247
2394	017664	000000		CURAD: .WORD	0
2395	017666	000000		RSPC: .WORD	0
2396	017670	000000		ESCFLG: .WORD	0
2397	017672	000000		OBJCNT: .WORD	0
2398	017674	000000		RSHOLD: .WORD	0
2399	017676	000000		TEMP: .WORD	0
2400	017700	000000		TEMP1: .WORD	0
2401	017702	000000		ERRO: .WORD	0

```

2402 017704 000000 ERR1: .WORD 0 J 6
2403 017706 000000 ERR2: .WORD 0
2404
2405
2406 ;ADDRESS SILO DATA BUFFER AREA
2407
2408 017710 000102 ADSILO: .BLKB 66. ;66. BYTE AREA FOR SILO DATA
2409
2410 ;RECEIVER ADDRESS TABLE
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 38-3
PCLEXR.P11 17-MAR-78 11:27 DATA AREAS SEQ 0103
2411
2412 000001 X = 1
2413 020012 RADB:
2414 000037 .REPT 31.
2415 .WORD 0 ;ACTIVITY FLAG
2416 .WORD X ;RECEIVER ADDRESS
2417 .WORD 0,0 ;ATTEMPTS ENTRY
2418 .WORD 0,0 ;SUCCESSSES ENTRY
2419
2420 X = X+1
2421 .ENDR
2422 020576 020576 RADEND: .WORD .
2423 017776 RADBO= RADB-14
2424
2425 ;TRANSMITTER DATA BUFFER:
2426
2427 020600 001020 DATBUF: .BLKW 1020
2428
2429
2430 ;RECEIVER DATA BUFFER:
2431
2432 022640 001000 RCBUF: .BLKW 1000
2433
2434 ;EXERCISER ERROR TABLE
2435
2436 000001 Y = 1 ;INITIAL ERROR NUMBER
2437 024640 ERTBL:
2438 000030 .REPT N-1
2439 .WORD Y ;ERROR NUMBER
2440 .WORD 0 ;ERROR ADDRESS
2441 .WORD 0 ;NO. OF OCCURRENCES SINCE INIT
2442
2443 Y = Y+1
2444 .ENDR
2445 025060 177777 .WORD -1 ;LAST ERROR # IS -1
2446
2447 024632 ERTBLO = ERTBL-6
2448
2449
2450 ; DETAILED ERROR TABLES FOR RCVR AND XMTR ERRORS:
2451
2452 025062 000623 TERTBL: .BLKW 37*15 ;RESERVE SPACE FOR XMTR ERRORS
2453
2454
2455 026530 000623 RERTBL: .BLKW 37*15 ;RESERVE SPACE FOR RCVR ERRORS
2456
2457
2458
2459
2460 030176 033333 FRELEM: .WORD 33333 ;STORAGE FOR DEQUEUED ELEMENT.
2461 030200 033333 BKELEM: .WORD 33333 ;STORAGE FOR ENQUEUED ELEMENT.

```

2462 030202 033333
 2463 030204 033333
 2464 030206 033333
 2465
 2466

TCBFPT: .WORD 33333
 TCBIN: .WORD 33333
 INTMP: .WORD 33333

:CMDBUF POINTER USED DURING SCAN
 :BINARY VALUE OF INPUT PARAMETER
 :TEMP STORAGE FOR INTERRUPT PROC

CZPLAAO PCL11 EXERCISER V-02
 PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 38-4
 KEYWORD TABLE

SEQ 0104

2467

2468

2469

:KEYWORD TABLE ASSOCIATING A COMMAND WITH ITS PROCESSING ROUTINE

2470 030210 004536

CMDTBL: .WORD CPADD

2471 030212 002 003

.BYTE 2,3

2472 030214 042101 020104

.ASCII /ADD /

2473 030220 005704

.WORD CPASS

2474 030222 002 006

.BYTE 2,6

2475 030224 051501 044523 047107

.ASCII /ASSIGN/

2476 030232 004764

.WORD CPCLR

2477 030234 002 005

.BYTE 2,5

2478 030236 046103 040505 020122

.ASCII /CLEAR /

2479 030244 005126

.WORD CPCNT

2480 030246 002 010

.BYTE 2,8

2481 030250 047503 052116 047111

.ASCII /CONTINUE/

2482 030256 042525

2483 030260 004644

.WORD CPDEL

2484 030262 001 006

.BYTE 1,6

2485 030264 042504 042514 042524

.ASCII /DELETE/

2486 030272 006612

.WORD CPERR

2487 030274 001 006

.BYTE 1,6

2488 030276 051105 047522 051522

.ASCII /ERRORS/

2489 030304 005522

.WORD CPGO

2490 030306 001 002

.BYTE 1,2

2491 030310 047507

.ASCII /GO/

2492 030312 005222

.WORD CPINIT

2493 030314 001 012

.BYTE 1,10

2494 030316 047111 052111 040511

.ASCII /INITIALIZE/

2495 030324 044514 042532

2496 030330 004130

.WORD CPMAS

2497 030332 001 006

.BYTE 1,6

2498 030334 040515 052123 051105

.ASCII /MASTER/

2499 030342 004400

.WORD CPRANG

2500 030344 002 005

.BYTE 2,5

2501 030346 040522 043516 020105

.ASCII /RANGE /

2502 030354 004306

.WORD CPRIB

2503 030356 002 003

.BYTE 2,3

2504 030360 044522 020102

.ASCII /RIB /

2505 030364 004214

.WORD CPSEC

2506 030366 002 011

.BYTE 2,9

2507 030370 042523 047503 042116

.ASCII /SECONDARY /

2508 030376 051101 020131

2509 030402 003420

.WORD CPSILO

2510 030404 002 004

.BYTE 2,4

2511 030406 044523 047514

.ASCII /SILO/

2512 030412 005066

.WORD CPSTAT

2513 030414 002 006

.BYTE 2,6

2514 030416 052123 052101 051525

.ASCII /STATUS/

2515 030424 005470

.WORD CPSUM

2516 030426 002 007

.BYTE 2,7

2517 030430 052523 046515 051101

.ASCII /SUMMARY /

2518 030436 020131

2519 030440 000000 000000

.WORD 0,0

2520

2521

2522

SEQ 0105

SEQ 0106

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 38-6
PCLEXR.P11 17-MAR-78 11:27 SOME MORE ASCII STORAGE:

2538	031064	042504	044526	042503	ERTMSG: .ASCIIZ /DEVICE ADDRESS ERROR. USE 'ASSIGN' COMMAND./
	031072	040440	042104	042522	
	031100	051523	042440	051122	
	031106	051117	020056	051525	
	031114	020105	040442	051523	
	031122	043511	021116	041440	
	031130	046517	040515	042116	
	031136	000056			
2539	031140	054105	051105	044503	EXRST: .ASCIIZ /EXERCISER STARTED/
	031146	042523	020122	052123	
	031154	051101	042524	000104	
2540	031162	054105	051105	044503	EXCNT: .ASCIIZ /EXERCISER CONTINUING/
	031170	042523	020122	047503	
	031176	052116	047111	044525	
	031204	043516	000		
2541	031207	052	025052	044124	MSTMG1: .ASCIIZ /***THIS UNIT IS NOT MASTER***/
	031214	051511	052440	044516	
	031222	020124	051511	047040	
	031230	052117	046440	051501	
	031236	042524	025122	025052	
	031244	000			
2542	031245	102	052125	044040	MSTMG2: .ASCIIZ /BUT HAS NOW BEEN MADE SECONDARY./
	031252	051501	047040	053517	
	031260	041040	042505	020116	
	031266	040515	042504	051440	
	031274	041505	047117	040504	
	031302	054522	000056		
2543	031306	044124	020105	044523	MSTMG3: .ASCIIZ /THE SILO YOU HAVE JUST LOADED WILL BE/
	031314	047514	054440	052517	
	031322	044040	053101	020105	
	031330	052512	052123	046040	
	031336	040517	042504	020104	
	031344	044527	046114	041040	
	031352	000105			
2544	031354	051525	042105	044440	MSTMG4: .ASCIIZ /USED IF YOU CLEAR THE CURRENT MASTER./
	031362	020106	047531	020125	
	031370	046103	040505	020122	
	031376	044124	020105	052503	
	031404	051122	047105	020124	
	031412	040515	052123	051105	
	031420	000056			
2545	031422	044124	020105	044523	MSTMG5: .ASCIIZ /THE SILO HAS BEEN PADDED WITH ADDRESS '0'/
	031430	047514	044040	051501	
	031436	041040	042505	020116	
	031444	040520	042104	042105	
	031452	053440	052111	020110	
	031460	042101	051104	051505	
	031466	020123	030042	000042	
2546	031474	044124	051511	052440	THUMST: .ASCIIZ /THIS UNIT IS -MASTER-/
	031502	044516	020124	051511	
	031510	026440	040515	052123	
	031516	051105	000055		
2547	031522	044124	051511	052440	THUSCN: .ASCIIZ /THIS UNIT IS -SECONDARY-/
	031530	044516	020124	051511	
	031536	026440	042523	047503	
	031544	042116	051101	026531	

CZPLAA0 PCL11 EXERCISER V-02
 PCLEXR.P11 17-MAR-78 11:27

30A(1052) 28-APR-78 13:50 PAGE 38-7
 SOME MORE ASCII STORAGE:

2548 031552 000
 031553 042 044522 021102 RBSTMG: .ASCIIZ /"RIB" IS -SET-/
 031560 044440 020123 051455
 031566 052105 000055

