

DX11-B

RESPONDER
CZDXIBO

AH-8777B-MC

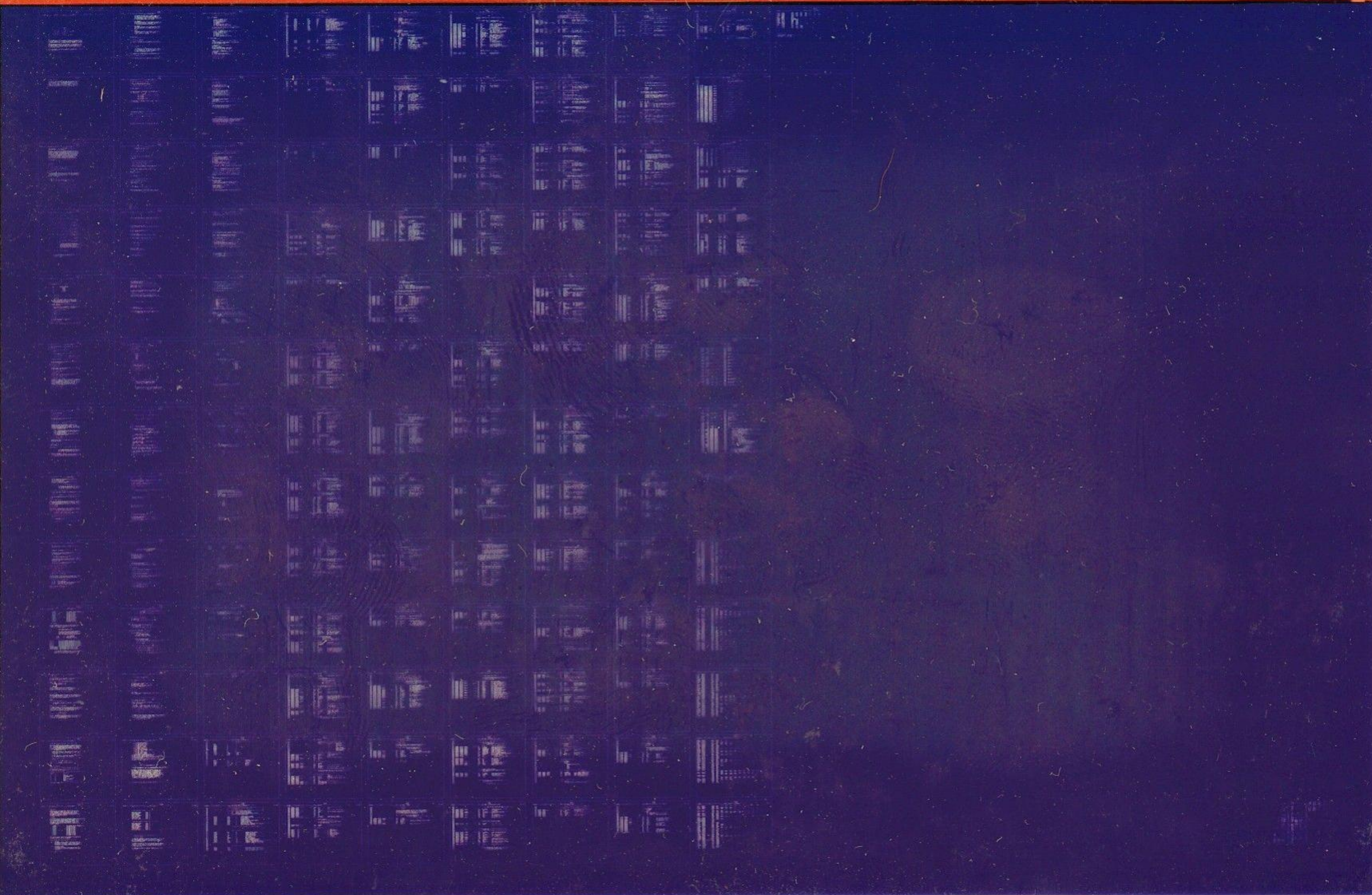
COPYRIGHT ©72-78

FICHE 1 OF 1

APR 1978

digital

MADE IN USA



B01

EOF10FFPCBSEQ 00010300 780330 PDP10 411 0#HDR10ZDX18SEQ 00010000 780330
PDP10 4880 0002 MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER MACY11 30A(1052) 01-FEB-78 09:46 PAGE 1
CZDX18.DOC 30-JAN-78 14:26 OPERATING PROCEDURE SEQ 0001
.REM

