

DX11-B

RESPONDER
CZDXIBO

AH-8777B-MC

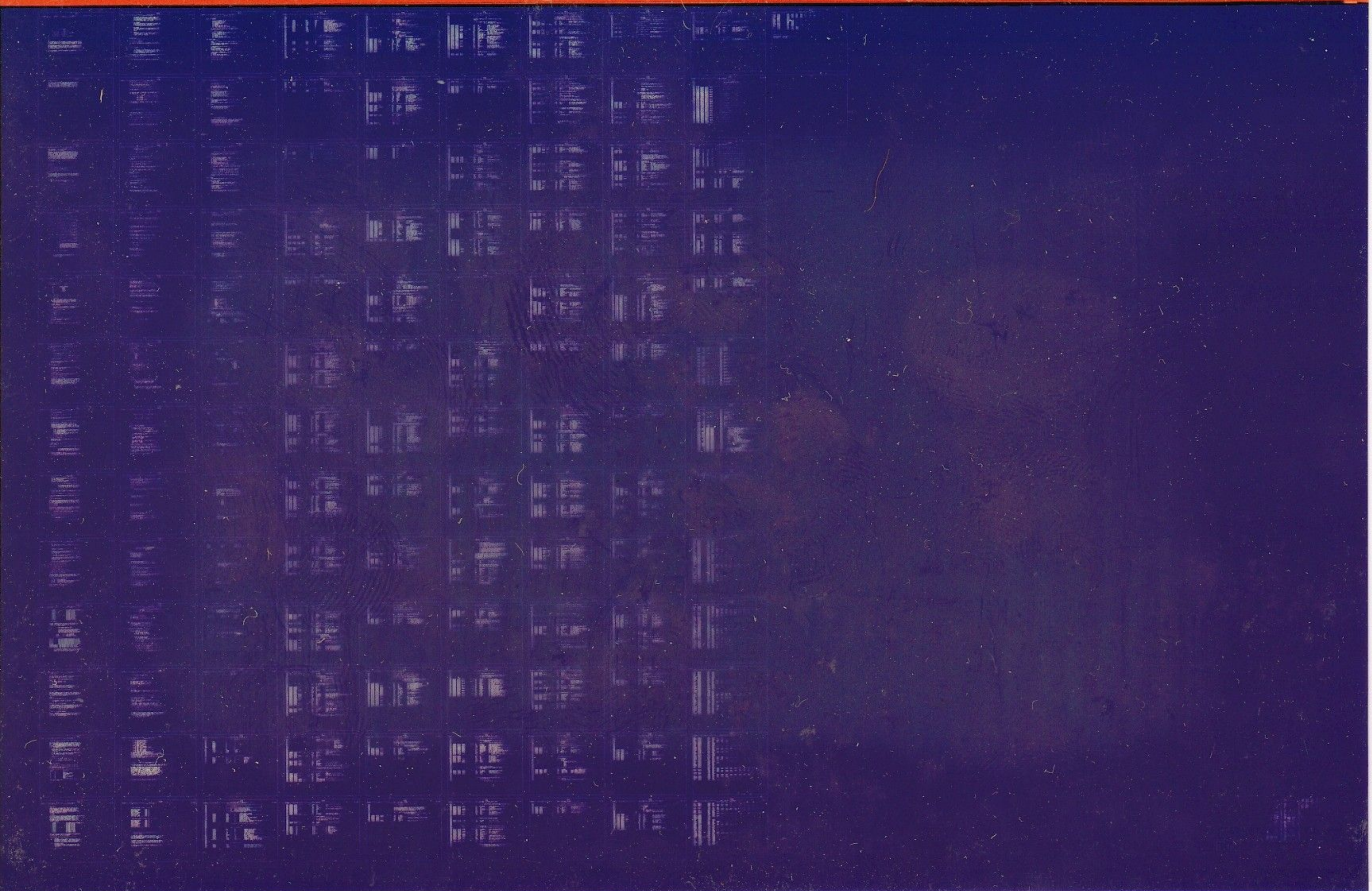
APR 1978

COPYRIGHT ©72-78

digital

FICHE 1 OF 1

MADE IN USA



B01

EOF10FFPCBSEQ

0001GJ00

780330

PDP10 411

0#HDRICZDXIBSEQ

00010000

780330

PDP10 480 0002

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER

MAY11 30A(1052) 01-FEB-78 09:46

PAGE 1

SEQ 0001

CZDXIB.DOC 30-JAN-78 14:26

OPERATING PROCEDURE

.REM :

IDENTIFICATION

PRODUCT CODE: AC-8776B MC
PRODUCT NAME: CZDXIB0 DX11-B RESPONDER
RELEASE DATE: MARCH 1978
MAINTAINER: DIAGNOSTIC ENGINEERING

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S DLT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1972, 1978 BY DIGITAL EQUIPMENT CORPORATION

THIS SYSTEM TEST PROGRAM EXERCISES THE INTERFACE BETWEEN THE PDP-11 AND AN IBM 360/370 COMMUNICATING VIA THE DX11-B CONTROL UNIT. THE PROGRAM EMULATES AN IBM CRT (2260) AND ITS CONTROL UNIT (2848) COMMUNICATING OVER EITHER A MULTIPLEXER OR SELECTOR CHANNEL. THE 360/370 EXERCISES THE INTERFACE BY RUNNING STANDARD IBM DIAGNOSTICS DESIGNED TO TEST THE 2260/2848; FRIEND OF THE

2848 RESPONDER. UP TO EIGHT 2260'S MAY BE EMULATED SIMULTANEOUSLY BY THE PROGRAM.

BASICALLY THE SYSTEM TEST PROGRAM COLLECTS THE TEST PARAMETERS NEEDED VIA A QUESTION AND RESPONSE TUTORIAL METHOD. VALIDATES THE PARAMETERS AND THEN INITIALIZES THE SYSTEM. AFTER THE SYSTEM HAS BEEN INITIALIZED THE OPERATOR IS THEN REQUIRED TO START THE TEST BY TYPING "R" AND THEN THE 360/370 BEGINS TO TEST A 2260/2848. THE SYSTEM TEST PROGRAM ONLY RECOGNIZES BASIC ERRORS; SUCH AS, PARITY ERROR, ILLEGAL DEVICE ADDRESS, ETC.. WITH THE 360 DIAGNOSTIC TESTING FOR MORE DETAILED ERRORS; SUCH AS, TIMING PROBLEMS, SEQUENCING ERRORS, ETC.

THIS PROGRAM COMPLETELY REPLACES AND OBSOLETEES MD-11-DZDXC.

2.0 REQUIREMENTS

2.1 EQUIPMENT

- A. PDP-11 COMPUTER WITH A MINIMUM OF 8K OF MEMORY.
- B. DX11-B 360 370 INTERFACE OPTION.
- C. ONE CONSOLE TELETYPE OR EQUIVALENT.

2.2 STORAGE

THE TEST PROGRAM LOADS INTO 4K OF MEMORY AND REQUIRES AT LEAST ANOTHER 4K FOR DATA BUFFERS. WITH 4K OF MEMORY FOR DATA BUFFERS, UP TO SIX DEVICES (6) MAY BE EMULATED. TO EMULATE EIGHT 2260/2848 DEVICES 8K OF MEMORY FOR DATA BUFFERS IS REQUIRED.

2.3 STORAGE MAP

THE FOLLOWING MAP ILLUSTRATES THE USAGE OF MEMORY BY THE DX11-B SYSTEM TEST PROGRAM.

C O R E M A P

G-777	I I I	INTERRUPT VECTORS (256 WORDS)	I I I
1000-17777	I I I	DX11-B TEST PROGRAM (4K WORDS)	I I I
/G-X777	I I I	SPW TABLE (256 WORDS)	I I I

X1000-X1777	TUMBLE TABLE (256 WORDS)
X2000-X2777	DUPLICATE TUMBLE TABLE (256 WORDS)
X3000-X3377	DST TABLE (128 WORDS)
X3400-X3475	SOFTWARE DEVICE STATUS TABLE (DEV 0) (31 WORDS)
X3476-X4437	INPUT BUFFER (DEV 0) (241 WORDS)
X4440-X5377	OUTPUT/DISPLAY BUFFER (DEV 0) (240 WORDS)
X5400-X5475	SOFTWARE DEVICE STATUS TABLE (DEV 1) (31 WORDS)
X5476-X6437	INPUT BUFFER (DEV 1) (241 WORDS)
X6440-X7377	OUTPUT/DISPLAY BUFFER (DEV 1) (240 WORDS)
	THE ABOVE SOFTWARE BUFFER LAYOUT (DEVICE STATUS TABLE, INPUT BUFFER + OUTPUT BUFFER) WILL BE REPEATED FOR EACH DEVICE SPECIFIED (UP TO 8). EACH DEVICE EMULATED REQUIRES 512 WORDS (2000 OCTAL) OF BUFFER SPACE
160000-177777	UNIBUS ADDRESSES

NOTE -- "X" IS DETERMINED BY THE BUFFER RELOCATION FACTOR
 INPUTTED AT SYSTEM CONFIGURATION TIME.
 THE DEFAULT VALUE OF "X" IS 20000. "X" IS ALWAYS A
 PHYSICAL ADDRESS.

3.0 LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES IS TO BE USED.

	STARTING ADDRESS FOR ABSOLUTE LOADER
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4.0 START UP PROCEDURE

4.1 CONTROL SWITCH SETTINGS -- NONE

4.2 STARTING ADDRESSES

1000 OR 200 NORMAL STARTING ADDRESS. FOR THE FIRST TIME AFTER LOADING ONLY, THE PROGRAM REQUESTS OPERATOR TO ENTER TEST PARAMETERS. EACH SUCCESSIVE RESTART USES THE PARAMETERS WHICH HAVE BEEN PREVIOUSLY ENTERED.

1002 RESTART ADDRESS WHICH REQUESTS OPERATOR TO ENTER TEST PARAMETERS AGAIN.

NOTE: AT ANY TIME WHILE THE PROGRAM IS RUNNING, A CONTROL P (↑P) TYPED ON THE TTY KEYBOARD WILL ALSO REQUEST THE OPERATOR TO REENTER THE TEST PARAMETERS.

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

1. LOAD PROGRAM INTO MEMORY USING ABSOLUTE LOADER.
2. LOAD ADDRESS 200.
3. PRESS START
4. THE PROGRAM WILL TYPE OUT "DZDXI-B NEW DX11-B RESPONDER"
5. THE SYSTEM NOW REQUESTS THE OPERATOR TO ENTER THE PARAMETERS NECESSARY TO RUN THE TEST.

4.3.2 ENTERING TEST PARAMETERS

BEFORE ANY TESTS MAY BE RUN OR WHENEVER A CHANGE IN PARAMETERS IS DESIRED, THE OPERATOR WILL BE REQUIRED TO ENTER

ALL THE TEST PARAMETERS. THE ENTERING OF THE PARAMETERS IS DONE VIA THE CONSOLE TELETYPE IN RESPONSE TO A SERIES OF QUESTIONS.

4.3.2.1 GENERAL RULES FOR ENTERING PARAMETERS

- A. ALL PARAMETERS MUST BE DELIMITED BY A CARRIAGE RETURN "(C R)"
- B. IF A TYPING ERROR IS DETECTED BEFORE ENTERING THE C/R, IT MAY BE CORRECTED BY:
 1. USING RUBOUT(S) TO DELETE THE LAST CHARACTER(S)
 2. HITTING CONTROL-U (↑U) TO DELETE THE ENTIRE ENTRY
 3. TO SELECT THE DEFAULT PARAMETER ENTRY, TYPE CARRIAGE RETURN (C/R) ONLY.
- D. IF THE PROGRAM DETECTS AN ERROR IN A PARAMETER IT WILL REPEAT THE QUESTION AGAIN AND REQUIRE THE OPERATOR TO REENTER THE PARAMETER.

4.3.2.2 PARAMETER DEFINITION

"UNIBUS ADDRESS -OCTAL-"

REQUESTS USER TO ENTER ADDRESS WHERE THE DX RESIDES ON THE UNIBUS. THIS MUST BE A 6 DIGIT OCTAL NUMBER BETWEEN 176200 AND 177700.

DEFAULT UNIBUS ADDRESS = 176200

"VECTOR ADDRESS -OCTAL-"

REQUESTS USER TO ENTER THE VECTOR ADDRESS FOR THE DX AS A 3 DIGIT OCTAL NUMBER BETWEEN 300 AND 770.

DEFAULT VECTOR ADDRESS = 300

"DEVICE ADDRESSES (XX,XX) -HEX-"

REQUESTS THE USER TO ENTER THE 360 CHANNEL ADDRESS(ES) OF THE 2260(S) TO BE EMULATED BY THE TEST. IF MORE THAN ONE DEVICE IS TO EMULATED, THEN THE USER ENTERS IN THE RANGE OF ADDRESSES TO BE EMULATED: SUCH AS, "A0 A3" --THIS INDICATES THAT UNITS A0, A1, A2, AND A3 CAN BE USED IN THE TEST. THE UNIT ADDRESSES ARE TO BE ENTERED IN HEX BETWEEN 00 AND FF. IF A RANGE OF DEVICES IS GIVEN, THERE CAN NOT BE MORE THAN 8 TOTAL.

DEFAULT DEVICE ADDRESS = 10,10

"CHANNEL TYPE (M OR S)"

REQUESTS THE USER TO INDICATE WHAT TYPE OF 360 CHANNEL

THE DX IS INTERFACED TO: M = MULTIPLEXER CHANNEL.
S = SELECTOR CHANNEL.

DEFAULT CHANNEL = S. SELECTOR CHANNEL

"MEMORY MANAGEMENT (Y OR N)"

REQUESTS THE USER TO INDICATE WHETHER THE PROGRAM IS TO
USE THE MEMORY MANAGEMENT OPTION.
Y = YES, N = NO

DEFAULT OPTION = N, DO NOT USE MEMORY MANAGEMENT

"BUFFER RELOCATION, IF SPECIFIED - IN EVEN ,000'S -OCTAL-"

REQUESTS THE PHYSICAL ADDRESS OF WHERE THE DX
FIRMWARE BUFFERS (TUMBLE TABLE, SPW + DST) AND
SOFTWARE DEVICE BUFFERS ARE TO RESIDE. THE RELOCATION
ADDRESS IS ENTERED IN OCTAL THOUSANDS, AND MUST BE
ON A 2000 BYTE ADDRESS BOUNDARY. EG: PHYSICAL ADDRESS
10000 IS ENTERED AS 100.

NOTE: THE BUFFER CANNOT BE CLOSER THAN 24000(8)
TO ANY 20000 BOUNDARY OR TO THE I/O PAGE. THE
THE DX IS NOT CAPABLE OF HAVING THESE BUFFERS
CROSS A 20000 BOUNDARY.
IT IS POSSIBLE TO OVERLAY THE ABSOLUTE LOADER
WHICH RESIDES IN THE HIGHEST AVAILABLE 4K(10)
OF THE FIRST 28K OF MEMORY.

DEFAULT BUFFER ADDRESS = 20 (20000)

"FRIEND (F) OR 2848 DIAG (D)"

REQUESTS THE USER TO INDICATE WHAT TYPE OF TEST WILL
BE RUN ON THE 360: F = IBM'S FRIEND OR D = THE 2848
RESPONDER DIAGNOSTICS.

DEFAULT OPTION = F -- FRIEND

IN FRIEND MODE, SEE PARA 5.0 FOR LIST OF VALID IBM
CHANNEL COMMANDS.

FRIEND MODE WILL ACCEPT THE SAME COMMAND STRINGS FORMERLY
USED WITH 'CTP'.

NOTE -- IF THE 2848 RESPONDER WAS SELECTED, NO
MORE PARAMETERS ARE NEEDED, SO THE SYSTEM
WILL BE INITIALIZED AND CONTROL PASSED TO
THE MONITOR. SEE MONITOR COMMANDS 4.4.

"SEPARATE I-O BUFFERS (Y OR N)"

REQUESTS THE USER TO INDICATE WHETHER SEPARATE INPUT
AND OUTPUT BUFFER SHOULD BE MAINTAINED FOR EACH CRT
UNIT EMULATED. SEPARATE INPUT/OUTPUT BUFFERS ALLOW
THE TRANSMISSION OF THE SAME DATA PATTERN TO THE

360 370 INDEPENDENT OF WHAT DATA IS RECEIVED.
 THIS IS USEFUL IN DETERMINING THE CAUSE OF BAD
 DATA BEING TRANSMITTED.

NOTE -- MOST TESTS USING 'FRIEND' WILL NOT UTILIZE
 SEPARATE I/O BUFFERS. THESE ARE ONLY FOR SPECIAL
 SITUATIONS AS MENTIONED ABOVE.

DEFAULT OPTION = N, NO USE THE SAME I-O BUFFER

NOTE -- IF THE SAME I-O BUFFER WAS SPECIFIED, NO
 MORE PARAMETERS ARE NEEDED, SO THE SYSTEM
 WILL BE INITIALIZED AND CONTROL PASSED TO
 THE MONITOR. SEE MONITOR COMMANDS 4.4.

"OUTPUT BUFFER FILL CHARACTER -HEX--"

REQUESTS THE USER TO ENTER THE CHARACTER WHICH IS USED
 TO FILL THE OUTPUT BUFFER. THIS CHARACTER IS ENTERED IN
 HEX (00 - FF).

DEFAULT FILL CHARACTER = 40, AN EBCDIC BLANK

NOW ALL TEST PARAMETERS HAVE BEEN ENTERED AND THE SYSTEM
 WILL BE INITIALIZED AND CONTROL WILL BE PASSED TO THE
 MONITOR.

4.3.3 SYSTEM INITIALIZATION

AFTER THE TEST PARAMETERS HAVE BEEN ENTERED THE SYSTEM IS
 INITIALIZED AND CONTROL PASSED TO THE MONITOR. BEFORE
 ANY COMMUNICATIONS MAY BE CONDUCTED TO THE 360 THE DX
 WILL NEED TO BE ENABLED VIA THE RUN "R" COMMAND. SEE SEC-
 TION 4.4 FOR MORE INFORMATION CONCERNING THIS AND OTHER
 MONITOR COMMANDS.

4.4 MONITOR COMMANDS

AFTER THE TEST PARAMETERS HAVE BEEN SUCCESSFULLY ENTERED,
 THE SYSTEM IS CONFIGURED AND INITIALIZED, THEN CONTROL IS
 PASSED TO THE MONITOR. ONCE IN THE MONITOR THE OPERATOR
 IS FREE TO ISSUE ANY COMMAND LISTED BELOW.

NOTE -- THE OPERATOR MUST ENABLE THE DX (RUN COMMAND)
 BEFORE ANY TESTS MAY BE PERFORMED WITH THE 360 370.

4.4.1 GENERAL RULES FOR ENTERING MONITOR COMMANDS

- A. ALL COMMANDS MUST BE DELIMITED BY A CARRIAGE RETURN
 "(C/R)"
- B. IF A TYPING ERROR IS DETECTED BEFORE ENTERING THE C P,
 IT MAY BE CORRECTED BY:
 1. USING RUBOUT(S) TO DELETE THE LAST CHARACTER(S).
 2. TYPING CONTROL-U (↑U) TO DELETE THE ENTIRE LINE.

- C. IF A USER WISHES TO ABORT A COMMAND, SUCH AS DUMPING DATA TO THE TELETYPE CONSOLE, HE DOES SO BY TYPING CONTROL-C (↑C).
- D. CONTROL-S (↑S) SIGNALS THAT CONSOLE OUTPUT SHOULD BE TEMPORARILY SUSPENDED.
- E. CONTROL-Q (↑Q) IS USED TO RESUME CONSOLE OUTPUT AFTER IT HAS BEEN STOPPED VIA A CONTROL-S.
- F. THE MONITOR MODE IS DENOTED BY THE ASTERICK (*) IN PRINT POSITION 1.
- G. IF AN ERROR IS DETECTED IN THE COMMAND BY THE PROGRAM, IT WILL PRINT A QUESTION MARK (?).
- H. IF THE OPERATOR TRIES TO ENTER DATA WHILE A COMMAND IS CURRENTLY ACTIVE OR HE OVERFLOWS THE INPUT BUFFER (64 CHARS) THE SYSTEM WILL PRINT A BACKSLASH (\) AND DELETE THE ENTIRE LINE.
- I. TYPING CTL-P (↑P) CAUSES THE SYSTEM TO BE REINITIALIZED AND NEW TEST PARAMETERS REQUESTED.

4.4.2 DESCRIPTION OF MONITOR COMMANDS

R -- ENABLE THE DX FOR TESTING - RUN COMMAND

THE RUN COMMAND DOES THE FOLLOWING:

- 1. INITIALIZES THE DX
- 2. CLEARS ALL TUMBLE TABLE ENTRIES.
- 3. ENABLES THE DX BY SETTING THE APPROPRIATE BITS IN DXCS.

S -- DISABLE THE DX - STOP COMMAND

THE STOP COMMAND ALLOWS THE USER TO DISABLE THE DX AFTER A SPECIFIC EVENT. THIS MAY EITHER BE IMMEDIATELY, AFTER AN INITIAL SELECTION SEQUENCE, AFTER A DATA TRANSFER, AFTER AN ENDING SEQUENCE, OR ON A PARITY ERROR.

THE FORMS OF THE STOP COMMAND ARE:

S(C/R) -- STOP IMMEDIATELY
 SI(C/R) -- STOP AFTER NEXT INITIAL SELECTION SEQUENCE
 SD(C/R) -- STOP AFTER NEXT DATA TRANSFER COMPLETION
 SE(C/R) -- STOP AFTER NEXT ENDING SEQUENCE
 SP(C/R) -- STOP ON NEXT PARITY ERROR

AFTER THE CONDITIONS OF STOP ARE MET, THE DX WILL BE DISABLED. TYPE "R" TO CONTINUE.
 THE FOLLOWING WILL BE PRINTED ON THE CONSOLE TELETYPE:

```

"CURRENT DEVICE -- XX"   THE CURRENT DEVICE ADDRESS IN HEX
"XXXXXX"                 THE DXDS IN OCTAL - PROBABLY ZERO
"XXXXXX"                 THE DXCA IN OCTAL
"XXXXXX"                 THE DXCS IN OCTAL
"XXXXXX"                 THE DXOS IN OCTAL
"XXXXXX"                 THE DXBA IN OCTAL
"XXXXXX"                 THE DXBC IN OCTAL
"XXXXXX"                 THE DXMO IN OCTAL
"XXXXXX"                 THE DXMI IN OCTAL
"XXXXXX"                 THE DXCB IN OCTAL
"XXXXXX"                 THE DXND IN OCTAL
"XXXXXX"                 THE DXES1 IN OCTAL
"XXXXXX"                 THE DXMOB IN OCTAL
"XXXXXX"                 THE DXES2 IN OCTAL

```

C -- DUMP COMMAND

THE DUMP COMMAND ALLOWS THE USER TO DUMP VARIOUS DATA BUFFERS, TABLES OR CORE LOCATIONS ON THE CONSOLE TELETYPE A VARIETY OF FORMATS. THE FOLLOWING DESCRIBES THE SYNTAXES OF THE DUMP COMMAND:

```

DTT,O      DUMP TUMBLE TABLE IN OCTAL
DTT,H      DUMP TUMBLE TABLE IN HEX

```

THE DUMP TUMBLE TABLE COMMAND REFERENCES A DUPLICATE TUMBLE TABLE MAINTAINED EXCLUSIVELY FOR THIS FUNCTION. THE TUMBLE TABLE IS DUMPED IN REVERSE CHRONOLOGICAL ORDER AND PRODUCES THE FOLLOWING REPORT:

```

XXXXXX      TT2 -- LAST OPERATION
XXXXXX      TT1 -- LAST OPERATION
XXXXXX      TT2 -- PREVIOUS T/T ENTRY
XXXXXX      TT1 -- PREVIOUS T/T ENTRY
ETC

```

```

DIN,O,XX    DUMP INPUT BUFFER FOR DEVICE XX IN OCTAL
DIN,H,XX    DUMP INPUT BUFFER FOR DEVICE XX IN HEX
DIN,E,XX    DUMP INPUT BUFFER FOR DEVICE XX IN EBCDIC
DIN,A,XX    DUMP INPUT BUFFER FOR DEVICE XX IN ASCII
DOT,O,XX    DUMP OUTPUT BUFFER FOR DEVICE XX IN OCTAL
DOT,H,XX    DUMP OUTPUT BUFFER FOR DEVICE XX IN HEX
DOT,E,XX    DUMP OUTPUT BUFFER FOR DEVICE XX IN EBCDIC
DOT,A,XX    DUMP OUTPUT BUFFER FOR DEVICE XX IN ASCII
DSSSSSS,EEEEEE,O  DUMP BETWEEN GIVEN LIMITS IN OCTAL
DSSSSSS,EEEEEE,H  DUMP BETWEEN GIVEN LIMITS IN HEX
DSSSSSS,EEEEEE,E  DUMP BETWEEN GIVEN LIMITS IN EBCDIC
DSSSSSS,EEEEEE,A  DUMP BETWEEN GIVEN LIMITS IN ASCII

```

NOTE -- XX IS THE DEVICE ADDRESS IN HEX ; IF NOT SPECIFIED, WILL DEFAULT TO 1ST DEVICE (CRT) # IN THE DEVICE TABLE.
SSSSSS IS THE STARTING MEMORY ADDRESS IN OCTAL
EEEEEE IS THE ENDING MEMORY ADDRESS IN OCTAL

F -- FILL COMMAND

THE FILL COMMAND ALLOWS THE USER TO FILL THE INPUT OR OUTPUT FOR A DEVICE WITH A SPECIFIC DATA PATTERN. THE FOLLOWING DESCRIBES THE SYNTAX FOR THE FILL COMMAND.

FIN,YY,XX FILL INPUT BUFFER FOR DEVICE XX WITH YY
FOT,YY,XX FILL OUTPUT BUFFER FOR DEVICE XX WITH YY
WHERE:
XX = THE DEVICE ADDRESS IN HEX
YY = THE FILL CHARACTER IN HEX

H -- HELP COMMAND

THE HELP COMMAND PRINTS OUT A SYNOPSIS OF THE MONITOR COMMANDS AND CONSOLE CONTROL CHARACTERS AVAILABLE FOR OPERATING THE DX11-B SYSTEM TEST PROGRAM. THE SYNTAX OF THE HELP COMMAND IS:

H PRINT OUT HELP MESSAGE

I -- INPUT COMMAND

THE INPUT COMMAND ALLOWS THE USER TO INPUT DATA FOR A PARTICULAR CRT AND SEND IT TO THE 360. IN THE SAME MANNER AS IF HE WERE ACTUALLY ON A 2260. THE INPUT COMMAND IS ONLY VALID WHEN THE IBM 2848 DIAGNOSTICS ARE BEING RUN. THE SYNTAX OF THE INPUT COMMAND IS:

IXX,D---D
WHERE:
XX IS THE DEVICE ADDRESS IN HEX

D---D IS THE DATA TO BE SENT TO THE 360. THE DATA WILL BE CONVERTED TO EBCDIC BEFORE BEING TRANSMITTED TO THE 360.

E -- ENABLE A DX-11 DEVICE ADDRESS

THE ENABLE COMMAND TURNS THE DEVICE INDICATED IN THE OPERAND TO AN ON-LINE STATUS. A DEVICE ADDRESS ONLY BECOMES OFF-LINE VIA THE "K" COMMAND. THE DEVICE ADDRESS MUST BE ENTERED IN HEX AND BE WITHIN THE LIMITS SPECIFIED BY THE TEST PARAMETERS. THE SYNTAX OF THE ENABLE COMMAND IS:

EYY ENABLE DEVICE XX

K -- DISABLE DX11-B DEVICE ADDRESS

THE KILL COMMAND SETS THE DEVICE INDICATED TO AN OFF-LINE STATUS. THE DEVICE ADDRESS ENTERED MUST BE IN HEX AND BE WITHIN THE LIMITS SPECIFIED BY THE TEST PARAMETERS. A DEVICE MAY ONLY BE ENABLED AGAIN VIA THE "E" COMMAND. THE SYNTAX OF THE KILL COMMAND IS:

KYY DISABLE DEVICE XX

A -- ACCESS AND DISPLAY LOCATIONS (QUICK LOOK + CHANGE)

THE ACCESS COMMAND ALLOWS THE USER TO DISPLAY AND ALTER MEMORY LOCATIONS WHILE THE PROGRAM IS RUNNING. AN ON-LINE QDT. THE ACCESS COMMAND SHOULD BE USED WITH EXTREME CAUTION. WHEN THE USER ENTERS THE ADDRESS TO BE ACCESSED, IN OCTAL, THE PROGRAM RESPONDS BY PRINTING THE CONTENTS OF THE REFERENCED LOCATION IN OCTAL ON THE CONSOLE TELETYPE. THE OPERATOR MAY THEN:

- A. CHANGE THE CONTENTS OF THE LOCATION BY TYPING IN THE NEW CONTENTS IN OCTAL, DELIMITED BY A (C/R). THE SYSTEM WILL THEN OPEN THE NEXT LOCATION AND DISPLAY ITS CONTENTS.
- S. TYPE A (C/R) ONLY. THIS WILL NOT AFFECT THE CONTENTS OF THE CURRENT LOCATION. THE SYSTEM WILL OPEN THE NEXT LOCATION AND DISPLAY ITS CONTENTS.
- C. TYPE (/) SLASH FOLLOWED BY A (C/R) TO ESCAPE TO THE MONITOR.

THE SYNTAX OF THE ACCESS COMMAND IS:

AYYYYY ACCESS + DISPLAY LOCATION YYYYY
NOTE: NO SPACE BETWEEN "A" AND LOCATION.

5.0 OPERATING PROCEDURE

REFER TO SECTION 4.4 "MONITOR COMMANDS" FOR DETAILS.

SEE MAINTENANCE MANUAL EK-DX11B-MM-002 FOR PROCEDURES FOR OPERATING THE IBM SYSTEM.

IN FRIEND OR 2848 DIAG.MODE, THE FOLLOWING IBM COMMANDS ARE VALID:

COMMAND		DESCRIPTION
OCTAL	HEX	
00	00	TEST I/O
01	01	WRITE FULL BUFFER
02	02	*READ MANUAL INPUT
03	03	NO OPERATION
04	04	SENSE
05	05	WRITE LINE ADDRESS
06	06	READ FULL BUFFER
07	07	ERASE
12	0A	*READ SHORT MANUAL INPUT

*DATA IN THE OUTPUT BUFFER IS ONLY TRANSMITTED ONCE FOR THESE COMMANDS.

6.0 ERRORS

6.1 ERROR HALTS

THERE ARE ONLY TWO CONDITIONS (MEMORY TIME-OUT AND MEMORY MANAGEMENT ERROR) WHICH WILL CAUSE THE PROGRAM TO HALT OUTSIDE OF THE TRAP CATCHER. BOTH ERRORS ARE ACCOMPANIED WITH A DESCRIPTIVE MESSAGE RELATING THE CAUSE OF THE ERROR. RECOVERY FROM ANY SYSTEM HALT REQUIRES THE OPERATOR TO RESTART THE PROGRAM AT LOCATION 200. SEE ERROR MESSAGES FOR DETAILS.

6.2 DX ERRORS

UPON RECEIPT OF AN ILLEGAL DX CONDITION (INVALID DEVICE ADDRESS, INVALID DX COMMAND, NON EXISTENT MEMORY ERROR) THE SYSTEM WILL PRINT A DESCRIPTIVE ERROR MESSAGE AND DISABLE THE DX. THE USER MAY THEN EXAMINE THE STATE OF THE DX. NOTE THAT THE DX MUST BE ENABLED BEFORE MORE TESTS CAN BE PERFORMED ON THE 360/370 (RUN COMMAND). AFTER THE DX HAS BEEN DISABLED THE FOLLOWING WILL BE PRINTED ON THE CONSOLE TELETYPE:

"CURRENT DEVICE -- XX"	THE CURRENT DEVICE ADDRESS IN HEX
"XXXXXX"	THE DXDS IN OCTAL -- PROBABLY ZERO
"XXXXXX"	THE DXCS IN OCTAL
"XXXXXX"	THE DXOS IN OCTAL
"XXXXXX"	THE DXBA IN OCTAL
"XXXXXX"	THE DXBC IN OCTAL
"XXXXXX"	THE DXMO IN OCTAL
"XXXXXX"	THE DXMI IN OCTAL
"XXXXXX"	THE DXCB IN OCTAL
"XXXXXX"	THE DXND IN OCTAL
"XXXXXX"	THE DXES1 IN OCTAL
"XXXXXX"	THE DXMOB IN OCTAL
"XXXXXX"	THE DXES2 IN OCTAL

NOTE -- THE DX WILL NOW BE IN A DISABLE STATE REQUIRING THE USER TO ENABLE THE DX VIA THE RUN "R" COMMAND BE ORE COMMUNICATIONS TO THE 360 CAN RESUME.

6.3 ERROR MESSAGES AND SUGGESTED CORRECTIVE ACTIONS

"MEMORY TIME OUT"

THE MEMORY TIME OUT ERROR INDICATES A TRAP WAS EXECUTED THRU LOCATION 4. THE SYSTEM HALTS AFTER THIS ERROR. THE MEMORY TIME OUT ERROR NORMALLY DENOTES THAT AN ILLEGAL ADDRESS WAS REFERENCED AND THE SYSTEM SHOULD PROBABLY BE RECONFIGURED.

"MEMORY MANAGEMENT ERROR"

THIS ERROR INDICATES A TRAP WAS EXECUTED THRU LOCATION 250. THE MEMORY MANAGEMENT TRAP VECTOR. THE SYSTEM WILL HALT AFTER REPORTING THE ERROR CONDITION.

"ILLEGAL DEVICE NUMBER"

THIS ERROR INDICATES THAT A TUMBLE TABLE ENTRY WAS MADE WHICH CONTAINED A DEVICE ADDRESS OUTSIDE THE VALID DEVICE ADDRESSES SPECIFIED BY THE TEST PARAMETERS. NOTE -- THIS CONDITION WILL NOT OCCUR ON A SYSTEM RESET FROM THE 360. SEE SECTION 6.3 FOR FURTHER DETAILS ON DX ERRORS.

"INVALID DX COMMAND"

THIS ERROR INDICATES THAT AN INVALID COMMAND WAS DETECTED FROM THE 360. THIS ERROR CAN ONLY OCCUR ON AN INITIAL SELECTION SEQUENCE. SEE SECTION 6.3 FOR FURTHER DETAILS ON DX ERRORS.

"NON EX-MEM ERROR"

THIS ERROR INDICATES THAT A NON-EXISTENT MEMORY ERROR WAS DETECTED IN A TUMBLE TABLE FROM THE DX. SEE SECTION 6.3 FOR FURTHER DETAILS ON DX ERRORS.