SEQ 0107

Line	Address	Source	Destination	Port	Operation	Details
2549	031572	051042	041111	020042	RBCLMG: .ASCIZ	/'RIB' IS -CLEAR-/ N 6
	031600	051511	026440	046103		
	031606	040505	026522	000		
2550	031613	105	040514	051520	ELPSTM: .ASCII	/ELAPSED TIME (HRS:MIN:SEC:TIK).../
	031620	042105	052040	046511		
	031626	020105	044050	051522		
	031634	046472	047111	051472		
	031642	041505	052072	045511		
	031650	027051	027056			
2551	031654	035060	035060	035060	TMLIN1: .ASCIZ	/0:0:0:0 /
	031662	020060	020040	020040		
	031670	020040	020040	000		
2552	031675	122	053103	020122	STITLE: .ASCIZ	/RCVR ADDRESS CONNECTION ATTEMPTS SUCCESSFUL CONNECTIONS/
	031702	042101	051104	051505		
	031710	020123	041440	047117		
	031716	042516	052103	047511		
	031724	020116	052101	042524		
	031732	050115	051524	020040		
	031740	052523	041503	051505		
	031746	043123	046125	041440		
	031754	047117	042516	052103		
	031762	047511	051516	000		
2553						
2554	031767				STLIN:	
2555	031767	116	047116	020116	STLIN1: .ASCII	/NNNN /
	031774	020040	020040	020040		
	032002	020040	020040	020040		
	032010	040				
2556	032011	116	047116	020116	STLIN2: .ASCII	/NNNN /
	032016	020040	020040	020040		
	032024	020040	020040	020040		
	032032	020040	020040	040		
2557	032037	116	047116	020116	STLIN3: .ASCII	/NNNN /
	032044	020040	020040	020040		
	032052	020040	000			
2558						
2559						
2560	032055	105	051122	051117	SMTITLE: .ASCIZ	/ERROR NUMBER ERROR ADDRESS NO. OF OCCURRENCES/
	032062	047040	046525	042502		
	032070	020122	020040	042440		
	032076	051122	051117	040440		
	032104	042104	042522	051523		
	032112	020040	020040	047516		
	032120	020056	043117	047440		
	032126	041503	051125	042522		
	032134	041516	051505	000		
2561						
2562	032141	040	025040	020052	NOERMG: .ASCIZ	/ ** NO ERRORS TO REPORT YET **/
	032146	047516	042440	051122		
	032154	051117	020123	047524		
	032162	051040	050105	051117		
	032170	020124	042531	020124		
CZPLAAO	PCL11	EXERCISER	V-02	MACY11	30A(1052)	28-APR-78 13:50 PAGE 38-8
PCLEXR.P11	17-MAR-78	11:27			SOME MORE ASCII STORAGE:	
2563	032176	025052	000			
	032201	124	040522	051516	TERHDR: .ASCIZ	/TRANSMITTER ERRORS:/
	032206	044515	052124	051105		
	032214	042440	051122	051117		
	032222	035123	000			
2564	032225	105	051122	051117	TRHLIN: .ASCIZ	/ERROR NO. CONCTD RCVR ERROR COUNT/
	032232	047040	027117	020040		
	032240	020040	047503	041516		

SEQ 0108

032246	042124	051040	053103	
032254	020122	020040	042440	
032252	051122	051117	041440	
032270	052517	052116	000	
2565	032275			TRELIN:
2566	032275	116	047116	047116
	032302	020040	020040	020040
	032310	020040		
2567	032312	047116	047116	020116
	032320	020040	020040	020040
	032326	020040	040	
2568	032331	116	047116	020116
	032336	020040	000	
2569	032341	122	041505	044505
	032346	042526	020122	051105
	032354	047522	051522	000072
2570	032362	051105	047522	020122
	032370	047516	020056	020040
	032376	041440	047117	052103
	032404	020104	046530	051124
	032412	020040	020040	051105
	032420	047522	020122	047503
	032426	047125	000124	
2571	032432			RCELIN:
2572	032432	047116	047116	020116
	032440	020040	020040	020040
	032446	040		
2573	032447	116	047116	047116
	032454	020040	020040	020040
	032462	020040	020040	
2574	032466	047116	047116	020040
	032474	000040		
2575	032476	020040	047050	047117
	032504	024505	000	
2576	032507			SMLIN:
2577	032507	116	047116	020116
	032514	020040	020040	020040
	032522	020040	020040	020040
	032530	040		
2578	032531	116	047116	020116
	032536	020040	020040	020040
	032544	020040	020040	020040
	032552	020040	040	
2579	032555	116	047116	020116
	032562	020040	000040	

RRHIN: .ASCII /ERROR NO. CONCTD XMTR ERROR COUNT/

RCTR: .ASCII /NNNNN /

RCERC: .ASCII /NNNN /

NONMG: .ASCII / (NONE)/

SMLIN: SMLIN1: .ASCII /NNNN /

SMLIN2: .ASCII /NNNN /

SMLIN3: .ASCII /NNNN /

CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 39
PCLEXR.P11 17-MAR-78 11:27 SOME MORE ASCII STORAGE:

SEQ 0109

2583 .EVEN
2584 .SBTTL AUXILIARY ROUTINES
2585 .SBTTL CHARACTER PROCESSOR

: CONVERT A CHARACTER TO ITS PROCESSING ROUTINE ADDRESS BASED
: UPON A TABLE OF ENTRIES IN THE FOLLOWING FORM:

-----	-----
(UNUSED)	CHARACTER
-----	-----
PROCESSING ROUTINE ADDR	
-----	-----

: THE TABLE MUST BE ARRANGED IN ASCENDING ORDER OF CHARACTER VALUES.

```

2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608 032566
2609 032566 121100
2610
2611 032570 001404
2612
2613 032572 101006
2614
2615 032574 062701 000004
2616
2617 032600 000772
2618
2619 032602 016101 000002
2620
2621 032606 000207
2622
2623
2624 032610 005001
2625 032612 000207
2626

```

```

: THE TABLE ENDS WITH A DUMMY ENTRY FOR CHARACTER FF (HEXADECIMAL).
:
: ON ENTRY:  R0 = CHARACTER
:             R1 = TABLE ADDRESS
:
: CALL:      JSR     PC,PROCHR
:
: ON RETURN: R0 = CHARACTER
:             R1 = PROCESSING ROUTINE ADDRESS, IF ANY
:             Z  = 1 ==> CHARACTER NOT IN TABLE
:
PROCHR:
PCLOOK: CMPB   @R1,R0
:             BEQ    PCCALL
:             BHI    PCQUIT
:             ADD    #4,R1
:             BR     PCLOOK
:
PCCALL: MOV     2(R1),R1
:             RTS    PC
:
PCQUIT: CLR     R1
:             RTS    PC

```

```

: **ENTRY POINT**
: COMPARE TABLE CHAR
: WITH ARG. CHAR.
: IF SAME, RETURN PROC.
: ROUTINE ADDR.
: IF >, THEN ARG. CHAR
: NOT IN TABLE.
: IF <, POINT TO NEXT
: TABLE ENTRY.
: TRY AGAIN.
:
: R1 = PROCESSING
: ROUTINE ADDRESS.
: RETURN: Z BIT IS
: OFF.
:
: R1 = 0
: RETURN: Z BIT IS ON.

```

CZPLA0 PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 40
BINARY TO ASCII CONVERSION

SEQ 0110

```

2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647

```

```

.SBTTL BINARY TO ASCII CONVERSION
:
: BINARY TO ASCII CONVERSION
:
: LOCAL MACROS
:
: SETUP CONVERSION CONTROL WORD ON STACK
:
: STCVT RADIX,WIDTH,SIGNED,COMPR,BLANKS
:
: WHERE:
:
: RADIX=NUMERIC VALUE SPECIFYING CONVERSION RADIX
: WIDTH=NUMERIC VALUE FROM 1 TO 7 SPECIFYING FIELD WIDTH
: SIGNED=SIGN OR MAGNITUDE FLAG. ASCII STRING 'SIGN' SPECI-
: FIES SIGNED CONVERSION. ANYTHING ELSE SPECIFIES MAGNI-
: TUDE.
:
: COMPR-COMPRESS LEADING ZEROS FLAG. ASCII STRING 'COMPRES' SPE-

```

```

2648      D 7
2649      : CTFIES COMPRESSION OF LEADING ZEROS. ANYTHING ELSE MEANS
2650      : INCLUDE LEADING ZEROS OR SPACES IN CONVERSION.
2651      :
2652      : BLANK=REPLACE LEADING ZEROS WITH BLANKS (SPACES). ASCII STRING
2653      : 'BLANKS' MEANS BLANK REPLACEMENT IF ZERO COMPRESS IS DIS-
2654      : ABLED. ANYTHING ELSE SPECIFIES ZERO PADDING.
2655      :
2656      : .MACRO STCVT RADIX,WIDTH,SIGN,COMPR,BLANK
2657      $BLK=0
2658      $SGN=0
2659      $SUP=1*1000
2660      .IF IDN <BLANK>,<BLANKS>
2661      $BLK=1*2000
2662      .ENDC
2663      .IF IDN <SIGN>,<SIGNED>
2664      $SGN=1*400
2665      .ENDC
2666      .IF IDN <COMPR>,<COMPRES>
2667      $SUP=0*1000
2668      .ENDC
2669      MOV #<WIDTH*4000>,$BLK.$SGN.$SUP.RADIX,-(SP)
2670      .ENDM