IDENTIFICATION

PRODUCT CODE: AC-8776B-MC
PRODUCT NAME: CZDX180 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 OBSOLETE
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

0-777	I	INTERRUPT VECTORS (256 WORDS)	I
	I		I
	I		I
1000-1777	I	DX11-B TEST PROGRAM (4K WORDS)	I
	I		I
	I		I
10-X777	I	SPW TABLE (256 WORDS)	I
	I		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 (IU) TO DELETE THE ENTIRE ENTRY
- C. 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
100000 IS ENTERED AS 100.

NOTE: THE BUFFER CANNOT BE CLOSER THAN 24000(8)
TO ANY 200000 BOUNDARY OR TO THE I/O PAGE. THE
THE DX IS NOT CAPABLE OF HAVING THESE BUFFERS
CROSS A 200000 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 (IU) 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:

KXX 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 OCT. 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 DY 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.

7.2 REM %
 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 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.

THIN: 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

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.DOC 30-JAN-78 14:26

MACY11 30A(1052
PROGRAM DESCRIPTION

01-FEB-78 09:46 PAGE 15

SEQ 0016
SEQ 0016

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.

GOGB: 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 SDEVIB
 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 IS 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 TCMAC -
 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. 0. 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.

0 = 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 PHONEY 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

PNXM: 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

PFARER: 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 VECTORING
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 IMPORTANT. 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)
SENSE 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 -----

SEY: 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 AND 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 IT 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 IT 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 IT 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
EBDMP: 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 *
      ---- SENSCH: = 4
      ---- SWRITE: & MWRITE: = 5
      ---- SREAD: & MREAD: = 6
      ---- ERASCH: = 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:
P/ISS:
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. SENSEM: MOV #SSENSE,DXBA

SET UP IN
-----PHIS:
PCHENC:USED IN

SENSECM:

CLEARED IN

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

SSTAT (3)

SSTAT	-----	UCHK	= 2
	-----	CEIDE	= 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,DX05

SET UP IN

USED IN

CLEARED IN

H03

MAINDEC-11-DZDAI-B NEW DX11-B RESPONDER
CZDXIB.DOC 30-JAN-78 14:26

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 32
PROGRAM DESCRIPTION

SEQ 0033
SEC 0033

PPARER:
ESEQ:
CONJNE:
SATIN:

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

INPUT:
SSRMI:

CLEARED IN

PSYSRT:
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

INIT: INIT14C:

USED IN

DUMP:
DISCTL:
MWRITE: & SWRITE:

SOUTB (10)

SOUTB ----- 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 (SRBYTC)--
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:

L03

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

SEQ 0037
SEQ 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

1966		:		
1967		:		
1968		:		
1969		:		
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 *      DEVICE BUFFER LAYOUT      (1 PER DEVICE)
1983 *
1984 *****
1985 *      LOC      0-61 = DEVICE STATUS TABLE
1986 *      LOC      62-543 = DEVICE INPUT BUFFER
1987 *      LOC      554-1023 = DEVICE OUTPUT/DISPLAY BUFFER
1988 *****
1989 *      LAYOUT OF DEVICE STATUS TABLE
1990 *
1991 *      SCMD      =      0      : CURRENT DEVICE COMMAND
1992 *      SLCMD     =      1      : LAST COMMAND, IF WRITE
1993 *      SSENSE    =      2      : DEVICE SENSE BYTE (NOTE -- MUST BE EVEN BYTE LOCATION)
1994 *      SSTAT     =      3      : DEVICE STATUS
1995 *      SCURS     =      4      : CURSOR POSITION
1996 *      SINBF     =      6      : ADDRESS OF DEVICE INPUT BUFFER
1997 *      SOUTB     =     10      : ADDRESS OF DEVICE OUTPUT BUFFER
1998 *      SBUFA     =     12      : CURRENT BUFFER PTR
1999 *      SRBYTC    =     14      : REMAINING BYTE COUNT
2000 *      SONLF     =     16      : DEVICE ONLINE - INDICATOR 0=ON-LINE 1=OFF-LINE
2001 *      SRDRQ     =     17      : READ MANUAL INPUT REQUEST -- IF NON-ZERO
2002 *      SMINS     =     20      : START OF MANUAL INPUT DATA
2003 *
2004 *      LOCATIONS 22-77 ARE AVAILABLE FOR EXPANSION PURPOSES
2005 *
2006 *
2007 *
2008 *
2009 *      OTHER      DX EQUATES
2010 *
2011 *      DEV       =      R0      : CURRENT DEVICE NUMBER
2012 *      DTAB      =      R3      : ADDRESS OF CUR DEV STATUS TABLE
2013 *      TT1       =      R4      : TUMBLE TABLE ENTRY 1
2014 *      TT2       =      R5      : TUMBLE TABLE ENTRY 2
2015 *      CEDE      =      3      : CHAN END & DEV END
2016 *      QCUE      =     10      : CODE TO QUE CONTROL UNIT END
2017 *      NOP       =      3      : NOP COMMAND
2018 *      NEWLINE   =     25      : NEW LINE CHARACTER
2019 *      EBCDSP    =     100      : EBCDIC SPACE CODE
2020 *      SMI       =     112      : START OF MESSAGE INDICATOR
2021 *      EOM       =     152      : END OF MESSAGE INDICATOR
2022 *      LINSZ     =     40      : NUMBER OF CHARACTERS PER LINE OF 2260 DISPLAY
2023 *      NOLIN     =     12      : NUMBER OF LINES PER 2260 DISPLAY
2024 *      DISPSZ    =     NOLIN*LINSZ : NUMBER OF CHARACTERS ON THE DISPLAY
2025 *      TTSIZE    =     512      : NUMBER OF ENTRIES IN TUMBLE TABLE
2026 *
2027 *
2028 *
2029 *      360 COMMAND EQUATES
2030 *
2031 *      CMWRT     =      1      : WRITE DATA (FROM 360 TO PDP-11)
2032 *      CMRMI     =      2      : READ MANUAL INPUT (PDP-11 TO 360)
2033 *      CMWTLA    =      5      : WRITE LINE ADDRESS (360 TO PDP-11)
2034 *      CMREAD    =      6      : READ FULL BUFFER (PDP-11 TO 360)
2035 *      CMSRMI    =     12      : SHORT READ MANUAL INPUT (PDP-11 TO 360)

```