"PARITY ERROR"

THIS ERROR INDICATES THAT A PARITY ERROR WAS DETECTED BY THE DX. TO STOP THE DX WHEN A PARITY ERROR IS DETECTED, THE USER SHOULD CONSULT THE "STOP" COMMAND.

7.0 RESTRICTIONS
SEE MEMORY REQUIREMENTS (SECTION 2.2)

7.1 MULTIPLE DEVICE ADDRESSES
ONLY 8 DEVICE ADDRESSES MAY BE EXERCISED SIMULTANEOUSLY OVER THE DX. ALL THE DEVICE ADDRESSES MUST BE CONTIGUOUS.

8.0 PROGRAM DESCRIPTION

PURPOSE

THE PURPOSE OF THIS PROGRAM IS TO GIVE INSIGHT ON FUNCTIONALITY OF THE HARDWARE AND TO GIVE AN EXAMPLE OF DX11 PROGRAMMING. IT WILL, BY DEFAULT, PROVE ON "WHICH SIDE OF THE FENCE" A PROBLEM LIES- SOFTWARE OR HARDWARE ,DEC OR IBM.

THE FOLLOWING IS A DESCRIPTION OF THE PROGRAMMING TECHNIQUES USED- IT IS BROKEN DOWN BY THE NEAREST DISCRIPTIVE ROUTINE-

-----KEYBOARD & PRINTER I/O -----

MSG: THIS ROUTINE PACKS THE TYPE OUT MESSAGE IN BUFFER AREA - LOOKS TO SEE IF PRINTER IS BUSY - IF NOT, PRINTS AND

RESTORES BUFFER AREA UNTIL MESSAGE IS COMPLETE.

IF BUSY, IT PACKS BUFFER AREA UNTIL FULL, WAITING FOR THE OTHER PRINTABLE TASK TO COMPLETE.

THIS APPROACH PROHIBITS MESSAGE INTERWEAVING. USES PROUT:

PROUT: THIS ROUTINE SENDS DATA TO PRINTER BASED UPON TTY FLAG IS BUSY OR NOT.

TRIN: THIS ROUTINE ACCEPTS CHARACTERS FROM KEYBOARD AND STUFFS THEM AWAY IN TBUF, BUT FIRST, IT CHECKS FOR CERTAIN CONTROL CHARACTERS.

↑P - JUMP TO RESTART TO RESELECT PARAMETERS.

↑C - WHEN COMMAND (TCMACT) ACTIVE = SET ABORT FLAG (TCMDAB)

↑C - WHEN COMMAND (TCMACT) NOT ACTIVE = PRINT & RESET BUFFER PTR.

A C/R DELIMITS TTY COMMAND - TCMACT IS SET - NOW IF YOU CONTINUE TYPING - TCMACT BEING SET WILL NOW THROW AWAY THOSE CHARACTERS.

-----MONITOR PARAMETER SETUP -----

SYSINT: THIS ROUTINE CLEARS THE THE WORLD, SETS UP TTY KEYBOARD & PRINTER VECTOR AREAS.

SETS UP MEMORY TIME OUT & MEMORY MANAGEMENT ERROR VECTOR AREAS.

CLEARS OUT SYSTEM BUFFER AREA & SETS UP TTY BUFFER POINTERS.

----GETS DX ADDRESS - CHECKS FOR LIMITS SAVES IT IN UNADDR:

----GETS DX VECTOR - DITTO

GETS DEVICE ADDRESS IN HEX - ACCEPTS RANGE OF DEVICE ADDRESSES MUST NOT EXCEED 8 - SEPARATED BY A COMMA

SAVES START DEV ADD IN SDEV
 SAVES END DEV ADD IN EDEV

----CHECKS FOR LEGAL TERMINATOR IE. C/R

----GETS CHANNEL TYPE M OR S

----GETS ANSWER WHETHER MEMORY MANAGEMENT? Y OR N

IF YES, SET UP VECTOR 4 AND TEST FOR EXISTANCE OF MEMORY
 MANAGEMENT.

----GET BUFFER RELOCATION IN ,000'S (THOUSANDS)

* CHECKS FOR BOUNDARY 20000 OR GREATER

* CHECKS FOR MULTIPLE OF 2000

* CHECKS TO SEE IF NUMBER IS VALID WITHIN MEMORY MANAGE-
 MENT AND COMPARES WHETHER M/M WAS SPECIFIED.

----GET TEST TYPE - FRIEND OR 2848 - STORE IN TSTTYP: - IF
 FRIEND ASK NEXT QUESTION, IF 2848 JUMP TO INIT:

----SEPARATE I/O BUFFERS? Y OR N
 STORE IN IOBUF:
 IF Y ASK

----FILL CHARACTER IN HEX
 SAVE IN FILLCH

-----MONITOR SETUP SUBROUTINES -----

NOMM: NO MEMORY MANAGEMENT AVAILABLE.

MMERR: MEMORY MANAGEMENT TRAP OUT ROUTINE
 CLEAR WORLD
 TYPE OUT ERROR MESSAGE
 HALT

INITRT: PRINTS MESSAGE - WAITS FOR INPUT - GETS IT OR IF IT IS
 A C/R - DEFAULTS.

CO7B: GOBBLES CHARACTERS FROM INPUT BUFFER AREA - CONVERTS TO
 OCTAL AND SAVES RESULT IN R3 - THIS ROUTINE DOES NO
 OTHER CHECKING THE CODE FOLLOWING UNIT EXAMINE R3 FOR

VALIDITY.

CHTS: GOBBLES CHARACTERS FROM INPUT BUFFER AREA CONVERTS HEX #
TO OCTAL AND SAVES RESULT. STORES AWAY TERMINATOR IN R4

THE TERMINATOR SHOULD BE EITHER A C/R OR A COMMA.

-----PROGRAM INITIALIZATION-----

INIT: SET UP MEMORY TIME OUT TRAP
----SET UP DX ADDRESS TABLE. SET UP VECTOR ADDRESS WITH
DXISR. WAS BUFFER RELOCATION SPECIFIED - IF NOT START
AT 20000.

----TEST FOR MEMORY MANAGEMENT.
----IF YES - SET UP MEMORY MANAGEMENT REGISTERS AND ENABLE
MEMORY MANAGEMENT.

----SET UP SPW TABLE
LOAD DXOS WITH BUFFER OFFSET (DEFAULT = 20000)
CALCULATE ADDRESS OF DST TABLE - SAVE AT DSTOFF

----SET UP SPW TABLE - MOVE UCHK FOR INVALID DEVICE #'S
MOVE DST ADDRESS TO VALID DEVICE #'S

SPW TABLE = 400(8) WORDS.

----CLR TUMBLE TABLE & DUPLICATE TUMBLE TABLE.

TT = 400(8) WORDS

DTT = 400(8) WORDS

----SET UP DST TABLE
FIRST 11. BYTE LOCATIONS FILL IN WITH VALID COMMANDS.
REMAINDER DST = UCHK = 2
DST = 128. WORDS = 256. BYTES

----SET UP FILL CHARACTER

----COMPUTE MAX NUMBER OF DEVICES +1
SAVE AT MAXDEV:
DEVCON = FIRST DEVICE -1

-----START SETTING UP DEVICE BUFFERS
 SAVE ADDRESS AT SDEV18
 MAKE THE FIRST DEVICE = 0 IN THIS TABLE
 CLEAR DEVICE STATUS BUFFER TABLE & INPUT BUFFER

----CREATE & SAVE ADDRESS OF INPUT/DISPLAY BUFFER IN DEVICE
 BUFFER AREA.
 CREATE & SAVE ADDRESS OF OUTPUT/DISPLAY BUFFER IN DEVICE
 BUFFER AREA.

----FILL OUTPUT/DISPLAY BUFFER WITH FILL CHARACTER
 NOW CHECK IF ALL DEVICES HAVE HAD THEIR DEVICE STATUS
 BUFFER TABLES GENERATED - IF NOT, REPEAT INT130: THRU
 INT150:

- REMEMBER MEMORY MANAGEMENT HAS BEEN TURNED ON-

CREATE EXTENDED ADDRESS BITS AND SAVE AT XADDR: SET
 FIRST TIME THRU FLAG - QUESTION/ANSWERS WILL ONLY BE
 GENERATED IF LA 1002 & START. OR HITTING ↑S ON TTY KEYBOARD

-----THE EXEC: SYSTEM EXECUTIVE BACKGROUND -----
 (A WAIT ROUTINE)

EXEC: CLR SYSTEM FLAGS
 ----ANY COMMANDS TO EXECUTE? IF YES GO TO EXEC20. DID THE
 DX ABORT AN OPERATION - IF NOT SPIN HERE

----ALWAYS COME HERE AFTER TELETYPE INPUT HAS SET TCMACT -
 THIS ROUTINE DISPATCHES YOU TO THE COMMAND TYPED IN - IF
 NOT AN ACCEPTABLE SYSTEM COMMAND = ? RETURN TO EXEC.
 (DISPATCH)

---TYPICAL DX COMMANDS---
 (ENTERED VIA TTY KEYBOARD)

1. 01 COMMAND

RLN: CHECK IF DX IS ENABLED -
IF YES, TYPE ? AND BELL)--RETURN TO EXEC AND
WAIT FOR ANOTHER TTY COMMAND.

IF NO, CONTINUE
RETURN TO EXEC.

CLR DXCS
INC DXCS - GO

CLR DEVICE STATUS BUFFER TABLE
(SCMD
SLCMD
SSENSE (NOT SCURS, SINTB, SOUBF, SONLF)
SSTAT
SBUFA
SRBYTC
SRDRQ
SMINS)

DO THIS FOR ALL DEVICE STATUS BUFFER TABLES (BASED ON
MAXDEV:)

CLR DXACT, CMDCHF, DXABFL

CLR TUMBLE TABLE & DUPLICATE TUMBLE TABLE
SET EXTENDED ADDRESS BITS IN DXCS

CHECK FOR CHANNEL TYPE

IF SELECTOR CHANNEL SET BUSY ENABLE IN DXCS

SET INTERRUPT ENABLE & ONLINE IN DXCS

RETURN TO EXEC

STOP DX COMMAND

STOP: PICK UP NEXT TTY INPUT CHARACTER FOR THE MODE.
WHAT IS IT?
C/P = CRUNCH DX, CONVERT AND PRINT CURRENT DEVICE #
IN HEX, PRINT 3 DX REGISTERS CONTENTS.
CLR ABORT FLAG (DXABFL), CLR DONE
RESET DX, SET GO, RETURN TO EXEC.

C = SET THE STOP FLAG (DXSTPF), TEST WHETHER STOP HAS
TAKEN PLACE, IF NOT, WAIT UNTIL DXSTPF HAS BEEN
CLEARED (TYPICALLY THE PCHEND: ROUTINE WILL CLEAR
DXSTPF (DXISR:)), DISABLE DX, RETURN TO EXEC

E = SAME AS D EXCEPT (TYPICALLY PESEND: OR
 PCHEND: ROUTINES WILL CLEAR DXSTPF (DXISR:))

: = SAME AS D EXCEPT (TYPICALLY PCHIS: ROUTINE
 WILL CLEAR DXSTPF (DXISR:))

ANY OTHER CHARACTER = AN ILLEGAL CHARACTER

DUMP COMMAND

DUMP: PICK UP THE NEXT SEQUENCE OF OCTAL NUMBERS OR NEXT CHARACTER
 FROM TTY INPUT BUFFER AREA.

GLIMIT:) 1ST CHECK IF THEY ARE OCTAL NUMBERS. IF YES, (SAVE IT); IF
 NOT, DETERMINE IF IT IS AN "I", "O", OR "T".
 IF NOT ONE OF THESE - TYPE ERROR MESSAGE

(SAVE IT) OCTAL NUMBERS, 1ST ADDRESS GIVEN = SADDR
 2ND ADDRESS GIVEN = EADDR.

IF "T" -CHECK FOR 2ND T - CREATE STARTING ADDRESS
 OF DUPLICATE TT (TTPTR +1000)
 (SAVE) DTT2 = SADDR

IF "I" - NOW CHECK FOR N - CREATE STARTING & ENDING ADDRESSES
 OF DEVICE 0 INPUT BUFFER TABLE
 SINBUF (DEV 0) = SADDR
 SADDR + 481. = EADDR

IF "O" - NOW CHECK FOR T - CREATE STARTING AND ENDING
 ADDRESSES OF DEVICE 0 OUTPUT BUFFER TABLE
 SOUTB (DEV0) = SADDR
 SADDR + 479. = EADDR

NOW SET UP DMPADR: TO CONTAIN THE ADDRESS OF THE
 CORRECT DUMP ROUTINE (IE ASCII DUMP, EBCDIC, HEX, OCTAL)

CHECK TO SEE IF IT IS A TT DUMP - IF YES, DUMP DTT
 IN REVERSE - USES ADDRESS IN DMPADR. CONTINUES DUMPING
 PRINTING) UNTIL BEGIN OF DTT IS SEEN.

IF NOT A TT DUMP - CHECK FOR A DEVICE # SPECIFIED - IF
 NOT JUST DUMP DEFAULTED LIMITS GET THE DEVICE #, CRUNCH THE
 CONTENTS OF SADDR & EADDR TO POINT TO THE PROPER DEVICE
 # SPECIFIED.

CONVERT AND DUMP IT, STOPPING @EADDR
 RETURN TO EXEC.: LOOKING FOR MORE COMMANDS TO EXECUTE.

FILL COMMAND

FILL: PICK UP CHARACTERS FROM TTY INPUT BUFFER AREA - PERFORMS
VERY SIMILAR TO THE DUMP COMMAND EXCEPT IF FILLS AREA WITH THE
SPECIFIED FILL CHARACTER (FILLCH)

USE ONLY THOSE FILL COMMANDS AS SPECIFIED IN THE TEXT - ANY
OTHERS MAY OBLITERATE THE CORE.

BASICALLY THIS IS USED TO FILL THE OUTPUT OR INPUT BUFFER AREA
WITH FILL CHARACTER (FILLCH)

ACCESS COMMAND

ACCESS: OPENS CORE LOCATION, ALLOWING IT TO BE MODIFIED WITH NEW CONTENTS.
A " " RETURNS YOU TO THE EXEC, A C/R OPENS NEXT LOCATION ETC.
-VERY SIMILAR TO "ODT" -

ENABLE DEVICE

ENABLE: GETS THE TYPED DEVICE # IN HEX
CLEARS THAT DEVICES STATUS TABLE
CLR SSENSE, CLR SONLF
RETURN TO EXEC

FILL DEVICE

FILL: GETS THE TYPED DEVICE # IN HEX
MOVES A "1" INTO SONLF
MOVES A UNIT CHECK INTO THE SPW TABLE
RETURN TO EXEC.

INPUT COMMAND

INPUT: CHECK FOR FRIEND OR 2848? - 2848 ONLY GET DEVICE #
IN HEX FROM TTY INPUT BUFFER.

PUT THE START CHARACTER IN DEVICE BUFFER AREA (SMI=112)
SAVE DATA LOCATION (SMINS)
INC CURSOR POSITION
CHECK FOR END OF SCREEN (SCURS=478.) IF YES, PUT EOM
(EOM=152) IN THE BUFFER AREA, INC CURSOR POSITION, QUEUE
A READ REQUEST (SRDRQ)

PUSH STACK (CREATE PONEY INTERRUPT)

JUMP DXEXEC

-----TYPICAL TT1 (TUMBLE TABLE) ENTRIES-----
 (THESE SERVICE ROUTINES ARE SELECTED BY THE DXISR
 ROUTINE WHEN THE TUMBLE TABLE ENTRY (TT1=DXDS) IS EXAMINED.)

SYSTEM RESET

PSYSRT: CLEAR DEVICE STATUS BUFFER TABLE. SETUP DISPLAY BUFFER
 AREA WITH FILL CHAR.
 DO THIS FOR ALL DEVICES
 CLR ACTIVE FLAGS, CMD CHAINING FLAG (DXACT & CMDCHF)
 CLR CUBUSY IN DXCS
 PROCESS NEXT ENTRY IN TT
 IF NO MORE TT ENTRIES - GO TO DXEXEC.

SELECTIVE RESET

PSELRT: CLR DEVICE STATUS BUFFER TABLE
 FOR THAT DEVICE + SENSE
 IT IS A SEL RESET ISSUED AGAINST THE CURRENT ACTIVE DEVICE.
 PROCESS ANY MORE TT ENTRIES THEN GO TO DXEXEC.

INTERFACE DISCONNECT:

PINDSC: IF DEVICE WAS ACTIVE, ITS DEVICE STATUS TABLE WILL BE
 CLEARED - IF NOT ACTIVE, IGNORE CMD.
 IF ACTIVE - QUEUE CE! DE IN SCMD
 (TYPICALLY IBM WILL INTERFACE DISCONNECT A DEVICE EVEN
 THO THE DEVICE WAS NOT ACTIVE)
 IF ACTIVE - CHECK FOR CMDCHF: & DXACT: FOR THAT PARTICULAR
 DEVICE - IF YES, CLR BOTH FLAGS - ONLY ONE DEVICE AT A TIME
 CAN HAVE CMD CHAINING AND/OR DX ACTIVE SET.
 IF NO MORE TT ENTRIES - GO TO DXEXEC.

STATUS ACCEPT

PESENT: WAS LAST CMD A WRITE? IF SO, FORMAT THE DISPLAY (DISCTL)
 WAS ATTN ACCEPTED? - IF YES, SET SRDRQ (READ MANUAL
 INPUT REQUEST)
 IF NO, CONTINUE
 CLR OUT SLCMD (LAST CMD)(SET ONLY ON A WRITE)
 CLR DXACT DXACTIVE FLAG
 CLR DEVICE STATUS BUFFER TABLE

TEST FOR CMDCHN (TT1)(DXDS) - IF YES, SAVE DEVICE # IN
IN CMDCHF (ONLY ONE DEVICE AT A TIME CAN
CMD CHAIN)
WAS A SE SPECIFIED? (STOP ON ENDING SEQ) - IF YES, CRUNCH
DX - IF NO, AND NO MORE TT ENTRIES GO TO
DXEXEC

NON-EXISTANT MEMORY - FATAL ERROR

FNXM: STOP THE DX FROM INTERRUPTING
SET ABORT FLAG
EXIT FROM DXISR - GO TO MONITOR WAIT STATE(EXEC).
(DO NOT PASS THRU DXEXEC ROUTINE - JUST ABORT)

PAPIT: ERROR

FFARER: WAS STOP ON PARITY ERROR SPECIFIED?
THE PROGRAM (PARSTP: =0) HAS BEEN PRESET TO YES
IF YES - CRUNCH DX
IF NO (PARSTP: =>0) QUEUE A UNIT CHK TO SSTAT (STATUS WORD)
RETURN TO DXISR AND CONTINUE CHECKING TT1

EVERYTHING OK UP TO THIS POINT
CHANNEL INITIATED SELECTION SEQUENCE

PCHIS: WAS A SI (STOP ON ISS) SPECIFIED?
IF YES, CRUNCH DX
CMDREJ? YES, IS DEVICE ONLINE?
NO, SET INTREQ IN SSENSE
CMDCHF? IF YES, CLR CMDCHF.
ANY MORE TT ENTRIES? - IF NO, GO TO DXEXEC

CMDREJ? YES, IS DEVICE ON LINE?
YES, TEST PARITY ERROR
IF NOT, MUST BE ILLEGAL CMD - SET BUS OUT IN SSENSE
IF YES, SET SCMDRJ (COMMAND REJECT) IN SSENSE

CMDCHF? YES, CLR CMDCHF
ANYMORE TT ENTRIES, NO, GO TO DXEXEC

CMDREJ? NO, THEN PROCESS CMD (TT2 CONTAINS CMD)

IS THIS A TIO CMD? IF YES, IGNORE, CHECK CMDCHF ETC.
ANYMORE TT? NO? GO TO DXEXEC

IS THIS A NOP CMD? IF YES, IGNORE, CHECK CMDCHF ETC.
TT ENTRIES?, NO GO TO DXEXEC

IS THIS A VALID CMD? NO - ABORT DX(DXAB:)...EXIT FROM DXISR &
RETURN TO EXEC:
YES - QUEUE CMD (TT2) TO SCMD

IS CMDCHF SET? YES, CLR CMDCHF
ANYMORE TT ENTRIES, NO? GO TO DXEXEC

CHANNEL END, PREPARE ENDING SEQUENCE RESPONSE

PCHEND: CLR DXACT

WAS STOP ON DATA TRANSFER DONE? YES, STOP DX
NO, QUEUE CEDE TO SCMD
SUBTRACT DXBYTE COUNT (DXBC) FROM SRBYTC
WAS THERE A PARITY ERROR? IF YES,
QUEUE EQPCHK TO SSENSE (EQPCHK = 20)

LOOP: ANYMORE TT ENTRIES? NO, GO TO DXEXEC

CONTROL UNIT END

PCJEND: CLR DXACT
USED TO KEEP TRACK OF REMAINING BYTE COUNT (SRBYTC)
AND TO KEEP TRACK OF CURRENT BUFFER POINTER (MULTIPLEXER CHANNEL)
JUMP TO PCHEND:

--- DXISR (DX11B INTERRUPT SERVICE ROUTINE) ---

THE DX SHOULD MAKE ENTRIES IN TT - INTERRUPTS VECTORED
THRU WHEN PSW IS < DX11B

DXISR: CHECK IF ZERO TT ENTRY UPON INTERRUPT
IF ZERO - ASSUME TT ENTRY HAS ALREADY BEEN PROCESSED -
RETURN FROM INTERRUPT

IF NON-ZERO, CLEAR "DONE" (DXCS) FOR EVERY TT ENTRY
- SAVE FIRST TT ENTRY IN DUPLICATE TT (DTT1) & TT1.
SAVE SECOND TT ENTRY IN DUPLICATE TT (DTT2) & TT2.
CLR BOTH TT ENTRIES TO SIGNIFY THAT THEY WERE PROCESSED.

NOTE: TT1 CONTAINS CONTENTS OF DXDS...TT2 CONTAINS CONTENTS
OF DXCA.

PICK UP DTT2 AND CHECK FOR VALID DEVICE # (TT2=DXCA)

THE ORDER IN WHICH THE FIRST TUMBLE TABLE ENTRY IS PROCESSED
IS IMPRTANT. CHECK FIRST FOR SYSTEM RESET, PARITY ERRORS,
ETC. THEN CHECK FOR CHANNEL INITIATED SEQUENCE, CHANNEL END,
CONTROL UNIT END. (PERFORM ACCORDING TO TT1 (DXDS))

- * CHECK FOR A SYSRST IN TT1 (DXDS)
IF YES, GO TO SYSTEM RESET (PSYSRT:)
- * SELECTIVE RESET? (DXDS)

IF YES, GO TO PSELRT:
 * CHECK FOR INTERFACE DISCONNECT? (DXDS)
 IF YES, GO TO PINDSC:
 * NON-EXISTANT MEMORY? (DXDS)
 IF YES, GO TO PNXM:
 * STATUS ACCEPTED? (DXDS)
 IF YES, GO TO PESEND:
 * PARITY ERROR? (DXDS)
 IF YES, GO TO PPARER:
 * CHANNEL INITIATED SEQUENCE? (DXDS)
 IF YES, GO TO TCHIS: (EVERYTHING OK UP TO THIS POINT).
 * CHANNEL END? (DXDS)
 IF YES, GO TO TCHEND:
 * CONTROL UNIT END? (DXDS)
 IF YES, GO TO TCUEND:
 * INITIAL SELECTION SEQUENCE REJECT? (DXDS)
 NO? IGNORE ENTRY...TREAT AS STACK STATUS
 GET NEXT TT ENTRY AND DO REST OF ABOVE..... IF, HOWEVER,
 INITIAL SELECTION SEQ WAS REJECTED, ENTER A QUEUE CONTROL
 UNIT END TO 360 (QUEUE A CONTROL UNIT END(QCUE=10) TO SCMD OF PROPER DEVICE
 STATUS BUFFER TABLE)
 -YOU WILL STAY IN THIS SECTION OF CODE UNTIL ALL TT ENTRIES
 HAVE BEEN PROCESSED. WHEN THERE ARE NO MORE TT ENTRIES TO
 PROCESSJUMP TO DXEXEC:.

---DXEXEC: OVERVIEW (CMD DISPATCH SECTION
 OF THE DXISR) ---

DXISR HAS THE PRIORITY LEVEL AT 7 PREVENTING ANY MORE INTERRUPTS.
 - IT HAS PROCESSED ALL THE TT ENTRIES BEFORE GETTING INTO THIS CODE

REMEMBER; THROUGHOUT THE DXISR INTERRUPT SERVICE ROUTINE,
 AS A RESULT OF SERVICING TT ENTRIES, THE PROGRAM HAS
 BEEN SETTING OR PUTTING SPECIFIC #'S IN THE DEVICE'S
 STATUS BUFFER AREA. THESE COMMANDS OR WHATEVER WERE
 BEING QUEUED FOR DXEXEC: PROCESSING. HOPEFULLY, AS THE TT
 WAS SERVICED SOME OF THESE WERE CANCELLED OR CHANGED TO
 REFLECT THE TRUE STATUS THAT MUST BE PRESENTED TO THE 360
 CHANNEL. (I KNOW THAT MAY BE DIFFICULT TO REMEMBER). WELL,
 NOW IS THE TIME TO PROCESS THESE QUEUED COMMANDS.
 YOU CAN EXIT FROM THE DXISR: BY SEVERAL PATHS; EXECUTING
 A COMMAND, SEND "ATTENTION" COMMAND CHAINING, OR A
 SYSTEM RESET, INTERFACE DISCONNECT, ETC..

THE DXEXEC: ROUTINE FIRST DETERMINES WHETHER THE CHANNEL
 WAS SELECTOR OR MULTIPLEXER (CHTYPE = "M" OR "S")

TYPICAL SELECTOR COMMANDS (FOR EACH DEVICE *)

WRITE FULL BUFFER	(SCMD = 1)
READ MANUAL INPUT	(SCMD = 2)
ENDING SEQUENCE	(SCMD = 3)
SENSR COMMAND	(SCMD = 4)
WRITE LINE ADDRESS	(SCMD = 5)
READ FULL BUFFER	(SCMD = 6)
ERASE COMMAND	(SCMD = 7)
CONTROL UNIT END	(SCMD =10)
SEND ATTN TO 360	(SCMD =11)
READ SHORT MANUAL INPUT	(SCMD =12)

TYPICAL MULTIPLEXER COMMANDS (FOR EACH DEVICE *)

WRITE FULL BUFFER	(SCMD = 1)
READ MANUAL INPUT	(SCMD = 2)
ENDING SEQUENCE	(SCMD = 3)
SENSE COMMAND	(SCMD = 4)
WRITE LINE ADDRESS	(SCMD = 5)
READ FULL BUFFER	(SCMD = 6)
ERASE COMMAND	(SCMD = 7)
CONTROL UNIT END	(SCMD =10)
SEND ATTENTION	(SCMD =11)
READ SHORT MANUAL INPUT	(SCMD =12)

----SELECTOR/MULTIPLEXER COMMAND DESCRIPTION ----

---- SELECTOR CHANNEL ----

SEX: IS THERE ANY COMMANDS TO EXECUTE (PER DEVICE)? IF NO, CHECK FOR COMMAND CHAINING; IF YES, EXIT FROM THE DXISR - WAIT FOR THE INTERRUPT (REMEMBER, YOU MUST EXIT IN ORDER TO DROP THE PROCESSOR LEVEL). RESULTANT DXISR INTERRUPT WILL PROCESS NEW TT ENTRIES.

IF CMDCHF = 0 CHECK TO SEE OF THE ATTENTION FLAG (SRDRQ) FOR THAT DEVICE IS SET. IF YES, QUEUE A "SEND ATTENTION" (SCMD=11). IF NO, RETURN TO DXEXEC AN REPEAT FOR NEXT DEVICE - REPEAT UNTIL ALL DEVICES HAVE BEEN SERVICED BEFORE EXITING FROM DXISR.

IF THERE WAS A COMMAND TO EXECUTE (SCMD=XX); GO TO THAT ROUTINE SPECIFIED BY THE COMMAND. WHEN COMPLETE...EXIT FROM DXISR

-----DESCRIPTION OF COMMAND ROUTINES (SELECTOR)-----

WRITE LINE ADDRESS
WRITE FULL BUFFER

SWRITE: SET UP THE ADDRESS OF INPUT BUFFER AREA (SINBF) INTO DXBA
 SUBTRACT PHYSICAL OFFSET
 . SET BYTE COUNT IN DXBC
 . SET DEVICE ADDRESS IN DXCA
 . SAVE COMMAND (SLCMD (----- SCMD)
 . CLR SSENSE
 . SET DEV ACTIVE FLAG (DXACT)
 . SET INPUT FUNCTION & GO IN DXCS
 . EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES

THE SAVING OF SLCMD SIGNIFIES TO THE PRESENT ENDING
 SEQUENCE (PESEND) THAT IT MUST FORMAT THE DISPLAY (DISCTL)

READ COMMAND (READ FULL BUFFER)

SREAD: SET UP THE ADDRESS OF THE OUTPUT BUFFER AREA (SOUTB)
 INTO DXBA. SUBTRACT PHYSICAL OFFSET.
 . SET BYTE COUNT IN DXBC
 . SET DEVICE ADDRESS IN DXCA
 . CLR SSENSE
 . SET DEV ACTIVE FLAG (DXACT)
 . SET OUTPUT FUNCTION & GO IN DXCS
 . EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES.

READ MANUAL INPUT
READ SHORT MANUAL INPUT

SSRMI: IS IT FRIEND? IF YES, TREAT AS READ FULL BUFFER
 (SREAD:)
 DID YOU SPECIFY A READ REQUEST? NO? ASSUME THE
 360 GAVE AN UNSOLICITATED REQUEST (POLL) AND SEND BACK
 AN ENDING SEQUENCE (ESEQ:)

IF READ REQUEST WAS SET-PROCEED -
 CLR SDRQ
 SAVE LAST COMMAND
 SET UP STARTING ADDRESS - MOVE SMINS TO DXBA
 SUBTRACT PHYSICAL OFFSET FROM DXBA
 CALCULATE BYTE COUNT AND SET DXBC

IF BYTE COUNT IS ERRONEOUS - JUST SEND AN ENDING SEQUENCE
 COMPUTE DEVICE ADDRESS AND SET DXCA
 CLR SSENSE
 SET DEVICE ACTIVE FLAG (DXACT)
 SET OUTPUT FUNCTION AND GO IN DXCS
 EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES.

PRESENT ENDING STATUS TO CHANNEL

ESEQ: QUEUE CE & DE TO SSTAT
 CALCULATE DEVICE ADDRESS AND SET DXCA

CHECK FOR UNIT CHECK BIT SET. IF YES, QUEUE SSTAT WITH
 UNIT CHECK ONLY

IF NO, MOVE SSTAT TO DXOS
 SET STATUS FUNCTION & GO TO DXCS
 SET DEVICE ACTIVE FLAG (DXACT)
 EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES.

PRESENT CONTROL UNIT END

CONJUNE: QUEUE A CONTROL UNIT TO SSTAT
 CALCULATE DEVICE ADDRESS AND SET DXCA
 CHECK FOR UNIT CHECK BIT SET
 IF YES, QUEUE SSTAT WITH UNIT CHECK ONLY
 IF NO, MOVE SSTAT TO DXOS
 SET STATUS FUNCTION & GO TO DXCS
 SET DEVICE ACTIVE FLAG (DXACT)
 EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES.

ERASE COMMAND

ERASCM: MOVE AN EBCDIC SPACE THROUGHOUT OUTPUT DATA BUFFER (SOUTB)
 CLEAR CURSOR POSITION (SCURS)
 CLEAR SSENSE
 QUEUE A CE & DE TO SCMD (CRUNCH WHATEVER WAS IN SCMD)
 DO AN ENDING SEQUENCE - (ESEQ:)

SENSE COMMAND

SENSCM: MOVE THE ADDRESS OF THE SENSE BYTE (SSENSE) TO DXBA
 COMPUTE DEVICE ADDRESS AND SET DXCA
 SET UP TO SEND ONE BYTE TO DXBC
 SET DEVICE ACTIVE FLAG (DXACT)
 EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES

----- MULTIPLEXER CHANNEL -----

ME: IS COMMAND CHAINING SPECIFIED? (CMDCHF) IF YES, EXIT FROM
 DXISR TO ALLOW PSW = 0 IF NO, PICK UP LAST DEVICE ADDRESS
 THAT HAS A COMMAND EXECUTED - HAS IT BEEN EXECUTED? (TYPICALLY
 SYSTEM RESET, SELECTIVE RESET, INTERFACE DISCONNECT,
 STATUS ACCEPTED, CHANNEL END, OR CONTROL UNIT END WILL
 TERMINATE DXACT IN A COMMAND SEQUENCE)

GO TO DEVICE AND FIND OUT IF THERE IS A JOB TO DO
 IF NOT, QUEUE "ATTENTION" IFF ATTENTION IS REQUESTED (SRDRG=1)

-GO EXECUTE COMMAND.

-----DESCRIPTION OF COMMAND ROUTINES (MULTIPLEXER)-----

THOSE THAT ARE COMMON TO THE SELECTOR CHANNEL WILL NOT
BE EXPLAINED HERE - REFER BACK TO SELECTOR

WRITE FULL BUFFER

MWRITE: IS THERE A WRITE IN PROGRESS? (SRBYTC)
IF NO, SET UP DXBA (DXBA <----- SUBFA)
SET UP BYTE COUNTER (SRBYTC)
SET UP DEVICE ADDRESS IN DXCA
SET UP FOR 4 BYTES MAXIMUM TRANSFER IN DYBC
CLR SSENSE
SAVE COMMAND (SLCMD <----- SCMD)
SET DEVICE ACTIVE (DXACT)
SET INPUT FUNCTION & GO IN DXCS
EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES

IF THERE WAS A WRITE IN PROGRESS JUST CONTINUE AS ABOVE
UNTIL SRBYTC = 0, THEN SET UP TO MAXIMUM INPUT BUFFER
SIZE.