```

CZPLA0 PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 41
 PCLEXR.P11 17-MAR-78 11:27 BINARY TO ASCII CONVERSION

SEQ 0111

```

2671      : INPUTS:
2672      :
2673      : R0=ADDRESS TO STORE FIRST BYTE IN OUTPUT STRING
2674      : R1=NUMBER TO BE CONVERTED
2675      : R2=ZERO COMPRESSION INDICATOR
2676      : IF R2 EQ 0, THEN SUPPRESS ZEROS
2677      : IF R2 NE 0, THEN DO NOT SUPPRESS ZEROS.
2678      :
2679      : IF CBTA IS CALLED, THEN R2 MUST CONTAIN THE FOLLOWING INFORMATION
2680      :
2681      : LOW BYTE=CONVERSION RADIX (2-10.)
2682      : BIT 8=MAGNITUDE/SIGNED CONVERSION (1=SIGNED)
2683      :
2684      : BIT 9 =ZERO COMPRESS FLAG (0=COMPRESS LEADING ZEROS)
2685      :
2686      : BIT 10=BLANK FILL FLAG (1=REPLACE LEADING ZEROS WITH BLANKS
2687      : IF ZERO COMPRESS DISABLED, 0=ZERO FILL).
2688      :
2689      : BITS 11-15=FIELD WIDTH (1-32)
2690      :
2691      :
2692      : OUTPUTS:
2693      :
2694      : R0=ADDRESS OF NEXT BYTE AFTER LAST DIGIT STORED.
2695      :
2696      : IF THE CONVERTED DIGIT EXCEEDS 9, THE RESULT IS BIASED TO FALL
2697      : IN THE RANGE A - Z
2698      :
2699      :
2700      : CONVERT 6 DIGIT OCTAL TO ASCII MAGNITUDE
2701      :
2702      : OCTJSP:
2703      : STCVT 8,,6,,MAGN,NOCOMP,BLANKS ;PUSH CONVERSION PARAMETERS
2704      : BR SETCN ;CONVERT TO ASCII
2705      :
2706      : CONVERT 6 DIGIT OCTAL TO ASCII (ZERO COMPR)
2707      :
2708      : OCTPNT:

```

032614
 032614
 032620 000411
 032622

				E 7		
2709	032622			STCVT	8.,6.,MAGN,COMPRES,NOBLKANK	
2710	032626	000406		BR	SETCN	
2711						
2712				; CONVERT 5 DIGIT DECIMAL TO ASCII MAGNITUDE		
2713						
2714	032630			DECJSP:		
2715	032630			STCVT	10.,5.,MAGN,NOCOMP,BLANKS	
2716	032634	000403		BR	SETCN	
2717						
2718				; CONVERT 5 DIGIT DECIMAL TO ASCII (ZERO COMPR)		
2719						
2720	032636			DECPNT:		
2721	032636			STCVT	10.,5.,MAGN,COMPRES,NOBLANK	
2722	032642	000400		BR	SETCN	
2723						
2724	032644			SETCN:		
2725	032644	005702		TST	R2	; SUPPRESS ZEROS?
2726	032646	001002		BNE	20\$	; IF NE, NO
CZPLAAO PCL11 EXERCISER V-02						
PCLEXR.P11 17-MAR-78 11:27				MACY11 30A(1052) 28-APR-78 13:50 PAGE 41-1		SEQ 0112
				BINARY TO ASCII CONVERSION		
2727	032650	042716	001000	20\$:	BIC #1*1000,(SP)	; ENABLE ZERO SUPPRESS
2728	032654					
2729	032654	012602		MOV	(SP)+,R2	; SET CONTROL WORD
CZPLAAO PCL11 EXERCISER V-02				MACY11 30A(1052) 28-APR-78 13:50 PAGE 42		SEQ 0113
PCLEXR.P11 17-MAR-78 11:27				GENERAL BINARY TO ASCII CONVERSION		
2731				.SBTTL GENERAL BINARY TO ASCII CONVERSION		
2732						
2733	032656			CBTA:		
2734	032656	004567	000406	JSR	R5,SAVRG	; SAVE THE NON-VOLATILE REGISTERS
2735	032662	110205		MOVB	R2,R5	; COPY RADIX BYTE
2736	032664	000302		SWAB	R2	; POSITION REMAINING TO LOW BYTE
2737	032666	106202		ASRB	R2	; SHIFT OFF MAG. FLAG
2738	032670	103005		BCC	10\$	; UNSIGNED IF C IS CLR
2739	032672	005701		TST	R1	; POSITIVE VALUE?
2740	032674	100003		BPL	10\$	; IF PL, YES
2741	032676	005401		NEG	R1	; MAKE VALUE POSITIVE
2742	032700	112720	000055	MOVB	#'-,(R0)+	; INSERT A MINUS SIGN
2743	032704			10\$:		
2744	032704	010004		MOV	R0,R4	; COPY STRING POINTER
2745	032706	000241		CLC		; CLEAR CARRY
2746	032710	106002		RORB	R2	; SHIFT OFF SUPPR FLAG
2747	032712	006002		ROR	R2	; TRANSFER TO R2
2748	032714	006003		ROR	R3	; GET BLANK/ZERO PAD FLAG
2749	032716	105003		CLRB	R3	; CLEAR COUNT BYTE
2750	032720	150203		BISB	R2,R3	; TRANSFER COUNT BYTE
2751	032722	105002		CLRB	R2	; CLEAR FILL BYTE
2752	032724	152702	000060	BISB	#'0,R2	; SET FILL BYTE
2753	032730	010100		MOV	R1,R0	; DIVIDEND TO R0
2754	032732			1\$:		
2755	032732	010501		MOV	R5,R1	; SET CONVERSION RADIX
2756	032734	004767	000272	JSR	PC,DIV	; DIVIDE EM UP
2757	032740	020127	000011	CMP	R1,#9.	; RESULT EXCEED NUMERICS?
2758	032744	101402		BLOS	15\$	; IF LOS, NO
2759	032746	062701	000007	ADD	#7,R1	; BIAS TO FALL IN ALPHA
2760	032752			15\$:		
2761	032752	060201		ADD	R2,R1	; ADD CHARACTER BIAS
2762	032754	010146		MOV	R1,-(SP)	; SAVE CHARACTER
2763	032756	105303		DECB	R3	; DECREMENT CHARACTER COUNT
2764	032760	003412		BLE	3\$	; IF LE NO DIGITS LEFT
2765	032762	005700		TST	R0	; ZERO QUOTIENT
2766	032764	001006		BNE	2\$	; IF NE, YES, GO AGAIN

2767	032766	005702		TST	R2	: SUPPRESS ZEROS
2768	032770	100006		BPL	38	: IF PL, YES, ALL DONE
2769	032772	005703		TST	R3	: SUBSTITUTE BLANKS?
2770	032774	100002		BPL	28	: IF PL, NO
2771	032776	042702	000020	BIC	#20,R2	: CONVERT FILL TO BLANK
2772	033002					
2773	033002	004767	177724	JSR	PC,18	: DIVIDE AGAIN
2774	033006					
2775	033006	112624		MOVB	(SP)+,(R4)+	: STORE A DIGIT
2776	033010	010400		MOV	R4,R0	: STORE TERMINAL ADDRESS
2777	033012			RETURN		
2778						