DX REGISTER ASSIGNMENTS & LAYOUTS

DXDS OR TUMBLE TABLE ENTRY 1 (TT1)

2036		:			
2037		:			
2038		:			
2039		:			
2040		:			
2041	100000	PARER =	100000	:	PARITY ERROR DETECTED
2042	040000	NXM =	40000	:	NON EXISTENT MEMORY CONDITION
2043	020000	SELST =	20000	:	IBM SELECTIVE RE-SET
2044	010000	SYSRST =	10000	:	IBM SYSTEM RESET
2045	004000	INFOSC =	4000	:	IBM PROGRAMMED INTERFACE DISCONNECT
2046	002000	UCHKS =	2000	:	UNIT CHECK WAS PRESENTED TO THE CHANNEL
2047	001000	CHENDS =	1000	:	CHANNEL END WAS PRESENTED TO THE CHANNEL
2048	000400	BYSS =	400	:	BUSY WAS PRESENTED TO THE CHANNEL
2049	000200	CHIS =	200	:	CHANNEL INIT SELECTION SEQ WAS COMPLETED
2050	000100	ESEND =	100	:	CHANNEL ACCEPTED LAST STATUS
2051	000040	CHEND =	40	:	CHANNEL DATA TRANSFER END
2052	000020	CUEND =	20	:	DX DATA TRANSFER END
2053	000010	ISSREJ =	10	:	INIT SELECTION SEQ WAS REJECTED
2054	000004	CMDCHN =	4	:	CHANNEL SPECIFIED COMMAND CHAINING
2055	000002	STKSTB =	2	:	CHANNEL COULD NOT ACCEPT LAST STATUS
2056	000001	CMDREJ =	1	:	CHANNEL COMMAND WAS REJECTED
2057					
2058					
2059					

DXCA OR TUMBLE TABLE ENTRY 2 (TT2)

BITS 15-8 = COMMAND (IF ANY)
BITS 7-0 = DEVICE ADDRESS

DXCS CONTROL UNIT STATUS REGISTER

2067		:			
2068	004000	BSYEN =	4000	:	BUSY RENABLE - FOR SELECTOR CHANNELS
2069	001000	DXONLN =	1000	:	ON-LINE INDICATION
2070	000400	CUBUSY =	400	:	CONTROL UNIT BUSY
2071	000200	DONE =	200	:	DONE FLAG
2072	000100	DXENB =	100	:	INTERRUPT ENABLE
2073					
2074					

BITS 4+3 ARE SET IF EXTENDED ADDRESS IS USED - 32K
BITS 2-0 ARE THE FUNCTION TO BE PERFORMED

2076		:			
2077	000001	DXRST =	1	:	DX RESET COMMAND
2078	000003	DXWR =	3	:	WRITE DATA TO THE 360
2079	000005	DXRD =	5	:	READ DATA FROM THE 360
2080	000007	DXST =	7	:	SEND STATUS TO THE 360
2081					
2082					

DXOS OFFSET AND STATUS REGISTER

BITS 15-10 OFFSET OF SPW TABLE
STATUS REGISTER DEF (SSTAT) - STATUS BYTE

2087		:			
2088		:			
2089		:			
2090	000200	ATTN =	200	:	ATTENTION
2091	000100	STAMOD =	100	:	STATUS MODIFIER

C04

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 41
PROGRAM EQUATES AND DEVICE ASSIGNMENTS

SEG 0041
SEG 0041

2092	000040	CUE	=	40	:CONTROL UNIT END
2093	000020	BSY	=	20	:BUSY
2094	000010	CE	=	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

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 42
TRAP CATCHER

SEQ 0042
SEQ 0042

2106			.SBTTL	TRAP CATCHER	
2107			:		
2108			:	THE TRAP CATCHER IS LOADED INTO LOW CORE	
2109					
2110		000000	.=0		
2111	000000	000000	HALT		;FOR MEMORY MANAGEMENT
2112	000002	000000	HALT		
2113		000000	.REPT	176	;TRAP CATCHER
2114		000176			
2115			.WORD	.+2	
2116			HALT		
2117			.ENDR		
2118			.=200		
2119	000200	000200	JMP	START	;ESTABLISH LOC 200 STARTING ADDRESS
2120	000200	00013*			
2121		001000			

E04

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 43
PROGRAM START-UP SEQUENCES

SEQ 0043
SEQ 0043