SRBYTC IS DECREASED BY THE FOLLOWING TT ENTERED
ROUTINE - (PREPARE CONTROL UNIT END (PCUEND))

SBUFA IS INCREASED BY THE SAME ROUTINE (PCUEND)

A 360 WRITE (MUX) WILL TRANSFER 4 BYTES AT A TIME

THE SAVING OF SLCMD SIGNIFIES TO THE PRESENT ENDING
SEQUENCE (PESEND) THAT IT MUST FORMAT THE DISPLAY
(DISCTL)

READ COMMAND

MREAD: SAME BASICALLY AS MWRITE EXCEPT IT USES SOUTB AND SETS
OUTPUT FUNCTION & GO IN DXCS

READ MANUAL INPUT COMMAND

MSRMI: FRIEND OR 2848
IF FRIEND--JUMP TO "READ FULL BUFFER" (MREAD:)
IF 2848, WAS READ REQUESTED ? NO- ASSUME NOP AND
QUEUE AN ENDING SEQUENCE TO CHANNEL (ESEQ:)

IF READ REQUESTED = YES (SRDRQ =1) SAVE CMD FOR DISPLAY
CONTROL (SLCMD)
COMPUTE ADDRESS OF OUTPUT BUFFER
COMPUTE THE BYTE COUNT
GO TO READ (MREAD:)

NOTE: AFTER TRANSFER OF THE 4 BYTES, THE DXBC WILL DECREMENT TO ZERO
CREATING A CONTROL UNIT END TT ENTRY (PCUEND:)
SRBYTC WILL BE DECREMENTED BY 4 AND SBUFA WILL BE

INCREMENTED BY 4-- THIS APPLIES TO ALL THE SELECTOR OF
 MULTIPLEXER READ OR WRITES IF THE DX HARDWARE IS
 FUNCTIONING CORRECTLY.

---MISCELLANEOUS ROUTINES ---

ASCDMP: THESE ROUTINES SPIT OUT THE CHARACTER
 EBCDMP: EQUIVALENT OF THE ORIGINAL OCTAL BYTE
 HEXDMP: IN ASCII, EBCDIC, HEXIDECIMAL, OR OCTAL..
 OCTDMP:

DISPLAY CONTROL ROUTINE

DISCTL: WAS IT A READ MANUAL INPUT COMMAND (SLCMD=2) IF YES, PICK
 UP SMINS. BACK UP. BLANK CHARACTER, SAVE SCURS & RETURN

SMINS: LOADED IN INPUT COMMAND (ENTER DATA ON A 7260 SCREEN)
 SMINS: USED IN READ MANUAL INPUT COMMAND
 SMINS: USED IN PERFORM READ MANUAL COMMANDS

WAS IT A SHORT READ MANUAL INPUT (SLCMD=12)
 IF YES, JUST RETURN

IF NEITHER, THE COMMAND MUST HAVE BEEN A 360 WRITE.

WAS IT FRIEND OR 2848?
 IF FRIEND AND NOT SEPARATE I/O BUFFERS (IOBUF=0)
 COPY INPUT BUFFER TO OUTPUT BUFFER
 IF FRIEND AND SEPARATE I/O BUFFERS (IOBUF=1)
 DON'T COPY INPUT BUFFER TO OUTPUT BUFFER

IF 2848, GET ADDRESS OF START OF INPUT (SINBF)
 WAS THE LAST CMD A WRITE LINE ADDRESS? (SLCMD=5)

DX ABORT

DXAB:

CLEAR DX INTERRUPT ENABLE TO PREVENT ANY MORE INTERRUPTS
 SET THE DXABLE FLAG TO ABORT
 EXIT FROM DXISR
 (TYPICALLY CAUSED BY A SYSTEM ERROR (NON EXISTANT MEMORY,
 INVALID COMMAND)
 -----DEVICE STATUS TABLE FLAGS-----

DESCRIPTION OF THE DEVICE STATUS TABLE FLAGS. (THERE IS ONE FULL SET PER SPECIFIED DEVICE).

- 1) THEY ARE BROKEN DOWN TO THEIR POSSIBLE CONTENTS
- 2) HOW THEY ARE USED BY THE PERTINENT ROUTINE (CLOSEST SIGNIFICANT ROUTINE)
- 3) A LISTING OF WHAT ROUTINE CLEARS THE FLAG, OR SET THE FLAG, OR USES THE FLAG.

THESE FLAGS ARE USED ACTIVELY BY THE PROGRAM TO KEEP TRACK OF SIGNIFICANT EVENTS.

SCMD (0)

```
SCMD  ---- IDLE = 0 (NO COMMAND)
      ---- SWRITE: & MWRITE: = 1
      <---- SRMI: & MRMI:      = 2
      <---- CEDE              = 3 *
      <---- SENSCHM:         = 4
      <---- SWRITE: & MWRITE: = 5
      <---- SREAD: & MREAD:  = 6
      <---- ERASCHM:        = 7
      <---- QCUE             = 10 *
      <---- "ATTENTION"     = 11 *
      <---- SSRMI: & MSRMI:  = 12
```

* PROGRAM GENERATED COMMANDS- THE REMAINING WERE AS A RESULT OF IBM 360/370 COMMANDS (TT2 ENTRIES)

HOW USED

PESEND: USED TO QUEUE INFORMATION IN SRDRQ & SLCMD FOR LATER PROCESSING
 MEX: & SEX: USED TO PERFORM THE 360 CMD - SET UP DX AND DO IT
 MWRITE: & SWRITE: USED TO SAVE LAST COMMAND IN SLCMD FOR LATER PROCESSING
 MSRMI: & SSRMI: USED TO SAVE LAST COMMAND IN SLCMD FOR LATER PROCESSING.

SET UP IN

TISSRJ:
PINDSC:
PVISS:
PCHEHC:

USED IN

PESEND:
SEX: & MEX:
SWRITE: & MWRITE:
SSRMI: & MSRMI:

CLEARED IN

CDEVST: (RUN:, ENABLE:.,
KILL:, PSYSRT:., PINDSC:.,
PESEND:.)

ERASCM:

SSENSE (2)

SSENSE	-----	INTREQ	=100
	-----	BUSOUT	=40
	-----	SCMDRJ	=200
	-----	EQPCHK	=20

HOW USED

USED BY 360 WHEN REQUESTING A SENSE CMD
IE. SENSCH: MOV #SSENSE,DXBA

SET UP IN

PHIS:
PCHENC:

USED IN

SENSCH:

CLEARED IN

RUN:
ENABLE:
PSYSRT:
PSELRT:
SWRITE:
SREAD:
ERASCM:
MWRITE:
MREAD:

SSTAT (3)

SSTAT	-----	UCHK	= 2
	-----	CE!DE	= 14
	-----	ATTN	= 200
	-----	CUE	=40

HOW USED

USED BY 360 WHEN REQUESTING STATUS WITH EXCEPTION
OF THE ASYNCHRONOUS PRESENTING OF STATUS (ATTN) TO
THE 360.
IE. STOUT: MOV SSTAT,DXOS

SET UP IN

USED IN

CLEARED IN

----- PPARER: ESEQ: CONJNE: SATYN:	----- STOUT:	----- PSYSRT: CDEVST:(RUN: ,ENABLE: , KILL: ,PSYSRT: ,PINDSC: , PESEND:)
--	-----------------	---

SCURS (4)

SCURS (---- ANY # FROM 0 TO 479. (CURSOR POSITION)

HOW USED

INPUT: USED TO CALCULATE CURSOR POSITION TO CREATE OUTPUT TABLE
(FOR IBM READ)
MSRMI: & SSRMI: USED TO CALCULATE BYTE COUNT FOR USE JDXBC

USED IN	CLEARED IN
-----	-----
INPJT:	PSYSRT:
SSRMI:	DISCTL:
	ERASCM:

SINBF (6)

SINBF (---- ADDRESS OF DEVICE INPUT DISPLAY BUFFER

HOW USED

DUMP: USED BY PROGRAM DUMP COMMAND TO ASCERTAIN BOUNDARIES
OF THE INPUT BUFFER
DISCTL: USED BY PROGRAM TO CALCULATE BOUNDARIES FOR INPUT BUFFER
MWRITE: & SWRITE: USED BY PROGRAM FOR CALCULATION

SET UP IN	USED IN
-----	-----
INIT: & INT140:	DUMP:
	DISCTL:
	MWRITE: & SWRITE:

SOOTB (10)

SOOTB ----- ADDRESS OF DEVICE OUTPUT/DISPLAY BUFFER

HOW USED

DUMP: USED BY PROGRAM DUMP COMMAND TO ASCERTAIN
BOUNDARIES OF THE OUTPUT BUFFER
INPUT: USED TO CALCULATE START OF DATA LOCATION FOR
LOADING OF THE OUTPUT BUFFER FOR A SUBSEQUENT
IBM READ
DISCTL: USED BY PROGRAM TO CALCULATE BOUNDARIES FOR OUTPUT BUFFER
MREAD: & SSRMI: & MSRMI: USED TO CALCULATE BYTE COUNT FOR
DXBC (IBM READ)
PSYSRT: USED TO CLEAR OUT OUTPUT BUFFER AREA (WITH FILLCH)
ERASCM: USED TO CLEAR OUT BUFFER AREA (WITH EBCDIC SPACE = 100)

SET UP IN

INIT: (INT140:)

USED IN

DUMP:
INPUT:
DISCTL:
SSRMI: & MSRMI:
PSYSRT:
ERASCM:
MREAD:

SBUFA (12)

SBUFA ----- CURRENT BUFFER ADDRESS (FOR MUX CHANNEL ONLY)

HOW USED

MSRMI: & MWRITE: & MREAD: USED TO KEEP TRACK OF CURRENT
BUFFER ADDRESS ,INCLUDING MEMORY MANAGEMENT--
LOADED IN DXBA
ALSO USED TO CALCULATE BYTE COUNT (SBYTC)--
LOADED IN DXBC

SET UP IN

PCUEND:
MWRITE:
MREAD:
MSPMI:

USED IN

MWRITE:
MREAD:
MSPMI:

CLEARED IN

CDEVST:(RUN:
ENABLE: ,KILL: ,PSYSRT:
PINUSC: ,PESEND:)

SONLF (16)

SONLF (---- ONLINE = 0
 ---- OFFLINE = 1

HOW USED

PHIS: IF DEVICE IS OFFLINE-- QUEUE AN INTERVENTION REQUEST
 TO IBM CHANNEL (SSENSE)
 - WHEN CHANNEL TIMES OUT WHEN DX DIDN'T RESPOND -
 IT WILL PROBABLY SEND A SENSE CMD . THEREBY
 READING THE SSENSE

SET UP IN

ENABLE: = 0
KILL: = 1

USED IN

PCHIS:

SRDRQ (17)

SRDRQ (---- READ REQUEST = 1
 ---- CLEARED = 0
 ---- READ REQUEST ACCEPTED(360) = 2

HOW USED

MEX: & SEX: USED TO FORCE AN ATTENTION (11) RESPONSE
TO IBM CHANNEL
MSRMI: & SSRMI: USED TO DETERMINE IF AN UNSOLICITED
IBM READ HAD TRANSPIRED-- IF YES, QUEUE AN
ENDING SEQUENCE

SET UP IN

INPUT: = 1
PESEND: = 1

USED IN

SEX: & MEX:
SSRMI:

CLEARED IN

RUN:
PSYSRT:
SSRMI:

SMINS (20)

SMINS <---- ADDRESS OF THE DATA POINTER (MANUAL INPUT READ)

HOW USED

DISCTL: USED TO CALCULATE THE RELATIVE CURSOR POSITION (SCURS)
MSRMI: & SSRMI: USED FOR STARTING DATA ADDRESS FOR DXBA

SET UP IN

INPUT:

USED IN

DISCTL:
SSRMI: & MSRMI:

CLEARED IN

RUN:
PSYSRT:

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 37
PROGRAM DESCRIPTION

SEG 0037
SEG 0037

1945
1946
1947

.TITLE MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
.ENABL ABS
.ENABL AMA

.SBTTL PROGRAM EQUATES AND DEVICE ASSIGNMENTS

SYSTEM EQUATES

1948		...		
1949		...		
1950		...		
1951		...		
1952	000000	R0	=	%0
1953	000001	R1	=	%1
1954	000002	R2	=	%2
1955	000003	R3	=	%3
1956	000004	R4	=	%4
1957	000005	R5	=	%5
1958	000006	R6	=	%6
1959	000006	SP	=	%6
1960	000007	PC	=	%7
1961	177776	PSW	=	177776
1962	172340	KISAR0	=	172340
1963	172356	KISAR7	=	172356
1964	172300	KISOR0	=	172300
1965	177572	MMSR0	=	177572

TELETYPE CHARACTER EQUATES

1970		...			
1971	000015	CR	=	15	: CARRIAGE RETURN
1972	000012	LF	=	12	: LINE FEED
1973	000040	SPACE	=	40	: SPACE CHARACTER
1974	000003	CTL.C	=	3	: CONTROL C
1975	000020	CTL.P	=	20	: CONTROL P
1976	000021	CTL.Q	=	21	: CONTROL Q
1977	000023	CTL.S	=	23	: CONTROL S
1978	000025	CTL.U	=	25	: CONTROL U
1979	000177	RUBOUT	=	177	: RUBOUT

```

1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991 000000
1992 000001
1993 000002
1994 000003
1995 000004
1996 000006
1997 000010
1998 000012
1999 000014
2000 000016
2001 000017
2002 000020
2003
2004
2005
2006
2007
2008
2009
2010
2011 000000
2012 000003
2013 000004
2014 000005
2015 000003
2016 000010
2017 000003
2018 000025
2019 000100
2020 000112
2021 000152
2022 000050
2023 000014
2024 000740
2025 001000
2026
2027
2028
2029
2030
2031 000001
2032 000002
2033 000005
2034 000006
2035 000012

*****
*
*   DEVICE BUFFER LAYOUT   (1 PER DEVICE)
*
*****
*   LOC   0-61 = DEVICE STATUS TABLE
*   LOC   62-543 = DEVICE INPUT BUFFER
*   LOC   554-1023 = DEVICE OUTPUT/DISPLAY BUFFER
*
*****
*   LAYOUT OF DEVICE STATUS TABLE
*
SCMD      =      0      ;CURRENT DEVICE COMMAND
SLCMD     =      1      ;LAST COMMAND, IF WRITE
SSENSE    =      2      ;DEVICE SENSE BYTE (NOTE -- MUST BE EVEN BYTE LOCATION)
SSTAT     =      3      ;DEVICE STATUS
SCURS     =      4      ;CURSOR POSITION
SINBF     =      6      ;ADDRESS OF DEVICE INPUT BUFFER
SOUTB     =     10      ;ADDRESS OF DEVICE OUTPUT BUFFER
SBUFA     =     12      ;CURRENT BUFFER PTR
SRBYTC    =     14      ;REMAINING BYTE COUNT
SONLF     =     16      ;DEVICE ONLINE - INDICATOR 0=ON-LINE 1=OFF-LINE
SRDRQ     =     17      ;READ MANUAL INPUT REQUEST -- IF NON-ZERO
SMINS     =     20      ;START OF MANUAL INPUT DATA

*
*   LOCATIONS 22-77 ARE AVAILABLE FOR EXPANSION PURPOSES
*
*
*   OTHER   DX EQUATES
*
DEV        =     R0      ;CURRENT DEVICE NUMBER
DTAB       =     R3      ;ADDRESS OF CUR DEV STATUS TABLE
TT1        =     R4      ;TUMBLE TABLE ENTRY 1
TT2        =     R5      ;TUMBLE TABLE ENTRY 2
CEDE       =      3      ;CHAN END & DEV END
QCUE       =     10      ;CODE TO QUE CONTROL UNIT END
NOP        =      3      ;NOP COMMAND
NEWLINE    =     25      ;NEW LINE CHARACTER
EBCDSP     =    100      ;EBCDIC SPACE CODE
SMI        =    112      ;START OF MESSAGE INDICATOR
EOM        =    152      ;END OF MESSAGE INDICATOR
LINSZ     =     40      ;NUMBER OF CHARACTERS PER LINE OF 2260 DISPLAY
NOLIN     =     12      ;NUMBER OF LINES PER 2260 DISPLAY
DISPSZ    =    NOLIN*LINSZ ;NUMBER OF CHARACTERS ON THE DISPLAY
TTSIZE    =     512     ;NUMBER OF ENTRIES IN TUMBLE TABLE

*
*   360 COMMAND EQUATES
*
CMWRT     =      1      ;WRITE DATA (FROM 360 TO PDP-11)
CMRMI     =      2      ;READ MANUAL INPUT (PDP-11 TO 360)
CMWTLA    =      5      ;WRITE LINE ADDRESS (360 TO PDP-11)
CMREAD    =      6      ;READ FULL BUFFER (PDP-11 TO 360)
CMSRMI    =     12      ;SHORT READ MANUAL INPUT (PDP-11 TO 360)

```


2092	000040	CUE	=	40	:	CONTROL UNIT END
2093	000020	BSY	=	20	:	BUSY
2094	000010	CF	=	10	:	CHANNEL END
2095	000004	DE	=	4	:	DEVICE END
2096	000002	UCHK	=	2	:	UNIT CHECK
2097	000001	UEXP	=	1	:	UNIT EXCEPTION
2098		:				
2099		:				
2100		:				
2101		:				
2102	000200	SCMDRJ	=	200	:	COMMAND REJECT
2103	000100	INTREQ	=	100	:	DEVICE OFF-LINE - INTERVENTION REQ
2104	000040	BUSOUT	=	40	:	BUS OUT -- PARITY ERROR DURING CHS
2105	000020	EQPCHK	=	20	:	EQUIPMENT CHECK - PARITY ERROR DUR DATA TRANS

2848 SENSE BYTE (SENSE) DEFINITION

```

2106          .SBTTL  TRAP CATCHER
2107          :
2108          :      THE TRAP CATCHER IS LOADED INTO LOW CORE
2109
2110
2111          .=0
2112 000000 000000  HALT          ;FOR MEMORY MANAGEMENT
2113 000002 000000  HALT
2114          000176  .REPT  176      ;TRAP CATCHER
2115
2116          .WORD  .+2
2117  HALT
2118  .ENDR
2119  .=200
2120 000200 000130 001000  JMP  START      ;ESTABLISH LOC 200 STARTING ADDRESS
2121
2122

```

2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177

001003 001000 000402
001002 005037 013132
001006 012706 012636
001012 000005
001014 012700 000060
001020 012720 010744
001024 012720 000340
001030 012720 011232
001034 012710 000340
001040 012737 011672 000004
001046 012737 000340 000006
001054 012737 011700 000250
001062 012737 000340 000252
001070 012700 012636
001074 012701 000272
001100 105020
001102 005301
001104 001375
001106 012737 012640 012742
001114 012737 012744 013046
001122 012737 012744 013050

.SBTTL PROGRAM START-UP SEQUENCES
NORMAL SYSTEM START LOCATION --1000
START: BR .=1000 SYSINT ;NORMAL START UP
RESTART ADDRESS -- REENTER ALL PARAMETERS -- 1002
RSTART: CLR FTIMFL ;RESET FIRST TIME FLAG TO FORCE PARAMETER REENTRY
SYSTEM GENERALIZED INITIALIZATION PROCEDURE
SET-UP STACK POINTER
TRAP/VECTOR AREA
SYSTEM GENERATED TRAPS
TELETYPE (CONSOLE) VECTORS + STATUS REGISTERS
CLEAR ALL LIVE SYSTEM VARIABLES
SET UP TELETYPE INPUT / OUTPUT BUFFERS
SYSINT: MOV #SSTACK,SP ;SET UP THE STACK POINTER
RESET ;RESET ALL DEVICES
SET UP CONSOLE VECTORS
MOV #60,R0 ;START OF CONSOLE VECTORS
MOV #TKIN,(R0)+ ;BEG OF TELE INPUT ISP
MOV #340,(R0)+ ;NEW PROC STATUS
MOV #PISR,(R0)+ ;BEG OF TELE PRINT ISR
MOV #340,(R0) ;NEW PROC STATUS
SET UP MISC TRAPS
MOV #MTO,4 ;MEMORY TIME OUT TRAP
MOV #340,6
MOV #MMERR,250 ;MEMORY MANAGEMENT ERROR
MOV #340,252
CLEAR ALL VARIABLES
MOV #VSTRT,R0 ;START OF VARIABLES
MOV #VEND-VSTRT+2,R1 ;# OF VARIABLES TO CLEAR
10\$: CLRB (R0)+ ;CLEAR A BYTE
DEC R1 ;DONE?
BNE 10\$;NO, CONTINUE CLEARING
SET UP TELE BUFFER POINTERS
MOV #TBUF,TPTR ;TELE INPUT POINTER
MOV #PBFS,PFPTTR ;TELE OUT FETCH PTR
MOV #PBFS,PPPTTR ;TELE OUT PUT PTR
ENABLE TTY

F04

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 44
PROGRAM START-UP SEQUENCES

SEQ 0044
SEQ 0044

2178					:			
2179	001130	052777	000100	011316		BIS	#100.2TKS	:ENABLE TELETYPE INPUT
2180	001136	052777	000100	011314		BIS	#100.2TFS	:ENABLE TTY OUTPUT INTERRUPTS
2181	001144	005037	177776			CLR	PSW	:CLEAR THE PROCESSOR STATUS WORD
2182	001150	005737	013132			TST	FTIMFL	:FIRST TIME THROUGH? (MUST PARAMETERS BE REENTERED)
2183	001154	001402				BEG	GETPRM	:YES, FORCE USER TO ENTER ALL PARAMETERS
2184	001156	000137	002026			JMP	INI	:NO, RESTART TEST USING SAME PARAMETERS

2185
2186
2187
2188
2189
2190 001162 004737 011330
2191 001166 004137 011472
2192 001172 013134
2193
2194
2195
2196
2197
2198
2199 001174 005037 013132
2200 001200 012737 176200 C12516
2201 001206 004137 002632
2202 001212 013175
2203 001214 001260
2204 001216 120427 000015
2205 001222 001364
2206 001224 020327 176200
2207 001230 002761
2208 001232 020327 177000
2209 001236 003356
2210 001240 032703 000037
2211 001244 001353
2212 001246 012737 001174 000004
2213 001254 00337 012516
2214 001260 005077 011232
2215
2216
2217
2218
2219
2220
2221
2222 001264 012737 011742 000004
2223 001272 012737 000300 012520
2224 001300 004137 002632
2225 001304 013232
2226 001306 001344
2227 001310 120427 000015
2228 001314 001363
2229 001316 020327 000300
2230 001322 002760
2231 001324 020327 000770
2232 001330 003355
2233 001332 032703 000001
2234 001336 001352
2235 001340 010337 012520
2236
2237
2238
2239
2240

.SBTTL TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)
:
HERE WE START GATHERING THE TEST INFORMATION
PRINT START-UP HERALD MESSAGE
GETPRM: JSR PC,CRLF :RESTORE THE CARRIAGE
JSR R1,IESG :PRINT START-UP MESS
.WORD STMSG
:
GET DX11 UNIBUSS ADDRESS (OCTAL ADDRESS INPUT)
VALID UNIBUS ADDRESSES (176200 - 177000)
DEFAULT UNIBUS ADDRESS 176200
NEWPRM: CLR FTIMFL :RESET FIRST TIME PARAMETERS FORCE ALL PARAMS TO BE ENTE
MOV #176200,UNADDR :SET UP DEFAULT ADDRESS
JSR R1,INOCT :GET UNIBUS ADDRESS
.WORD UNMSG
.WORD 5\$:ADDRESS OF DEFAULT ROUTINE
CMPB R4,#CR :WAS LINE DELIMITED PROPERLY?
BNE NEWPRM :NO, TELL HIM TO REENTER
CMP R3,#176200 :VALID UNIBUS ADDRESS? BETWEEN 176200 AND 177000
BLT NEWPRM :NO, GET AGAIN
CMP R3,#177000 :UNIBUS ADDRESS GT 177000?
BGT NEWPRM :YES, ERROR -- REENTER
BIT #37,R3 :MAKE SURE 40 OCTAL WORD BOUNDARY
BNE NEWPRM :ILLEGAL REENTER
MOV #NEWPRM,4 :SET UP TRAP OUT TO VALIDATE ADDRESS
MOV R3,UNADDR :SAVE UNIBUS ADDRESS
CLR UNADDR :VALIDATE THE UNIBUS ADDRESS
:
GET THE DX11 INTERRUPT VECTOR ADDRESS (OCTAL ADDRESS INPUT)
VALID VECTOR ADDRESSES (300 - 770)
DEFAULT VECTOR ADDRESS 300
10\$: MOV #UNTRP,4 :RESTORE MEMORY TIME-OUT TRAP
MOV #300,VECTAD :SET UP DEFAULT VECTOR ADDRESS
JSR R1,INOCT :GET VECTOR ADDRESS
.WORD VECTMS
.WORD 20\$:ADDRESS OF THE DEFAULT ENTRY
CMPB R4,#CR :WAS LINE DELIMITED PROPERLY?
BNE 10\$:NO, REENTER
CMP R3,#300 :CHECK VECTOR ADDRESS BETWEEN 300 AND 770
BLT 10\$:TOO LOW GIVE AN ERROR AND REENTER
CMP R3,#770 :LT 770?
BGT 10\$:YES, REENTER
BIT #1,R3 :WORD ADDRESS?
BNE 10\$:NO, REENTER
MOV R3,VECTAD :SAVE IT
:
GET STARTING AND ENDING DEVICE CHANNEL ADDRESSES (HEX INPUT)
VALID DEVICE CHANNEL ADDRESSES (00 - FF)
DEFAULT DEVICE CHANNEL ADDRESS 10,10

```

2241
2242
2243
2244
2245
2246 001344 012737 000020 012522 20$: MOV #20,SDEV ;DEFAULT TO HEX ADDRESS 10
2247 001352 012737 000020 012524 MOV #20,EDEV ;
2248 001360 004137 002624 JSR R1,INHEX ;GET DEVICE ADDRESSES IN HEX
2249 001364 013300 .WORD DEVMS ;
2250 001366 001460 .WORD NEWP10 ;ADDRESS OF THE DEFAULT ROUTINE
2251 001370 010337 012522 MOV R3,SDEV ;SAVE START DEV ADDR
2252 001374 010337 012524 MOV R3,EDEV ;
2253 001400 005703 TST R3 ;BE SURE POSITIVE
2254 001402 100760 BMI 20$ ;
2255 001404 020327 000377 CMP R3,#377 ;AND NOT GREATER THAN 377 -- HEX FF
2256 001410 003355 BGT 20$ ;ILLEGAL ENTRY
2257 001412 120427 000054 CMPB R4,#'.' ;MORE THAN ONE DEV? (COMMA, PARAMETER DELIMITER)
2258 001416 001015 BNE 30$ ;
2259 001420 004737 011604 JSR PC,CHTB ;GET ENDING DEVICE
2260 001424 010337 012524 MOV R3,EDEV ;SAVE ENDING ADDRESS
2261 001430 023737 012522 012524 CMP SDEV,EDEV ;IS START LT END?
2262 001436 003342 BGT 20$ ;YES, ERROR
2263 001440 163703 012522 SUB SDEV,R3 ;MORE THAN 8 DEVICES?
2264 001444 020327 000007 CMP R3,#7 ;
2265 001450 003335 BGT 20$ ;YES, ERROR
2266 001452 120427 000015 30$: CMPB R4,#CR ;WAS DEVICE ADDRESSES DELIMITED PROPERLY?
2267 001456 001332 BNE 20$ ;NO, REENTER
2268
2269
2270
2271
2272
2273
2274
2275 001460 105037 012526 NEWP10: CLRB CHTYPE ;0 = M, 1 = S
2276 001464 004137 002632 JSR R1,INOCT ;GET CHANNEL TYPE
2277 001470 013344 .WORD CHTYMS ;
2278 001472 001510 .WORD 50$ ;DEFAULT TO SELECTOR CHANNEL
2279 001474 120427 000115 CMPB R4,#'M' ;M? -- MULTIPLEXER CHANNEL --
2280 001500 001414 BEQ 60$ ;YES, MULTIPLEXER CHANNEL
2281 001502 120427 000123 CMPB R4,#'S' ;S? -- SELECTOR CHANNEL --
2282 001506 001364 BNE NEWP10 ;NOT S OR M -- ERROR
2283 001510 105237 012526 50$: INCB CHTYPE ;SELECTOR CHANNEL
2284 001514 000406 BR 60$ ;GET MEMORY MANAGEMENT FACILITIES
2285
2286
2287
2288
2289
2290
2291 001516 022626 55$: CMP (SP)+,(SP)+ ;DUMP PC AND PSW SAVED BY INTERRUPT
2292 001520 005037 177776 CLR PSW ;TURN DOWN PROCESSOR STATUS
2293 001524 004137 011472 JSR R1,MESG ;PRINT "NO MEM MANAGEMENT AVAIL"
2294 001530 014102 .WORD PNOMM ;
2295
2296
    
```

;ASK TO HAVE QUESTION REENTERED

```

2297
2298
2299
2300
2301
2302
2303 001532 105037 012527
2304 001536 004137 002632
2305 001542 013376
2306 001544 001600
2307 001546 120427 000116
2308 001552 001412
2309 001554 120427 000131
2310 001560 001364
2311 001562 105237 012527
2312 001566 012737 001516 000004
2313 001574 005037 177572
2314
2315
2316
2317
2318
2319
2320
2321 001600 004137 002632
2322 001604 013436
2323 001606 001704
2324 001610 120427 000015
2325 001614 001371
2326 001616 032703 000001
2327 001622 001366
2328 001624 020327 000020
2329 001630 002763
2330 001632 005703
2331 001634 100761
2332 001636 020327 000734
2333 001642 002356
2334 001644 105737 012527
2335 001650 001412
2336 001652 010304
2337 001654 042704 000600
2338 001660 020427 000154
2339 001664 003407
2340 001666 004137 011472
2341
2342
2343
2344 001672 017051
2345 001674 000741
2346
2347 001676 020327 000134
2348 001702 002336
2349 001704 010337 012530
2350
2351
2352

```

: DETERMINE IF MEMORY MANAGEMENT IS TO BE USED
: Y = YES, MEMORY MANAGEMENT TO BE USED
: N = NO, DO NOT USE MEMORY MANAGEMENT
: DEFAULT IS 'N', DO NOT USE MEMORY MANAGEMENT
60\$: CLRB MMRESP ; DEFAULT TO NO MEMORY MANAGEMENT
: JSR R1, INOCT ; GET MEM MANAGEMENT
: .WORD MMRES
: .WORD 70\$; DEFAULT ROUTINE ADDRESS
: CMPB R4, #'N ; N? --DO NOT USE MEMORY MANAGEMENT
: BEQ 70\$; IF EQ, NO MEMORY MANAGEMENT
: CMPB R4, #'Y ; Y? --MEMORY MANAGEMENT TO BE USED
: BNE 60\$; ERROR
: INCB MMRESP ; MEMORY MANAGEMENT SPEC
: MOV #55\$ 4 ; SET UP TRAP TO TEST MEMORY MANAGEMENT
: CLR MMSRD ; CHECK FOR MEMORY MANAGEMENT
:
: GET BUFFER RELOCATION IN OCTAL ,000'S
: VALID RELOCATION ADDRESSES (20 - 700)
: (20000 THROUGH 734000)
: DEFAULT RELOCATION ADDRESS 20 --- (20000)
70\$: JSR R1, INOCT ; GET BUFFER RELOC. IN ,000'S
: .WORD BFAEMS
: .WORD NEWP20 ; ADDRESS OF DEFAULT ROUTINE
: CMPB R4, #CR ; WAS LINE DELIMITED PROPERLY?
: BNE 70\$; NO, REENTER
: BIT #1 R3 ; MUST BE A MULTIPLE OF 2000
: BNE 70\$
: CMP R3, #20
: BLT 70\$; ILLG BUFFER CONST -- LT 20000
: TST R3 ; IS NUMBER NEGATIVE?
: BMI 70\$; YES, REENTER ADDRESS
: CMP R3, #734 ; IS ADDRESS TOO LARGE?
: BGE 70\$; YES, REENTER ADDRESS
: TSTB MMRESP ; WAS MEMORY MANAGEMENT SPECIFIED?
: BEQ 71\$; NO, CHECK FOR 28K
: MOV R3, R4 ; PUT VALUE IN WORK REG
: BIC #600 R4 ; IGNORE ADDRESS EXTENSION BITS
: CMP R4, #154 ; IS IT TOO CLOSE TO 200000 BOUNDARY?
: BLE NEWP20 ; BRANCH IF OK
: JSR R1, MMSG ; PRINT ERROR. CANNOT SET BUFFER SO
: CLOSE TO A 200000 BOUNDARY THAT A CARRY WOULD BE NEEDED TO CHANGE
: THE EXTENDED ADDRESS BITS. THE DX CANNOT WORK ACROSS 200000
: BOUNDARIES.
: .WORD TOOC ; ADDRESS OF TOO CLOSE MESSAGE
: BR 70\$; ASK FOR INPUT AGAIN
71\$: CMP R3, #134 ; NO, IS IT TOO CLOSE TO I/O PAGE?
: BGE 70\$; YES, REENTER THE ADDRESS
NEWP20: MOV R3, BUFREL ; SAVE REL CONST
:
: GET TYPE OF TEST TO BE RUN

J04

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
 CZDXIB.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 48
 TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)

SEQ 0048
 SEQ 0048

```

2353          :
2354          :
2355          :
2356          :
2357          :
2358          :
2359 001710 105037 012532          :
2360 001714 004137 002632          :
2361 001720 013534          :
2362 001722 001752          :
2363 001724 112737 000100 012534 :
2364 001732 120427 000104          :
2365 001736 001433          :
2366 001740 120427 000106          :
2367 001744 001361          :
2368 001746 105037 012534          :
2369 001752 105237 012532          :
2370          :
2371          :
2372          :
2373          :
2374          :
2375          :
2376          :
2377 001756 105037 012533          :
2378 001762 004137 002632          :
2379 001766 013574          :
2380 001770 002026          :
2381 001772 120427 000116          :
2382 001776 001413          :
2383 002000 120427 000131          :
2384 002004 001364          :
2385 002006 105237 012533          :
2386          :
2387          :
2388          :
2389          :
2390          :
2391          :
2392 002012 004137 002624          :
2393 002016 013636          :
2394 002020 002026          :
2395 002022 110337 012534          :
  
```

```

D = 2848 RESPONDER DIAGNOSTIC
F = FRIEND
DEFAULT = 'F', FRIEND
90$: CLR B TSTYP ;RESET TEST TYPE
      JSR R1, INOCT ;GET TEST TYPE
      .WORD TESTMS
      .WORD 100$ ;DEFAULT TO FRIEND
      MOV B #EBCDSP, FILLCH ;FOR 2848 SET FILL CHAR TO EBCDIC SPACE
      CMP B R4, #'D ;D? --2848 RESPONDER DIAGNOSTIC --
      BEQ INIT ;YES, 2848 TEST
      CMP B R4, #'F ;F? -- FRIEND TEST --
      BNE 90$ ;ILLEGAL ENTRY
      CLR B FILLCH ;FRIEND MODE -- DEFAULT FILL CHAR TO NULL
      INCB TSTYP ;SET TEST TO FRIEND
  
```

```

FRIEND TEST ONLY
DETERMINE IF SEPARATE INPUT / OUTPUT BUFFERS ARE TO BE USED
Y = YES, MAINTAIN SEPARATE INPUT / OUTPUT BUFFERS
N = NO, USE SAME BUFFER FOR INPUT AND OUTPUT
DEFAULT IS 'N', NO, USE SAME BUFFER FOR INPUT / OUTPUT
  