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 43
 PCLEXR.P11 17-MAR-78 11:27 DOUBLE PRECISION BINARY TO ASCII

SEQ 0114

2780				.SBTTL	DOUBLE PRECISION BINARY TO ASCII	
2781				:	CONVERT A DOUBLE PRECISION UNSIGNED QUANTITY TO DECIMAL ASCII	
2782				:		
2783				:	LOCAL MACROS	
2784				:		
2785				:	SET ASCII CONVERSION PARAMETERS	
2786				:		
2787				:	CBTAS RADIX,WIDTH,SIGN,BLANK	
2788				:		
2789				:	WHERE:	
2790				:		
2791				:	RADIX=CONVERSION RADIX	
2792				:		
2793				:	WIDTH=FIELD WIDTH	
2794				:		
2795				:	SIGN='SIGNED' FOR SIGNED CONVERSION. ANYTHING ELSE IMPLIES	
2796				:	UNSIGNED CONVERSION	
2797				:		
2798				:	BLANK='BLANKS' TO CONVERT LEADING ZEROS TO BLANKS. ANYTHING ELSE	
2799				:	IMPLIES NO CONVERSION OF ZEROS .	
2800						
2801				.MACRO	CBTAS RADIX,WIDTH,SIGN,BLANK	
2802				\$BLKS=0		
2803				\$SGNS=0		
2804				.IF	IDN <BLANK>,<BLANKS>	
2805				\$BLKS=1*2000		
2806				.ENDC		
2807				.IF	IDN <SIGN>,<SIGNED>	
2808				\$SGNS=1*400		
2809				.ENDC		
2810				MOV	#<WIDTH*400>,\$SGNS,\$BLKS RADIX,R5	
2811				TST	R2	
2812				BEQ	.+4	
2813				BIS	#1*1000,R5	
2814				.ENDM		
2815						
2816						
2817				:	INPUTS:	
2818				:		
2819				:	R0=POINTER TO ASCII OUTPUT STRING	
2820				:	R1=ADDRESS OF DOUBLE PRECISION VALUE	
2821				:	R2=ZERO COMPRESS FLAG	
2822						
2823	033014			CDDMG:		
2824	033014	004567	000250	JSR	R5,SAVRG	: SAVE THE NON-VOLATILE REGISTERS
2825	033020	010003		MOV	R0,R3	: COPY THE STRING POINTER
2826	033022	012704	023420	MOV	#10000.,R4	: SET DIVISOR
2827	033026			CBTAS	10.,0,NOSIGN,BLANKS	: SET CONVERSION PARAMETERS

```

2828 033042 022104      CMP      (R1)+,R4      ;TEST FOR OVERFLOW
2829 033044 103042      BHIS     40$      ;IF HIS, OVERFLOW
2830 033046 011102      MOV      (R1),R2      ;GET LOW PART OF NUMBER
2831 033050 014101      MOV      -(R1),R1      ;GET HIGH PART OF NUMBER
2832 033052 010400      MOV      R4,R0      ;COPY DIVISOR
2833 033054 004767 000110 JSR      PC,DDIV      ;DO DOUBLE PREC. DIVIDE
2834 033060 010046      MOV      R0,-(SP)      ;SAVE REMAINDER
2835 033062 010201      MOV      R2,R1      ;COPY QUOTIENT
CZPLAAO PCL11 EXERCISER V-02 MACY'1 30A(1052) 28-APR-78 13:50 PAGE 43-1
PCLEXR.P11 17-MAR-78 11:27 DOUBLE PRECISION BINARY TO ASCII
SEQ 0115

2836 033064 001011      BNE      11$      ;IF NE, SOMETHING TO PRINT
2837 033066 012702 000005 MOV      #5,R2      ;OTHERWISE, FILL FIELD WITH BLANKS
2838 033072 112723 000040 21$: MOVB     #'',(R3)+
2839 033076 005302      DEC      R2
2840 033100 001374      BNE      21$
2841 033102 052705 003000 BIS      #3000,R5      ;DISABLE BLANK SUPPRESSION
2842 033106 000411      BR       20$
2843 033110 012702 024000 11$: MOV      #5*4000,R2      ;SET FIELD WIDTH
2844 033114 004767 000020 JSR      PC,30$      ;OUTPUT HIGH ORDER DIGITS
2845 033120 052705 001000 BIS      #1*1000,R5      ;DISABLE ZERO COMPRESS
2846 033124 042705 002000 BIC      #1*2000,R5      ;DISABLE BLANKS
2847 033130 010003 31$: MOV      R0,R3      ;SET STRING POINTER
2848 033132 20$:
2849 033132 012601      MOV      (SP)+,R1      ;GET LOW ORDER VALUE
2850 033134 012702 020000 MOV      #4*4000,R2      ;SET FIELD WIDTH
2851 033140 30$:
2852 033140 010300      MOV      R3,R0      ;GET STRING POINTER
2853 033142 050502      BIS      R5,R2      ;INCLUDE RADIX & BLANK SUPPRESS
2854 033144 004767 177506 JSR      PC,CBTA      ;CONVERT TO ASCII
2855 033150 000406      BR       60$      ;EXIT
2856 033152 40$:
2857 033152 012702 000011 MOV      #9.,R2      ;GET COUNT
2858 033156 50$:
2859 033156 112720 000052 MOVB     #'*',(R0)+      ;FILL FIELD WITH ASTERISKS
2860 033162 005302      DEC      R2
2861 033164 001374      BNE      50$      ; "SOB R2,50$"
2862 033166 60$:
2863 033166      RETURN

.SBTTL DOUBLE PRECISION DIVIDE ROUTINE
:
: INPUTS:
: R2=LOW ORDER OF DIVIDEND
: R1=HIGH ORDER OF DIVIDEND
: R0=DIVISOR (15 BITS UNSIGNED)
:
: OUTPUTS:
: R2=LOW ORDER OF QUOTIENT
: R1=HIGH ORDER OF QUOTIENT
: R0=REMAINDER
:
2882 033170 DDIV:      MOV      R3,-(SP)      ;SAVE R3
2883 033170 010346      MOV      #32.,R3      ;SET ITERATION COUNT IN R3
2884 033172 012703 000040 MOV      R0,-(SP)      ;STACK DIVISOR
2885 033176 010046      CLR      R0      ;SET REMAINDER TO 0
2886 033200 005000 1$:
2887 033202

```

```

      2888 033202 006302          ASL    R2          ;SHIFT THE ENTIRE DIVIDEND
      2889 033204 006101          ROL    R1          ;... ONE BIT TO THE LEFT AND...
      2890 033206 006100          ROL    R0          ;... INTO THE REMAINDER
      2891 033210 020016          CMP    R0,(SP)      ;IS REMAINDER GE DIVISOR?
CZPLAAO PCL11 EXERCISER V-02      MACV11 30A(1052) 28-APR-78 13:50 PAGE 43-2
PCLEXR.P11 17-MAR-78 11:27      DOUBLE PRECISION DIVIDE ROUTINE
                                     SEQ 0116

      2892 033212 103402          BLO    2$          ;NO, SKIP TO ITERATION CONTROL
      2893 033214 161600          SUB    (SP),R0      ;YES, SUB DIVISOR OUT
      2894 033216 005202          INC    R2          ;AND INCR THE QUOTIENT
      2895 033220                2$:                ;REPEAT AS LONG AS NECESSARY
      2896 033220 005303          DEC    R3
      2897 033222 003367          BGT    1$          ;PURGE DIVISOR FROM STACK
      2898 033224 005726          TST    (SP)+
      2899 033226 012603          MOV    (SP)+,R3     ;RESTORE R3
      2900 033230          RETURN

                                     .SBTTL  INTEGER DIVIDE MAGNITUDE NUMBERS
      2901
      2902
      2903
      2904
      2905
      2906
      2907
      2908
      2909
      2910
      2911
      2912
      2913
      2914
      2915
      2916
      2917
      2918
      2919
      2920
      2921
      2922
      2923
      2924
      2925
      2926
      2927
      2928
      2929
      2930
      2931
      2932
      2933
      2934
      2935
      2936
      2937
      2938
      2939
      2940
      2941
      2942
      2943
      2944
      2945
      2946

      033232 012746 000020
      033236 010146
      033240 005001
      033242 006300
      033244 006101
      033246 020116
      033250 103402
      033252 161601
      033254 005200
      033256 005366 000002
      033262 003367
      033264 022626
      033266

      DIV:
      MOV    #20,-(SP)          ;SET LOOP COUNT
      MOV    R1,-(SP)          ;SAVE DIVISOR FOR SUBTRACTS
      CLR    R1                ;CLEAR REMAINDER
      30$:
      ASL    R0                ;DOUBLE LEFT SHIFT
      ROL    R1
      CMP    R1,(SP)           ;SUBTRACT OUT DIVISOR
      BLO    40$               ;IF LO, NO
      SUB    (SP),R1           ;SUBTRACT OUT DIVISOR
      INC    R0                ;ADD IN LOW BIT
      40$:
      DEC    2(SP)             ;DECREMENT REPEAT COUNT
      BGT    30$               ;IF GT, MORE TO GO
      50$:
      CMP    (SP)+,(SP)+       ;CLEAN STACK
      RETURN

      ; SAVE/RESTORE NONVOLATILE REGISTERS

      SAVRG: MOV    R4,-(SP)    ;SAVE R4 & R3
      MOV    R3,-(SP)
      MOV    R5,-(SP)
      MOV    6(SP),R5          ;PUT RETURN ADDRESS ON STACK
      JSR    PC,@(SP)+         ;RETRIEVE REAL R5
      MOV    (SP)+,R3          ;CALL THE CALLER ...
      MOV    (SP)+,R4          ;RESTORE NON VOLATILE REGISTERS
      MOV    (SP)+,R5
      RETURN

CZPLAAO PCL11 EXERCISER V-02      MACV11 30A(1052) 28-APR-78 13:50 PAGE 44

```

Address	Op Code	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

; SUBROUTINE TO SCAN AN INPUT COMMAND & CALL ITS
; PROCESSING ROUTINE.

.LIST MEB

.MACRO SPAN REG,CHAR,?L

; THIS MACRO SCANS THE STRING OF CHARACTERS STARTING AT
; @REG UNTIL IT FINDS ONE NOT EQUAL TO CHAR. REG IS
; SET POINTING TO THAT CHARACTER.

L: CMPB (REG)+,CHAR
BEQ L
DEC REG

.ENDM

.MACRO BREAK REG,CHRSET,?HH,?JJ

; THIS MACRO SCANS THE STRING STARTING AT @REG UNTIL
; IT FINDS A CHARACTER THAT IS A MEMBER OF CHRSET.
; REG IS SET POINTING TO THAT CHARACTER.

; EACH MEMBER OF CHRSET IS AN ADDRESSABLE QUANTITY.

HH: .IRP LS,<CHRSET>
CMPB @REG,LS
BEQ JJ
.ENDM
INC REG
BR HH

JJ: .ENDM

.MACRO SYNCLS CHAR,CLASS,?CC,?DD,?EE,?FF,?GG

; THIS MACRO DETERMINES THE SYNTACTIC CLASS OF AN
; OBJECT BEGINNING WITH CHAR. THE CLASS IS RETURNED
; IN CLASS AS FOLLOWS:
; CLASS = 0 (WORD) IF CHAR = (A,...,Z,-)
; 2 (NUMBER) IF CHAR = (0,...,9)
; 6 (END OF LINE) IF CHAR = CARRIAGE RETURN
; 4 (CHARACTER STRING) OTHERWISE

CC: CMPB CHAR,#'A
BLT EE
CMPB CHAR,#'Z
BGT DD
CLR CLASS
BR GG

DD: MOV #4,CLASS
BR GG

```

EE:  CMPB  CHAR,#'0
      BLT   FF
      CMPB  CHAR,#'9
      BGT   DD
      MOV   #2,CLASS
      BR    GG

FF:  CMPB  CHAR,#'-
      BEQ   CC
      CMPB  CHAR,#15
      BNE   DD
      MOV   #6,CLASS

GG:

```

.ENDM

CZPLA0 PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 46
COMMAND PROCESSOR INITIATING ROUTINE

SEQ 0121

.SBTTL COMMAND PROCESSOR INITIATING ROUTINE

```

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

```

COMAND: ;**ENTRY POINT**

```

: THIS ROUTINE CAUSES THE COMMAND SPECIFIED BY INPLIN TO BE
: PROCESSED, AS DESCRIBED BELOW. INPLIN IS A STRING OF ASCII
: CHARACTERS ENDED BY A CARRIAGE RETURN CODE.