```

2122      .SBTTL  PROGRAM START-UP SEQUENCES
2123      :
2124      :      NORMAL SYSTEM START LOCATION  --1000
2125      :
2126      :      .=1000
2127      001003  001000      START:  BR      SYSINT      ;NORMAL START UP
2128      :      000402
2129      :      RESTART ADDRESS  -- REENTER ALL PARAMETERS  -- 1002
2130      :
2131      001002  005037  013132  RSTART: CLR      FTIMFL      ;RESET FIRST TIME FLAG TO FORCE PARAMETER REENTRY
2132      :
2133      :      SYSTEM GENERALIZED INITIALIZATION PROCEDURE
2134      :
2135      :      SET-UP  STACK POINTER
2136      :      TRAP/VECTOR AREA
2137      :      SYSTEM GENERATED TRAPS
2138      :      TELETYPE (CONSOLE) VECTORS + STATUS REGISTERS
2139      :      CLEAR ALL LIVE SYSTEM VARIABLES
2140      :      SET UP TELETYPE INPUT / OUTPUT BUFFERS
2141      :
2142      001006  012706  012636  SYSINT: MOV      #SSACK,SP      ;SET UP THE STACK POINTER
2143      001012  000005      RESET      ;RESET ALL DEVICES
2144      :
2145      :
2146      :
2147      :
2148      :      SET UP CONSOLE VECTORS
2149      :
2150      001014  012700  000060      MOV      #60,R0      ;START OF CONSOLE VECTORS
2151      001020  012720  010744      MOV      #TKIN,(R0)+    ;BEG OF TELE INPUT ISP
2152      001024  012720  000340      MOV      #340,(R0)+    ;NEW PROC STATUS
2153      001030  012720  011232      MOV      #PISR,(R0)+    ;BEG OF TELE PRINT ISR
2154      001034  012710  000340      MOV      #340,(R0)      ;NEW PROC STATUS
2155      :
2156      :      SET UP MISC TRAPS
2157      :
2158      001040  012737  011672  000004      MOV      #MTO,4      ;MEMORY TIME OUT TRAP
2159      001046  012737  000340  000006      MOV      #340,6      ;
2160      001054  012737  011700  000250      MOV      #MMERR,250    ;MEMORY MANAGEMENT ERROR
2161      001062  012737  000340  000252      MOV      #340,252    ;
2162      :
2163      :      CLEAR ALL VARIABLES
2164      :
2165      001070  012700  012636      MOV      #VSTART,R0      ;START OF VARIABLES
2166      001074  012701  000272      MOV      #VEND-VSTART+2,R1 ;# OF VARIABLES TO CLEAR
2167      001100  105020      10$:  CLRB      (R0)+    ;CLEAR A BYTE
2168      001102  005301      DEC      R1      ;DONE?
2169      001104  001375      BNE      10$      ;NO, CONTINUE CLEARING
2170      :
2171      :      SET UP TELE BUFFER POINTERS
2172      :
2173      001106  012737  012640  012742      MOV      #TBUF,TPTR    ;TELE INPUT POINTER
2174      001114  012737  012744  013046      MOV      #PBFS,PFPTTR    ;TELE OUT FETCH PTR
2175      001122  012737  012744  013050      MOV      #PBFS,PPPTR    ;TELE OUT PUT PTR
2176      :
2177      :      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

G04

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 45
TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)

SEG 0045
SEG 0045

```

2185 .SBTTL TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)
2186
2187 :
2188 :
2189 :
2190 :
2191 :
2192 :
2193 :
2194 :
2195 :
2196 :
2197 :
2198 :
2199 :
2200 :
2201 :
2202 :
2203 :
2204 :
2205 :
2206 :
2207 :
2208 :
2209 :
2210 :
2211 :
2212 :
2213 :
2214 :
2215 :
2216 :
2217 :
2218 :
2219 :
2220 :
2221 :
2222 :
2223 :
2224 :
2225 :
2226 :
2227 :
2228 :
2229 :
2230 :
2231 :
2232 :
2233 :
2234 :
2235 :
2236 :
2237 :
2238 :
2239 :
2240 :

```

001162 004737 011330
001166 004137 011472
001172 013134

001174 005037 013132
001200 012737 176200 C12516
001206 004137 002632
001212 013175
001214 001260
001216 120427 000015
001222 001364
001224 020327 176200
001230 002761
001232 020327 177000
001236 003356
001240 032703 000037
001244 001353
001246 012737 001174 000004
001254 00337 012516
001260 005077 011232

GE*PRM: JSR PC,CRLF :RESTORE THE CARRIAGE
JSR R1,IESG :PRINT START-UP MSG
.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 ENTERED
MOV #176200,UNADDR :SET UP DEFAULT ADDRESS
JSR R1,INOCT :GET UNIBUS ADDRESS
.WORD UNMSG
.WORD 55 :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
:TRAP WILL OCCUR IF INVALID 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 205 :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