```

```

110$: CLR B IOBUF ;0 = NO, 1 = YES
      JSR R1, INOCT ;SEPARATE I/O BUFFERS?
      .WORD FIOMS
      .WORD INIT ;DEFAULT TO NO
      CMP B R4, #'N ;N? -- NO, SAME I/O BUFFER --
      BEQ INIT ;IF EQ, USE SAME I/O BUFFER FOR INPUT AND OUTPUT
      CMP B R4, #'Y ;Y? --YES, SEPARATE I/O BUFFERS--
      BNE 110$ ;ERROR, REQUEST INPUT AGAIN
      INCB IOBUF ;SET SEPARATE I/O BUFFER INDICATOR
  
```

```

FRIEND TEST MODE ONLY
GET BUFFER FILL CHARACTER (HEX INPUT REQUIRED)
ANY VALUE WILL BE ACCEPTED
  
```

```

JSR R1, INHEX ;FILL CHARACTER
.WORD FILLMS
.WORD INIT
MOV B R3, FILLCH
  
```

```

2396 .SBTTL PROGRAM INITIALIZATION
2397
2398 I N I T I A L I Z A T I O N
2399
2400
2401 SET UP ALL DX BUFFERS, MEMORY MANAGEMENT REGISTERS
2402 AND DX REGISTERS
2403
2404 002026 012737 011672 000004 INIT: MOV #MTO,4 ;SET UP MEMORY TIME OUT TRAP
2405 002034 013701 012516 MOV UNADDR,R1 ;SET UP DX UNIBUS ADDRESSES
2406 002040 012702 012464 MOV #DXDS,R2
2407 002044 012703 000015 MOV #13,R3 ;13 ADDRESSES (REGISTERS)
2408 002050 010122 10$: MOV R1,(R2)+ ;SET UP UNIBUS ADDRESS
2409 002052 005721 TST (R1)+ ;INCR TO NEXT DX REGISTER
2410 002054 005303 DEC R3 ;DONE?
2411 002056 001374 BNE 10$ ;NO, SET UP NEXT REGISTER
2412
2413 SET UP DX VECTOR ADDRESS
2414
2415 002060 013701 012520 MOV VECTAD,R1
2416 002064 012721 005334 MOV #DXISR,(R1)+ ;TRAP TO DX ISR
2417 002070 012711 000340 MOV #340,(R1) ;SET UP PROC STATUS AT INTER.
2418
2419 COMPUTE ADDRESSES OF DX BUFFERS
2420 CURRENTLY THIS INCLUDES DATA AREA, TUMBLE TABLE, AND SPW TABLE
2421
2422 002074 005737 012530 TST BUFREL ;WAS BUFFER RELOC SPECIFIED?
2423 002100 001003 BNE 20$ ;YES
2424 002102 012737 000020 012530 MOV #20,BUFREL ;NO, MAKE BUFFERS START AT 20000
2425 002110 013737 012530 013100 20$: MOV BUFREL,PBUFA ;SAVE PHYSICAL ADDRESS
2426 002116 013737 012530 013102 MOV BUFREL,VBUFA ;SAVE VIRTUAL ADDRESS
2427 002124 105737 012527 TSTB MMRESP ;WAS MEMORY MANAGEMENT SPECIFIED?
2428 002130 001436 BEQ 40$ ;NO, SET UP BUFFERS
2429
2430 MEMORY MANAGEMENT WAS SPECIFIED
2431 SET UP KERNEL REGISTERS
2432 0-17777 = PROGRAM
2433 20000-157777 = BUFFERS (VIRTUAL ADDRESSES)
2434 160000-177777 = UNIBUS ADDRESSES
2435 ONLY I SPACE REGISTERS WILL BE USED
2436
2437 002132 012704 172340 MOV #KISAR0,R4 ;I-SPACE PAR
2438 002136 012705 172300 MOV #KISDR0,R5 ;I-SPACE PDR
2439 002142 005024 CLR (R4)+ ;VA 0-17777 = PA 0-17777
2440 002144 012725 077406 MOV #77406,(R5)+ ;64 BLOCKS, UNLIMITED ACCESS
2441 002150 013703 013100 MOV PBUFA,R3 ;PHYSICAL ADDR * 2-6
2442 002154 006303 ASL R3
2443 002156 006303 ASL R3
2444 002160 006303 ASL R3
2445 002162 010324 30$: MOV R3,(R4)+ ;SET UP PA FOR VA 20000-157777
2446 002164 012725 077406 MOV #77406,(R5)+ ;64 BLOCKS, UNLIMITED ACCESS
2447 002170 062703 000200 ADD #200,R3 ;INCREMENT TO NEXT 4K BANK
2448 002174 020427 172356 CMP R4,#KISAR7 ;ALL BUFFER ADDRESSES SET UP?
2449 002200 001370 BNE 30$ ;NO, SET UP NEXT REGISTER
2450 002202 012714 007600 MOV #7600,(R4) ;SET UP UNIBUS ADDRESS REGISTER
2451 002206 012715 077406 MOV #77406,(R5) ;64 BLOCKS, UNLIMITED ACCESS

```

2452 002212 012737 000001 177572
 2453 002220 012737 000020 013102
 2454
 2455
 2456
 2457
 2458
 2459
 2460
 2461
 2462
 2463
 2464
 2465
 2466
 2467
 2468
 2469 002226 013705 013100
 2470 002232 000305
 2471 002234 105005
 2472 002236 006305
 2473 002240 010577 010226
 2474 002244 013701 013102
 2475 002250 000301
 2476 002252 105001
 2477 002254 006301
 2478 002256 010137 013114
 2479 002262 010137 013104
 2480 002266 160537 013104
 2481 002272 062705 003000
 2482 002276 010537 013116
 2483 002302 005000
 2484 002304 120037 012522
 2485 002310 002405
 2486 002312 120037 012524
 2487 002316 003002
 2488 002320 010521
 2489 002322 000402
 2490 002324 012721 000002
 2491 002330 005200
 2492 002332 020027 000400
 2493 002336 001362
 2494
 2495
 2496
 2497 002340 010137 013062
 2498 002344 010137 013060
 2499 002350 012702 001000
 2500 002354 005021
 2501 002356 005302
 2502 002360 001375
 2503
 2504
 2505
 2506
 2507

```

MOV #1,MMSRO ;ENABLE MEMORY MANAGEMENT
MOV #20,VBUFA ;TO BK BANK OR 20000 AND UP

START SETTING UP SPW TABLE
1 ENTRY PER DEVICE (256 DEVICES)
ENTRY DESCRIPTION
----FOR VALID DEVICE NUMBERS
BITS 15-8 = OFFSET TO DST TABLE (PHYSICAL ADDR
7-0 = 0
----FOR INVALID DEVICE NUMBERS
BITS 15-8 = 0
7-0 = 2 -- UNIT CHECK

THIS TABLE IS REFERENCED ON EACH 360 ACTION TO DETERMINE
IF DEVICE NUMBER IS VALID. THIS AUTOMATICALLY DONE
BY THE DX CONTROL UNIT

40$: MOV FBUFFA,R5 ;COMPUTE OFFSET PHYSICALLY
SWAB R5 ;*1000
CLRB R5
ASL R5
MOV R5,DXOS ;OFFSET TO SPW TABLE
MOV VBUFA,R1 ;COMPUTE VIRT ADDR OF SPW TABLE
SWAB R1 ;*1000
CLRB R1
ASL R1
MOV R1,STSPW ;SAVE START OF SPW TABLE
MOV R1,PHYOFF ;COMPUTE THE OFFSET FOR PHYSICAL ADDRESSES
SUB R5,PHYOFF ;VERSES VIRTUAL ADDRESS - FOR MEM MANAGEMENT
ADD #3000,R5 ;COMPUTE THE OFFSET TO THE DST TABLE
MOV R5,DSTOFF ;SAVE OFFSET TO DST TABLE
CLR DEV ;START AT DEVICE 0
50$: CMPB DEV,SDEV ;IS DEVICE NUMBER VALID
BLT 60$ ;NO
CMPB DEV,EDEV
BGT 60$ ;NO
MOV R5,(R1)+ ;VALID DEVICE DST OFFSET TO ENTRY
BR 70$
60$: MOV #UCHK,(R1)+ ;INVALID DEV # UNIT CHECK TO ENTRY
70$: INC DEV ;TO NEXT DEVICE
CMP DEV,#256. ;ALL DEVICES DONE?
BNE 50$ ;NO, SET UP SPW FOR NEXT DEVICE

NEXT SET UP TUMBLE TABLE AND DUPLICATE TUMBLE TABLE

MOV R1,TTADDR ;TUMBLE TABLE ADDRESS
MOV R1,TIPTR ;TUMBLE TABLE FETCH POINTER
MOV #TTSIZE,R2 ;CLEAR T/T + DUPLICATE T/T (WORD POINTER)
80$: CLR (R1)+ ;CLEAR NEXT WORD
DEC R2 ;DONE?
BNE 80$ ;NO, CLEAR NEXT WORD

SET UP DST TABLE
THE DST TABLE IS USED TO VERIFY COMMANDS FROM THE
360, THIS IS DONE BY THE HARDWARE
THE DST TABLE IS A BYTE TABLE, 1 BYTE PER POSSIBLE

```

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

COMMAND 0-255. THE ENTRY IN THE DST TABLE IS
SENT TO THE 360.
THE FOLLOWING ARE A LIST OF VALID COMMANDS AND RESPONSES

COMMAND	RESPONSE	DESCRIPTION
0	0	TEST I/O
1	0	WRITE BUFFER
2	0	READ MANUAL INPUT
3	CE!DE	NO
4	0	SENSE COMMAND
5	0	WRITE LINE ADDRESS
6	0	READ FULL BUFFER
7	0	ERASE COMMAND
12	0	SHORT READ MANUAL INPUT

ALL OTHER COMMANDS ARE RESPONDED WITH UNIT CHECK

```
INIT10: MOV #13,R2 ;NUMBER OF VALID 360 COMMANDS
        MOV #VCMDTB,R3 ;VALID COMMAND TABLE
        MOVB (R3)+,(R1)+ ;TO DST TABLE
        DEC R2 ;DONE?
        BNE INIT10 ;NO, MOVE IN NEXT RESPONSE
100$: MOV #245.,R2 ;MOVE UNIT CHECK TO INVALID ENTRIES
      MOVB #UCHK,(R1)+
      DEC R2
      BNE 100$
```

COMPUTE MAX NUMBER OF DEVICES

```
MOV EDEV,R3
SUB SDEV,R3
INC R3 ;START AT DEVICE NUMERO UNO
MOVB R3,MAXDEV
MOV SDEV,DEVCON ;SET UP DEVICE NUMBER -1
DEC DEVCON
MOV #1,SELDEV ;INIT DEVICE NUMBER FOR MUX AND SEL EXECUTORS
```

NOTE -- THE DEVICE BUFFERS ARE USED BY THE SOFTWARE ONLY TO CONTAIN
POINTERS AND INPUT AND OUTPUT DATA FOR EACH DEVICE;

START SETTING UP DEVICE BUFFERS

```
120$: MOV R1,SDEVTB ;SAVE START OF DEVICE BUFFERS
      CLR DEV ;DEV # 0
      MOV R1,DTAB ;SAVE ADDR OF DEVICE STATUS TABLE
122$: MOV #272.,R2 ;CLEAR DEVICE STATUS TABLE + INPUT BUFFER
      CLR (R1)+
      DEC R2 ;DONE?
      BNE 122$ ;NO, CLEAR NEXT WORD
      MOV DTAB,SINBF(DTAB)
      ADD #62.,SINBF(DTAB) ;COMPUTE ADDRESS OF INPUT BUFFER
      MOV DTAB,SOUTB(DTAB)
      ADD #544.,SOUTB(DTAB) ;COMPUTE ADDRESS OF OUTPUT BUFFER
      MOV #DISPSZ,R2
```

2564	002524	113721	012534	125\$	MOVB	FILLCH, (R1)+	; FILL OUTPUT/DISPLAY BUFFER
2565	002530	005302			DEC	R2	; DONE?
2566	002532	001374			BNE	125\$; NO
2567	002534	005200			INC	DEV	; HAVE ALL DEVICE BUFFERS BEEN SET JP?
2568	002536	120037	013067		CMPB	DEV, MAXDEV	
2569	002542	001346			BNE	120\$; NO, SET UP NEXT DEVICE BUFFERS
2570	002544	013705	013100		MOV	PBUFA, R5	; SET UP EXTENDED ADDRESS BITS
2571	002550	006205			ASR	R5	
2572	002552	006205			ASR	R5	
2573	002554	006205			ASR	R5	
2574	002556	006205			ASR	R5	
2575	002560	042705	177747		BIC	#177747, R5	; SAVE ONLY H.O. 2 BITS
2576	002564	010537	013112		MOV	R5, XADDR	; SAVE EXTENDED ADDRESS BITS FOR DX CONTROL REG
2577	002570	012737	000001 013132		MOV	#1, FTIMFL	; SET FIRST TIME THROUGH FLAG

INITIALIZATION COMPLETE
TELL OPERATOR WE ARE ALREADY TO GO

NOTE: AT THIS POINT THE DX HAS NOT BEEN STARTED
AND THE OPERATOR MUST TYPE R (RUN COMMAND)
TO THE SHOW UNDER WAY

2586	002576	004137	011472		JSR	R1, MMSG	; TELL OPERATOR WE ARE READY TO GO
2587	002602	014224			.WORD	RNMMSG	
2588	002604	000137	002720		JMP	EXEC	; GET THE SHOW ON THE ROAD

VALID COMMAND TABLE

2592	002610	000		VCMDTB:	.BYTE	0	; 0 = TEST I/O
2593	002611	000			.BYTE	0	; 1 = WRITE BUFFER
2594	002612	000			.BYTE	0	; 2 = READ MANUAL INPUT
2595	002613	014			.BYTE	CE!DE	; 3 = NOP
2596	002614	000			.BYTE	0	; 4 = SENSE COMMAND
2597	002615	000			.BYTE	0	; 5 = WRITE LINE ADDRESS
2598	002616	000			.BYTE	0	; 6 = READ FULL BUFFER
2599	002617	000			.BYTE	0	; 7 = ERASE COMMAND
2600	002620	002			.BYTE	UCHK	; 10 = INVALID
2601	002621	002			.BYTE	UCHK	; 11 = INVALID
2602	002622	000			.BYTE	0	; 12 = SHORT READ MANUAL INPUT
2603		002624			.EVEN		

```

2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618 002624 012705 011604
2619 002630 000402
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633 002632 012705 011540
2634 002636 012102
2635 002640 004737 011404
2636 002644 105037 013053
2637 002650 105037 013054
2638 002654 105737 013054
2639 002660 001367
2640 002662 105737 013053
2641 002666 001772
2642 002670 012702 012640
2643 002674 004715
2644 002676 005705
2645 002700 001005
2646 002702 120427 000015
2647 002706 001002
2648 002710 011101
2649 002712 000201
2650 002714 005721
2651 002716 000201

```

```

.SBTTL  INITIALIZATION PARAMETER INPUT AND CONVERSION ROUTINES
INHEX  -- PRINT MESSAGE, WAIT FOR INPUT, GET IT AND CONVERT THE HEX TO BINARY

CALLING SEQUENCE
JSR    R1,INHEX
.WORD  ADDRESS OF MESSAGE TO BE PRINTED
.WORD  ADDRESS OF DEFAULT ROUTINE
.....RETURN
R2 = NEXT CHAR POINTER
R3 = BINARY RESULT
R4 = (BITS 0-7) FIRST NON-OCTAL CHARACTER
R5 = NUMBER OF CHARACTERS CONVERTED

INHEX:  MOV    #CHTB,R5      ;MOVE ADDRESS OF CONVERSION ROUTINE TO R5
        BR     INRS

INOC   -- PRINT MESSAGE, WAIT FOR INPUT, + GET IT AND CONVERT OCTALL TO BINARY

CALLING SEQUENCE
JSR    R1,INOC
.WORD  ADDRESS OF MESSAGE TO BE PRINTED
.WORD  ADDRESS OF THE DEFAULT ROUTINE
.....RETURN
R2 = NEXT CHAR PTR
R3 = BINARY RESULT
R4 = (BITS 0-7) FIRST NON-OCTAL CHARACTER
R5 = NUMBER OF CHARS CONVERTED

INOC:  MOV    #COTB,R5      ;SET UP ADDRESS OF THE CONVERSION ROUTINE
        INRS:  MOV    (R1)+,R2  ;GET ADDRESS OF THE MESSAGE
        10$:   JSR    PC,FRMSG   ;PRINT THE DESIRED MESSAGE
        CLRB  TCMACT        ;RESET ACTIVE FLAG
        CLRB  TCMDAB       ;RESET ABORT FLAG
        30$:   TSTB  TCMDAB    ;COMMAND ABORT?
        BNE  10$           ;YES, REASK QUESTION
        TSTB  TCMACT       ;WAS ENTRY COMPLETED?
        BEQ  30$           ;NO, WAIT
        MOV   #TBUF,R2     ;SET UP ADDRESS OF BEG OF INPUT BUFFER
        JSR  PC,IR5        ;CONVERT INPUT TO BINARY
        TST  R5            ;LOOK FOR DEFAULT RESP -- C/R
        BNE  40$           ;NOT DEFAULT TAKE NORMAL RETURN
        CMPB R4,#CR        ;ILLEGAL CHAR MUST BE A C/R
        BNE  40$           ;ITS NOT A DEFAULT
        MOV  (R1),R1       ;---TAKE THE DEFAULT RETURN
        RTS  R1
        40$:   TST  (R1)+    ;INCR FOR NORMAL RETURN
        RTS  R1

```

2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666 002720 004737 011330
2667 002724 012702 000052
2668 002730 004737 011346
2669 002734 012706 012636
2670 002740 105037 013053
2671 002744 105037 013054
2672 002750 105037 013055
2673 002754 105737 013072
2674 002760 001402
2675 002762 000137 003340
2676 002766 105737 013053
2677 002772 001001
2678 002774 000767
2679
2680
2681
2682 002776 012702 012640
2683 003002 112203
2684 003004 042703 177400
2685 003010 012704 003044
2686 003014 020324
2687 003016 001411
2688 003020 022404
2689 003022 005714
2690 003024 001373
2691
2692
2693
2694 003026 012702 137607
2695 003032 004737 011346
2696 003036 000137 002720
2697
2698
2699
2700 003042 000134
2701
2702
2703
2704 003044 000101
2705 003046 004050
2706 003050 000104
2707 003052 003432

```

.SBTTL BACKGROUND TELETYPE COMMAND DISPATCHER (EXECUTIVE)
SYSTEM EXECUTIVE/BACKGROUND
THE SYS EXEC EXECUTES THE SYSTEM TELETYPE COMMANDS
ENTRY TO THE TELETYPE COMMAND EXEC IS PERFORMED
BY EXECUTING A JUMP TO EXEC. THE CALLER
SHOULD NOT EXPECT ANY REGISTERS TO BE SAVED OR CONTROL
RTS PCED TO HIS PROGRAM.
ENTRY TO THE TELETYPE COMMAND EXEC CAUSES THE STACK POINTER
TO BE RESET. THUS, MOST COMMAND HANDLERS WILL NOT
WITH LEAVING UN"POPPED" DATA ON THE STACK.
EXEC: JSR PC,CR/LF ;PRINT CR/LF
MOV #1,R2
JSR PC,PRINT2 ;PRINT * -- DENOTE COMMAND MODE
MOV #SSTACK,SP ;RE-ESTABLISH PUSH STACK
CLRB TCMACT ;CLEAR TELE CMD ACT
CLRB TCMDOB ;CLEAR TELE CMD ABORT
CLRB LINECT ;RESET LINE COUNTER
10$: TSTB DXABFL ;DID THE DX ABORT AN OPERATION ?
BEQ 20$ ;NO, CONTINUE
JMP STOPOX ;YES IT DID, PRINT THE DX REGISTERS
20$: TSTB TCMACT ;IS THERE A COMMAND TO EXECUTE
BNE 30$ ;YES, EXECUTE IT
BR 10$ ;NO, WAIT AGAIN IF NOTHING TO DO

THERE IS A TELETYPE COMMAND TO BE EXECUTED
30$: MOV #TBUF,R2 ;SET UP PTR TO START OF TELE BUFFER
MOVB (R2)+,R3 ;GET COMMAND IDENTIFIER
BIC #177400,R3 ;SAVE L.O. BYTE
MOV #TCMDTB,R4 ;SET UP PTR TO COMMAND TABLE
40$: CMP R3,(R4)+ ;DOES COMMAND MATCH TABLE ENTR.?
BEQ EXECMD ;YES, WE GOT A MATCH - START EXECUTION
CMP (R4)+,R4 ;INCR TO NEXT COMMAND
TST (R4) ;END OF TABLE?
BNE 40$ ;NO, TEST NEXT ENTRY

COMMAND ERROR - NOTIFY OPERATOR WITH ? AND "BELL"
ERR: MOV #137607,R2 ;PRINT ? AND "BELL"
JSR PC,PRINT2
JMP EXEC ;RETURN TO EXEC

EXECUTE COMMAND
EXECMD: JMP 2(R4)+ ;EXECUTE COMMAND

TELETYPE COMMAND TABLE
TCMDTB: .WORD 'A ;A = ACCESS
.WORD ACCESS
.WORD 'D ;D = DUMP
.WORD DUMP

```

D05

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 55
BACKGROUND TELETYPE COMMAND DISPATCHER (EXECUTIVE)

SEQ 0055
SEQ 0055

2708	003054	000105	.WORD	'E	:E = ENABLE DEVICE
2709	003056	004222	.WORD	ENABLE	
2710	003060	000106	.WORD	'F	:F = FILL
2711	003062	003716	.WORD	FILL	
2712	003064	000110	.WORD	'H	:H = HELP COMMAND
2713	003066	004004	.WORD	HELP	
2714	003070	000111	.WORD	'I	:I = INPUT
2715	003072	004330	.WORD	INPUT	
2716	003074	000113	.WORD	'K	:K = KILL
2717	003076	004266	.WORD	KILL	
2718	003100	000122	.WORD	'R	:R = RUN
2719	003102	003112	.WORD	RUN	
2720	003104	000123	.WORD	'S	:S = STOP
2721	003106	003264	.WORD	STOP	
2722	003110	000000	.WORD	0	:END OF TABLE

E05

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 56
BACKGROUND -- RUN COMMAND

SEQ 0056
SEQ 0056

```

2723          .SBTTL BACKGROUND -- RUN COMMAND
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          :
003112 032777 001000 007350 RUN: BIT      #DXONLN,DXCS      ; IS DX ENABLED?
003120 001342          BNE      CERR          ; YES, ERROR
003122 005077 007342 CLR      DXCS          ; INITIALIZE THE DX
003126 005277 007336 INC      DXCS          ; SET GO
003132 012700 000001 MOV      #1,DEV        ; START CLEARING DEVICE TABLES
003136 004737 010256 10$: JSR      PC,CDEVST      ; CLEAR DEV STATUS TABLE
003142 004737 010310 JSR      PC,CSPWST      ; RESET THE APPR 2 W STATUS ENTRY FOR THE DEVICE
003146 105063 000002 CLRB     SSENSE(DTAB)  ; CLEAR SENSE BYTE
003152 105063 000017 CLRB     SDRQ(DTAB)   ; CLEAR THE READ REQUEST
003156 005063 000020 CLR      SMINS(DTAB)  ; CLEAR THE START OF MANUAL INPUT
003162 005200          INC      DEV          ; INCR TO NEXT DEVICE
003164 120037 013067 CMPB     DEV,MAXDEV    ; ARE WE DONE
003170 003762          BLE      10$          ; NO, DO NEXT DEVICE
003172 105037 013070 CLRB     DXACT         ; CLEAR DX ACTIVE FLAG
003176 105037 013074 CLRB     CMDCHF        ; CLEAR COMMAND CHAINING FLAG
003202 105037 013072 CLRB     DXABFL        ; CLEAR DX ABORT FLAG
003206 013701 013062 MOV      TTADDR,R1     ; GET THE TUMBLE TABLE ADDRESS
003212 010137 013060 MOV      R1,TTPTR      ; RESET THE SOFTWARE T/T POINTER
003216 012702 001000 MOV      #TTSIZE,R2    ; SET UP CLEAR CONSTANT (WORD COUNTER)
003222 005021          20$: CLR      (R1)+          ; CLEAR T/T AND DUP T/T
003224 005302          DEC      R2          ; ARE WE DONE?
003226 001375          BNE      20$          ; NO, KEEP ON CLEARING
003230 012737 000001 013076 MOV      #1,MDEV       ; INIT THE DEVICE NUMBER FOR MUX
                                ; AND SEL EXECUTOR ROUTINES
003236 053777 013112 007224 BIS      XADDR,DXCS    ; SET UP THE EXTENDED ADDRESS BITS
003244 052777 004000 007216 BIS      #BSYEN,DXCS   ; SEL CHANNEL - SET BUSY ENABLE
003252 052777 001100 007210 30$: BIS      #DXENB,DXONLN,DXCS ; ENABLE THE DX
003260 000137 002720 JMP      EXEC

```

2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781 003264 111204
2782 003266 120427 000015
2783 003272 001422
2784 003274 120427 000104
2785 003300 001413
2786 003302 120427 000105
2787 003306 001410
2788 003310 120427 000111
2789 003314 001405
2790 003316 120427 000120
2791 003322 001402
2792 003324 000137 003026
2793 003330 110437 013066
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804 003334 000137 002720
2805
2806
2807
2808
2809 003340 042777 000100 007122
2810 003346 004737 011330
2811 003352 004137 011472
2812 003356 014016
2813 003360 013702 013106
2814 003364 004737 004760
2815 003370 012703 000015

.SBTTL BACKGROUND -- STOP COMMAND

S = STOP DX COMMAND

STOP DISABLES THE DX IMMEDIATELY, AFTER THE NEXT CHIS
AFTER THE NEXT DATA TRANSFER COMPLETION, OR AFTER THE
NEXT ENDING SEQUENCE

THE FOLLOWING FORMATS ARE ALLOWED

- SI(C/R) -- STOP DX IMMEDIATELY
- SI(C/R) -- STOP DX AFTER NEXT INITIAL SELECTION SEQUENCE
- SD(C/R) -- STOP DX AFTER NEXT DATA TRANSFER IS COMPLETED
- SE(C/R) -- STOP DX AFTER NEXT ENDING SEQUENCE IS RECEIVED
- SP(C/R) -- STOP ON NEXT PARITY ERROR RECEIVED FROM CHANNEL

STOP WAITS UNTIL THE SPECIFIED CONDITION IS MET. THEN, THE
DX IS DISABLED AND THE DX STATUS REGISTERS ARE
DUMPED ON THE CONSOLE TELETYPE.

A RUN COMMAND (R) MUST BE EXECUTED BEFORE ANY MORE
ACTIONS WILL BE PERFORMED ON THE DX.

```

STOP:  MOVB    (R2),R4          ;GET THE TYPE OF STOP INDICATED
        CMPB   R4,#CR        ;IMMEDIATELY?
        BEQ   STOPDX        ;YES, DISABLE DX AND PRINT REGISTERS
        CMPB   R4,#D        ;D = AFTER NEXT DATA TRANSFER?
        BEQ   10$           ;YES, SET STOP FLAG
        CMPB   R4,#E        ;E = AFTER THE NEXT ENDING SEQUENCE
        BEQ   10$           ;YES, SET STOP FLAG
        CMPB   R4,#I        ;I = AFTER THE CHIS SEQUENCE
        BEQ   10$           ;YES, SET STOP FLAG
        CMPB   R4,#P        ;P = STOP ON PARITY ERROR??
        BEQ   10$           ;YES, SET STOP FLAG
        JMP   CERR          ;ILLEGAL FORMAT -- GIVE ERROR
10$:   MOVB   R4,DXSTPF      ;SET THE STOP FLAG

```

WHEN THE STOP CONDITION IS SATISFIED,
THE DX ISR WILL ABORT ALL DX ACTIVITY AND
SET A FLAG CAUSING ALL DX REGISTERS TO BE
DUMPED BY "STOPDX", BELOW

THE STOP CONDITION WILL REMAIN IN EFFECT
UNTIL IT IS SATISFIED OR ANOTHER REQUEST
SUPERCEDES IT.

JMP EXEC ;RETURN TO THE EXEC

STOP THE DX AND PRINT THE REGISTERS
NOTE THE PRINT OUTS WILL BE IN OCTAL

```

007122 STOPDX: BIC    #DXENB,DXCS    ;DISABLE THE DX
              JSR   PC,CRLF    ;START AT NEW LINE
              JSR   R1,MSG     ;PRINT "CURRENT DEVICE -- "
              .WORD STAMES
              MOV   CDEV,R2    ;CONVERT AND PRINT THE CURRENT
              JSR   PC,HDMP    ;DEVICE NUMBER IN HEX
              MOV   #13,R3     ;PRINT THE 13 DX REGISTERS IN OCTAL

```

2816	003374	012701	012464
2817	003400	013102	
2818	003402	004737	004670
2819	003406	005303	
2820	003410	001373	
2821	003412	105037	013072
2822	003416	005077	007046
2823	003422	005277	007042
2824	003426	000137	002720

10S:	MOV	#DXDS,R1
	MOV	2(R1)+,R2
	JSR	PC, OCTDMP
	DEC	R3
	BNE	10S
	CLRB	DXABFL
	CLR	DXCLS
	INC	DXYCS
	JMP	E-EC

:	STARTING POINT
:	GET THE REGISTER CONTENTS
:	PRINT IN OCTAL
:	ARE WE DONE
:	NO DUMP NEXT WORD
:	YES RESET THE ABORT FLAG
:	RESET THE DX
:	AND RETURN TO THE E-EC

H05

MAINDEC-11-DZDAI-B NEW D:11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 59
BACKGROUND -- DUMP COMMAND

SEC 0059
SEC 0059

2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880

.SBTTL BACKGROUND -- DUMP COMMAND
DUMP COMMAND
THE DUMP COMMAND DUMPS THE SPECIFIED DATA AREA ON THE
CONSOLE TELETYPE IN THE SPECIFIED FORMAT.
THE FOLLOWING COMMAND SYNTAXES ARE AVAILABLE:
DTT,X DUMP DUPLICATE TUMBLE TABLE IN CODE X
DIN,X,YY DUMP INPUT BUFFER FOR DEVICE YY IN CODE X
DOT,X,YY DUMP OUTPUT BUFFER FOR DEVICE YY IN CODE X
DSSSSSS,EEEEEE,X DUMP BETWEEN THE OCTAL LIMITS GIVEN
IN CODE X
WHERE: X = A-ASCII, E-EBCDIC, H-HEX, O-OCTAL
YY = THE DEVICE ADDRESS IN HEX

THE DUMPS ARE PERFORMED IN A COLUMN FASHION FOR
OCTAL AND HEX MODES (ONE WORD PER LINE) AND IN A LINE
FASHION FOR ASCII AND EBCDIC MODES (60 CHARACTERS PER LINE)

DUMP: JSR PC, GLIMIT ; GET BUFFER LIMITS
MOV (R2)+, R4 ; GET DUMP MODE A/E/O/H
MOV #ASCDMP, R5 ; SET UP FOR ASCII DUMP
CMPB R4, #'A ; IS IT ASCII?
BEQ 10\$; YES, START DUMP
MOV #EBCDMP, R5 ; SET UP FOR EBCDIC DUMP
CMPB R4, #'E ; IS IT EBCDIC?
BEQ 10\$; YES, CONTINUE DUMP
MOV #HEXDMP, R5 ; SET UP FOR HEX DUMP
CMPB R4, #'H ; IS IT HEX?
BEQ 10\$; YES, CONTINUE DUMP
MOV #OCTDMP, R5 ; SET UP FOR OCTAL DUMP
CMPB R4, #'O ; IS IT OCTAL?
BEQ 10\$; YES, CONTINUE DUMP
JMP CERR ; ILLLEGAL ENTRY -- ERROR
10\$: MOV R5, DMPADR ; SAVE ADDRESS OF DUMP ROUTINE
TST R0 ; WAS THIS A TUMBLE TABLE DUMP?
BNE DTUMTB ; YES, DUMP THE TUMBLE TABLE
MOV (R2)+, R4 ; WAS A DEV # SPECIFIED
CMPB R4, #' ; IS NEXT POSITION A COMMA
BNE 50\$; NO, DUMP GIVEN LIMITS
JSR PC, GDEV ; GET THE DEVICE NUMBER -- IN HE
:
COMPUTE RELOCATION CONSTANT FOR DEVICE
30\$: CLR R4 ; RELOCATION CONSTANT
DEC R3 ; DONE?
BEQ 40\$; YES, ADD TO START + END ADDRESSES
ADD #2000, R4 ; TO NEXT DEVICE TABLES
BR 30\$
40\$: ADD R4, SADDR ; ADD RELOCAT TO START ADDRESS
ADD R4, EADDR ; ADD RELOCAT TO END ADDRESS
50\$: MOV #SADDR, R2 ; GET WORD
JSR PC, DMPADR ; CONVERT AND DUMP IT
60\$: TSTB PC, R
BNE 60\$

2881	003602	062737	000002	013120		ADD	#2,SADDR	: INCR TO NEXT WORD
2882	003610	023737	013120	C13122		CMP	SADDR,EADDR	: DUMP DONE
2883	003616	003003				BGT	70\$: YES, EXIT
2884	003620	105737	013054			TSTB	TCMDAB	: COMMAND ABOPT?
2885	003624	001757				BEQ	50\$: NO, PRINT NEXT WORD
2886	003626	000137	002720		70\$:	JMP	EXEC	: YES, RETURN TO EXEC
2887								
2888								
2889								
2890								
2891								
2892								
2893	003632	012700	000400			MTB: MOV	#TTSIZE/2,R0	: SET UP COUNTER TO DUMP ENTIRE TUMBLE TABLE
2894	003636	017702	007256		5\$:	MOV	SADDR,R2	: GET STARTING ADDRESS
2895	003642	004777	007256			JSR	PC,DMPAUR	: PRINT THE CONTENTS
2896	003646	105737	013071		10\$:	TSTB	PC↑R	: IS PRINT COMPLETE?
2897	003652	001375				BNE	10\$: NO, WAIT TILL DONE
2898	003654	032737	000777	013120		BIT	#TTSIZE-1,SADDR	: CHECK FOR WRAP AROUND
2899	003662	001003				BNE	20\$	
2900	003664	062737	001000	013120		ADD	#TTSIZE,SADDR	: WRAP AROUND TO TOP OF TABLE
2901	003672	162737	000002	013120	20\$:	SUB	#2,SADDR	: DECREMENT TO NEXT ENTRY
2902	003700	005300				DEC	R0	: HAS ENTIRE TUMBLE TABLE BEEN DUMPED?
2903	003702	001403				BEQ	30\$: YES, EXIT TO THE EXEC
2904	003704	105737	013054			TSTB	TCMDAB	: ARE WE TO ABORT?
2905	003710	001752				BEQ	5\$: NO, KEEP ON DUMPING
2906	003712	000137	002720		30\$:	JMP	EXEC	: UP, BACK TO THE EXEC

J05

MAINDEC-11-DZDAI-B NEW DX11-B RESPONDER
CZDXIB.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 61
BACKGROUND -- FILL COMMAND

SEG 0061
SEG 0061

2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921 003716 004737 005050
2922 003722 004737 011604
2923 003726 110337 012534
2924 003732 004737 005302
2925
2926
2927
2928 003736 005004
2929 003740 005303
2930 003742 001403
2931 003744 062704 002000
2932 003750 000773
2933 003752 060437 013120
2934 003756 060437 013122
2935 003762 013701 013120
2936
2937
2938
2939 003766 113721 012534
2940 003772 020137 013122
2941 003776 101773
2942 004000 000137 002720

.SBTTL BACKGROUND -- FILL COMMAND
FILL COMMAND
THE FILL COMMAND LOADS THE SPECIFIED BYTE
INTO THE GIVEN DATA AREA.
THE FOLLOWING SYNTAXES ARE AVAILABLE FOR THE FILL COMMAND:
FIN,XX,YY FILL INPUT BUFFER FOR DEVICE YY WITH XX
FOT,XX,YY FILL OUTPUT BUFFER FOR DEVICE YY WITH XX
WHERE: XX IS THE FILL CHARACTER IN HEX
YY IS THE DEVICE ADDRESS IN HEX
FILL: JSR PC,GLIMIT ;GET BUFFER LIMITS
JSR PC,CHTB ;GET THE FILL CHARACTER
MOV B R3,FILLCH ;SAVE FILL CHAR
JSR PC,GDEV ;GET THE DEVICE ADDRESS
COMPUTE RELOCATION FOR DEVICE
10\$: CLR R4
DEC R3 ;DONE?
BEQ 20\$;YES, ADD TO START AND END ADDR
ADD #2000,R4
BR 10\$
20\$: ADD R4,SADDR ;ADD RELOC CONST TO START
ADD R4,EADDR ;ADD RELOC CONST TO END ADDR
MOV SADDR,R1
FILL BUFFER WITH SPECIFIED CHARACTER
30\$: MOV B FILLCH,(R1)+ ;FILL CHARACTER
CMP R1,EADDR ;DONE?
BLOS 30\$;NOPE, FILL NEXT CHAR
JMP EXEC ;DONE, RETURN TO EXEC

2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965

004004 012701 014300
004010 012702 002551

004014 112100
004016 123727 013071 000004
004024 003374
004026 004737 011366
004032 105737 013054
004036 001002
004040 005302
004042 001364
004044 000137 002720

.SBTTL BACKGROUND -- HELP COMMAND
:
THE HELP COMMAND PROVIDES THE OPERATOR WITH A SYNOPSIS OF
COMMANDS WHICH MAY BE USED FOR OPERATING THIS SYSTEM.
:
THE SYNTAX FOR THE HELP COMMAND IS:
H
HELP: MOV #HELPM5,R1 ;SET UP ADDRESS OF HELP MESSAGE
MOV #HELPLN,R2 ;LENGTH OF HELP MESSAGE
:
START OUTPUTTING THE HELP MESSAGE UNDER OUR CONTROL
SO THE COMMAND MAY BE ABORTED QUICKLY.
10\$: MOVB (R1)+,R0 ;GET BYTE TO OUTPUT
15\$: CMPB PCTR,#4 ;MORE THEN FOUR CHARACTER IN OUTPUT BUFFER??
BGT 15\$; YES, WAIT TIL DOWN A LITTLE
JSR PC,PCHAR ;PRINT IT ON CONSOLE
TSTB TCMDAB ;HAS OPERATOR INDICATED A DESIRE TO STOP?
BNE 20\$; YES, ABORT HELP MESSAGE
DEC R2 ;HAS ENTIRE MESSAGE BEEN OUTPUTTED??
BNE 10\$; NO, OUTPUT ANOTHER BYTE
20\$: JMP EXEC ; YES, RETURN TO THE EXECUTIVE

```

2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979 004050 004737 011540
2980 004054 005705
2981 004056 001403
2982 004060 032703 000001
2983 004064 001402
2984 004066 000137 003026
2985 004072 010337 013120
2986 004076 013702 013120
2987 004102 105037 013053
2988 004106 004737 004670
2989 004112 012702 020040
2990 004116 004737 011346
2991 004122 017702 006772
2992 004126 004737 004674
2993 004132 012702 020040
2994 004136 004737 011346
2995 004142 105737 013053
2996 004146 001775
2997 004150 012702 012640
2998 004154 004737 011540
2999 004160 005705
3000 004162 001007
3001 004164 120427 000057
3002 004170 001412
3003 004172 120427 000015
3004 004176 001403
3005 004200 000736
3006 004202 010377 006712
3007 004206 062737 000002 013120
3008 004214 000730
3009 004216 000137 002720

```

.SBTTL BACKGROUND -- ACCESS COMMAND

ACCESS SPECIFIED LOCATIONS AND CHANGE IF DESIRED

THE ACCESS COMMAND IS A QUICK LOOK AND CHANGE ROUTINE MAINLY USED FOR PROGRAM DEBUGGING.