```

- INPLIN IS LEXICALLY SCANNED USING THE LXSCAN ROUTINE.
- THE 1ST OBJECT IS CONVERTED TO A PROCESSING ROUTINE ADDRESS USING THE KEYWD ROUTINE & THE KEYWORD TABLE KWTABL SUPPLIED BY THE CALLING PROGRAM.
- THE PROCESSING ROUTINE IS CALLED WITH THE OUTPUT FROM LXSCAN ON THE STACK STARTING AT 2(SP). (THE RETURN ADDRESS OF THIS CALL OCCUPIES THE TOP WORD OF THE STACK.)
- THE LXSCAN OUTPUT IS REMOVED FROM THE STACK.

```

: IF LXSCAN OR KEYWD OR THE PROCESSING ROUTINE RETURN AN ERROR
: CONDITION, THEN C = 1; OTHERWISE, C = 0.

```

```

MOV  R5,-(SP)          ;SAVE PAR LIST POINTER.
MOV  SP,CMMARK         ;SAVE STACK POINTER.
LXSCAN INPLIN(R5)      ;LEXICALLY SCAN INPLIN.
MOV  INPLIN(R5),R1
JSR  R1,LXSCAN
BCS  CMRET             ;IF ERROR, RETURN WITH C=1.
TST  @SP              ;HAVE WE ANY OBJECTS?
BGT  1$               ;NO, IGNORE BLANK COMMAND.
MOV  #-1,ESCFLG
JMP  CMRET
1$:  MOV  CMMARK,R0     ;YES, DETERMINE ADDRESS
      SUB  #6,R0       ;OF FIRST OBJECT.
      MOV  @CMMARK,R5   ;RESTORE PAR LIST POINTER.
      KEYWD R0,KWTABL(R5) ;GET ADDR OF COMMAND PROCESSOR.
      MOV  R0,-(SP)
      MOV  KWTABL(R5),-(SP)
      JSR  PC,KEYWD
      BCS  CMRET
      ;IF INVALID COMMAND, MAKE
      ;ERROR RETURN.
      MOV  #NULPAR,R5  ;LOAD NULL PAR LIST ADDRESS.

```

```

3071
3072
3073 033454      000002
      (3)      000004
      (3)
      (1)
      (1) 033454
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091 033454 010546
3092 033456 010667 000076
3093 033462
      (1) 033462 016501 000002
      (1) 033466 004167 000310
3094 033472 103426
3095 033474 005716
3096 033476 003005
3097 033500 012767 177777 164162
3098 033506 000167 000036
3099 033512 016700 000042
3100 033516 162700 000006
3101 033522 017705 000032
3102 033526
      (1) 033526 010046
      (1) 033530 016546 000004
      (1) 033534 004767 000024
3103 033540 103403
3104
3105 033542 012705 033562

```

```

3106 033546 004736          JSR      PC,@(SP)+      L 7      ;PROCESS COMMAND & POP ADDR.
3107 033550 016706 000004 CMRET: MOV      CMMARK,SP      ;RESTORE STACK TO ENTRY STATUS.
3108 033554 030026          BIT      R0,(SP)+      ;THROW AWAY SAVED R5 (WITHOUT
3109                                ; AFFECTING C BIT).
3110                                RETURN      PC      ;RETURN TO CALLING PROGRAM.
(1) 033556 000207          RTS
3111
3112                                ; DATA AREAS.
3113 033560 033333          CMMARK: .WORD 33333      ;STORAGE FOR STACK PTR ON ENTRY.
3114 033562 000000          NULPAR: .WORD 0          ;PAR LIST CONTAINING NO PARS.
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 47
PCLEXR.P11 17-MAR-78 11:27      KEYWORD PROCESSING ROUTINE

```

SEQ 0122

```

3116                                .SBTTL KEYWORD PROCESSING ROUTINE
3117
3118                                ; SUBROUTINE TO DETERMINE THE ADDRESS OF THE PROCESSING
3119                                ; ROUTINE ASSOCIATED WITH THE KEYWORD REPRESENTED BY
3120                                ; THE SYNTACTIC OBJECT POINTED TO BY THE ARGUMENT
3121                                ; SRC. CONVERSION FROM KEYWORD TO ROUTINE ADDRESS
3122                                ; IS DONE AS DEFINED IN THE KEYWORD TABLE
3123                                ; POINTED TO BY THE ARGUMENT TABLAD.
3124
3125                                ; ON ENTRY, THE TOP OF THE STACK IS AS FOLLOWS:
3126                                ; (SP): RETURN ADDRESS
3127                                ; 2(SP): TABLAD
3128                                ; 4(SP): SRC
3129
3130                                ; CALLING INSTRUCTION: JSR PC,KEYWD
3131
3132                                ; ON RETURN, THE TOP OF THE STACK IS AS FOLLOWS:
3133                                ; (SP): ROUTINE ADDRESS, IF KEYWORD IN TABLE; 0 IF NOT.
3134
3135                                ; IF THE KEYWORD IS IN THE TABLE, C=0 ON RETURN. IF NOT, C=1.
3136
3137                                ; STACK POINTER OFFSETS
3138                                ;ROUTAD = 0
3139                                ;ROUTINE ADDR FOR
3140                                ; CURRENT TABLE
3141                                ; ELEMENT.
3142                                ;ADDR OF INPUT WORD
3143                                ;#CHAR IN INPUT WORD
3144                                ;SUBROUTINE RETURN ADDR
3145                                ;ADDR OF KEYWORD TABLE
3146                                ;ADDR OF INPUT OBJECT
3147                                ;RESULT RETURNED
3148                                ;**ENTRY POINT**
3149                                ;SAVE REGISTERS.
(1) 033564 004567 000160          JSR      R5,REGSAV
3150 033570 162706 000006          SUB      #6,SP      ;ALLOCATE STACK SPACE
3151 033574 016602 000026          MOV      SRC(SP),R2    ;GET ADDR OF INPUT
3152                                ; OBJECT
3153 033600 005722          TST      (R2)+      ;IF OBJECT NOT A WORD
3154 033602 001051          BNE      NOTHER      ; THEN EXIT: NOT FOUND
3155 033604 012205          MOV      (R2)+,R5      ;GET # CHAR IN OBJECT.
3156 033606 010566 000004          MOV      R5,LENINP(SP) ;STORE ON STACK.
3157 033612 011204          MOV      @R2,R4      ;GET ADDR OF INPUT WORD.
3158 033614 010466 000002          MOV      R4,ADRINP(SP) ;STORE ON STACK.
3159 033620 016603 000024          MOV      TABLAD(SP),R3 ;GET ADDR. OF KEYWORD
3160                                ; TABLE.
3161 033624 012316          GTLENS: MOV      (R3)+,@SP    ;SAVE ROUTINE ADDR OF
3162                                ; 1ST ELEMENT.
3163 033626 112302          MOV      (R3)+,R2      ;GET MINIMUM LENGTH
3164 033630 042702 177400          BIC      #177400,R2    ; OF TABLE WORD.

```

```

3165 033634 112301          MOVB (R3)+,R1      M 7
3166 033636 042701 177400    BIC  #177400,R1      :GET FULL LENGTH.
3167 033642 001431          BEQ  NOTHER          :IF 0, THEN NO MORE
3168                                :TABLE TO SEARCH
3169 033644 122423          NXTCH: CMPB (R4)+,(R3)+ :COMPARE INPUT CHAR
3170                                :WITH TABLE CHAR.

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 47-1
PCLEXR.P11 17-MAR-78 11:27      KEYWORD PROCESSING ROUTINE                      SEQ 0123

3171 033646 002427          BLT  NOTHER          :IF<, THEN NO MATCH
3172                                :EXISTS.
3173 033650 003005          BGT  NXTWD           :IF>, THEN TRY NEXT
3174                                :TABLE WORD.
3175 033652 005305          DEC  R5              :IF INPUT STRING
3176                                :EXHAUSTED,
3177 033654 001413          BEQ  THFND           :WE MAY HAVE FOUND
3178                                :MATCH.
3179 033656 005301          DEC  R1              :IF MORE CHAR IN TABLE WORD
3180 033660 001371          BNE  NXTCH           :TO TEST, GO & TEST THEM.
3181 033662 005201          INC  R1
3182 033664 060103          NXTWD: ADD R1,R3      :GET ADDR OF NEXT
3183 033666 042703 000001    BIC  #1,R3        :TABLE ENTRY.
3184 033672 016604 000002    MOV  ADRINP(SP),R4 :POINT TO BEGINNING
3185                                :OF INPUT WORD.
3186 033676 016605 000004    MOV  LENINP(SP),R5 :GET LENGTH OF INPUT WORD.
3187 033702 000750          BR   GTLENS         :GET LENGTHS OF TABLE WORD.
3188
3189 033704 026602 000004    THFND: CMP LENINP(SP),R2 :IF LEN(INP.WD) < MIN LEN (TABLE
3190 033710 002406          BLT  NOTHER          :WORD), WORD IS NOT IN TABLE.
3191 033712 011666 000026    MOV  (SP),RESULT(SP) :SAVE ROUTINE ADDR. OF
3192                                :MATCH.
3193 033716 062706 000006    ADD  #6,SP          :FREE LOCAL STACK SPACE.
3194 033722 000241          CLC                   :CLEAR CARRY BIT.
3195 033724 000405          BR   KWEXIT
3196
3197                                : WORD IS NOT IN TABLE. SET RESULT TO 0 & SET Z BIT ON.
3198 033726 005066 000026    NOTHER: CLR RESULT(SP) :CLEAR RESULT.
3199 033732 062706 000006    ADD  #6,SP          :FREE LOCAL STACK SPACE.
3200 033736 000261          SEC                   :SET CARRY BIT.
3201 033740          KWEXIT: REGRES              :RESTORE REGISTERS.
3202 033744 012616          JSR  R5,REGRES
3203 033746 000207          MOV  (SP)+,@SP
3204                                :POP AN ARGUMENT.
3205                                :RETURN.
3206                                :
3207                                :
3208                                :
3209 033750          REGSAV: MOV R4,-(SP)
3210 033750 010446          MOV  R3,-(SP)
3211 033752 010346          MOV  R2,-(SP)
3212 033754 010246          MOV  R1,-(SP)
3213 033756 010146          MOV  R0,-(SP)
3214 033760 010046          JMP  @R5
3215 033762 000115
3216
3217                                :
3218                                : SUBROUTINE TO RESTORE R0-R5 FROM STACK
3219                                : THE CONDITION CODE BITS IN THE PS ARE DESTROYED,
3220                                : EXCEPT FOR THE CARRY BIT, WHICH IS PRESERVED.
3221
CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 48
PCLEXR.P11 17-MAR-78 11:27      REGISTER SAVE & RESTORE ROUTINES                      SEQ 0124

3205                                .SBITL REGISTER SAVE & RESTORE ROUTINES
3206
3207                                : SUBROUTINE TO SAVE R0 - R5 ON STACK
3208                                : CALLING SEQUENCE: JSR R5,REGSAV
3209                                :
3210                                :
3211                                :
3212                                :
3213                                :
3214                                :
3215                                :
3216                                :
3217                                :
3218                                :
3219                                :
3220                                :
3221                                :