H04

 MAINDEC-11-D2D\1-B NEW D\11-B RESPONDER
 C2D\1B.F11 30-JAN-78 14:19

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

 SEQ 0046
 SE% 0046

```

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 CHTMS ;
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
2292 001516 022626 55$: CMP (SP)+,(SP)+ ;DUMP PC AND PSW SAVED BY INTERRUPT
2293 001520 005037 177776 CLR PSW ;TURN DOWN PROCESSOR STATUS
2294 001524 004137 011472 JSR R1,MESG ;PRINT "NO MEM MANAGEMENT AVAIL"
2295 001530 014102 .WORD PNOMM ;
2296 ;ASK TO HAVE QUESTION REENTERED

```

```

2297      :
2298      :
2299      :
2300      :
2301      :
2302      :
2303      :
2304      :
2305      :
2306      :
2307      :
2308      :
2309      :
2310      :
2311      :
2312      :
2313      :
2314      :
2315      :
2316      :
2317      :
2318      :
2319      :
2320      :
2321      :
2322      :
2323      :
2324      :
2325      :
2326      :
2327      :
2328      :
2329      :
2330      :
2331      :
2332      :
2333      :
2334      :
2335      :
2336      :
2337      :
2338      :
2339      :
2340      :
2341      :
2342      :
2343      :
2344      :
2345      :
2346      :
2347      :
2348      :
2349      :
2350      :
2351      :
2352      :

```

001532 105037 012527
 001536 004137 002632
 001542 013376
 001544 001600
 001546 120427 000116
 001552 001412
 001554 120427 000131
 001560 001364
 001562 105237 012527
 001566 012737 001516
 001574 005037 177572

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 BFREMS
 .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, MSG ; 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 T00C ; 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-DZDI-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          :          D = 2848 RESPONDER DIAGNOSTIC
2354          :          F = FRIEND
2355          :
2356          :          DEFAULT = 'F', FRIEND
2357          :
2358 001710 105037 012532 90$: CLR B TSTYP ;RESET TEST TYPE
2359 001714 004137 002632 JSR R1, INOCT ;GET TEST TYPE
2360 001720 013534 .WORD TESTMS
2361 001722 001752 .WORD 100$ ;DEFAULT TO FRIEND
2362 001724 112737 000100 012534 MOVB #EBCDSP, FILLCH ;FOR 2848 SET FILL CHAR TO EBCDIC SPACE
2363 001732 120427 000104 CMPB R4, #'D ;D? --2848 RESPONDER DIAGNOSTIC --
2364 001736 001433 BEQ INIT ;YES, 2848 TEST
2365 001740 120427 000106 CMPB R4, #'F ;F? -- FRIEND TEST --
2366 001744 001361 BNE 90$ ;ILLEGAL ENTRY
2367 001746 105037 012534 CLR B FILLCH ;FRIEND MODE -- DEFAULT FILL CHAR TO NULL
2368 001752 105237 012532 100$: INCB TSTYP ;SET TEST TO FRIEND
2369
2370          :
2371          :          FRIEND TEST ONLY
2372          :          DETERMINE IF SEPARATE INPUT / OUTPUT BUFFERS ARE TO BE USED
2373          :          Y = YES, MAINTAIN SEPARATE INPUT / OUTPUT BUFFERS
2374          :          N = NO, USE SAME BUFFER FOR INPUT AND OUTPUT
2375          :          DEFAULT IS 'N', NO, USE SAME BUFFER FOR INPUT / OUTPUT
2376          :
2377 001756 105037 012533 110$: CLR B IOBUF ;0 = NO, 1 = YES
2378 001762 004137 002632 JSR R1, INOCT ;SEPARATE I/O BUFFERS?
2379 001766 013574 .WORD FIOMS
2380 001770 002026 .WORD INIT ;DEFAULT TO NO
2381 001772 120427 000116 CMPB R4, #'N ;N? -- NO, SAME I/O BUFFER --
2382 001776 001413 BEQ INIT ;IF EQ, USE SAME I/O BUFFER FOR INPUT AND OUTPUT
2383 002000 120427 000131 CMPB R4, #'Y ;Y? --YES, SEPARATE I/O BUFFERS--
2384 002004 001364 BNE 110$ ;ERROR, REQUEST INPUT AGAIN
2385 002006 105237 012533 INCB IOBUF ;SET SEPARATE I/O BUFFER INDICATOR
2386
2387          :
2388          :          FRIEND TEST MODE ONLY
2389          :          GET BUFFER FILL CHARACTER (HEX INPUT REQUIRED).
2390          :          ANY VALUE WILL BE ACCEPTED
2391          :
2392 002012 004137 002624 JSR R1, INHEX ;FILL CHARACTER
2393 002016 013636 .WORD FILLMS
2394 002020 002026 .WORD INIT
2395 002022 110337 012534 MOVB R3, FILLCH

```

K04

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

 PROGRAM MACY11 30A(1052) 01-FEB-78 09:46 PAGE 49
 INITIALIZATION

 SEQ 0049
 SEQ 0049