BASICALLY THE FOLLOWING ACTIONS ARE PERMITTED:

- AXXXXX -- OPEN AND PRINT SPECIFIED OCTAL LOCATION
- [XXXXXX](C/R) -- CHANGE CURRENT LOCATION IF DATA SPECIFIED [XXXXXX] AND OPEN NEXT LOCATION
- / -- RETURN TO EXEC MODE

```

ACCESS: JSR PC,COTB ;GET THE START ADDRESS
        TST R5 ;WAS A VALID ADDRESS ENTERED?
        BEQ 5$ ;NO, GIVE OPERATOR AN ERROR
        BIT #1,R3 ;WAS ADDRESS SPECIFIED A WORD ADDRESS?
        BEQ 7$ ;YES, OPEN SPECIFIED LOCATION
        JMP CERR ;NO, GIVE OPERATOR AN ERROR INDICATION
5$: MOV R3,SADDR ;SAVE STARTING ADDRESS
7$: MOV SADDR,R2 ;GET OBJECT WORD
10$: CLR TCMACT ;CLEAR TELE ACTIVE FLAG
        JSR PC,OCTDMP ;PRINT ADDRESS IN OCTAL
        MOV #2,R2 ;PRINT 2 SPACES
        JSR PC,PRINT2
        MOV JSADDR,R2 ;GET CONTENTS OF OBJECT LOCATION
        JSR PC,ODMP ;PRINT CONTENTS IN OCTAL
        MOV #2,R2 ;PRINT 2 SPACES
        JSR PC,PRINT2
20$: TST TCMACT ;ACTIVE COMMAND?
        BEQ 20$ ;NO
        MOV #TBUF,R2 ;SET UP INPUT BUFFER ADDRESS
        JSR PC,COTB ;WAS LOCATION CHANGED?
        TST R5 ;ANY CHANGE?
        BNE 30$ ;YES, STORE IT
        CMPB R4,#' / ;EXIT TO EXEC
        BEQ 50$ ;YES, RETURN TO EXEC
        CMPB R4,#CR ;CR, GO TO NEXT LOCATION?
        BEQ 40$ ;YES, OPEN AND PRINT NEXT LOC.
        BR 10$ ;ERROR, PRINT CONTENTS OF CURRENT LOC.
30$: MOV R3,JSADDR ;CHANGE OPEN LOCATION
40$: ADD #2,SADDR ;OPEN NEXT LOCATION
        BR 10$
50$: JMP EXEC ;RETURN TO THE EXEC

```



```

3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022 004222 004737 005302
3023 004226 004737 010256
3024 004232 010005
3025 004234 063705 013110
3026 004240 060505
3027 004242 063705 013114
3028 004246 013715 013116
3029 004252 105063 000002
3030 004256 105063 000016
3031 004262 000137 002720

```

```

.SBTTL BACKGROUND -- ENABLE DEVICE COMMAND

E = ENABLE DEVICE

THE ENABLE COMMAND TURNS THE DEVICE SPECIFIED INTO AN
ON-LINE MODE. THIS IS ONLY NECESSITATED BECAUSE A KILL
COMMAND WAS PERFORMED ON THE DEVICE IN QUESTION.

THE ENABLE COMMAND HAS THE FOLLOWING SYNTAX:
  EXX      -- ENABLE DEVICE ADDRESS XX
             THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX

ENABLE: JSR      PC,GDEV      ;GET THE DEVICE NUMBER
        JSR      PC,CDEVST    ;CLEAR THE DEVICE STATUS TABLE
        MOV      DEV,R5       ;COMPUTE THE ADDRESS OF THE SPW TABLE ENTRY
        ADD      DEVCON,R5    ;COMPENSATE FOR OFFSET DEVICE ADDRESS
        ADD      R5,R5
        ADD      STSPW,R5
        MOV      DSTOFF,(R5)  ;ENABLE THE DEVICE NUMBER
        CLRB     SSENSE(DTAB)
        CLRB     SONLF(DTAB)
        JMP      EXEC         ;RETURN TO THE EXEC

```

3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054

004266 004737 005302
004272 004737 010256
004276 112763 000001 000016
004304 010005
004306 063705 013110
004312 060505
004314 063705 013114
004320 012715 000002
004324 000137 002720

.SBTTL BACKGROUND -- KILL DEVICE COMMAND
K = KILL A DEVICE
THE KILL COMMAND DISABLES THE SPECIFIED DEVICE ADDRESS FROM PERFORMING TRANSFERS OVER THE DX. IT PUTS THE SPECIFIED DEVICE ADDRESS INTO AN OFF-LINE STATE. AN ENABLE COMMAND MUST BE ISSUED BEFORE DATA TRANSFERS MAY BE PERFORMED WITH THE DX FOR THE SPECIFIED DEVICE ADDRESS.
THE KILL COMMAND HAS THE FOLLOWING SYNTAX:
KXX -- KILL DEVICE ADDRESS XX
THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX
KILL: JSR PC,GDEV ;GET THE DEVICE NUMBER
JSR PC,CDEVST
MOVB #1,SONLF(DTAB)
MOV DEV,R5 ;COMPUTE THE ADDRESS OF THE SPW TABLE
ADD DEVCON,R5 ;COMPONSATE FOR OFFSET DEVICE ADDRESS
ADD R5,R5
ADD STSPW,R5
MOV #UCHK,(R5) ;MAKE THE DEVICE OFF-LINE SEND UNIT CHECK
KILLEX: JMP EXEC ;RETURN TO THE EXEC

```

3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067 004330 105737 012532
3068 004334 001005
3069 004336 004737 005302
3070 004342 120427 000054
3071 004346 001402
3072 004350 000137 003026
3073 004354 004737 010230
3074 004360 026327 000004
3075 004366 002370
3076 004370 032763 000001
3077 004376 001002
3078 004400 005263 000004
3079 004404 016305 000010
3080 004410 066305 000004
3081 004414 112725 000112
3082 004420 010563 000020
3083 004424 005263 000004
3084 004430 026327 000004
3085 004436 001423
3086 004440 112204
3087 004442 042704 177600
3088 004446 020427 000015
3089 004452 001415
3090 004454 020427 000040
3091 004460 002410
3092 004462 020427 000137
3093 004466 003005
3094 004470 162704 000040
3095 004474 116425 012354
3096 004500 000751
3097 004502 005004
3098 004504 000773
3099
3100
3101
3102
3103
3104 004506 112715 000152
3105 004512 005263 000004
3106 004516 105263 000017
3107
3108
3109
3110 004522 105737 013070

```

```

.SBTTL BACKGROUND -- INPUT DISPLAY DATA COMMAND

I = INPUT

THE INPUT COMMAND IS USED TO ENTER DATA ONTO A 2260
SCREEN AND THEN SEND IT TO THE 360 VIA THE READ MANUAL INPUT
COMMANDS

THE INPUT COMMAND HAS THE FOLLOWING SYNTAX:
  IXX,DDD...DDD -- SEND DATA DDD TO DEVICE XX
                  THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX

INPUT:  TSTB      TSTYP      ; ILLEGAL ON FRIEND TEST
        BNE      IC$        ; FRIEND -- GIVE AN ERROR
        JSR      PC,GDEV     ; GET THE DEVICE NUMBER
        CMPB    R4,#' '     ; THE NEXT CHAR MUST BE A COMMA
        BEQ     20$        ; IT IS, CONTINUE
        JMP     CERR        ; AN ERROR WAS FOUND GIVE INDICATION
10$:    JSR      PC,SUDEV     ; SET UP THE DEVICE STATUS TABLE PCINTERS
20$:    CMP     SCURS(DTAB),#DISPSZ-4 ; ARE WE AT THE END OF THE BUFFER?
        BGE     10$        ; YES, GIVE AN ERROR
        BIT     #1,SCURS(DTAB) ; START INPUT ON EVEN BYTE ADDRESS
        BNE     30$        ; START SOM ON ODD BYTE ADDRESS
        INC     SCURS(DTAB)  ; INCR CURSOR TO ODD BYTE ADDRESS
30$:    MCV     SOUTB(DTAB),R5 ; COMPUTE STARTING ADDRESS
        ADD     SCURS(DTAB),R5
        MOVB   #SMI,(R5)+   ; START CHARACTER TO BUFFER
        MCV     R5,SMINS(DTAB) ; SAVE START OF DATA LOCATION
40$:    INC     SCURS(DTAB)  ; INCREMENT CURSOR POSITION
        CMP     SCURS(DTAB),#DISPSZ-3 ; ARE WE AT THE END OF BLFFEP
        BEQ     70$        ; YES, TERMINATE INPUT
        MOVB   (R2)+,R4     ; GET NEXT INPUTTED CHARACTER
        BIC     #177600,R4  ; SAVE L.O. 7 BITS
        CMP     R4,#CR     ; END OF INPUT?
        BEQ     70$        ; YES, SET UP TO EXIT
        CMP     R4,#SPACE  ; CAN CHARACTER BE CONVERTED?
        BLT     60$        ; NO, MUST BE BETWEEN 40 - 137
        CMP     R4,#'+     ;
        BGT     60$        ; NO, MUST BE BETWEEN 40 - 137
        SUB     #SPACE,R4  ; SCALE DOWN FOR INDEXING
50$:    MOVB   ATOETB(R4),(R5)+ ; CONVERT CHARACTER AND MOVE TO DISPLAY BUFFER
        BR     40$        ; GET AND CONVERT NEXT CHARACTER
60$:    CLR     R4         ; ILLEGAL CHARACTER -- TREAT AS SPACE
        BR     50$

        ;
        ; SET UP TO EXIT
        ; SET EOM INDICATOR
        ; QUEUE READ MANUAL INPUT REQUEST
70$:    MOVB   #EOM,(R5)   ; SET EOM INDICATOR
        INC     SCURS(DTAB) ; INCREMENT CURSOR POINTER
        INCB   SDRDQ(DTAB) ; QUEUE READ REQUEST
        ;
        ; SEE IF THE DX IS CURRENTLY ACTIVE
        TSTB   DXACT      ; IS D+ ACTIVE?

```

```

3111 004526 001402          BEQ      BOS          ;NO, START ASYNCHRONOUS PROCESSING TO SEND ATTENTION
3112 004530 000137 002720  JMP      EXEC         ;YES, ATTENTION WILL BE TAKEN CARE OF BY DX
3113
3114          :
3115          :
3116          :
3117 004534 013746 177776    BOS:   MOV      PSW, -(SP)      ;PSW TO PUSH STACK
3118 004540 012746 002720    MOV      #EXEC, -(SP)   ;RETURN ADDRESS TO PUSH STACK
3119 004544 012737 000340 177776  MOV      #340, PSW      ;INHIBIT INTERRUPTS
3120 004552 010046          MOV      R0, -(SP)      ;SET UP PUSH STACK FOR FAKE INTERRUPT
3121 004554 010146          MOV      R1, -(SP)
3122 004556 010246          MOV      R2, -(SP)
3123 004560 010346          MOV      R3, -(SP)
3124 004562 010446          MOV      R4, -(SP)
3125 004564 010546          MOV      R5, -(SP)
3126 004566 000137 006554    JMP      DYEXEC         ;START PROCESSING THE ATTENTION

```

3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182

004572 010237 013056
004576 113703 013056
004602 042703 177400
004606 116337 011754 013056
004614 113703 013057
004620 042703 177400
004624 116337 011754 013057
004632 013702 013056

004636 105737 013055
004642 001005
004644 004737 011330
004650 112737 000036 013055
004656 105337 013055
004662 004737 011346
004666 000207

004670 004737 011330

```
.SBTTL BACKGROUND SUBROUTINES -- PRINT FORMATTING

DUMP WORD IN EBCDIC ON TTY
CALLING SEQUENCE
.....R2 CONTAINS WORD TO BE PRINTED
JSR PC,EBCDMP
.....RETURN

REGISTERS 2 + 3 ARE DESTROYED BY THIS SUBROUTINE
EBCDMP: MOV R2,WK ;SAVE WORD TO BE PRINTED
        MOVB WK,R3 ;GET LO BYTE
        BIC #177400,R3
        MOVB EBCDTB(R3),WK ;CONVERT EBCDIC TO ASCII
        MOVB WK1,R3 ;GET HI BYTE AND CONVERT
        BIC #177400,R3
        MOVB EBCDTB(R3),WK1 ;CONVERT CHAR TO ASCII
        MOV WK,R2
        FALL THROUGH TO ASCII PRINT ROUTINE

DUMP WORD IN ASCII ON TTY
CALLING SEQUENCE
.....R2 CONTAINS WORD TO BE PRINTED
JSR PC,ASCDMP
.....RETURN

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
ASCDMP: TSTB LINECT ;NEW LINE?
        BNE 10$ ;NO
        JSR PC,CRLF ;YES, PRINT CR/LF
        MOVB #30,LINECT ;60 CHARACTERS PER LINE
        10$: DECB LINECT ;DECR LINE COUNTER
        JSR PC,PRINT2 ;PRINT 2 CHARS
        RTS ;RETURN TO CALLER

DUMP WORD IN OCTAL ON TTY
CALLING SEQUENCE
.....R2 CONTAINS WORD TO BE PRINTED
JSR PC,OCTDMP OR ODMP
.....RETURN

OCTDMP PERFORMS A CR/LF BEFORE PRINTING OCTAL DATA
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
OCTDMP: JSR PC,CRLF ;GIVE A CRLF
```

E06

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
 DZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 69
 BACKGROUND SUBROUTINES -- PRINT FORMATTING

SEG 0069
 SEQ 0069

```

3183 004674 010046      ODMP:  MOV    R0,-(SP)      :SAVE IMPORTANT REGISTERS
3184 004676 010246      MOV    R2,-(SP)
3185 004700 010446      MOV    R4,-(SP)
3186 004702 012704 000006  MOV    #6,R4      :EXTRACT 6 OCTAL DIGITS
3187 004706 005000      CLR    R0          :CLEAR THE WORKING REGISTER
3188 004710 006102      ROL    R2          :MOVE HIGH ORDER BIT TO C-BIT
3189 004712 006100      10$:  ROL    R0          :GET THE REMAINING BIT STILL IN LINK
3190 004714 042700 177770  BIC    #177770,R0  :ONLY 3 LOW ORDER BITS
3191 004720 062700 000060  ADD    #'0,R0      :MAKE ASCII
3192 004724 004737 011366  JSR    PC,PCHAR    :PRINT IT ON THE TTY
3193 004730 006102      ROL    R2          :ROTATE THE NEXT OCTAL CHAR INTO POSITION
3194 004732 006102      ROL    R2
3195 004734 006102      ROL    R2
3196 004736 010200      MOV    R2,R0      :DATA TO WORKING REGISTER
3197 004740 005304      DEC    R4          :ARE WE DONE?
3198 004742 001363      BNE    10$        :NO, PRINT ANOTHER CHARACTER
3199 004744 012604      OCTEX: MOV    (SP)+,R4   :RESTORE USED REGISTERS
3200 004746 012602      MOV    (SP)+,R2
3201 004750 012600      MOV    (SP)+,R0
3202 004752 000207      RTS    PC          :RETURN TO THE CALLER
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217 004754 004737 01133C  HEXDMP: JSR    PC,CRLF   :DO A CR LF
3218 004760 010046  HDMP:  MOV    R0,-(SP)   :SAVE THE WORKING REGISTERS
3219 004762 010246      MOV    R2,-(SP)
3220 004764 010446      MOV    R4,-(SP)
3221 004766 012704 000004  MOV    #4,R4      :4 CHARACTERS PER WORD
3222 004772 006102      10$:  ROL    R2          :ROTATE HIGH ORDER 4 BITS TO LOW ORDER 4 BITS
3223 004774 006102      ROL    R2
3224 004776 006102      ROL    R2
3225 005000 006102      ROL    R2
3226 005002 010200      MOV    R2,R0      :TO WORKING REG
3227 005004 006100      ROL    R0          :GET THE LINK BIT TOO
3228 005006 042700 177760  BIC    #177760,R0  :ONLY LOW ORDER 4 BITS
3229 005012 062700 000060  ADD    #'0,R0      :MAKE ASCII IF NUMBER
3230 005016 020027 000071  CMP    R0,#'9      :SHOULD IT BE A-F?
3231 005022 003402      BLE    20$        :NO, SKIP IT
3232 005024 062700 000007  ADD    #'7,R0      :YES, MAKE ALPHA
3233 005030 004737 011366  20$:  JSR    PC,PCHAR    :PRINT THE HEX CHARACTER
3234 005034 005304      DEC    R4          :ARE WE DONE?
3235 005036 001355      BNE    10$        :NO, CONVERT AND PRINT NEXT CHARACTER
3236 005040 012604      MOV    (SP)+,R4   :YES, RESTORE REGISTERS AND EXIT
3237 005042 012602      MOV    (SP)+,R2
3238 005044 012600      MOV    (SP)+,R0
  
```

.....
 DUMP WORD IN HEX ON THE TTY
 CALLING SEQUENCE
R2 CONTAINS THE WORD TO BE PRINTED
 JSR PC,HEXDMP OR HDMP
RETURN
 HEXDMP PERFORMS A CRLF BEFORE OUTPUTTING THE DATA
 NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

F06

MAINDEC-11-DZDNI-B NEW D:11-B RESPONDER
CZD:1B.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 70
BACKGROUND SUBROUTINES -- PRINT FORMATTING

SEC 0070
SEC 0070

3239 005046 000207

RTS PC

:RETURN TO THE CALLER

```

3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259 005050 005000
3260 005052 004737 011540
3261 005056 005705
3262 005060 001014
3263 005062 120427 000124
3264 005066 001425
3265 005070 120427 000111
3266 005074 001436
3267 005076 120427 000117
3268 005102 001453
3269
3270
3271
3272 005104 012601
3273 005106 000137 003026
3274
3275
3276
3277
3278
3279 005112 010337 013120
3280 005116 120427 000054
3281 005122 001370
3282 005124 004737 011540
3283 005130 005705
3284 005132 001764
3285 005134 010337 013122
3286 005140 000454
3287
3288
3289
3290
3291
3292 005142 112204
3293 005144 120427 000124
3294 005150 001355
3295 005152 013737 013060 013120

```

```

.SBTTL BACKGROUND SUBROUTINES -- COMPUTE SPECIFIED BUFFER LIMITS AND DEVICE ADD
GLIMIT -- SET UP BUFFER LIMITS FOR TELE COMMANDS

CALLING SEQ
.....R2 = ADDRESS OF FIRST PARAMETER
JSR PC, GLIMIT
.....RETURN IF NO ERRORS DETECTED IN BUFFER LIMIT SYNTAX
IF AN ERROR IS DETECTED, CONTROL WILL BE
PASSED TO "CERR" TO ABORT THE TELETYPE COMMAND.
UPON GOOD RETURN:
R0 = 0 = NOT T/T, 1 = T/T
R2 = NEXT CHAR POSITION IN COMMAND STRING
SADDR = BEG ADDR TO BE DUMPED
EADDR = END ADDR TO BE DUMPED

REGISTERS R5, R4, R3 WILL BE DESTROYED.

IF AN ERROR IS FOUND CONTROL IS PASSED TO CERR

GLIMIT: CLR R0 ; RESET BUFFER TYPE
JSR PC, COTB ; GET FIRST PARAMETER
TST R5 ; WAS AN OCTAL NUMBER ENTERED?
BNE GLOCT ; YES, OCTAL PARAMS
CMPB R4, #'T ; T = TUMBLE TABLE
BEQ GLMTT ; YES, SET UP T/T LIMITS
CMPB R4, #'I ; I = INPUT BUFFER
BEQ GLMIN ; YES, SET UP INPUT BUFFER LIMITS
CMPB R4, #'O ; O = OUTPUT BUFFER
BEQ GLMOT ; YES, SET UP OUTPUT BUFFER LIMITS

:
:
: ERROR DETECTED - POP OFF RETURN ADDR AND GIVE ERROR
GLERR: MOV (SP)+, R1
JMP CERR

:
:
: OCTAL LIMITS SPECIFIED
GLOCT: MOV R3, SADDR ; SAVE START ADDR
CMPB R4, #',' ; CHECK FOR COMMA
BNE GLERR
JSR PC, COTB ; GET END ADDR
TST R5 ; WAS SECOND PARAM GIVEN?
BEQ GLERR ; NO, ERROR
MOV R3, EADDR ; SAVE END ADDR
BR GLEX ; PREPARE TO EXIT

:
:
: SET UP LIMITS OF TUMBLE TABLE
GLMTT: MOVB (R2)+, R4
CMPB R4, #'T ; MUST BE T
BNE GLERR ; ILLEGAL ENTRY
MOV TPTR, SADDR

```


H06

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 72
BACKGROUND SUBROUTINES -- COMPUTE SPECIFIED BUFFER LIMITS AND DEVICE ADDRESSES

SEC 0072
SEC 0072

```
3296 005160 062737 000776 013120 ADD #TTSIZE-2,SADDR ; COMPUTE ADDRESS OF APPR DUPLICATE TT ENTRY
3297 005166 005200 INC R0 ; INDICATE DUMP TUMBLE TABLE
3298 005170 000437 BR GLEX1 ; SET UP TO EXIT
3299
3300
3301
3302
3303
3304 005172 112204 GLMIN: MOVB (R2)+,R4
3305 005174 120427 000116 CMPB R4,#'N ; MUST BE IN
3306 005200 001341 BNE GLERR ; ILLEGAL ENTRY
3307 005202 013704 013064 MOV SDEVTB,R4 ; GET ADDR OF DEV 0 STATUS TABLE
3308 005206 016437 000006 013120 MOV SINBF(R4),SADDR
3309 005214 013737 013120 013122 MOV SADDR,EADDR
3310 005222 062737 000741 013122 ADD #DISPSZ+1,EADDR ; DISPLAY SIZE + ROOM FOR LINE ADDRESS
3311 005230 000417 BR GLEX1
3312
3313
3314
3315
3316
3317 005232 112204 GLMOT: MOVB (R2)+,R4
3318 005234 120427 000124 CMPB R4,#'T ; MUST BE OT
3319 005240 001321 BNE GLERR ; ILLEGAL ENTRY
3320 005242 013704 013064 MOV SDEVTB,R4 ; GET ADDR OF DEV 0 STATUS TABLE
3321 005245 016437 000010 013120 MOV SOUTB(R4),SADDR ; COMPUTE STARTING AND ENDING ADDRESSES OF SPECIFIED BUF
3322 005254 013737 013120 013122 MOV SADDR,EADDR
3323 005262 062737 000737 013122 ADD #DISPSZ-1,EADDR ; DISPLAY CHAR BUFFER
3324 005270 112204 GLE>1: MOVB (R2)+,R4 ; GET NEXT INPUT CHARACTER AND UPDATE POINTER
3325
3326
3327
3328 005272 120427 000054 GLE>: CMPB R4,#'
3329 005276 001307 BNE GLERR ; CHECK FOR
3330 005280 000207 RTS PC ; ENTRY NOT PROPERLY DELIMITED (ERROR)
```

```

3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345 005307 004737 011604
3346 005306 163703 012522
3347 005312 100406
3348 005314 005203
3349 005316 120337 013067
3350 005322 101002
3351 005324 010300
3352 005326 000207
3353 005330 000137 003026
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372 005334 010046
3373 005336 010146
3374 005340 010246
3375 005342 010346
3376 005344 010446
3377 005346 010546
3378 005350 013702 013060
3379 005354 005712
3380 005356 001002
3381
3382
3383
3384
3385
3386 005360 000137 010212

```

```

:
: GDEV -- GET THE THE DEVICE NUMBER FROM THE HEX INPUT
:
: CALLING SEQUENCE
: ..... R2 = ADDRESS OF DEVICE ADDRESS IN HEX
: JSR PC GDEV
: ..... RETURN IF NO ERRORS DETECTED
: ..... IF ERROR DETECTED, COMMAND IS ABORTED BY GOING
: ..... TO "CERR"
:
: UPON VALID RETURN
: R3 AND DEV (R0) WILL CONTAIN THE DEVICE ADDRESS
: SCALED TO 1 - 8, NOTATION USED BY SYSTEM.
: R2 WILL POINT TO THE NEXT CHARACTER FOLLOWING DEVICE ADDRESS
:
:DEV: JSR PC,CHTB ; CONVERT THE HEX TO BINARY
SUB SDEV,R3 ; -STARTING ADDRESS
BMI 10$ ; ERROR ON INPUT
INC R3 ; MAKE BETWEEN 1 AND 8
CMPB R3,MAXDEV ; IS DEVICE NUMBER TOO BIG?
BHI 10$ ; YES, GIVE ERROR
MOV R3,DEV ; SET UP THE DEVICE NUMBER
RTS PC
:
:10$: JMP CERR ; INPUT PARAM ERROR
.SBTTL DX11-B ISR (INTERRUPT REQUEST LOGIC AND TUMBLE TABLE DECODE LOGIC)
:
: D X 1 1 - B I S R
:
: DX11 ISR AND RELATED SUBROUTINE REGISTER USAGE
:
: R0 DEV DEVICE NUMBER
: R1 UN ASSIGNED
: R2 UNASSIGNED
: R3 DTAB ADDRESS OF CURRENT DEVICE TABLE
: R4 TT1 TUMBLE TABLE ENTRY 1
: R5 TT2 TUMBLE TABLE ENTRY 2
:
: THE ABOVE REGISTER DESIGNATIONS REPRESENT WHAT USUALLY WILL
: BE CONTAINED IN A REGISTER DURING DX ISR PROCESSING. HOWEVER,
: AS SITUATIONS DICTATE REGISTERS MAY BE USED FOR DIFFERENT
: PURPOSES.
:
:DXISR: MOV R0, -(SP) ; SAVE HARDWARE REGISTERS
MOV R1, -(SP)
MOV R2, -(SP)
MOV R3, -(SP)
MOV R4, -(SP)
MOV R5, -(SP)
MOV TT1PTR, R2 ; CHECK FOR ZERO T/T ENTRY UPON INTERRUPT
TST (R2)
BNE LOOP ; NON-ZERO -- WERE OK
:
: NOTE -- AN INTERRUPT OCCURRED WITHOUT A TUMBLE TABLE
: ENTRY, THE ASSUMPTION IS THEN MADE THAT THE TUMBLE TABLE
: ENTRY HAS ALREADY BEEN PROCESSED
:
: JMP DXEXIT

```

3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411 005364 013702 013060
3412 005370 005712
3413 005372 001002
3414 005374 000137 006554
3415
3416
3417
3418
3419
3420
3421 005400 042777 090200 005062
3422 005406 010203
3423 005410 062703 001000
3424 005414 011223
3425 005416 011204
3426 005420 005022
3427 005422 011223
3428 005424 011205
3429 005426 005022
3430
3431
3432
3433 005430 032702 000777
3434
3435 005434 001002
3436 005436 013702 013062
3437 005442 010237 013060
3438
3439
3440
3441
3442

```

PROCESS TUMBLE TABLE ENTRIES
FOR CONVIENCE THE PROCESSING IS BEING PERFORMED
AT THE INTERRUPT LEVEL. IT IS SUGGESTED THAT IN
NORMAL PROCESSING ENVIRONMENTS THIS PROCESSING
BE DISTRIBUTED TO LESS PRIVILEGED PRIORITY LEVELS
SUCH AS THE FORK LEVEL IN RSX11-M.

THE INTERRUPT SERVICE LEVEL PROCESSING REQUIRED
BY THE DX11-B IS TO RELIEVE THE INTERRUPT (DONE BIT
IN DXCS) AND SCHEDULE A REQUEST FOR PROCESSING
AT ANOTHER LEVEL. THE LEVEL SCHEDULED TO PERFORM
THE PROCESSING SHOULD BE HIGH ENOUGH
TO PROTECT AGAINST TUMBLE TABLE OVERFLOW

THE TUMBLE TABLE ENTRIES ARE PROCESSED SEQUENTIALLY FROM THE
CIRCULAR BUFFER FILLED BY THE DX. AS EACH ENTRY IS RETRIEVED
FROM THE TUMBLE TABLE IT IS ZEROED. IT IS THIS MECHANISM
THAT ALLOWS THE PROGRAMMER TO DISCERN WHEN ALL ENTRIES HAVE
BEEN PROCESSED. WHEN ALL ENTRIES HAVE BEEN
RETRIEVED FROM THE TUMBLE TABLE THEN THE NEXT ACTION
IS PERFORMED TO THE DX. THE DX11-B NEVER ENTERS A ZERO
IN TUMBLE TABLE ENTRY 1.

LOOP:  MOV     TTPTR,R2           ;GET T/T PTR
       TST     (R2)           ;ANY ENTRIES LEFT IN T/T?
       BNE     10$           ;NO
       JMP     DXEXEC        ;NO, EXECUTE NEXT DX COMMAND

TUMBLE TABLE ENTRY AVAILABLE FOR PROCESSING
RESET THE DONE BIT (RELIEVE INTERRUPT)
COPY TUMBLE TABLE ENTRY TO DUPLICATE TUMBLE TABLE (FOR SYSTEM TESTS PUPP
RESET TUMBLE TABLE ENTRY (2 WORDS)

10$:   BIC     #DONE,DXCS      ;CLEAR DONE
       MOV     R2,R3         ;SET UP PTR TO DUP T/T
       ADD     #TTSIZE,R3
       MOV     (R2),(R3)+    ;SAVE T/T ENTRY #1
       MOV     (R2),TT1
       CLR     (R2)+         ;CLEAR T/T ENTRY #1
       MOV     (R2),(R3)+    ;SAVE T/T ENTRY #2
       MOV     (R2),TT2
       CLR     (R2)+         ;CLEAR T/T ENTRY #2

CHECK FOR POINTER WRAP AROUND

20$:   BIT     #TTSIZE-1,R2    ;AT END OF BUFFER?
       BNE     20$          ;NOTE -- POWER OF 2 BOUNDARY
       MOV     TTADDR,R2    ;YES, RESET PTR
       MOV     R2,TTPTR     ;SAVE T/T PTR

START PROCESSING TUMBLE ENTRY ENTRY
SAVE DEVICE ADDRESS
CHECK FOR SYSTEM RESET
VALIDATE DEVICE ADDRESS