```

```

; CALLING SEQUENCE: N 7
REGRES: JSR R5,REGRES ;**ENTRY POINT**
; THROW AWAY OLD R5 VALUE.
BIT R0,(SP)+
MOV (SP)+,R0
MOV (SP)+,R1
MOV (SP)+,R2
MOV (SP)+,R3
MOV (SP)+,R4
RTS R5

```

CZPLAAO PCL11 EXERCISER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 49
PCLEXR.P11 17-MAR-78 11:27 LEXICAL SCAN ROUTINE

SEQ 0125

.SBTTL LEXICAL SCAN ROUTINE

```

; PERFORM LEXICAL SCAN OF INPUT COMMAND IN
; BUFFER.

; THREE CLASSES OF SYNTACTIC OBJECTS ARE RECOGNIZED:
; 1. WORD: A STRING OF CHARACTERS BEGINNING WITH
; A LETTER & TERMINATED WITH A
; BLANK OR CARRIAGE RETURN.
; 2. NUMBER: A STRING OF OCTETS TERMINATED WITH A
; BLANK OR CARRIAGE RETURN, OR A STRING OF DIGITS
; TERMINATED WITH A DOT.
; 3. CHARACTER STRING: A STRING SURROUNDED BY 2 INSTANCES
; (1 ON EACH END) OF A SPECIAL CHARACTER.

; SYNTACTIC OBJECTS ARE SEPARATED BY 1 OR MORE BLANKS.
; THE COMMAND IS ENDED BY A CARRIAGE RETURN.

; THIS LEXICAL SCANNER DETERMINES THE LOCATIONS OF THE
; SYNTACTIC OBJECTS & DETERMINES THEIR CLASSES.
; NUMBERS ARE CONVERTED TO THEIR BINARY VALUES.

; AFTER THE LEXICAL SCAN IS PERFORMED, EACH
; SYNTACTIC OBJECT WILL BE REPRESENTED ON THE STACK
; AS A 3 WORD QUANTITY AS FOLLOWS:
; TOP WORD = SYNTACTIC CLASS:
; 0==>WORD
; 2==>NUMBER
; 4==>CHARACTER STRING
; 2ND WORD = LENGTH OF OBJECT IN CHARACTERS; NOT
; SIGNIFICANT FOR NUMBERS; DOES NOT
; INCLUDE SURROUNDING DELIMITERS FOR CHAR.
; STRINGS.
; 3RD WORD = VALUE OF NUMBER, OR POINTER TO 1ST
; LETTER OF KEYWORD OR 1ST SIGNIFICANT
; CHARACTER OF STRING (IE: NOT SUR-
; ROUNDING DELIMITER)

; AT THE END OF THE LEXICAL SCAN THE STACK WILL BE
; ARRANGED AS FOLLOWS (N = NO. OF SYNTACTIC OBJECTS):
;
; |-----|
; | N | <--- SP
; |-----|
; | NTH OBJECT |
; |-----|
; | : |
; |-----|
; | 1ST OBJECT |
; |-----|
;
; WHERE EACH OBJECT IS REPRESENTED AS ABOVE.

```

3283
3284
3285
3286
3287
CZPLA0 PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACV11

8 8
: 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.
30A(1052) 28-APR-78 13:50 PAGE 49-1
LEXICAL SCAN ROUTINE

SEQ 0126

3288
3289
3290
3291
3292
3293
3294 034002
3295 034002 005002
3296 034004 012605
3297 034006
(1) 034006 122527 000040
(1) 034012 001775
(1) 034014 005305
3298 034016
(1) 034016 121527 000101
(1) 034022 002410
(1) 034024 121527 000132
(1) 034030 003002
(1) 034032 005004
(1) 034034 000424
(1) 034036 012704 000004
(1) 034042 000421
(1) 034044 121527 000060
(1) 034050 002406
(1) 034052 121527 000071
(1) 034056 003367
(1) 034060 012704 000002
(1) 034064 000410
(1) 034066 121527 000055
(1) 034072 001757
(1) 034074 121527 000015
(1) 034100 001356
(1) 034102 012704 000006
3299 034106 000174 034112
3300
3301 034112 034122
3302 034114 034162
3303 034116 034326
3304 034120 034366
3305
3306 034122 010503
3307 034124
(2) 034124 121527 000040
(2) 034130 001405
(2) 034132 121527 000015
(2) 034136 001402
(1) 034140 005205
(1) 034142 000770
3308 034144 010346
3309 034146 160503
3310 034150 005403
3311 034152 010346
3312 034154 010446
3313
3314 034156 005202

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

LXSCAN:

CLR R2
MOV (SP)+,R5
NXTL: SPAN R5,<# '>
64\$: CMPB (R5)+,#'
BEQ 64\$
DEC R5
SYNCLS @R5,R4
CMPB @R5,#'A
BLT 67\$
CMPB @R5,#'Z
BGT 66\$
65\$: CLR R4
BR 69\$
66\$: MOV #4,R4
BR 69\$
67\$: CMPB @R5,#'0
BLT 68\$
CMPB @R5,#'9
BGT 66\$
MOV #2,R4
BR 69\$
68\$: CMPB @R5,#'-
BEQ 65\$
CMPB @R5,#15
BNE 66\$
MOV #6,R4
JMP @SJT(R4)

:::ENTRY POINT:::

:CLEAR #OBJECTS.
:GET ADDR. OF COMMAND BUFFER.
:R5 = ADDR (1ST NONBLANK CHAR)

:DETERMINE SYNTACTIC CLASS.

:PROCESS OBJECT

SJT: .WORD SCWORD
.WORD SCNO
.WORD SCCHAR
.WORD SCEOL

:WORD PROCESSOR.
:NUMBER PROCESSOR.
:CHARACTER STRING PROCESSOR.
:END OF LINE PROCESSOR.

SCWORD: MOV R5,R3
BREAK R5,<# '>,<#15>>

:R3 = START ADDR OF OBJECT
:R5 = ADDR (NEXT BLANK OR CR)

CMPB @R5,#'
BEQ 65\$
CMPB @R5,#15
BEQ 65\$
INC R5
BR 64\$
MOV R3,-(SP)
SUB R5,R3
PLAC: NEG R3
MOV R3,-(SP)
MOV R4,-(SP)

:PUSH ADDR OF OBJECT ONTO STACK.
:R3 = (-LENGTH OF OBJECT)
:NEGATE TO GET LENGTH.
:PUSH LENGTH ONTO STACK.
:PUSH SYNTACTIC CLASS ONTO
:STACK
:INCREMENT

LXINCN: INC R2

3315 034160 000712
CZPLAAO PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 49-2
LEXICAL SCAN ROUTINE

C 8

:GO TO SCAN NEXT ELEMENT

SEQ 0127

3316
3317 034162 005000
3318 034164 010504
3319 034166 121527 000060
3320 034172 002415
3321 034174 121527 000071
3322 034200 003012
3323 034202
(2) 034202 006300
(2) 034204 010003
(2) 034206 006300
(2) 034210 006300
(2) 034212 060300
3324 034214 112503
3325 034216 142703 000060
3326 034222 060300
3327 034224 000760
3328
3329 034226 121527 000040
3330 034232 001413
3331 034234 121527 000015
3332 034240 001410
3333 034242 122527 000056
3334 034246 001052
3335 034250 010046
3336 034252 005046
3337 034254 012746 000002
3338
3339 034260 000736
3340
3341
3342 034262 005000
3343 034264 121427 000060
3344 034270 002413
3345 034272 121427 000067
3346 034276 003010
3347 034300
(2) 034300 006300
(2) 034302 006300
(2) 034304 006300
3348 034306 112403
3349 034310 142703 000060
3350 034314 060300
3351 034316 000762
3352
3353 034320 020405
3354 034322 001024
3355 034324 000751
3356
3357 034326 112500
3358 034330 010503
3359
3360 034332
(2) 034332 121500
(2) 034334 001405
(2) 034336 121527 000015