```

```

3443
3444
3445
3446
3447 005446 010500
3448 005450 042700 177400
3449 005454 010037 013106
3450 005460 042777 000200 005002
3451 005466 032704 010000
3452 005472 001066
3453 005474 163700 012522
3454 005500 100403
3455 005502 120C37 013067
3456 005506 103405
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466 005510 004137 011506
3467 005514 013762
3468 005516 000137 006536
3469
3470
3471
3472 005522 00520C
3473 005524 004737 01023C

```

```

: NOTE -- IF SYSTEM RESET OCCURRED, THERE IS NO GUARANTEE
: THAT THE DEVICE ADDRESS WILL BE VALID.
:
MOV TT2,DEV ;GET DEV #
BIC #177400,DEV
MOV DEV,CDEV ;SAVE CURRENT DEVICE NUMBER
BIC #DONE,DXCS ;CLEAR DONE
BIT #SYSRST,TT1 ;SYSTEM RESET?
BNE PSYSRT ;YES, PERFORM SYSTEM RESET FUNCTION
SUB SDEV,DEV ;GET IN 0-7 RANGE - IF VALID
BMI 30$ ;INVALID DEVICE NUMBER
CMPB DEV,MAXDEV ;VALID DEVICE?
BLO 40$ ;YES, NOT TOO BIG
:
INVALID DEVICE ADDRESS - BITCH
:
AN INVALID DEVICE ADDRESS WILL GENERALLY INDICATE
A PROBLEM IN THE CONFIGURATION OF DX DEVICE
ADDRESSES. BASICALLY THE DX HAS BEEN STRAPPED
TO HANDLE DEVICE ADDRESSES WHICH OVERLAP WITH
OTHER DEVICES ON THE CHANNEL.
:
30$: JSR R1,INMES ;PRINT "INVALID DEVICE"
;WORD ILLMES
JMP DXAB ;ABORT DX11
:
COMPUTE ADDRESS OF SPECIFIED DEVICES STATUS TABLE
:
40$: INC DEV ;MAKE DEVICE NUMBER 1 -8
JSR PC,SUDEV ;SET UP ADDR OF DEV STAT TABLE

```

```

3474
3475
3476
3477
3478
3479
3480
3481
3482 005530 032704 020000
3483 005534 001402
3484 005536 000137 005776
3485 005542 032704 004000
3486 005546 001402
3487 005550 000137 006026
3488 005554 032704 040000
3489 005560 001402
3490 005562 000137 006102
3491 005566 032704 000100
3492 005572 001402
3493 005574 000137 006114
3494 005600 032704 100000
3495 005604 001402
3496 005606 000137 006244
3497 005612 032704 000200
3498 005616 001402
3499 005620 000137 006274
3500 005624 032704 000040
3501 005630 001402
3502 005632 000137 006456
3503 005636 032704 000020
3504 005642 001650
3505 005644 000137 006442

```

DECODE DX TUMBLE TABLE STATUS ENTRY

THE FOLLOWING PROCESS INDICATES THE ORDER IN WHICH THE TUMBLE TABLE STATUS ENTRY SHOULD BE DECODED. THIS ORDER IS IMPORTANT AND SHOULD BE ADHERED TO FOR MOST EMULATIONS.

```

SELRST: BIT #SELRST,TT1 :SELECTIVE RESET?
          BEQ TINDSC :NO, TEST INTERFACE DISC.
          JMP PSELRT :YES, PERFORM SELECTIVE RESET
TINDSC: BIT #INFDC,TT1 :INTERFACE DISCONNECT?
          BEQ TNXM :NO, CHECK NON-EXISTENT MEMOR
          JMP PINDC :YES, PERFORM INTER DISC
TNXM: BIT #NXM,TT1 :NON-EXISTENT MEMORY ERROR?
        BEQ TESEND :NO, ES END
        JMP PNXM :YES, PROCESS NON-EXISTENT MEMOR; ERROR
TESEND: BIT #ESEND,TT1 :WAS STATUS ACCEPTED?
        BEQ TPARER :NO, CHECK FOR PARITY ERROR
        JMP PESEND :YES, PERFORM STATUS ACCEPT
TPARER: BIT #PARER,TT1 :DID A PARITY ERROR OCCUR?
        BEQ TCHIS :NO, CHECK FOR CHIS
        JMP PPARER :YES, PROCESS PARITY ERROR
TCHIS: BIT #CHIS,TT1 :DID CHANNEL START A SELECTION SEQ?
        BEQ TCHEND :NO, CHECK FOR CHANNEL DATA END
        JMP PCHIS :YES, PROCESS SELECTION SEQUENCE
TCHEND: BIT #CHEND,TT1 :DID CHANNEL END OCCUR?
        BEQ TCUEND :NO, CHECK FOR CONTROL UNIT END
        JMP PCHEND :YES, PROCESS CHANNEL END
TCUEND: BIT #CUEND,TT1 :DID A CONTROL UNIT END OCCUR?
        BEQ LOOP :NO, IGNORE ENTRY -- ASSUME STACK STATUS
        JMP PCUEND :ES, PROCESS CONTROL UNIT END

```

.SBTTL DX11-B ISR (TUMBLE TABLE ENTRY PROCESSING LOGIC)

SYSTEM RESET OCCURRED FROM 360
CLEAR ALL DEVICE STATUS TABLES AND RESPECTIVE
SENSE BYTES
RESET DX ACTIVE FLAGS AND COMMAND CHAIN FLAG

3506
3507
3508
3509
3510
3511
3512
3513 005650 012700 000001
3514 005654 004737 010256
3515 005660 004737 010310
3516 005664 105063 000002
3517 005670 105063 000017
3518 005674 005063 000020
3519 005700 005063 000004
3520 005704 105063 000003
3521 005710 016301 000010
3522 005714 012702 000740
3523 005720 112704 000100
3524 005724 105737 012532
3525 005730 001402
3526 005732 113704 012534
3527
3528 005736 110421
3529 005740 005302
3530 005742 001375
3531 005744 005200
3532 005746 120037 013067
3533 005752 003740
3534 005754 105037 013070
3535 005760 105037 013074
3536 005764 042777 000400 004476
3537 005772 000137 005364
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547 005776 004737 010256
3548 006002 004737 010310
3549 006006 105063 000002
3550 006012 105037 013070
3551 006016 105037 013074
3552 006022 000137 005364
3553
3554
3555
3556
3557
3558
3559
3560
3561

```
PSYSRT: MOV #1,DEV ; START AT FIRST DEVICE
10$: JSR PC,CDEVST ; CLEAR DEVICE STATUS TABLE
JSR PC,CSPWST ; RESET SPW STATUS WORD UPON SYSTEM RESET
CLRB SSENSE(DTAB) ; CLEAR SENSE BYTE
CLRB SDRQ(DTAB) ; CLEAR THE READ REQUEST
CLR SMINS(DTAB) ; CLEAR THE BEG OF MANUAL INPUT ADDRESS
CLR SCURS(DTAB) ; RESET THE CURSOR
CLRB SSTAT(DTAB) ; CLEAR THE STATUS REGISTER
MOV SOUTB(DTAB),R1 ; SET UP TO CLEAR THE DISF AY BUFFER
MOV #DISPS2,R2 ; SET UP NUMBER OF CHARACTERS IN DISPLAY
MOVB #EBCDSP,R4 ; ASSUME 2848 DIAGNOSTIC TEST MODE
TSTB TSTYP ; WHAT TYPE OF TEST?
BEQ 20$ ; IF 2848, USE EBCDIC SPACE
MOVB FILLCH,R4 ; FRIEND TEST -- USE CURRENT FILL CHARACTER

20$: MOVB R4,(R1)+ ; USE THE FILL CHARACTER
DEC R2 ; ARE WE DONE?
BNE 20$ ; NO, LOOP TILL DONE
INC DEV ; TO NEXT DEVICE
CMPB DEV,MAXDEV ; ARE WE DONE?
BLE 10$ ; NO, CLEAR NEXT DEV STAT TABLE
CLRB DXACT ; CLEAR DX ACTIVE FLAG
CLRB CMDCHF ; CLEAR COMMAND CHAINING FLAG
BIC #CUBUSY,DXCS ; RESET CU BUSY FLAG
JMP LOOP ; PROCESS NEXT T/T ENTRY
```

CHANNEL ISSUED A SELECTIVE RESET
RESET THE DEVICE STATUS TABLE FOR THAT DEVICE + SENSE
NOTE: THE SEL RESET IS ISSUED AGAINST THE CURRENT
ACTIVE DEVICE

```
PSELRT: JSR PC,CDEVST ; CLEAR DEVICE STATUS TABLE
JSR PC,CSPWST ; RESET SPW STATUS RESPONSE
CLRB SSENSE(DTAB) ; CLEAR SENSE BYTE
CLRB DXACT ; CLEAR DX ACTIVE FLAG
CLRB CMDCHF ; CLEAR COMMAND CHAIN FLAG
JMP LOOP
```

INTERFACE DISCONNECT WAS ISSUED FROM THE 360
THIS IS DIRECTED TO A SPECIFIC DEVICE AND IS UNDER
360 PROGRAM CONTROL

IF THE DEVICE WAS ACTIVE
ITS DEVICE STATUS TABLE WILL BE CLEARED

```

3562      :
3563      :
3564      :
3565 006026 004737 010310      PINOSC: JSR      PC,CSPWST      ;CLEAR THE SPW STATUS RESPONSE
3566 006032 105763 000000      TSTB     SCMD(DTAB)      ;IS DEVICE ACTIVE?
3567 006036 001417              BEQ      20$          ;NO, IGNORE
3568 006040 004737 010256      JSR      PC,CDEVST      ;CLEAR THE DEVICE STATUS TABLE
3569 006044 012763 000003 000000  MOV     #CEDE,SCMD(DTAB);QUE DEV END + CHAN END
3570 006052 120037 013070      CMPB     DEV,DXACT      ;IS DEVICE USING DX NOW?
3571 006056 001002              BNE     10$          ;NO
3572 006060 105037 013070      CLRB     DXACT          ;YES, RELEASE DX
3573 006064 120037 013074      10$:   CMPB     DEV,CMDCHF ;DOES DEVICE HAVE CMD CHAIN SPEC?
3574 006070 001002              BNE     20$          ;NO, GET NEXT T/T ENTRY
3575 006072 105037 013074      CLRB     CMDCHF        ;YES, CLEAR FLAG
3576 006076 000137 005364      20$:   JMP      LOOP      ;GET NEXT T/T ENTRY
3577
3578
3579      :
3580      :
3581      :
3582      :
3583 006102 004137 011506      PNXM:   JSR      R1,INMES ;PRINT "NON EX MEM"
3584 006106 013706              .WORD   NXMMMSG
3585 006110 000137 006536      JMP     DXAB          ;ABORT DX AND RETURN TO EXEC
3586
3587
3588
3589      :
3590      :
3591      :
3592      :
3593      :
3594 006114 004737 010310      PESEND: JSR      PC,CSPWST ;RESET THE SPW STATUS BYTE
3595 006120 105763 000001      TSTB     SLCMD(DTAB)    ;DOES LAST COMMAND REQUIRE 2260 DISPLAY EMULATION?
3596 006124 001402              BEQ     10$          ;NO
3597 006126 004737 010402      JSR      PC,DISCTL      ;YES, FORMAT THE DISPLAY
3598 006132 126327 000000 000011 10$:   CMPB     SCMD(DTAB),#11 ;WAS ATTN ACCEPTED?
3599 006140 001003              BNE     20$          ;NO, CONTINUE
3600 006142 112763 000002 000017 20$:   MOV     #2,SRDRQ(DTAB);YES, INDICATE 360 ACCEPTANCE
3601 006150 105063 000001      CLRB     SLCMD(DTAB)
3602 006154 105037 013070      CLRB     DXACT          ;CLEAR DX ACTIVE FLAG
3603 006160 004737 010256      JSR      PC,CDEVST      ;CLEAR THE DEVICE STATUS TABLE
3604 006164 032704 000004      BIT     #CMDCHN,TT1     ;WAS COMMAND CHAINING SPECIFIED?
3605 006170 001402              BEQ     30$          ;NO
3606 006172 110037 013074      MOV     DEV,CMDCHF      ;YES, SAVE THE DEVICE NUMBER
3607 006176 123727 013066 000105 30$:   CMPB     DXSTPF,#'E     ;WAS STOP ON END SEQ SPEC?(SE)
3608 006204 001552              BEQ     STPDX          ;YES, DISABLE THE DX
3609
3610 006206 032704 000010      BIT     #ISSREJ,TT1    ;WAS AN ISS REJ DETECTED?
3611 006212 001412              BEQ     50$          ;NO, EXIT
3612
3613      :
3614      :
3615      :
3616      :
3617      :
INIT SELECTION SEQUENCE WAS REJECTED BY DX (FAST CU BUSY SEQUENCE)
IF FREIND TEST MODE -- QUEUE CONTROL UNIT END
ON UNIT COMPLETING TRANSFER
IF 2848 DIAGNOSTIC TEST MODE - QUEUE CONTROL UNIT END

```

RESPONSE ON LOW ORDER CHANNEL ADDRESS

3618
3619
3620
3621
3622
3623
3624
3625
3626 006214 105737 012532
3627 006220 001004
3628 006222 012700 000001
3629 006226 004737 010230
3630 006232 112763 000010 000000
3631 006240 000137 005364
3632
3633
3634
3635
3636
3637 006244 004137 011506
3638 006250 013736
3639 006252 123727 013066 000120
3640 006260 001524
3641 006262 152763 000002 000003
3642 006270 000137 005612
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652 006274 004737 010310
3653
3654 006300 123727 013066 000111
3655 006306 001511
3656 006310 032704 000001
3657 006314 001022
3658
3659
3660
3661 006316 105005
3662 006320 000305
3663 006322 105705
3664 006324 001437
3665 006326 120527 000003
3666 006332 001434
3667 006334 020527 000012
3668 006340 003405
3669 006342 004137 011506
3670 006346 014050
3671 006350 000137 006536
3672 006354 110563 000000
3673 006360 000421

THE 2648 DEVICE EMULATION IS EXPECTED TO ISSUE
A CONTROL UNIT END ON THE LOW ORDER DEVICE ADDRESS
OF THE CONTROL UNIT.
MOST OTHER 360/370 DEVICES ARE EXPECTED TO ISSUE
CONTROL UNIT END ON THE DEVICE COMPLETING THE OPERATION.

TSTB TSTTYP ;PROCESS SEPARATELY IF FRIEND
BNE 40\$;FRIEND --QUEUE CU END ON BUSY UNIT
MOV #1,DEV ;SET UP TO SEND CUE ON LOW ORDER CONTROLLER ADDR
JSR PC,SUDEV ; X
MOVB #QCUE,SCMD(DTAB) ;QUEUE CONTROL UNIT END
JMP LOOP ;LOOP BACK AND PROCESS NEXT TUMBLE TABLE ENTRY

40\$:
50\$:

PARITY ERROR WAS DETECTED

PPARER: JSR R1,INMES ;PRINT "PARITY ERROR"
 .WORD PARMES
CMPB DXSTPF,#'P ;STOP ON PARITY ERROR??
BEQ STPDX ;YES, DISABLE THE DX
BISB #UCHK,SSTAT(DTAB) ;SET UNIT CHECK IN STATUS WORD
JMP TCHIS ;CONTINUE WITH TUMBLE TABLE INTERROGATION

CHANNEL INITIATED SELECTION SEQUENCE
THUS FAR THE DEVICE NUMBER HAS BEEN VALIDATED
AND THE COMMAND CHECKED BY THE DX

TT2 CONTAINS THE COMMAND TO BE EXECUTED

PCHIS: JSR PC,CSPWST ;RESET THE SPW STATUS BYTE
 ;ON NEXT CHANNEL INITIATED SELECTION SEQUENCE
CMPB DXSTPF,#'I ;WAS STOP ON ISS SPECIFIED 'SI'
BEQ STPDX ;YES, DISABLE DX
BIT #CMOREJ,TT1 ;WAS COMMAND REJECTED BY DX?
SNE 20\$;YES, COMMAND REJECTED BY THE DX

VALID COMMAND, SET UP TO PROCESS IT

CLRB TT2 ;RESET DEVICE ADDRESS BITS
SWAB TT2 ;COMMAND TO L.O. BYTE
TSTB TT2 ;TEST I/O COMMAND?
BEQ 50\$;YES, IGNORE
CMPB TT2,#NOP ;WAS COMMAND A NOP?
BEQ 50\$;YES, IGNORE IT
CMP TT2,#12 ;IS THIS A VALID COMMAND?
BLE 10\$;YES, QUEUE TO BE EXECUTED
JSR R1,INMES ;NO -- REPORT AN ILLEGAL COMMAND RECIEVED FROM THE DX
 .WORD INVLDC
JMP DXAB ;AND ABORT THE PROGRAM
MOVB TT2,SCMD(DTAB) ;QUEUE COMMAND TO BE PROCESSED
BR 50\$;EXIT + PROCESS NEXT T/T ENTRY

10\$:


```

3674
3675
3676
3677 006362 105763 000016
3678 006366 001404
3679
3680
3681
3682 006370 052763 000100 000002
3683 006376 000412
3684 006400 032704 100000
3685 006404 001404
3686
3687
3688
3689
3690 006406 052763 000040 000002
3691 006414 000403
3692
3693
3694
3695
3696 006416 052763 000200 000002
3697
3698
3699
3700
3701 006424 120037 013074
3702 006430 001002
3703 006432 105037 013074
3704 006436 000137 005364
3705
3706
3707
3708
3709
3710
3711
3712
3713 006442 105037 013070
3714 006446 004737 010342
3715 006452 103017
3716
3717
3718
3719
3720 006454 000404
3721
3722
3723
3724
3725
3726
3727
3728 006456 105037 013070
3729 006462 004737 010342

```

COMMAND WAS REJECTED, DETERMINE WHY
20\$: TSTB SONLF(DTAB) ; IS DEVICE ON LINE?
BEQ 30\$; YES, TEST PARITY ERROR

DEVICE OFF-LINE -- RESPOND INTERVENTION REQUIRED SENSE CONDITION

BIS #INTREQ, SSENSE(DTAB) ; SET INTERVENTION REQUIRED IN SENSE BYTE
BR 50\$; FINISH UP CHANNEL INITIATED SELECTION PROCESS
30\$: BIT #PARER, TTI ; WAS A PARITY ERROR DETECTED?
BEQ 40\$; NO, MUST BE ILLEGAL COMMAND

COMMAND WAS REJECTED BECAUSE OF A PARITY ERROR
SET UP BUS OUT SENSE RESPONSE

BIS #BUSOUT, SSENSE(DTAB) ; SET BUS OUT FLAG
BR 50\$; EXIT

INVALID COMMAND RECEIVED FROM 360
SET UP COMMAND REJECT SENSE RESPONSE

40\$: BIS #SCMDRJ, SSENSE(DTAB) ; SET CMD REJ FLAG

CHANNEL INITIATED SELECTION SEQUENCE COMMON EXIT LOGIC
IF SELECTED DEVICE HAS COMMAND CHAINING IN AFFECT -- KILL IT

50\$: CMPB DEV, CMDCHF ; DOES DEVICE HAVE COMMAND CHAINING SPECIFIED?
BNE 60\$; NO, GET NEXT TUMBLE TABLE ENTRY
CLRB CMDCHF ; YES, CLEAR THE COMMAND CHAINING FLAG
60\$: JMP LOOP ; AND GET THE NEXT T/T ENTRY

CONTROL UNIT END OF DATA TRANSFER WAS DETECTED
IF TRANSFER COMPLETE PREPARE ENDING SEQ RESP
IF TRANSFER INCOMPLETE INCR BUFFER ADDRESS
DECR BYTE COUNT

PCUEND: CLRB DXACT ; CLEAR DX ACTIVE FLAG
JSR PC, MUXEND ; HANDLE MUX DATA TRANSFER COMPLETION
BCC PCHEX ; IF SEL CHAN OR MUX DAT NOT DONE, MERELY EXIT

MUX DATA TRANSFER COMPLETE
TREAT SAME AS SEL CHANNEL DONE

BR PCHEX

CHANNEL END WAS DETECTED
PREPARE ENDING SEQUENCE RESPONSE

PCHEX: CLRB DXACT ; CLEAR DX ACTIVE FLAG
JSR PC, MUXEND ; IF MUX CHANNEL HANDLE DATA TRANSFER

```

3730 006466 123727 013066 000104 PCHEM1: CMPB DXSTPF,#'D ;STOP ON DATA TRANSFER DONE" SD.
3731 006474 001416 BEQ STPDX ;YES, DISABLE DX
3732 006476 112763 000003 000000 MOVB #CEDE,SCMD(DTAB);QUE" END SEQ RESPONSE
3733 006504 167763 003766 000014 SUB #DXBC,SRBYTC(DTAB);SAVE REMAINING BYTE COUNT
3734 006512 032704 100000 PCHEX: BIT #PAREA,TT1 ;WAS A PARITY ERROR SENSED?
3735 006516 001403 BEQ IOS ;NO, PROCESS NEXT TUMBLE TABLE ENTRY
3736 006520 152763 000020 000002 IOS: BISB #EQCHK,SSENSE(DTAB);YES, SET EQUIP CHECK IN SENSE
3737 006526 000137 005364 JMP LOOP ;LOOP BACK + PROCESS NEXT TT ENTRY
3738
3739
3740
3741
3742
3743
3744 006532 105037 013066 STPDX: CLRB DXSTPF ;CLEAR STOP FLAG
3745
3746
3747
3748
3749
3750
3751 006536 042777 001100 003724 DXAB: BIC #DXONLN:DXENB,#DXCS ;DISABLE THE DX
3752 006544 105237 013072 INCB DXABFL ;SET THE DX ABORT FLAG SO THE
3753 ;DX REGISTERS WILL BE PRINTED
3754 006550 000137 010212 JMP DXEXIT ;EXIT FROM INTERRUPT
3755 .SBTTL DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)
3756
3757 EXECUTE NEXT COMMAND FOR THE DX
3758
3759 006554 105737 012526 DXEXEC: TSTB CHTYPE ;CHANNEL TYPE 0=M, 1=S
3760 006560 001002 BNE SEX ;SELECTOR CHANNEL EXEC
3761 006562 000137 007210 JMP MEX ;MULTIPLEXER EXEC
3762
3763 SEX -- SELECTOR CHANNEL EXECUTIVE
3764
3765 SEX EXECUTES COMMANDS FOR THE DX TO A SELECTOR CHANNEL
3766
3767 ON A SELECTOR CHANNEL A COMMAND WILL BE COMPLETED
3768 BEFORE ATTEMPTING TO EXECUTE A COMMAND ON ANOTHER
3769 DEVICE, ISS-DATA TRANSFER-ES.
3770
3771 DATA TRANSFERS ARE COMPLETED IN ONE BURST
3772
3773 SEX: MOV SELDEV,DEV ;GET SEL DEV #
3774 006572 004737 010230 IOS: JSR PC,SUDEV ;SET UP DEV STATUS TABLE ADDR
3775 006576 105763 000000 TSTB SCMD(DTAB) ;ANY JOB TO DO?
3776 006602 001030 BNE 60$ ;YES, EXECUTE IT
3777 006604 105737 013074 TSTB CMDCHF ;WAS COMMAND CHAINING SPECIFIED
3778 006610 001402 BEQ 30$ ;NO
3779 006612 000137 010212 JMP DXEXIT ;YES, WAIT FOR COMMAND
3780 006616 126327 000017 000001 30$: CMPB SRDRQ(DTAB),#1 ;IS ATTENTION TO BE SENT?
3781 006624 001004 BNE 40$ ;NO, CONTINUE
3782 006626 112763 000011 000000 MOVB #11,SCMD(DTAB) ;YES, SET UP TO SEND THE ATTENTION
3783 006634 000413 BR 60$ ;FOR THE READ MANUAL INPUT
3784 006636 005200 40$: INC DEV ;TO NEXT DEV
3785 006640 120037 007367 CMPB DEV,MAXDEV ;HAVE WE TRIED THE HIGHEST DEVICE?

```

E07

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
 CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 82
 DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)

SEG 0082
 SEG 0082

3786	006644	003402		BLE	SOS	:NO
3787	006646	012700	000001	MOV	#1,DEV	:YES, RESTART AT FIRST DEVICE
3788	006652	020037	013076	SOS:	CMP DEV,SELDEV	:IS THIS WHERE IT ALL STARTED?
3789	006656	001345		BNE	IOS	:NOPE, TEST THIS DEVICE
3790	006660	000137	010212	JMP	DXEXIT	:EXIT -- NO TASKS PENDING
3791				:		
3792				:	THERE IS A JOB TO DO, LETS DO IT	
3793				:		
3794	006664	116304	000000	SOS:	MOVB SCMD(DTAB),R4	:COMMAND TO INDEX
3795	006670	005304		DEC	R4	:SCALE TO 0 - 11
3796	006672	006304		ASL	R4	:MAKE WORD ADDRESS
3797	006674	010037	013076	MOV	DEV,SELDEV	:SAVE CURRENT DEVICE ADDR
3798	006700	000174	006704	JMP	JSCMDTB(R4)	:EXECUTE THE COMMAND
3799	006704	006730		SCMDTB:	.WORD	WRITE
3800	006706	007020			.WORD	SRMI
3801	006710	010042			.WORD	ESEQ
3802	006712	010140			.WORD	SENSCM
3803	006714	006730			.WORD	SWRITE
3804	006716	007120			.WORD	SREAD
3805	006720	007770			.WORD	ERASCM
3806	006722	007750			.WORD	COMUNE
3807	006724	007760			.WORD	SATIN
3808	006726	007320			.WORD	SSRMI

:1 = WRITE FULL BUFFER
 :2 = READ MANUAL INPUT
 :3 = ENDING SEQUENCE
 :4 = SENSE COMMAND
 :5 = WRITE LINE ADDRESS
 :6 = READ FULL BUFFER
 :7 = ERASE COMMAND
 :10 = CONTROL UNIT END
 :11 = SEND ATTENTION TO 360
 :12 = READ SHORT MANUAL INPUT

3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864

.....
: COMMANDS SPECIFICALLY FOR THE SELECTOR CHANNEL
:

.....
: WRITE COMMAND RECEIVED FROM 360
: PREPARE TO GET DATA FROM 360
: BOTH WRITE AND WRITE LINE ADDRESS COME HERE

```
SWRITE: MOV SINBF(DTAB),DXBA ;SET UP BUFFER ADDRESS
SUB PHYOFF,DXBA ;FOR VIRTUAL MEMORY -- OFFSET FOR PHYSICAL ADDRESS
MOV DEVCON,R2 ;COMPUTE DEVICE ADDRESS
ADD DEV,R2
MOV R2,DXCA
MOV #DISPSZ-1,DXBC ;SET UP BYTE COUNT FOR MAX. WRITE LINE ADDRESS
MOV SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
MOV DEV,DXACT ;SET DX ACTIVE FLAG
CLR SRBYTC(DTAB) ;RESET REMAINING BYTE COUNT
BIS DXWR,DXCS ;START TRANSFER
JMP DXEXIT ;RETURN FROM INTERRUPT
```

.....
: PERFORM READ MANUAL INPJT COMMANDS

```
SRMI:
SSRMI: TSTB TSTTYP ;IS TEST FOR FRIEND?
BNE SREAD ;YES, TREAT ALL READS AS READ FULL BUFFER
CLRB SSENSE(DTAB) ;RESET THE SENSE BYTE
TSTB SRDRQ(DTAB) ;WAS A READ REQUESTED?
BNE IOS ;YES, CONTINUE
JMP ESEQ ;NO, TREAT AS A NOP -- END SEG ONLY
IOS: CLRB SRDRQ(DTAB)
MOV SMINS(DTAB),DXBA ;SET UP STARTING ADDRESS
MOV SOUTB(DTAB),R2 ;DETERMINE ENDING ADDRESS
ADD SCURS(DTAB),R2
DEC R2
SUB DXBA,R2 ;COMPUTE BYTE COUNT
BPL IOS ;INSURE VALID BYTE COUNT
JMP ESEQ ;ILLEGAL
20$: NEG R2
MOV R2,DXBC ;SET UP DX'S BYTE COUNT
SUB PHYOFF,DXBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
BR SRDIO ;START THE READ
```

.....
: READ COMMAND RECEIVED FROM 360
: PREPARE TO SEND DISP BUFFER TO 360

```
SREAD: MOV SOUTB(DTAB),DXBA ;SET UP BUFFER ADDRESS
SUB PHYOFF,DXBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
```

3865	007134	012777	177040	003334		MOV	#-DISPSZ,DXBC	;SET UP BYTE COUNT
3866	007142	116363	000000	000001	SPC10:	MOV	SCMD(DTAB),SLCMD(DTAB)	;SAVE CODE OF LAST COMMAND
3867	007150	013702	013110			MOV	DEVCON,F2	;COMPUTE DEVICE ADDRESS
3868	007154	060002				ADD	DEV,R2	
3869	007156	110277	003304			MOV	R2,DXCA	
3870	007162	105063	000002			CLRB	SSENSE(DTAB)	;CLEAR SENSE BYTE
3871	007166	110037	013070			MOV	DEV,DXACT	;SET DX ACTIVE FLAG
3872	007172	005063	000014			CLR	SRBYTC(DTAB)	;RESET REMAINING BYTE COUNT
3873	007176	052777	000005	003264		BIS	#DXRD,DXCS	;START TRANSFER
3874	007204	000137	010212			JMP	DXEXIT	;RETURN FROM INTERRUPT
3875								
3876								

```

3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891 007210 105737 013074
3892 007214 001402
3893 007216 000137 010212
3894 007222 013700 013076
3895 007226 004737 010230
3896 007232 105763 000000
3897 007236 001023
3898 007240 126327 000017 000001
3899 007246 001004
3900 007250 112763 000011 000000
3901 007256 000413
3902
3903
3904
3905
3906
3907
3908 007260 005200
3909 007262 120037 013067
3910 007266 003402
3911 007270 012700 000001
3912 007274 020037 013076
3913 007300 001352
3914 007302 000137 010212
3915
3916
3917
3918 007306 116304 000000
3919 007312 005304
3920 007314 006304
3921 007316 010037 013076
3922 007322 000174 007326
3923 007326 007352
3924 007330 007502
3925 007332 010042
3926 007334 010140
3927 007336 007352
3928 007340 007600
3929 007342 007770
3930 007344 007750
3931 007346 007760
3932 007350 007502

```

```

.SBTTL DX11-B ISR (MULTIPLEXER CHANNEL COMMANDS)
MEX-- MULTIPLEXER CHANNEL EXECUTIVE
MEX EXECUTES COMMANDS FROM THE DX ON A MULTIPLEXER CHANNEL
ON A MULTIPLEXER MULTIPLE DEVICE REQUESTS WILL BE
INTERLEAVED. THIS WILL PROHIBIT A TIME OUT TO OCCUR
IF A DEVICE IS NOT SERVICED UNTIL ALL OTHER DEVICES
BEFORE IT.
DATA TRANSFERS ARE DONE IN 4 BYTE BLOCKS, SO AS TO NOT
HOG THE CHANNEL
MEX: TSTB CMDCHF ; IS COMMAND CHAINING SPECIFIED?
      BEQ 10$ ; NO, CONTINUE
      JMP DXEXIT ; YES, LEAVE DX FREE
10$: MOV MDEV,DEV ; GET LAST DEVICE ADDR THAT HAD A COMMAND
30$: JSR PC,SUDEV ; COMPUTE ADDR OF DEV STAT TABLE
      TSTB SCMD(DTAB) ; ANY JOB TO DO?
      BNE 50$ ; YES, EXECUTE IT
      CMPB SRDRQ(DTAB),#1 ; IS ATTENTION REQUESTED?
      BNE 40$ ; NO, CONTINUE
      MOVB #11,SCMD(DTAB) ; YES, QUEUE ATTENTION
      BR 50$ ; FOR THE READ MANUAL INPUT
      NO TASK PENDING FOR CURRENT DEVICE
      BUMP TO INTERROGATE NEXT DEVICE ON CONTROL UNIT
      THIS CODE WILL REPEAT SEQUENCES WHICH MAY HAVE RUN INTO
      A LOCKOUT CONDITION IN THE DX.
40$: INC DEV ; INCR TO NEXT DEVICE NUMBER
      CMPB DEV,MAXDEV ; WAS DEVICE NUMBER WRAPPED AROUND?
      BLE 45$ ; NO, SEE IF ALL DEVICES HAVE BEEN INTERROGATED
      MOV #1,DEV ; YES, RESET THE DEVICE NUMBER
45$: CMP DEV,MDEV ; NO JOB HERE, HAVE WE CHECKED ALL DEVICES?
      BNE 30$ ; NO, EXAMINE NEXT DEVICE
      JMP DXEXIT ; YES, EXIT FROM ISR
      THIS DEVICE HAS A JOB TO DO, EXECUTE IT
50$: MOVB SCMD(DTAB),R4 ; COMMAND TO INDEX
      DEC R4 ; SCALE TO 0 - 11
      ASL R4 ; MAKE INTO WORD ADDRESS
      MOV DEV,MDEV ; SAVE CURRENT DEVICE ADDRESS
      JMP @MCMDBT(R4) ; EXECUTE THE COMMAND
MCMDBT: .WORD MWRITE ; 1 = WRITE FULL BUFFER
        .WORD MRMI ; 2 = READ MANUAL INPUT
        .WORD ESEQ ; 3 = ENDING SEQUENCE
        .WORD SENSCH ; 4 = SENSE COMMAND
        .WORD MWRITE ; 5 = WRITE LINE ADDRESS
        .WORD MREAD ; 6 = READ FULL BUFFER
        .WORD ERASC ; 7 = ERASE COMMAND
        .WORD CONUNE ; 10 = CONTROL UNIT END
        .WORD SATTN ; 11 = SEND ATTENTION TO 360
        .WORD MSRMI ; 12 = READ SHORT MANUAL INPUT