SCNO: CLR R0
MOV R5,R4
SNXTDG: CMPB @R5,#'0
BLT SNTDIG
CMPB @R5,#'9
BGT SNTDIG
MULT 10.,R0,R3
ASL R0
MOV R0,R3
ASL R0
ASL R0
ADD R3,R0
MOVB (R5)+,R3
BICB #60,R3
ADD R3,R0
BR SNXTDG

SNTDIG: CMPB @R5,#'
BEQ TRYOCT
CMPB @R5,#15
BEQ TRYOCT
CMPB (R5)+,#'.
BNE LXERR
PUSHNO: MOV R0,-(SP)
CLR -(SP)
MOV #2,-(SP)

BR LXINCN

TRYOCT: CLR R0
ONXTDG: CMPB @R4,#'0
BLT ODELIM
CMPB @R4,#'7
BGT ODELIM
MULT 8.,R0
ASL R0
ASL R0
ASL R0
MOVB (R4)+,R3
BICB #60,R3
ADD R3,R0
BR ONXTDG

ODELIM: CMP R4,R5
BNE LXERR
BR PUSHNO

SCCHAR: MOVB (R5)+,R0
MOV R5,R3

BREAK R5,<R0,#15>
CMPB @R5,R0
BEQ 655
CMPB @R5,#15

:CLEAR ACCUMULATED NO.
:SAVE POINTER TO 1ST DIGIT.
:IF CHAR <'0'
: THEN TREAT AS DELIMITER.
:IF CHAR >'9'
: THEN TREAT AS DELIMITER.
:MULTIPLY PREVIOUS DIGITS BY 10.

:CLEAR TOP BITS OF ASCII CODE.
:ADD DIGIT.
:GET NEXT DIGIT.

:IF DELIMITER = SPACE, TRY
: CONVERTING OCTAL NO.
:IF DELIMITER = CR, TRY
: CONVERTING OCTAL NO.
:IF DELIMITER IS NOT DOT,
: THEN SIGNAL LXSCAN ERROR.
:PUT CONVERTED NO. ON STACK.
:SET OBJECT LENGTH TO ZERO.
:SET OBJECT CLASS TO 2
: (NUMBER).
:INCREMENT OBJ. COUNT &
: SCAN NEXT OBJ.

:CLEAR ACCUMULATED NO.
:IF CHAR <'0'
: THEN TREAT AS DELIMITER.
:IF CHAR >'7'
: THEN TREAT AS DELIMITER.
:MULTIPLY PREVIOUS DIGITS BY 8.

:GET CHARACTER.
:CLEAR TOP BITS OF ASCII CODE.
:ADD DIGIT.
:GET NEXT DIGIT.

:IF NOT AT END OF NO.(DUE TO
: '8' OR '9') , LXSCAN ERROR.
:GENERATE OBJECT FOR NUMBER.

:R0 = DELIMITER OF STRING.
:R3 = ADDR (1ST CHAR OF
: STRING ITSELF)
:R5 = ADDR (NEXT DELIM OR CR).

CZPLAAO PCL11 EXERCISER V-02
PCLEXR.P11 17-MAR-78 11:27

MACY11 30A(1052) 28-APR-78 13:50 PAGE 49-3
LEXICAL SCAN ROUTINE

SEQ 0128

```

(2) 034342 001402
(1) 034344 005205
(1) 034346 000771
3361 034350 121527 000015
3362 034354 001407
3363 034356 010346
3364
3365 034360 160503
3366
3367 034362 005205
3368
3369 034364 000671
3370
3371 034366 010246
3372 034370 000241
3373 034372 000111
3374
3375
3376 034374 000261
3377 034376 005302
3378 034400 002774
3379 034402 062706 000006
3380 034406 000772
3381
3382
3383
000001
CZPLAAO PCL11 EXERC!SER V-02 MACY11 30A(1052) 28-APR-78 13:50 PAGE 50
PCLEXR.P11 17-MAR-78 11:27 CROSS REFERENCE TABLE -- USER SYMBOLS

```

```

BEQ 658
INC R5
BR 648
CMPB @R5,#15
BEQ LXERR
MOV R3,-(SP)

SUB R5,R3

INC R5

BR PLAC

SCEOL: MOV R2,-(SP)
CLC
LXIT: JMP @R1

; PROCESS LEXICAL SCAN ERROR .
LXERR: SEC
DEC R2
BLT LXIT
ADD #6,SP
BR LXERR

```

```

; WAS CR FOUND BEFORE DELIMITER
; YES, PROCESS ERROR
; PUSH ADDR. OF STRING ONTO
; STACK.
; R3 = (-LENGTH OF STRING EXCL.
; DELIMITER)
; R5 = ADDR (CHAR AFTER CLOSING
; DELIMITER).
; FINISH OFF PROCESSING STRING.

```

```

; PUSH NO. OF OBJECTS ONTO STACK
; INDICATE NO ERROR .
; RETURN TO CALLING PROGRAMME .

```

```

; INDICATE ERROR OCCURRED .
; CLEAN UP STACK & EXIT .
; ..
; ..
; ..

```

.END

SEQ 0129

```

ADQLD 006740 503 1212#
ADRINP= 000002 3141# 3158* 3184
ADSILO 017710 566 613 2408#
AQAREA 016352 2344#
AQEND = 016452 2344#
AQO 016336 851* 852* 853* 986* 991* 992* 1223 1232 1344 1348 2344#
AOSIZE= 000040 308# 2344
ASEXT 006304 1092 1097#
A.BKSL 016074 2177 2309#
A.CR 015764 2257 2269#
A.LF 015760 2258 2268#
A.S 016072 2157 2308#
BKELEM 030200 1231* 1232 2200* 2201 2243* 2245 2461#
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= 016314 2196 2325#

```

CBTA	032656	2733#	2854				E 0		
CBUFPT	016314	2081#	2158#	2171	2176#	2196	2198#	2199#	2326#
CDDMG	033014	1749	1758	2823#					
CHAR =	000002	2187#	2190						
CLEAV	004106	570	639#						
CLKEXT	012322	1703	1707	1711	1714#				
CLKINT	012230	457	1700#						
CMCHTB	015754	2063	2266#						
CMDBUF	016110	1113	2158	2171	2324#				
CMDENT	017553	424#	495	1111#	2082#	2159#	2351#		
CMCRET	006430	1118	1120#						
CMDRTN	006406	1114	1117#						
CMDTBL	030210	1113	2470#						
CMMARK	033560	3092#	3099	3101	3107	3113#			
CMISC	015020	2065	2069#						
CMRET	033550	3094	3098	3103	3107#				
COMAND	033454	1113	3073#						
COMENT	006322	497	1110#						
CPADD	004536	766#	2470						
CPALP	004552	771#	786						
CPARTN	004634	788#	790						
CPASS	005704	1023#	2473						
CZPLAAO PCL11	EXERCISER V-02	MACY11	30A(1052)	28-APR-78	13:50	PAGE 50-1			
PCLEXR.P11	17-MAR-78 11:27		CROSS REFERENCE TABLE	--	USER SYMBOLS				

SEQ 0130

CPCLR	004764	834#	2476		
CPCLRC	005014	843#	850		
CPCLRT	005056	854#	856		
CPCNRT	005212	888	890#	892	
CPCNT	005126	881#	2479		
CPDEL	004644	794#	2482		
CPDLP	004660	799#	819		
CPDRTN	004754	821#	823		
CPERR	006612	1177#	2485		
CPERTN	006730	1188	1194#	1197	
CPGO	005522	982#	2488		
CPINIT	005222	905#	2491		
CPINRT	005460	947#	949		
CPMAST	004130	649#	2494		
CPMLP	003520	583#	599		
CPMOK	004144	651	654#		
CPMRET	004212	653	663#		
CPRANG	004400	723#	2497		
CPRFIL	004502	747#	751		
CPRIB	004306	694#	2500		
CPROK	004322	696	699#		
CPRRET	004376	698	711#		
CPRRTN	004526	753#	756		
CPSEC	004214	669#	2503		
CPSEX	004076	633#	636	642	
CPSILO	003420	565#	2506		
CPSLV	004034	621	627#		
CPSOK	004230	671	674#		
CPSRET	004300	673	684#	688	
CPSRTN	005116	868#	870		
CPSTAT	005066	862#	2509		
CPSUM	005470	956#	2512		
CPSURT	005512	961#	963		
CRRET	015100	2080	2085#		
CR. =	000015	333#	2081	2269	
CTLCMG	016324	2126	2330#		
CTLOMG	016316	2094	2328#		
CTLUMG	016321	2135	2329#		