```

```

3933 .....
3934 .....
3935 .....
3936 .....
3937 .....
3938 .....
3939 .....
3940 .....
3941 .....
3942 .....
3943 007352 005763 000014 MWRITE: TST SRBYTC(DTAB) ;WRITE IN PROGRESS?
3944 007356 001011 BNE 10$ ;YES, SEND OUT MORE DATA
3945 007360 016363 000006 000012 MOV SINBF(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
3946 007366 163763 013104 000012 SUB PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS
3947 007374 012763 000741 000014 MOV #DISPSZ+1,SRBYTC(DTAB) ;SET UP BUFFER FOR MAX SIZE
3948 007402 016377 000012 003064 10$: MOV SBUFA(DTAB),DXBA ;OUTPUT BUFFER ADDR TO DX
3949 007410 013702 013110 MOV DEVCON,R2 ;COMPUTE DEVICE ADDRESS
3950 007414 060002 ADD DEV,R2
3951 007416 110277 003044 MOVB R2,DXCA
3952 007422 012777 177774 003046 MOV #-4,DXBC ;START BYTE COUNT AT 4
3953 007430 026327 000014 000004 CMP SRBYTC(DTAB),#4 ;IS LESS THEN 4 BYTES LEFT?
3954 007436 002005 BGE 20$ ;NO, START TRANSFER
3955 007440 016302 000014 MOV SRBYTC(DTAB),R2 ;YES, USE REMAINING BYTE COUNT
3956 007444 005402 NEG R2
3957 007446 010277 003024 MOV R2,DXBC
3958 007452 105063 000002 20$: CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
3959 007456 116363 000000 000001 MOVB SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
3960 007464 110037 013070 MOVB DEV,DXACT ;SET ACTIVE FLAG
3961 007470 152777 000003 002772 BISB #DXWR,DXCS ;START THE TRANSFER
3962 007476 000137 010212 JMP DXEXIT ;RETURN FROM INTERRUPT
3963 .....
3964 .....
3965 .....
3966 .....
3967 .....
3968 .....
3969 .....
3970 .....
3971 007502 105737 012532 MRM1 =
3972 007506 001034 MSRMI: TST ISTTYP ;FRIEND OR 2848 DIAG?
3973 007510 005763 000014 BNE MREAD ;FRIEND -- TREAT AS READ FULL BUFFER
3974 007514 001031 TST SRBYTC(DTAB) ;ANY DATA LEFT TO TRANSFER?
3975 007516 105063 000002 BNE MREAD ;BRANCH IF YES TO CONTINUE
3976 007522 105763 000017 CLRB SSENSE(DTAB) ;RESET THE SENSE BYTE
3977 007526 001002 TST SRDRQ(DTAB) ;WAS THE READ REQUESTED?
3978 007530 000137 010042 BNE 20$ ;YES, CONTINUE
3979 007534 105063 000017 10$: JMP ESEQ ;NO, RETURN AN ENDING SEQ RESP DEICE
3980 007540 016363 000020 000012 20$: CLRB SRDRQ(DTAB) ;CLEAR THE READ REQUEST
3981 007546 016302 000010 MOV SMINS(DTAB),SBUFA(DTAB) ;SET UP THE ADDRESS OF THE DATA
3982 007552 066302 000004 MOV SOUTB(DTAB),R2 ;COMPUTE THE BYTE COUNT
3983 007556 005302 ADD SCURS(DTAB),R2 ;END - START
3984 007560 166302 000012 DEC R2
3985 007564 100761 SUB SBUFA(DTAB),R2 ;COMPUTE THE BYTE COUNT
3986 007566 010263 000014 BMI 10$ ;NEGATIVE -- SOMETHING IS WRONG
3987 007572 163763 013104 000012 MOV R2,SRBYTC(DTAB) ;SAVE FOR READ DRIVER
3988 SUB PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS

```

```

3989      :      FALL THROUGH TO NORMAL READ BUFFER ROUTINE
3990      :
3991      :
3992      :
3993      :
3994      :      READ COMMAND RECEIVED FROM 360
3995      :      PREFARE TO SEND 4 BYTES OF DATA TO THE 360
3996      :
3997 007600 116363 000000 000001 MREAD: MOVB  SCMD(DTAB),SLCMD(DTAB) ;SAVE CODE OF LAST COMMAND FOR DISPLAY CONTROL
3998 007606 005763 000014          TST  SRBYTC(DTAB) ;READ IN PROGRESS?
3999 007612 001011          BNE  10$ ;YES SEND OUT MORE DATA
4000 007614 016363 000010 000012  MOV  SOUTB(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
4001 007622 163763 013104 000012  SUB  PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS
4002 007630 012763 000740 000014  MOV  #DISPS2,SRBYTC(DTAB) ;SET UP TOTAL BYTE COUNT
4003 007636 016377 000012 002630 10$:  MOV  SBUFA(DTAB),DXBA ;SEND BUFFER ADDR TO DX
4004 007644 013702 013110          MOV  DEVCON,R2 ;COMPUTE DEVICE ADDR
4005 007650 060002          ADD  DEV,R2
4006 007652 110277 002610          MOVB R2,DXCA ;OUTPUT THE DEVICE ADDRESS
4007 007656 012777 177774 002612  MOV  #-4,DXBC ;OUTPUT THE BYTE COUNT -4-
4008 007664 026327 000014 000004  CMP  SRBYTC(DTAB),#4 ;SEE IF REMAINING BYTE COUNT LESS THAN 4
4009 007672 002005          BGE  20$
4010 007674 016302 000014          MOV  SRBYTC(DTAB),R2 ;SET UP BYTE COUNT
4011 007700 005402          NEG  R2
4012 007702 010277 002570          MOV  R2,DXBC ;OUTPUT THE NEW BYTE COUNT -- LT 4
4013 007706 105063 000002 20$:  CLRB SSSENSE(DTAB) ;CLEAR SENSE AND SET DX ACTIVE FLAG
4014 007712 110037 013070          MOVB DEV,DXACT ;SET DEVICE ACTIVE FLAG FOR SOFTWARE
4015      :
4016      :      BEFORE TRANSMIT IS STARTED SET BUSY FLAG IN DX11 STATUS
4017      :      TABLE FOR DEVICE
4018      :
4019      :
4019 007716 010002          MOV  DEV,R2 ;COMPUTE ADDRESS OF SPW ENTRY
4020 007720 063702 013110          ADD  DEVCON,R2 ; X
4021 007724 060202          ADD  R2,R2 ; X
4022 007726 063702 013114          ADD  STSPW,R2 ;ADD IN SPW BASE ADDRESS
4023 007732 052712 000020          BIS  #BSY,(R2) ;SET UNIT BUSY FLAG
4024 007736 152777 000005 002524  BISB #DYR0,DXCS ;START THE DX READING
4025 007744 000137 010212          JMP  DXEXIT
    
```



```

4026 .SBTTL DX11-B ISR (MULTIPLEXER AND SELECTOR CHANNEL COMMANDS)
4027 .....
4028
4029
4030
4031 .....
4032
4033
4034
4035
4036 .....
4037 007750 152763 000040 000003 CONUNE: B1SB #CUE,SSTAT(DTAB) ;PUT IN STATUS BYTE
4038 007756 000434 BR STOUT ;OUTPUT TO CHANNEL
4039
4040
4041 .....
4042
4043
4044 007760 152763 000200 000003 SATTN: B1SB #ATTN,SSTAT(DTAB) ;PUT IN STATUS BYTE
4045 007766 000430 BR STOUT ;OUTPUT TO THE 360
4046
4047
4048 .....
4049
4050
4051 007770 016304 000010 ERASCM: MOV SOUTB(DTAB),R4 ;SET UP BEG OF DISPLAY BUFFER
4052 007774 012705 000740 MOV #DISPSZ,R5 ;SET UP COUNTER
4053 010000 112702 000100 MOVB #EBCDSP,R2 ;SET BUFFER FILL FOR 2848 DIAG
4054 010004 105737 012532 TSTB TSTYP ;IS TEST BEING RUN FOR 2848 RESPONDER
4055 010010 001402 BEQ 10$ ;YES, FILL BUFFER WITH EBCDIC SPACE
4056 010012 113702 012534 MOVB FILLCH,R2 ;NO, USE CURRENT FILL CHARACTER
4057 010016 110224 10$: MOVB R2,(R4)+ ;MOVE FILL CHARACTER TO BUFFER
4058 010020 005305 DEC R5 ;DECR COUNTER
4059 010022 001375 BNE 10$ ;NOT DONE, DO NEXT CHAR
4060 010024 005063 000004 CLR SCURS(DTAB) ;RESET THE CURSOR
4061 010030 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
4062 010034 112763 000003 000000 MOVB #CEDE,SCMD(DTAB) ;CHANGE COMMAND TO PRESENT END SEQ
4063
4064 .....
4065
4066
4067
4068 .....
4069
4070
4071 010042 152763 000014 000003 ESEQ: B1SB #CE!DE,SSTAT(DTAB) ;SET CH END + DEV END
4072
4073
4074
4075 .....
4076
4077
4078
4079
4080
4081

```

PRESENT CONTROL UNIT END TO CHANNEL

SEND THE ATTENTION BIT TO THE 360

ERASE THE DISPLAY

FALL THROUGH TO PRESENT ENDING STATUS

PRESENT ENDING STATUS TO CHANNEL

PRESENT STATUS TO CHANNEL

THE STATUS IS BOTH PUT IN THE DX11-B SPW TABLE AND SENT TO THE CHANNEL. CONDITIONS CAN OCCUR WHICH CAUSE THE STATUS ENTRY TO THE CHANNEL TO BE IGNORED.

```

4082 010050 013702 013110          STOUT:  MOV     DEVCON,R2          ;OUTPUT DEVICE ADDRESS
4083 010054 060002                    ADD     DEV,R2
4084 010056 110277 002404          MOVSB  R2,DXCF
4085 010062 132763 000002 0000C3  BITB   #UCHK,SSTAT(DTAB) ;IS THE UNIT CHECK BIT SET?
4086 010070 001403                    BEQ    10$                ;NO, TRANSMIT THE STATUS
4087 010072 112763 000002 0000C3  MOVSB  #UCHK,SSTAT(DTAB) ;YES, THEN SEND ONLY UNIT CHECK
4088
4089      :
4090      IF MULTIPLEXER CHANNEL
4091      CLEAR ANY PENDING STATUS IN SPW STATUS ENTRY
4092      (PROBABLY "BUSY")
4093 010100 105737 012526          10$:  TSTB   CHTYPE                ;SELECTOR CHANNEL?
4094 010104 001004                    BNE   20$                ;YES, DON'T CLEAR STATUS IN SPW TABLE
4095 010106 060202                    ADD   R2,R2              ;COMPUTE ADDRESS OF SPW STATUS ENTRY
4096 010110 063702 013114          ADD   STSPW,R2           ;OFFSET BY BASE OF SPW TABLE
4097 010114 105012                    CLRB  (R2)               ;CLEAR SPW STATUS ENTRY
4098
4099      :
4100      OUTPUT THE STATUS TO THE CHANNEL
4101 010116 116377 000003 002346  20$:  MOVSB  SSTAT(DTAB),DXOS ;OUTPUT STATUS TO CHANNEL
4102 010124 152777 000007 002336  BISH  #DXST,DXCS        ;PRESENT TO CHANNEL
4103 010132 110037 013070          MOVSB  DEV,DXACT         ;SET DX ACTIVE FLAG
4104 010136 000425                    BR    DXEXIT             ;RETURN FROM INTERRUPT
4105
4106      :
4107      :
4108      :
4109      SENSE COMMAND DESIRED BY 360
4110 010140 012777 000002 002326  SENSEM: MOV    #SENSE,DXBA      ;SET UP ADDRESS OF SENSE BYTE
4111 010146 060377 002322          ADD    DTAB,DXBA
4112 010152 163777 013104 002314  SUB    PHYOFF,DXBA       ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
4113 010160 013702 013110          MOV    DEVCON,R2        ;COMPUTE DEVICE ADDRESS
4114 010164 060002                    ADD    DEV,R2
4115 010166 110277 002274          MOVSB  R2,DXCA
4116 010172 012777 177777 002276  MOV    #-1,DXBC         ;TRANSFER 1 BYTE
4117 010200 110037 013070          MOVSB  DEV,DXACT        ;SET DX ACTIVE FLAG
4118 010204 052777 000005 002256  BIS    #DXRD,DXCS       ;START TRANSFER
4119
4120      :
4121      :
4122      :
4123      EXIT FROM THE DX ISR
4124 010212 012605          DXEXIT: MOV    (SP)+,R5      ;RESTORE REGISTERS
4125 010214 012604          MOV    (SP)+,R4
4126 010216 012603          MOV    (SP)+,R3
4127 010220 012602          MOV    (SP)+,R2
4128 010222 012601          MOV    (SP)+,R1
4129 010224 012600          MOV    (SP)+,R0
4130 010226 000002          RTI
  
```

```

4131 .SBTTL DX11-B ISR (UTILITY SUBROUTINES)
4132
4133 SET UP ADDR OF DEVICE STATUS TABLE
4134
4135 CALLING SEQUENCE
4136 ..... RO = DEV #
4137 JSR PC,SUDEV
4138 ..... RETURN
4139 ..... R3 = ADDRESS OF DEVICE TABLE
4140
4141 ONLY REGISTER R3 IS MODIFIED BY THIS SUBROUTINE
4142
4143 SUDEV: MOV SDEVTB,DTAB ;START AT DEV 1
4144 010230 013703 013064 MOV R1,-(SP) ;SAVE R1
4145 010234 010146 MOV DEV,R1
4146 010236 010001
4147 010240 005301 SUD10: DEC R1 ;DEC DEVICE NUMBER
4148 010242 001403 BEQ SUDEX ;DONE, EXIT
4149 010244 062703 002000 ADD #2000,DTAB ;INCR TO NEXT DEV TABLE
4150 010250 000773 BR SUD10 ;TRY AGAIN
4151 010252 012601 SUDEX: MOV (SP)+,R1 ;RETURN TO CALLER
4152 010254 000207 RTS PC
4153
4154
4155 CLEAR DEVICE STATUS TABLE
4156
4157 CALLING SEQUENCE
4158 ..... RO = DEV #
4159 JSR PC,CDEVST
4160 ..... RETURN
4161 ..... R3 = ADDRESS OF DEVICE TABLE
4162 THE FOLLOWING TABLE ENTRIES ARE CLEARED
4163 SCMD
4164 SSTAT
4165 SBUFA
4166 SRBYTC
4167 SLCMD
4168
4169 ONLY REGISTER R3 IS AFFECTED BY THIS SUBROUTINE
4170
4171 CDEVST: JSR PC,SUDEV ;SET UP ADDR OF DEVICE STAT TABLE
4172 010256 004737 010230 CLRB SCMD(DTAB) ;RESET CURRENT COMMAND ENTRY
4173 010262 105063 000000 CLRB SSTAT(DTAB) ;RESET DEVICE STATUS ENTRY
4174 010266 105063 000003 CLR SBUFA(DTAB) ;RESET CURRENT BUFFER ADDRESS POINTER
4175 010272 005063 000012 CLR SRBYTC(DTAB) ;RESET REMAINING BYTE COUNT
4176 010302 105063 000001 CLRB SLCMD(DTAB) ;RESET LAST COMMAND ENTRY
4177 010306 000207 RTS PC ;RETURN TO THE CALLER
4178
4179
4180
4181 CSPWST -- CLEAR SPW STATUS BYTE
4182
4183 CALLING SEQUENCE
4184 .....DTAB (R3) POINTS TO CURRENT DEVICE STATUS TABLE
4185 .....DEV (RO) CONTAINS CURRENT DEVICE NUMBER
4186 JSR PC,CSPWST
    
```

```

4187
4188
4189
4190
4191 010310 010546
4192 010312 105763 000016
4193 010316 001007
4194 010320 010005
4195 010322 063705 013110
4196 010326 060505
4197 010330 063705 013114
4198 010334 105015
4199 010336 012605
4200 010340 000207

```

```

: .....RETURN TO CALLER WITH DEVICE STATUS BYTE RESET
:
: ALL REGISTERS AFE PRESERVED ACCROSS THIS SUBROUTINE
:
:SPWST: MOV R5, -(SP) ;SAVE REGISTER FOR SUBROUTINE USAGE
: TSTB SONLF(DTAB) ;IS DEVICE ON-LINE?
: BNE 10$ ;NO JUST EXIT
: MOV DEV, R5 ;GET DEVICE NUMBER AND COMPUTE
: ADD DEVCON, R5
: ADD R5, R5 ;ADDRESS OF SPW STATUS BYTE
: ADD STSPW, R5
: CLRB (R5) ;RESET SPW STATUS BYTE
: 10$: MOV (SP)+, R5 ;RESTORE REGISTER
: RTS PC

```

```

4201      :
4202      : MUXEND -- HANDLE DATA TRANSFER COMPLETIONS FOR MUX
4203      :
4204      : CALLING SEQUENCE
4205      : .....R3 (DTAB) CONTAINS THE ADDRESS OF THE DEVICE STATUS TABLE
4206      : JSR      PC,MUXEND
4207      : .....RETURN  C-BIT SET   - MUX DATA TRANS DONE
4208      :                   C-BIT RESET - SEL CHAN OR DATA TRANSFER NOT DONE
4209      :
4210      : NO REGISTERS ARE AFFECTED BY THIS SUBROUTINE
4211      :
4212 010342 105737 012526 MUXEND: TSTB  CHTYPE      ;SELECTOR OF MULTIPLEXER CHANNEL??
4213 010346 001006      BNE      5$          ;SELECTOR CHANNEL -- EXIT
4214      :
4215      : MULTIPLEXER CHANNEL
4216      :
4217 010350 162763 000004 000014 SUB      #4,SRBYTC(DTAB) ;DECR REMAINING BYTE COUNT
4218 010356 003004      BGT      10$          ;IF > 1, DATA TRANSFER NOT COMPLETE LET
4219      :
4220      : DATA TRANSFER COMPLETE ON MUX CHANNEL
4221      :
4222 010360 005063 000014 CLR      SRBYTC(DTAB)   ;INSURE REMAINING BYTE COUNT ZERO
4223 010364 000261      SEC      5$          ;SET MUX TRANSFER COMPLETE FLAG
4224 010366 000404      BR       30$          ;GOTO COMMON EXIT
4225      :
4226      : DATA TRANSFER INCOMPLETE
4227      :
4228 010370 062763 000004 000012 10$: ADD      #4,SBUFA(DTAB) ;BUMP BUFFER ADDRESS
4229 010376 000241      CLC      20$          ;RESET FLAG TO INDICATE MUX CHAN NOT DONE
4230 010400 000207      RTS      PC          ;RETURN TO THE CALLER

```

4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286

010402 010046
010404 010146
010406 010246
010410 126327 000001 000002
010416 001535
010420 126327 000001 000012
010426 001542
010430 126327 000001 000006
010436 001522

010440 105737 012532
010444 001102

010446 016301 000006
010452 126327 000001 000005
010460 001016

010462 005263 000014
010466 112102
010470 042702 177760

.SBTTL DX11-B ISR (2260 DISPLAY CONTROL SUBROUTINE)
DISPLAY CONTROL ROUTINE
THIS ROUTINE IS ENTERED AFTER DATA HAS BEEN
RECEIVED FROM OR WRITTEN TO THE 360.
DISCTL THEN FORMATS THE DATA TO CONFORM TO
A 2260 DISPLAY SCREEN IF THE 2848 DIAG IS RUN
CALLING SEQUENCE
.....DTAB(R3) POINTS TO CURRENT DEVICE STATUS TABLE
JSR PC DISCTL
.....RETURN
THIS SUBROUTINE IS ONLY USED TO COMPLETELY EMULA
A 2260'S DISPLAY. THIS ALLOWS THIS PROGRAM TO BE USED
WITH THE 2848 RESPONDER DIAGNOSTIC.
NOTE -- THE REMAINING BYTE COUNT (SRBYTC) IS USED TO
INDICATE THE NUMBER OF CHARACTERS RECEIVED FROM THE CHANNEL.
IT IS SET UP AT THE COMPLETION OF AN I/O
TRANSFER TO THE NUMBER OF CHARACTERS REMAINING IN
THE DX BYTE COUNT REGISTER.
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
DISCTL: MOV R0,-(SP) ;SAVE REGSITERS USED BY SUBROUTINE
MOV R1,-(SP)
MOV R2,-(SP)
CMPB SLCMD(DTAB),#CMRMI ;WAS IT A READ MANUAL INPUT COMMAND?
BEQ DRMI ;IF YES, PERFORM READ MANUAL INPUT PROCEEDURE
CMPB SLCMD(DTAB),#CMSRMI ;WAS IT A SHORT READ MANAUL INPUT?
BEQ DSRMI ;IF YES, EXIT
CMPB SLCMD(DTAB),#CMREAD ;WAS IT A READ FULL BUFFER COMMAND?
BEQ DREAD ;YES, RESET CURSOR ON READ FULL BUFFER

THE COMMAND MUST HAVE BEEN A 360 WRITE
DETERMINE TYPE OF TEST BEING RUN
TSTB TSTTYP ;TYPE OF TEST 0 = 2848 1 = FRIEND
BNE DISFRN ;FRIEND
FORMAT DISPLAY ALA 2260
MOV SINBF(DTAB),R1 ;GET ADDR OF START OF INPUT
CMPB SLCMD(DTAB),#CMWTLA ;WAS LAST CMD A WRITE LINE ADDRESS?
BNE 20\$;NO, NORMAL WRITE
WRITE LINE ADDRESS COMMAND
FIRST BYTE OF DATA BLOCK IS CURSOR LINE ADDRESS
INC SRBYTC(DTAB) ;INCR BYTE COUNT
MOV (R1)+,R2 ;GET LINE NUMBER
BIC #177760,R2 ;GET ONLY LINE NUMBER

```

4287 010474 005063 000004 CLR SCURS(DTAB) ;
4288 :
4289 : COMPUTE CURSOR ADDRESS CURS = LINE * LINESIZE
4290 :
4291 010500 005702 10$: TST R2 ;DONE?
4292 010502 001405 BEQ 20$ ;YES MORE DATA INTO DISPLAY BUF
4293 010504 062763 000050 000004 ADD #LINSZ,SCURS(DTAB) ;INCR TO NEXT LINE
4294 010512 005302 DEC R2 ;DECR LINE COUNT
4295 010514 000771 BR 10$
4296 :
4297 : MOVE DATA RECEIVED INTO DISPLAY BUFFER
4298 : THE DATA RECEIVED MAY BE ANY CONFIGURATION
4299 : OF EIGHT BITS, 0 - 255. TO EMULATE A 2260 DISPLAY
4300 : THE RECEIVED CHARACTER IS "FOLDED" INTO ONE OF
4301 : THE 64 CHARACTERS USED BY THE 2260. THIS FOLDING
4302 : IS PERFORMED BY TRANSLATING THE CHARACTER TO ASCII
4303 : AND THEN BACK TO EBCDIC.
4304 :
4305 010516 016302 000010 20$: MOV SOUTB(DTAB),R2 ;COMPUTE DISPLAY ADDR
4306 010522 066302 000004 ADD SCURS(DTAB),R2
4307 010526 026327 000014 000741 CMP SRBYTC(DTAB),#DISPSZ+1 ;ALL CHARS PROCESSED?
4308 010534 103077 BHS DISCEX ;YES EXIT
4309 010536 005263 000014 INC SRBYTC(DTAB) ;INCREMENT THE BYTE COUNT
4310 010542 112100 MOVB (R1)+,R0 ;GET THE NEXT BYTE RECEIVED AND BUMP POINTER
4311 010544 042700 177400 BIC #177400,R0 ;STRIP SIGN EXTENSION BITS (IF ANY)
4312 010550 116000 011754 MOVB EBCDTB(R0),R0 ;FOLD CHARACTER INTO ASCII CHARACTER SET
4313 010554 042700 177400 BIC #177400,R0 ;STRIP SIGN EXTENSION BITS, IF ANY
4314 010560 162700 000040 SUB #40,R0 ;SCALE INTO ASCII TABLE RANGE
4315 010564 116012 012354 MOVB ATOETB(R0),(R2) ;COMPLETE FOLDING BY RETRANSLATING TO EBCDIC
4316 010570 005263 000004 INC SCURS(DTAB) ;INCR CURSOR PTR
4317 010574 121227 000025 CMPB (R2),#NEWLINE ;WAS A NEW LINE SPECIFIED?
4318 010600 001015 BNE 60$
4319 :
4320 : NEW LINE COMMAND - ADVANCE CURSOR TO BEG OF NEW LINE
4321 : CURSOR = (CURSOR/LINESIZE + 1) * LINESIZE.
4322 :
4323 010602 005002 40$: CLR R2 ;CLEAR LINE CTR
4324 010604 005202 INC R2 ;INCR LINE CTR
4325 010606 162763 000050 000004 SUB #LINSZ,SCURS(DTAB)
4326 010614 003373 BGT 40$ ;KEEP DIVIDING
4327 010616 005063 000004 CLR SCURS(DTAB) ;CLEAR CURSOR
4328 010622 062763 000050 000004 50$: ADD #LINSZ,SCURS(DTAB)
4329 010630 005302 DEC R2
4330 010632 001373 BNE 50$
4331 :
4332 : CHECK FOR WRAP AROUND
4333 :
4334 010634 026327 000004 000740 60$: CMP SCURS(DTAB),#DISPSZ ;CURSOR OVERFLOW DISPLAY BUFFER?
4335 010642 002725 BLT 20$ ;CURSOR OK, PROCESS NEXT CHAR
4336 010644 005063 000004 CLR SCURS(DTAB) ;OVERFLOW, RESTART CURSOR AT POS 0
4337 010650 000722 BR 20$
4338 :
4339 :
4340 :
4341 : FRIEND TEST, IF SEPARATE I/O BUFFERS DON'T COPY
4342 : INPUT TO OUTPUT BUFFER

```

```

4343
4344 010652 0105737 012533      DISFRN: TSTB IOBUF      ;SEPARATE I/O BUFFERS?
4345 010656 001026      BNE DISCEX      ;YES, DON'T COPY INPUT TO OUTPUT
4346 010660 016301 000006      MOV SINBF(DTAB),R1 ;SET UP INPUT BUFFER ADDRESS
4347 010664 016302 000010      MOV SOUTB(DTAB),R2 ;SET UP OUTPUT BUFFER ADDRESS
4348 010670 012700 000360      MOV #DISPSZ/2,R0  ;TRANSFER THE INPUT BUFFER TO THE OUTPUT BUFFER
4349
4350      ;
4351      ; PERFORM COPY
4352 010674 012122      IOS: MOV (R1)+,(R2)+ ;INPUT TO OUTPUT
4353 010676 005300      DEC R0          ;ARE WE DONE?
4354 010700 001375      BNE IOS        ;NO, CONTINUE COPY
4355 010702 000414      BR DISCEX     ;PREPARE TO RETURN TO CALLER
4356
4357      ;
4358      ; A READ FULL BUFFER WAS PERFORMED
4359      ; THE CURSOR MUST BE RESET TO THE BEGINNING OF THE SCREEN
4360
4361 010704 005063 000004      DREAD: CLR SCURS(DTAB) ;RESET THE CURSOR
4362 010710 000411      BR DISCEX     ;AND PREPARE TO EXIT
4363
4364      ;
4365      ;
4366      ; A READ MANUAL INPUT WAS PERFORMED
4367      ; TO EMULATE THE 2260 SCREEN THE START OF MANUAL INPUT CHARACTER
4368      ; MUST BE DELETED FROM THE SCREEN
4369
4370
4371 010712 016301 000020      DRMI: MOV SMINS(DTAB),R1 ;GET THE STARTING ADDRESS
4372 010716 005301      DEC R1          ;DECREMENT TO THE SMI CHAR
4373 010720 112711 000100      MOVB #EBCDSP,R1 ;BLANK OUT THE CHARACTER
4374 010724 166301 000010      SUB SOUTB(DTAB),R1 ;AND COMPUTE THE CURSOR POSITION
4375 010730 010163 000004      MOV R1,SCURS(DTAB)
4376
4377      ;
4378      ;
4379      ; A SHORT READ MANUAL INPUT WAS PERFORMED
4380      ; NO ACTION REQUIRED BY DISPLAY CONTROL ROUTINE
4381
4382 010734      OSRMI:
4383
4384      ;
4385      ;
4386      ; RESTORE REGISTERS AND RETURN TO CALLER
4387
4388 010734 012602      DISCEX: MOV (SP)+,R2 ;RESTORE SAVED REGISTERS
4389 010736 012601      MOV (SP)+,R1
4390 010740 012600      MOV (SP)+,R0
4391 010742 000207      RTS PC        ;RETURN TO THE CALLER
4392      .SBTTL TELETYPE (CONSOLE) INPUT ISR
4393
4394      ;
4395      ; TELETYPE INPUT HANDLER (ISR)
4396      ;
4397      ; CONTROL PASSES HERE ON A TELETYPE INPUT INTERRUPT
4398      ;
4399      ; DATA IS INPUT FROM THE CONTROL CONSOLE AND STORED INTO

```


4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454

```

010744 010046
010746 010146
010750 017700 001502
010754 042700 177600
010760 013701 012742
010764 020027 000020
010770 001002
010772 000137 001002
010776 020027 000023
011002 001003
011004 105237 013073
011010 000503
011012 020027 000021
011016 001010
011020 105037 013073
011024 105737 013071
011030 001473
011032 004737 011252
011036 000470
011040 020027 000003
011044 001011
011046 105737 013053
011052 001457
011054 105237 013054
011060 012701 012640
011064 000137 011220
011070 105737 013053
011074 001051
011076 110021
011100 020027 000015
011104 001005
011106 012701 012640
011112 105237 013053
011116 000440
011120 020027 000177
    
```

44 IN: 3%: 5%: 10%: 20%: 30%:

THE TELETYPE INPUT BUFFER (TBUF). WHEN ALL THE DATA IS ENTERED, THE OPERATOR HITS A C/R TO END THE LINE, THEN AN ACTIVE FLAG IS SET AND THE COMMAND EXECUTED BY THE SYSTEM.

THE FOLLOWING CONTROL FUNCTIONS ARE AVAILABLE FOR OPERATOR CONVENIENCE.

- C/R = LINE DELIMITER
- ^ = DELETE LAST CHARACTER
- ␣ = (BACKSLASH SHIFT L) = DELETE LAST LINE
- (CONTROL-C) = ABORT CURRENT COMMAND -- FOR COMPS
- (RUB OUT) = DELETE LAST CHARACTER
- (CTL-P) = REENTER ALL PARAMETERS
- (CTL-U) = DELETE CURRENT INPUT LINE
- (CTL-S) = TEMPORARILY STOP OUTPUT TO CONSOLE
- (CTL-Q) = RESUME OUTPUT TO CONSOLE

NOTE -- A CONTROL Q MUST BE ISSUED AFTER A CONTROL S TO RESUME CONSOLE OUTPUT

```

MOV RO, -(SP) ;SAVE REGISTERS
MOV R1, -(SP)
MOV #TKB, RO ;GET TELE CHARACTER
BIC #177600, RO ;INSURE 7-BIT ASCII
MOV TPTR, R1 ;BUFFER PTR
CMP RO, #CTL.P ;CONTROL -P ?
BNE 3% ;NO
JMP RSTART ;YES, ALLOW OPERATOR TO REENTER ALL PARAMETERS
3%: CMP RO, #CTL.S ;CONTROL-S, TEMPORARILY STOP CONSOLE OUTPUT
BNE 6% ;NO, CONTINUE
INCB TTYSTP ;YES, SET FLAG TO STOP TTY OUTPUT
BR 100% ;AND EXIT FROM INTERRUPT
5%: CMP RO, #CTL.Q ;CONTROL-Q, RESUME CONSOLE OUTPUT?
BNE 10% ;NO, CONTINUE
CLRB TTYSTP ;YES, RESET CONSOLE STOP FLAG
TSTB PCTR ;CHECK TO INSURE OUTPUT TO RESUME
BEQ 100% ;NO OUTPUT -- EXIT
JSR PC, PROUT ;RESTART CONSOLE OUTPUT
BR 100% ;AND EXIT FROM THE INTERRUPT
10%: CMP RO, #CTL.C ;COMMAND ABORT -- CTL C?
BNE 20% ;NO
TSTB TCMACT ;IS A COMMAND ACTIVE?
BEQ 90% ;NO, TREAT AS A DELETE LAST LINE
INCB TCMDAB ;YES, SET ABORT FLAG
MOV #TBUF, R1 ;SET UP BUFFER POINTER
JMP 100% ;EXIT
20%: TSTB TCMACT ;TELE CMD CURRENTLY ACTIVE?
BNE 100% ;YES, IGNORE CHARACTER
MOVB RO, (R1)+ ;STORE CHAR INTO BUFFER - INC PTR
CMP RO, #CR ;LINE DELIMITER -- C/R?
BNE 30% ;NO
MOV #TBUF, R1 ;RESET BUFFER PTR
INCB TCMACT ;YES, SET COMMAND ACTIVE FLAG
BR 100% ;DON'T PRINT THE LINE DELIMITER
30%: CMP RO, #RUBOUT ;A RUBOUT?
    
```

4455	011124	001002		BNE	40\$:NOPE
4456	011126	012700	000137	MOV	*+,RO	:YES, TREAT AS A DELETE LAST CHARACTER
4457	011132	120027	000025	40\$: CMPB	RO,CTL.U	:CONTROL-U? (DELETE CURRENT INPUT LINE
4458	011136	001002		BNE	50\$:NOPE, CONTINUE
4459	011140	112700	000134	MOVb	*+,RO	:YES, TREAT AS DELETE LAST LINE (BACKSLASH
4460	011144	004737	011366	50\$: JSR	PC,PCHAR	:ECHO THE CHARACTER BACK
4461	011150	020027	000137	CMP	RO,*+	:DELETE LAST CHAR -- BACK ARROW
4462	011154	001004		BNE	60\$:NO
4463	011156	124141		CMPB	-(R1),-(R1	:YES, DECR POINTER BY 2
4464	011160	020127	012640	CMP	R1,*TBUF	:ARE WE BEYOND BEG OF THE BUFFER?
4465	011164	003403		BLE	70\$:YES, RESET TO BEG OF BUFFER
4466	011166	020027	000134	60\$: CMP	RO,*+	:DELETE CUR LINE -- BACK SLASH?
4467	011172	001004		BNE	80\$:NO
4468	011174	012701	012640	70\$: MOV	*TBUF,R1	:YES, RESET BUFFER PTR
4469	011200	004737	011330	JSR	PC,CRLF	:NEW LINE FOR NEW COMMAND
4470	011204	020127	012740	80\$: CMP	R1,*TBUFE	:WERE LIMITS EXCEEDED?
4471	011210	001003		BNE	100\$:NOPE, EXIT
4472	011212	012700	000134	90\$: MOV	*+,RO	:THEY WERE -- TREAT AS A LINE ABOVE
4473	011216	000740		BR	30\$	
4474	011220	010137	012742	100\$: MOV	R1,TPTR	:SAVE BUFFER PTR
4475	011224	012601		MOV	(SP)+,R1	:RESTORE REGISTERS + E-IT
4476	011226	012600		MOV	(SP)+,PO	
4477	011230	000002		RTI		

4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544

011252 105737 013052
011256 001023
011260 105737 013073
011264 001020
011266 105237 013052
011272 105337 013071
011276 117777 001544 001156
011304 005237 013046
011310 023727 013046 013046
011316 001003
011320 012737 012744 013046
011326 000207