ERRX	010406	1413	1417	1425	1429	1433	1435	1441	1449	1451	1453#		
ERRO	017702	1190#	1829	1845	1861#	1883	1896	1912#	2401#				
ERR1	017704	1191#	1834#	1848	1856	1858#	1860#	1899	1907	1909#	1911#	2402#	
ERR2	017706	1192#	1835#	1841	1851	1855#	1893	1902	1906#	2403#			
ERTBCL	005314	919#	923										
ERTBL	024640	918	1961	2437#	2447								
ERTBLO=	024632	1990	2028	2447#									
ERTCD	005336	920	924#										
ERTMSG	031064	537	2538#										
ERTRAP	003340	421	536#										
ESCLG	017670	468#	630#	640#	662#	683#	710#	752#	787#	820#	839#	908#	1112# 1117
		1187#	2396#	3097#									
EXCNT	031162	886	2540#										
EXRST	031140	997	2539#										
FIRST	017622	581#	600	2373#									
FRELEM	030176	1348	1349	1359	2054	2055#	2056	2062#	2066	2069	2216	2217	2460#
GTLENS	033624	3161#	3187										
HDRFLG	017630	1193#	1823	1825#	1833#	1877	1879#	2376#					
HOURS	017560	914#	990#	1713#	1782	2356#							
INPLIN=	000002	3073#	3093										
INTENB=	000100	345#											
INTMP	030206	2227#	2228	2464#									
ITEM =	000002	2955#	2962	2975#	2983#	2993#							
CZPLA0	PCL11 EXERCISER V-02		MACY11	30A(1052)	28-APR-78	13:50	PAGE 50-3						
PCLEXP	P11 17-MAR-78 11:27						CROSS REFERENCE TABLE -- USER SYMBOLS						
KEYWD	033564	656	676	701	3102	3148#							
KWDFRT	004302	657	677	687#	702								
KWEXIT	033740	3195	3201#										
KWFLG	017562	454#	456#	461#	887	998	1778	2148	2357#				
KWTABL=	000004	3073#	3102										
LCS	016070	455	889#	1000#	1700#	2150#	2303#						
LENINP=	000004	3142#	3156#	3186	3189								
LF. =	000012	332#	2268										
LXERR	034374	3334	3354	3362	3376#	3380							
LXINCN	034156	3314#	3339										
LXIT	034372	3373#	3378										
LXSCAN	034002	3093	3294#										
MDNER	030472	1139	2529#										
MESSAG=	000002	2241#	2242										
MINUTS	017556	912#	988#	1709#	1710	1712#	1786	2354#					
MLI	007336	1279	1295#	1672									
MLSD	017642	939	2385#										
MSCHNG	030544	1159	2530#										
MSGLSO	017644	939#	1262#	1264#	1265	1273	1277	1283#	2386#				
MSTMG1	031207	622	2541#										
MSTMG2	031245	623	2542#										
MSTMG3	031306	624	2543#										
MSTMG4	031354	625	2544#										
MSTMG5	031422	629	2545#										
MTQMSG	030745	1239	2535#										
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# 2438
NEXT	017624	582#	586	596#	600	2374#							
NMINT	010774	1395	1526#										
NOERMG	032141	1186	2562#										
NONMG	032476	1838	1888	2575#									
NOTHER	033726	3154	3167	3171	3190	3198#							
NOUMST=	000004	1379#	1393	1526									
NOFULL	033366	2961	2971#										
NONOWP	033360	2965	2967#										
NORET	033364	2969#	2972										

SEQ 0132

NULPAR	033562	3105	3114#
NXTCM	033644	3169#	3180
NXTEL	034006	3297#	3315
NXTWD	033664	3173	3182#
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	017672	575*	589*
OCTJSP	032614	451	483
CZPLAAO PCL11	EXERCISER V-02		MACY11
PCLEXR.P11	17-MAR-78 11:27		

603*	604	609	612	617	618	2397#	
1744	1846	1849	1897	1900	1939	1943	2702#
30A(1052) 28-APR-78 13:50 PAGE 50-4							
CROSS REFERENCE TABLE -- USER SYMBOLS							

SEQ 0133

[illegible]

30A(1052) 28-APR-78 13:50 PAGE 50-7
CROSS REFERENCE TABLE -- USER SYMBOLS

SEO 0136

3329#

1141* 1440* 2371*

386 420 485 543

2556#

2557#

1734 1737 1741* 1742 1745* 1747 1754* 1756 1764* 1799* 1802*

1800 1803 1807#

1004

942* 1143* 1805* 2152* 2364*

959* 1144* 1967* 2153* 2362*

1931 1933 1957* 1961* 1964* 23664

482 2363M

502 505 406 435#

392	393	608	633M	
733	736	738	760	755M

133 130 130 140 133

776 789#

801 8224

804 822#

SYNXR	006316	1028 1100# 3144#	1033 3159 MACY11	1038	1041	1049	L 1054 8	1057	1065	1070	1073	1081	1086	1089
TABLAD=	000024													
CZPLAAO PCL11 EXERCISER V-02														
PCLEXR.P11	17-MAR-78 11:27													
TCBFPT	030202	2462#												
TCBIN	030204	2463#												
TCR	016016	388*	867*	910*	1343*	1359*	1360*	1382*	1444	1453*	1462*	1487*	1513*	1528*
TEMP	017676	1981	2280#											
TEMP1	017700	1981*	1982*	1983*	1999*	2000*	2001*	2002*	200*	2019*	2020*	2021*	2038*	2039*
TEREV	017564	2040*	2041*	2042*	2399#									
TERHDR	032201	1985*	1996*	1997*	1998*	2001	2023*	2034*	2035*	2036*	2037*	2040	2400#	
TEROS	013100	525	943*	1189*	1831*	2154*	2358#							
TERTBL	025062	1826	2563#											
TERTN	013436	527	1819#											
THFND	033704	925	1180	1192	2002	2452#								
THLTFL	017634	1822	1828	1841	1859	1862#								
THUMST	031474	3177	3189#											
THUSCN	031522	2107*	2117*	2213	2225*	2378#								
TIAREA	016466	1769	2546#											
TICKS	017554	1772	2547#											
TIEND =	016536	2344#												
TIQ	016452	913*	989*	1701*	1702	1704*	1794	2352#						
TISIZE=	000024	2344#												
TMLIN1	031654	433*	492	2054	2228	2344#								
TMMR	016030	306#	2344											
TMMRH	016032	1781	2551#											
		398*	614*	2285#										
		400*	607*	620	626*	631*	639*	658	661*	678	682*	1393	1526*	1767
		1770	2286#											
TOAREA	016552	2344#												
TOEND =	017552	2344#												
TOQ	016536	433*	498	1729	1820	1874	1927	2092*	2093*	2157	2177	2201	2216	2245
		2257	2258	2259	2344#									
TORET	015560	2212	2215	2218#										
TOSIZE=	000400	307#	2344											
TRAP4	002702	465	479#											
TRELIN	032275	1853	2565#											
TRERC	032331	1850	2568#											
TRHLIN	032225	1827	2564#											
TRNKF	017654	471*	1261*	1267*	1271*	1470	2390#							
TRPDMG	031020	484	2536#											
TRP4AD	031053	481	2537#											

PCLEXR.P11		17-MAR-78 11:27		CROSS REFERENCE TABLE -- USER SYMBOLS												SEQ 0138			
TTVCTR= 000060		296#	2302																
TTVECT 016066		437*	442	2302#															
TTXBUF 016064		2217*	2301#																
TTXCSR 016062		2211	2300#																
TXBSY = 000020		1376#	1396	1446	1460														
TXDEV 016100		387	1090*	1093*	2318#														
TXDGEN 007566		1365	1367#																
TXER = 100000		1375#	1384																
TXMEV 017606		507	940*	1226*	1284*	1342*	1399*	1454*	1461*	1486*	1512*	2367#							
TXMIT 007402		509	1339#																
TXML 017656		1276*	1346	2391#															
TXMST 007400		703	709*	915*	1337#	1360	1773												
TXMSV 017610		2368#																	
TXMVEC 016052		403*	435*	438	2295#														
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 016102		403	1074*	1094*	2319#														
TXVECT= 000170		301#																	
X = 000040		2412#	2421#																
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 017620		473*	522	1161*	1527*	2372#													
XSPEC 006536		524	1157#																
Y = 000031		2436#	2444#																
ZERO 016076		2259	2310#																
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#	2168#	2187#	2210#	2241#	2256#	2955#					
		2975#	3073#	3323#	3347#														
ZX2 = 000000		578#	1997#	1998#	2000#	2036#	2037#	2039#	3323#	3347#									
\$BLK = 000000		2703#	2709#	2715#	2721#														
\$BLKS = 002000		2827#																	
\$SGN = 000000		2703#	2709#	2715#	2721#														
\$SGNS = 000000		2827#																	
\$SUP = 000000		2703#	2709#	2715#	2721#														
\$TR 002550		453	460#																
. = 034410		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	2324#	2325	2344#	2355#	2408#	2422	2427#					
		2432#	2452#	2455#	2827														
CZPLAAO PCL11 EXERCISER V-02		MACY11 30A(1052) 28-APR-78 13:50 PAGE 51																	
PCLEXR.P11		17-MAR-78 11:27		CROSS REFERENCE TABLE -- MACRO NAMES												SEQ 0139			

								N	8							
CBTAS	2801#	2827														
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	3102											
LXSCAN	69#	3093														
MULP	217#	1279	1672													
MULT	151#	578	1997	1998	2000	2036	2037	2039	3323	3347						
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	2168	2187	2210	2241	2256	
	2955	2975	3073													
REGRES	280#	542	1227	1401	1455	1463	1488	1514	1529	1563	1588	1629	1643	1681	1692	
	2229	3201														
REGSAV	274#	536	1213	1381	1557	1573	2226	3149								
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	2160	2178	2202	2218	2260	2777	2863	2900	2933	2946	2969	
	2990	3110														
SPAN	3002#	3297														
STCVT	2655#	2703	2709	2715	2721											
SYNCLS	3035#	3298														

. ABS. 034410 000

ERRORS DETECTED: 0

PCLEXR,PCLEXR/CR/NL:TTM<PCLEXR

RUN-TIME: 11 16 1 SECONDS

RUN-TIME RATIO: 496/29=16.7

CORE USED: 10K (19 PAGES)