011330 010246
011332 012702 105215
011336 004737 011346
011342 012602
011344 000207

.SBTTL TELETYPE OUTPUT HANDLING SUBROUTINES
SEND DATA TO PRINTER, IF NOT BUSY
CALLING SEQUENCE
JSR PC,PROUT
.....RETURN
IF TELETYPE OUTPUT IS CURRENTLY IN PROGRESS OR HAS BEEN SUSPENDED BY A CONTROL -
CONTROL IS RETURNED IMMEDIATELY WITH NO ACTION BEING INITIATED.
IF TELETYPE OUTPUT IS NOT CURRENTLY IN PROGRESS
THE PRINTER BUSY FLAG IS SET AND A CHARACTER IS SENT TO THE TERMINAL
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
PROUT: TSTB PIUFL ;IS IT BUSY?
BNE 20\$;YES, EXIT
TSTB TTYSTP ;HAS CONSOLE OUTPUT BEEN SUSPENDED?
BNE 20\$;YES, RETURN IMMEDIATELY TO CALLER
INCB PIUFL ;NO, SET BUSY FLAG
DECB PCTR ;DECR CHAR COUNTER
MOVB JPFSTR,JTPB ;OUTPUT NEXT CHAR
INC PFPTR ;INCR PRINT FETCH POINTER
CMP PFPTR,#PBFE ;TIME TO WRAP AROUND?
BNE 20\$;NO, EXIT
MOV #PBFS,PFPTR ;YES, RESTORE TO START OF BUFFER
RTS PC ;RETURN TO CALLER

PRINT A CR/LF
CALLING SEQUENCE
JSR PC,CRLF
.....RETURN
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
CRLF: MOV R2,-(SP) ;SAVE THE R2 REGISTER
MOV #105215,R2 ;DO A CRLF
JSR PC,PRINT2 ;PRINT IT
MOV (SP)+,R2 ;RESTORE THE R2 REGISTER
RTS PC ;RETURN TO THE CALLER

PRINT 2 CHARACTERS ON THE TTY
CALLING SEQUENCE
.....R2 CONTAINS DATA TO BE PRINTED (2 BYTES)
JSR PC,PRINT2
.....RETURN
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

J08

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 100
TELETYPE OUTPUT HANDLING SUBROUTINES

SEC 0:00
SEC 0:00

4545	011346	010237	011362
4546	011352	004137	011472
4547	011356	011362	
4548	011360	000207	
4549	011362	000000	
4550	011364	377	377
4551			
4552			
4553			
4554			
4555			
4556			
4557			
4558			
4559			
4560			
4561			
4562			
4563	011366	110037	011402
4564	011372	004137	011472
4565	011376	011402	
4566	011400	000207	
4567	011402	000	377

```

PRINT2: MOV     R2,P2BF
        JSR     R1,MSG
        .WORD  P2BF
        RTS     PC
P2BF:   .WORD  0
        .BYTE  377,377

```

```

:
: PRINT 1 CHARACTER
:
: CALLING SEQUENCE
: .....R0 CONTAINS THE CHARACTER TO BE PRINTED
: JSR     PC,PCHAR
: .....RETURN WITH THE DATA IN THE PRINT BUFFER
:
: NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

```

```

PCHAR:  MOVB    R0,P1BF
        JSR     R1,MSG
        .WORD  P1BF
        RTS     PC
P1BF:   .BYTE  0,377
:RETURN TO THE CALLER

```

4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579 011404 010246
4580 011406 010346
4581 011410 013703 013050
4582 011414 121227 000377
4583 011420 001417
4584 011422 112223
4585 011424 105237 013071
4586 011430 020327 013046
4587 011434 001002
4588 011436 012703 012744
4589 011442 004737 011252
4590 011446 123737 013071 013130
4591 011454 001774
4592 011456 000756
4593 011460 010337 013050
4594 011464 012603
4595 011466 012602
4596 011470 000207
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611 011472 010246
4612 011474 012102
4613 011476 004737 011404
4614 011502 012602
4615 011504 000201
4616
4617
4618
4619
4620
4621
4622
4623

```

PRMSG PRINT A CHARACTER STRING

CALLING SEQ
.....R2 CONTAINS THE STARTING ADDRESS OF THE MESSAGE
JSR PC PRMSG
.....RETURN

NOTE -- MESSAGE MUST BE TERMINATED BY A 377

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
PRMSG: MOV R2, -(SP) ;SAVE REGS
MOV R3, -(SP)
MOV PPTR, F3 ;GET PRINT OUTPUT POINTER
10$: CMPB (R2), #377 ;END OF MESSAGE?
BEQ 40$ ;YES, EXIT
MOV (R2)+, (R3)+ ;NO MOVE NEXT CHAR TO PRINT BUFFER
INCB PCTR ;INCR CHAR COUNTER
CMPB R3, #PBFE ;AT END OF BUFFER?
BNE 20$ ;NO
MOV #PBFS, R3 ;YES, WRAP AROUND TO BEG OF BUFFER
JSR PC, PROUT ;CAN WE START PRINT?
20$: CMPB PCTR, PMAX ;IS PRINT BUFFER FULL?
30$: BEQ 30$ ;YES, WAIT TILL ROOM AVAILABLE
BF 10$ ;GET NEXT CHAR
40$: MOV R3, PPTR ;EXIT, RESTORE PUT PTR
MOV (SP)+, R3 ;RESTORE REGS
MOV (SP)+, R2
RTS PC ;RETURN TO THE CALLER

MSG -- PRINT A CHARACTER STRING ON THE SYSTEM CONSOLE

CALLING SEQUENCE
JSP R1, MSG
.WORD ADDRESS OF START OF MESSAGE
.....RETURN

NOTE -- MESSAGE MUST BE TERMINATED BY A 377

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
MSG: MOV R2, -(SP) ;SAVE REGISTER
MOV (R1)+, R2 ;GET ADDRESS OF MESSAGE AND BUMP FOR RETURN
JSR PC, PRMSG ;MORE MESSAGE PROCESSING
MOV (SP)+, R2 ;RESTORE SOILED REGISTER
RTS R1 ;RETURN TO THE CALLER

INMES PRINT A CHARACTER STRING

CALLING SEQUENCE
JSR R1, INMES
.WORD ADDRESS OF MESSAGE

```

```

4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634 011506 010246
4635 011510 113746 013130
4636 011514 112737 000377 013130
4637 011522 012102
4638 011524 004737 011404
4639 011530 112637 013130
4640 011534 012602
4641 011536 000201

```

```

.....RETURN
INMES IS USED FOR ROUTINES AT THE ISR LEVEL AND DOES
NOT CHECK TO SEE IF DATA WILL BE OVERLAPPED IN
TELEBUFFER

NOTE -- THE MESSAGE MUST BE TERMINATED BY A 377

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

INMES:  MOV    R2, -(SP)
        MOVB  PMAX, -(SP)      ;CHEAT, SAVE PMAX
        MOVB  #377, PMAX      ;AND MAKE VERY LARGE
        MOV   (R1)+, R2
        JSR   PC, PRMSG      ;USE STANDARD MESSAGE PROCESSOR
        MOVB (SP)+, PMAX     ;RESTORE PRINT MAX
        MOV   (SP)+, R2
        RTS   R1             ;RETURN TO CALLER

```

M08

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
 CZDXIB.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 103
 UTILITY SUBROUTINES (CONVERT OCTAL OR HEX TO BINARY)

SEQ 0103
 SEQ 0103

```

4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657 011540 005003
4658 011542 005005
4659 011544 112204
4660 011546 120427 000067
4661 011552 003013
4662 011554 120427 000060
4663 011560 002410
4664 011562 042704 177770
4665 011566 006303
4666 011570 006303
4667 011572 006303
4668 011574 060403
4669 011576 005205
4670 011600 000761
4671 011602 000207
  
```

```

.SBTTL UTILITY SUBROUTINES (CONVERT OCTAL OR HEX TO BINARY)
COTB -- CONVERT ASCII OCTAL TO BINARY (COTB)
CALLING SEQUENCE
.....R2 = CHAR ADDRESS OF FIRST CHARACTER TO BE CONVERTED
JSR PC,COTB
.....RETURN

UPON RETURN THE FOLLOWING REGISTERS WILL CONTAIN
R2 = NEXT CHAR POSITION AFTER LAST ILLG CHAR
R3 = BINARY RESULT OF CONVERSION
R4 = (BITS 0-7) FIRST NON-OCTAL CHARACTER
R5 = NUMBER OF CHARACTERS CONVERTED

COTB: CLR R3
      CLR R5
10$:  MOVB (R2)+,R4 ;GET NEXT CHAR
      CMPB R4,#'7 ;CHAR GT 7?
      BGT 20$ ;YES EXIT
      CMPB R4,#'0 ;CHAR LT 0?
      BLT 20$ ;YES, EXIT
      BIC #177770,R4 ;SAVE ONLY L.S. 3 BITS
      ASL R3 ;SHIFT OLD RESULT BY 8
      ASL R3
      ADD R4,R3 ;ADD IN NEW NUMBER
      INC R5 ;INCR CHAR COUNT
      BR 10$ ;GET NEXT CHAR
20$:  RTS PC ;RETURN TO CALLER
  
```



```

4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685 011604 005003
4686 011606 005005
4687 011610 112204
4688 011612 162704 000060
4689 011616 100422
4690 011620 020427 000012
4691 011624 002410
4692 011626 162704 000007
4693 011632 020427 000020
4694 011636 002012
4695 011640 020427 000012
4696 011644 002407
4697 011646 006303
4698 011650 006303
4699 011652 006303
4700 011654 006303
4701 011656 060403
4702 011660 005205
4703 011662 000752
4704 011664 005302
4705 011666 112204
4706 011670 000207

```

```

:
: CHTB -- CONVERT ASCII HEX TO BINARY
:
: CALLING SEQUENCE
: JSR R2 = ADDRESS OF FIRST CHARACTER TO BE CONVERTED
: PC CHTB
: RETURN
: UPON RETURN
: R2 = NEXT CHAR POSITION NOT CONVERTED
: R3 = BINARY RESULT
: R4 = (BITS 0-7) FIRST NON HEX CHARACTER
: R5 = NUMBER OF CHARACTERS CONVERTED
:
: CHTB: CLR R3
: CLR R5
10$: MOVB (R2)+,R4 ;GET THE FIRST CHARACTER
: SUB #0,R4 ;SCALE RELATIVE TO ASCII ZERO
: BMI 30$ ;NOT A VALID HEX CHAR
: CMP R4,#10. ;IS RESULT STILL GT 10?
: BLT 20$ ;YES, WE HAVE A VALID HEX DIGIT
: SUB #7,R4
: CMP R4,#16. ;IS IT A LETTER?
: BGE 30$ ;NO, INVALID CHAR
: CMP R4,#10. ;AND GT 10
: BLT 30$ ;NO, ILLEGAL CHAR
20$: ASL R3 ;MAKE ROOM FOR NEW ENTRY
: ASL R3
: ASL R3
: ASL R3
: ADD R4,R3 ;INSERT NEW ENTRY
: INC R5 ;INCR CHAR COUNT
: BR 10$ ;AND CONVERT NEXT CHAR
30$: DEC R2 ;GET THE ILLEGAL CHARACTER
: MOVB (R2)+,R4 ;AND PUT IT R4
: RTS PC ;ITS TIME TO RETURN TO THE CALLER

```

4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740

.SBTTL PROCESSOR ERROR TRAP HANDLERS
TRAP OUT ROUTINES

MEMORY TIME OUT ROUTINE

ATO: MOV #PMTG,R2 ;SET UP ADDRESS OF THE PRINT ROUTINE
BR TOUTRT ;TO GENERALIZED TRAP OUT ROUTINE

MEMORY MANAGEMENT TRAP OUT ROUTINE

MMERR: CLR MMSRD ;CLEAR THE MEMORY MANAGEMENT BIT
MOV #PMMERR,R1 ;SET UP ADDRESS OF ERROR MESSAGE
TOUTRT: RESET ;CLEAR ALL DEVICES
CLRB PIUFL ;CLEAR PRINT IN USE FLAG
CLR PSW ;LOWER PROCESSOR STATUS TO ALLOW INTERRUPTS TO CUM
JSR PC,PRHESG ;PRINT THE ERROR MESSAGE
IOS: TSTB PCIR ;IS PRINTING DONE?
BNE IOS
HALT ;YES, HALT
JMP START

INVALID UNIBUS ADDRESS TRAP

UNTRP: CMP (SP)+,(SP)+ ;POP THE PUSH STACK
CLR PSW ;CLEAR THE PROCESSOR STATUS WORD
JMP NEWPRM ;ASK OPERATOR TO REENTER THE DATA

.SBTTL CODE CONVERSION TABLES
EBCDIC TO ASCII CODE CONVERSION TABLE
THIS TABLE FOLDS ALL INPUT INTO A 64 CHARACTER SET
NOTE -- BACKARROW IS USED TO DENOTE A NEWLINE

4741	011754	040440	041502	042504
4742	011754	043506	044510	027136
4743	011770	024074	020453	
4744	011774	045046	046113	047115
4745	012002	050117	051121	022137
4746	012010	024452	056473	
4747	012014	027455	052123	053125
4748	012022	054127	055131	026042
4749	012030	055445	037476	
4750	012034	030460	031462	032464
4751	012042	033466	034470	021472
4752	012050	023500	056075	
4753	012054	040440	041502	042504
4754	012062	043506	044510	027136
4755	012070	024074	020453	
4756	012074	045046	046113	047115
4757	012102	050117	051121	022137
4758	012110	024452	056473	
4759	012114	027455	052123	053125
4760	012122	054127	055131	026042
4761	012130	055445	037476	
4762	012134	030460	031462	032464
4763	012142	033466	034470	021472
4764	012150	023500	056075	
4765	012154	040440	041502	042504
4766	012162	043506	044510	027136
4767	012170	024074	020453	
4768	012174	045046	046113	047115
4769	012202	050117	051121	022137
4770	012210	024452	056473	
4771	012214	027455	052123	053125
4772	012222	054127	055131	026042
4773	012230	055445	037476	
4774	012234	030460	031462	032464
4775	012242	033466	034470	021472
4776	012250	023500	056075	
4777	012254	040440	041502	042504
4778	012262	043506	044510	027136
4779	012270	024074	020453	
4780	012274	045046	046113	047115
4781	012302	050117	051121	022137
4782	012310	024452	056473	
4783	012314	027455	052123	053125
4784	012322	054127	055131	026042
4785	012330	055445	037476	
4786	012334	030460	031462	032464
4787	012342	033466	034470	021472
4788	012350	023500	056075	

EBCDIB:

.ASCII / ABCDEFGHIJ.((+!)	:00 - 0F
.ASCII "&JKLMN+OPQR+S*);]"	:10 - 1F
.ASCII '-/STUVWXYZ",%[]>?"	:20 - 2F
.ASCII "0123456789: @'=\`"	:30 - 3F
.ASCII / ABCDEFGHIJ.((+!)	:40 - 4F
.ASCII "&JKLMN+OPQR+S*);]"	:50 - 5F
.ASCII '-/STUVWXYZ",%[]>?"	:60 - 6F
.ASCII "0123456789: @'=\`"	:70 - 7F
.ASCII / ABCDEFGHIJ.((+!)	:80 - 8F
.ASCII "&JKLMN+OPQR+S*);]"	:90 - 9F
.ASCII '-/STUVWXYZ",%[]>?"	:A0 - AF
.ASCII "0123456789: @'=\`"	:B0 - BF
.ASCII / ABCDEFGHIJ.((+!)	:C0 - CF
.ASCII "&JKLMN+OPQR+S*);]"	:D0 - DF
.ASCII '-/STUVWXYZ",%[]>?"	:E0 - EF
.ASCII "0123456789: @'=\`"	:F0 - FF

```

48000 0123354 100 117 152
48001 0123357 113 133 154
48002 0123362 120 175
48003 0123364 115 135 134
48004 0123367 116 153 140
48005 0123372 113 141
48006 0123374 360 361 362
48007 0123377 363 364 365
48008 012402 366 367
48009 012404 370 371 172
48010 012407 136 114 176
48011 012412 156 157
48012 012414 174 301 302
48013 012417 303 304 305
48014 012422 306 307
48015 012424 310 311 321
48016 012427 322 323 324
48017 012432 325 326 327
48018 012434 327 330 331 342 343 344 345 346 320 327
48019 012437 342 343 344
48020 012442 345 346
48021 012444 347 350 351 137
48022 012447 155 177
48023 012452 112 025
48024
48025
48026
48027
48028 012454 177560
48029 012456 177562
48030 012460 177564
48031 012462 177566
48032
48033
48034
48035
48036
48037
48038 012464 000000
48039 012466 000000
48040 012470 000000
48041 012472 000000
48042 012474 000000
48043 012476 000000
48044 012500 000000
48045 012502 000000
48046 012504 000000
48047 012506 000000
48048 012510 000000
48049 012512 000000
48050 012514 000000
48051
48052

```

```

: ASCII TO EBCDIC CONVERSION TABLE
ATOE^B: .BYTE 100,117,152,173,133,154,120,175 ;240-247
:
: .BYTE 115,135,134,116,153,140,113,141 ;250-257
:
: .BYTE 360,361,362,363,364,365,366,367 ;260-267
:
: .BYTE 370,371,172,136,114,176,156,157 ;270-277
:
: .BYTE 174,301,302,303,304,305,306,307 ;300-307
:
: .BYTE 310,311,321,322,323,324,325,326 ;310-317
:
: .BYTE 327,330,331,342,343,344,345,346 ;320,327
:
: .BYTE 347,350,351,155,177,137,112,025 ;330-337
:
: .SBTTL PROGRAM CONSTANTS AND VARIABLES
:
: CONSOLE UNIBUS ADDRESS CONSTANTS
TKS: .WORD 177560 ;KEYBOARD CONTROL STATUS REGISTER
TKB: .WORD 177562 ;KEYBOARD DATA BUFFER
TPS: .WORD 177564 ;PRINTER STATUS/CONTROL REGISTER
TPB: .WORD 177566 ;PRINTER DATA BUFFER
:
: DX REGISTERS - ADDRESS GENERATED BY INITIALIZATION
DXDS: .WORD 0 ;DEVICE STATUS -- TT1
DXCA: .WORD 0 ;COMMAND AND ADDRESS -- TT2
DXCS: .WORD 0 ;CONTROL UNIT STATUS
DXOS: .WORD 0 ;OFFSET AND STATUS
DXBA: .WORD 0 ;BUS ADDRESS
DXBC: .WORD 0 ;BYTE COUNT
DXMO: .WORD 0 ;MAINTANCE OUT
DXMI: .WORD 0 ;MAINTANCE IN
DXCB: .WORD 0 ;CONTROL BITS
DXND: .WORD 0 ;NPR DATA
DXES1: .WORD 0 ;EXTRA SIGNALS
DXMOB: .WORD 0 ;BUFFERED BUS OUT
DXES2: .WORD 0 ;EXTRA SIGNALC

```

```

4853      :
4854      : CONFIGURATION CONSTANTS
4855      :
4856 012516 000000 UNADDR: .WORD 0      : UNIBUS ADDRESS
4857 012520 000000 VECTAD: .WORD 0      : DX VECTOR ADDRESS
4858 012522 000000 SDEV: .WORD 0      : STARTING DEV NUMBER
4859 012524 000000 EDEV: .WORD 0      : ENDING DEV NUMBER
4860 012526 000 CHTYPE: .BYTE 0      : CHANNEL TYPE 0 = MPX - 1 = SEL
4861 012527 000 MMRESP: .BYTE 0      : MEMORY MANAGEMENT 0 = NO - 1 = YES
4862 012530 000000 BUFREL: .WORD 0      : BUFFER RELOCATION ADDRESS
4863 012532 000 TSTTYP: .BYTE 0      : TEST TYPE 0 = 2848 - 1 = FRIEND
4864 012533 000 IOBUF: .BYTE 0      : SEPERATE I/O BUFFER 0 = NO - 1 = YES
4865 012534 000 FILLCH: .BYTE 0      : FILL CHARACTER
4866 012535 000 CONEND: .BYTE 0      : EXTRA
4867      :
4868      :
4869      :
4870      : SYSTEM PUSH STACK
4871      :
4872      :
4873      :
4874      :
4875      :
4876      :
4877      : SYSTEM VARIABLES
4878      :
4879      : THE FOLLOWING VARIABLES ARE RESET UPON START-JP
4880      :
4881 012636 000000 VSTRT: .WORD 0      : DUMMY
4882      : 012640 TBUF = .      : START OF TELETYPE INPUT BUFFER
4883      : 012740      :
4884      : =. +100      :
4885 012740 000000 TBUFE: .WORD 0      : END OF TELETYPE INPUT BUFFER
4886 012742 000000 TPTR: .WORD 0      : TELE IN PTR
4887 012744 000000 PBFS: .WORD 0      : START OF PRINT BUFFER
4888      : 013046      :
4889      : =. +100      :
4890 013046 000000 PBFE = .      : END OF PRINT BUFFER
4891 013050 000000 PFPTR: .WORD 0      : PRINT FETCH PTR
4892 013052 000 PPTR: .WORD 0      : PRINT PUT PTR
4893 013053 000 PIUFL: .BYTE 0      : PRINTER IN USE FLAG
4894 013054 000 TCMACT: .BYTE 0      : TELE COMMAND ACTIVE FLAG 0 = NON-ACT
4895 013055 000 TCMAB: .BYTE 0      : TEL COMMAND ABORT 1 = ABORT
4896 013056 000 LINECT: .BYTE 0      : LINE CTR - CHARS LINE
4897 013057 000 WK: .BYTE 0      : WORK LOC
4898 013060 000000 WK1: .BYTE 0      : WORK LOC
4899 013062 000000 TTPTR: .WORD 0      : TUMBLE TABLE PTR
4900 013064 000000 TTADDR: .WORD 0      : BEG OF TUMBLE TABLE
4901 013066 000 SDEVTB: .WORD 0      : START OF DEVICE TABLES
4902 013067 000 DXSTPF: .BYTE 0      : DX STOP FLAG
4903 013070 000 MAXDEV: .BYTE 0      : HIGHEST DEV # 1 - 8
4904 013071 000 DXACT: .BYTE 0      : DXACTIVE FLAG
4905 013072 000 PCTR: .BYTE 0      : PRINT BUFFER COUNTER
4906 013073 000 DXABFL: .BYTE 0      : DX ABORT FLAG 0 = NO ABORT, 1 = ABORT
4907      : TTYSTP: .BYTE 0      : CONSOLE OUTPUT STOP FLAG 0 = OUTPUT; 1 = NO OUTPUT
4908      : .EVEN
4909 013074 000000 CMDCHF: .WORD 0      : COMMAND CHAIN FLAG
4910 013076 000000 MDEV: .WORD 0      : DEV # IN MPXR EXEC

```

```

4909      013076
4910      013100 000000
4911      013102 000000
4912      013104 000000
4913      013106 000000
4914      013110 000000
4915      013112 000000
4916      013114 000000
4917      013116 000000
4918      013120 000000
4919      013122 000000
4920      013124 000000
4921      013126 000000
4922
4923
4924
4925
4926
4927      013130 000002
4928      013132 000000

```

```

SELDEV = MDE.
PBUFA: .WORD 0
VBUFA: .WORD 0
PHYOFF: .WORD 0
CDEV: .WORD 0
DEVCON: .WORD 0
XADDR: .WORD 0
STSPW: .WORD 0
DSTOFF: .WORD 0
SADDR: .WORD 0
EADDR: .WORD 0
DMPADR: .WORD 0
VEND: .WORD 0

```

```

:DEV # IN SEL EXEC
:PHYSICAL BUFF ADDR - IN '000'S
:VIRTUAL BUFF ADDR - IN '000'S
:PHY OFFSET FOR MEMORY MANAGEMENT
:CURRENT DX DEVICE -- INTER SERVICE ROUTINE
:DEVICE ADDED TO THE DEVICE NUMBER = STARTING DEV NUMB -
:EXTENDED ADDRESS BITS FOR THE DX CONTROL REGISTER -- IN
:START OF THE PSW TABLE
:OFFSET TO THE DST TABLE
:TELETYPE COMMAND STARTING BUFFER ADDRESS
:TELETYPE COMMAND ENDING BUFFER ADDRESS
:POINTER TO DUMP ROUTINE CURRENTLY BEING UTILIZED BY TEL

```

```

:
: THE FOLLOWING VARIABLES ARE NOT RESET ON START-UP
:
PMAX: .WORD PBFE-PBFS :SIZE OF PRINT BUFFER
FTMFL: .WORD 0 :FIRST TIME FLAG

```

4929
4930

MESSAGES
:SBTTL MESSAGES
:NLIST BEX

:
: SYSTEM MESSAGES

013134	215	212	
013136	055104	054104	026511
013173	377	377	
013175	215	212	
013177	125	044516	052502
013230	377		
	013232		
013232	215	212	
013234	047111	042524	051122
013277	377		
013300	215	212	
013302	042504	044526	042503
013342	377		
	013344		
013344	215	212	
013346	044103	047101	042516
013375	377		
013376	215	212	
013400	042515	047515	054522
013434	377		
	013436		
013436	215	212	
013440	052502	043106	051105
013533	377		
013534	215	212	
013536	051106	042511	042116
013572	377		
	013574		
013574	215	212	
013576	042523	040520	040522
013635	377		
013636	215	212	
013640	052517	050124	052125
013705	377		
013706	207	207	215
013712	047516	020116	054105
013732	215	212	377
	013736		
013736	207	207	215
013742	040520	044522	054524
013756	212	215	377
	013762		
013762	207	207	215
013766	046111	042514	040507
014013	212	215	377

```

:STMSG: .BYTE 215,212
         .ASCII /DZDAI-B NEW D-11-B RESPONDER
UNMSG:  .BYTE 377,377
         .BYTE 215,212
         .ASCII /UNIBUS ADDRESS -OCTAL- :
         .BYTE 377
         .EVEN
VECTMS: .BYTE 215,212
         .ASCII /INTERRUPT VECTOR ADDRESS -OCTAL- :
         .BYTE 377
         .EVEN
DEVMES: .BYTE 215,212
         .ASCII /DEVICE ADDRESSES -HEX- XX,XXX:
         .BYTE 377
         .EVEN
CHTYMS: .BYTE 215,212
         .ASCII /CHANNEL TYPE (M OR S):
         .BYTE 377
         .EVEN
MMMES:  .BYTE 215,212
         .ASCII /MEMORY MANAGEMENT (Y OR N):
         .BYTE 377
         .EVEN
BFREMS: .BYTE 215,212
         .ASCII /BUFFER RELOCATION, IF SPECIFIED - IN EVEN .000'S -OCTAL- :
         .BYTE 377
         .EVEN
TESTMS: .BYTE 215,212
         .ASCII /FRIEND (F) OR 2848 DIAG(D):
         .BYTE 377
         .EVEN
FIOMS:  .BYTE 215,212
         .ASCII /SEPARATE I-O BUFFERS (Y OR N):
         .BYTE 377
         .EVEN
FILLMS: .BYTE 215,212
         .ASCII /OUTPUT BUFFER FILL CHARACTER -HEX- :
         .BYTE 377
         .EVEN
NXMMSG: .BYTE 207,207,215,212
         .ASCII /NON EX-MEM ERROR
         .BYTE 215,212,377
         .EVEN
PARMES: .BYTE 207,207,215,212
         .ASCII /PARITY ERROR
         .BYTE 212,215,377
         .EVEN
ILLMES: .BYTE 207,207,215,212
         .ASCII /ILLEGAL DEVICE NUMBER
         .BYTE 212,215,377
         .EVEN

```

MESSAGES

```

014016 052503 051122 047105 STPMES: .ASCII /CURRENT DEVICE NUMBER --
014047 377 .BYTE 377
.EVEN
014050 207 207 215 INVLDC: .BYTE 207,207,215,212
014054 047111 040526 044514 .ASCII /INVALID DX COMMAND/
014076 212 215 377 .BYTE 212,215,377
.EVEN
014102 014102 047516 046440 PNOHM: .ASCII / NO MEMORY MANAGEMENT AVAILABLE
014142 020040 377 .BYTE 377
014143 215 212 207 PMTO: .BYTE 215,212,207,207
014147 115 046505 051117 .ASCII /MEMORY TIME OUT/
014166 377 .BYTE 377
014167 215 212 207 PMMERR: .BYTE 215,212,207,207
014173 115 046505 051117 .ASCII /MEMORY MANAGEMENT ERROR
014222 377 .BYTE 377
.EVEN
014224 014224 215 212 RNMSG: .BYTE 215,212
014226 054523 052123 046505 .ASCII /SYSTEM INITIALIZED, TYPE "R" TO ENABLE DX
014277 377 .BYTE 377
014300 015 012 HELPMS: .BYTE CR,LF
014302 054104 030461 041055 .ASCII /DX11-B 2848 EMULATOR TEST PACKAGE - OPERATIONAL INFORMATION
014375 015 012 .BYTE CR,LF
014377 015 012 .BYTE CR,LF
014401 104 026440 020055 .ASCII /D -- DUMP COMMAND/<CR><LF>
014424 020040 020040 020040 .ASCII / DTT,C DUMP TUMBLE TABLE IN CODE "C"/<CR><LF>
014504 020040 020040 020040 .ASCII / DIN,C,XX DUMP INPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014602 020040 020040 020040 .ASCII / DOT,C,XX DUMP OUTPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014701 105 026440 020055 .ASCII /E -- ENABLE DEVICE ON DX/<CR><LF>
014733 040 020040 020040 .ASCII / EXX ENABLE DEVICE XX/<CR><LF>
014776 020106 026455 043040 .ASCII /F -- FILL BUFFER COMMAND/<CR><LF>
015030 020040 020040 020040 .ASCII / FIN,HH,XX FILL INPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015116 020040 020040 020040 .ASCII / FOT,HH,XX FILL OUTPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015205 110 026440 020055 .ASCII /H -- HELP COMMAND/<CR><LF>
015230 020040 020040 020040 .ASCII / THIS TEXT/<CR><LF>
015252 020113 026455 045440 .ASCII /K -- KILL A DEVICE ON THE DX/<CR><LF>
015310 020040 020040 020040 .ASCII / KXX KILL DEVICE XX/<CR><LF>
015351 122 026440 020055 .ASCII /R -- ENABLE DX (RUN)/<CR><LF>
015377 040 020040 020040 .ASCII / R RUN TEST/<CR><LF>
015432 020123 026455 042040 .ASCII /S -- DISABLE DX (STOP)/<CR><LF>
015462 020040 020040 020040 .ASCII / S STOP IMMEDIATELY/<CR><LF>
015525 040 020040 020040 .ASCII / SD STOP AFTER NEXT DATA TRANSFER/<CR><LF>
015605 040 020040 020040 .ASCII / SE STOP AFTER NEXT ENDING SEQUENCE/<CR><LF>
015667 040 020040 020040 .ASCII / SI STOP ON NEXT SEL SEQ (ISS)/<CR><LF>
015744 020040 020040 020040 .ASCII / SP STOP ON NEXT PARITY ERROR/<CR><LF>
016020 005015 044127 051105 .ASCII /<CR><LF>/WHERE: <CR><LF>
016032 020040 020040 041442 .ASCII / "C" IS CODE FORMAT 0 = OCTAL/<CR><LF>
016100 020040 020040 020040 .ASCII / A = ASCII/<CR><LF>
016146 020040 020040 020040 .ASCII / E = EBCDIC/<CR><LF>
016215 040 020040 020040 .ASCII / H = HEX/<CR><LF>
016261 040 020040 021040 .ASCII / "XX" IS DX-11 DEVICE NUMBER IN HEX/<CR><LF>
016331 040 020040 021040 .ASCII / "HH" IS A HEX CHARACTER/<CR><LF><LF>
016367 103 047117 047523 .ASCII /CONSOLE CONTROL CHARACTERS/<CR><LF>
016423 103 046124 041455 .ASCII /CTL-C (+C) ABORT CURRENT COMMAND/<CR><LF>
016465 103 046124 050055 .ASCII /CTL-P (+P) REQUESTS THE REENTRY OF CONTROL PARAMETERS/<CR><LF>
016554 052103 026514 020121 .ASCII /CTL-Q (+Q) RESUME OUTPUT AFTER TEMPORARILY STOPPING BY CTL-S/<CR><LF>
016661 103 046124 051455 .ASCII /CTL-S (+S) TEMPORARILY STOP OUTPUT TO CONSOLE/<CR><LF>

```


MAINDEC-11-DZDNI-B NEW D4:11-B RESPONDER
DZDNI.B.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 112

SEP 01:12
SEP 01:12

MESSAGES

016730	052103	026514	020125
016776	052522	047502	052125
017045	015	012	012
	002551		
017051	215	212	
017059	124	047517	041440
017107	377		

HELPLN
TOOC:

```

.ASCII /CTL-U (^U) DELETE CURRENT INPUT LINE (^CP ^LF
.ASCII /RUBOUT -- DELETE LAST CHARACTER INPUT (^CP ^LF
.BYTE CR,LF,LF,LF
= -HELPLN
.BYTE 215,212
.ASCII /TOO CLOSE TO 200000 BOUNDAR-
.BYTE 377
.LIST BEY
.EVEN
.ENC

```

433
433

00001

TS* TYP	012532	2358*	2368*	3067	3524	3626	3839	3971	4054	4272	4863*		
TTADDR	013062	2497*	2748	3436	4898*								
TTPTR	013060	2498*	2749*	3295	3378	3411	3437*	4897*					
TTSIZE =	001000	2025*	2499	2750	2893	2898	2900	3296	3423	3433			
TTYSTP	013073	4430*	4434*	4507	4905*								
UCHK =	000002	2096*	2490	2530	2600	2601	3053	3641	4085	4087			
UCHKS =	002000	2046*											
UEXP =	000001	2097*											
UNADDR	012516	2200*	2213*	2214*	2405	4856*							
UNMSG	013175	2202	4930*										
UNTRP	011742	2222	4738*										
VBUFA	013102	2426*	2453*	2474	4911*								
VCMOTB	002610	2525	2592*										
VECTAD	012520	2223*	2235*	2415	4857*								
VECTMS	013232	2225	4930*										
VENO	013126	2166	4921*										
VSTRT	012636	2165	2166	4881*									
WK	013056	3139*	3140	3142*	3146	4895*							
WK1	013057	3143	3145*	4896*									
XADDR =	013112	2576*	2756	4915*									
.	= 017110	2111*	2119*	2126*	2603*	3970	4872*	4873	4882	4863*	4887*	4888	4930*

. ABS. 017110 000

ERRORS DETECTED: 0

CZDXIB, CZDXIB/SOL/CRF=CZDXIB.DOC, CZDXIB.P11
RUN-TIME: 5 10 .7 SECONDS
RUN-TIME RATIO: 356 17=20.8
CORE USED: 13 PAGES

C10