```

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      MOV      #1,MMSRO      ;ENABLE MEMORY MANAGEMENT
2453 002220 012737 000020 013102      MOV      #20,VBUFA    ;TO 8K BANK OR 20000 AND UP
2454                                     :
2455                                     : START SETTING UP SPW TABLE
2456                                     : 1 ENTRY PER DEVICE (256 DEVICES)
2457                                     : ENTRY DESCRIPTION
2458                                     : ----FOR VALID DEVICE NUMBERS
2459                                     : BITS 15-8 = OFFSET TO DST TABLE (PHYSICAL ADDR
2460                                     : 7-0 = 0
2461                                     : ----FOR INVALID DEVICE NUMBERS
2462                                     : BITS 15-8 = 0
2463                                     : 7-0 = 2 -- UNIT CHECK
2464                                     :
2465                                     : THIS TABLE IS REFERENCED ON EACH 360 ACTION TO DETERMINE
2466                                     : IF DEVICE NUMBER IS VALID. THIS AUTOMATICALLY DONE
2467                                     : BY THE DX CONTROL UNIT
2468                                     :
2469 002226 013705 013100      40$:      MOV      FBUFA,R5      ;COMPUTE OFFSET PHYSICALLY
2470 002232 000305                                     : *1000
2471 002234 105005      SWAB      R5
2472 002236 006305      CLRB     R5
2473 002240 010577 010226      ASL      R5
2474 002244 013701 013102      MOV      R5,DXOS      ;OFFSET TO SPW TABLE
2475 002250 000301      MOV      VBUFA,R1      ;COMPUTE VIRT ADDR OF SPW TABLE
2476 002252 105001      SWAB      R1
2477 002254 006301      CLRB     R1
2478 002256 010137 013114      ASL      R1
2479 002262 010137 013104      MOV      R1,STSPW     ;SAVE START OF SPW TABLE
2480 002266 160537 013104      MOV      R1,PHYOFF    ;COMPUTE THE OFFSET FOR PHYSICAL ADDRESSES
2481 002272 062705 003000      SUB      R5,PHYOFF    ;VERSES VIRTUAL ADDRESS - FOR MEM MANAGEMENT
2482 002276 010537 013116      ADD      #3000,R5     ;COMPUTE THE OFFSET TO THE DST TABLE
2483 002302 005000      MOV      R5,DSTOFF          ;SAVE OFFSET TO DST TABLE
2484 002304 120037 012522      CLR      DEV        ;START AT DEVICE 0
2485 002310 002405      50$:      CMPB     DEV,SDEV    ;IS DEVICE NUMBER VALID
2486 002312 120037 012524      BLT      60$          ;NO
2487 002316 003002      CMPB     DEV,EDEV
2488 002320 010521      BGT      60$          ;NO
2489 002322 000402      MOV      R5,(R1)+          ;VALID DEVICE DST OFFSET TO ENTRY
2490 002324 012721 000002      BR       70$          ;
2491 002330 005200      60$:      MOV      #UCHK,(R1)+ ;INVALID DEV # UNIT CHECK TO ENTRY
2492 002332 020027 000400      70$:      INC      DEV    ;TO NEXT DEVICE
2493 002336 001362      CMP      DEV,#256.         ;ALL DEVICES DONE?
2494                                     : NO, SET UP SPW FOR NEXT DEVICE
2495                                     :
2496                                     : NEXT SET UP TUMBLE TABLE AND DUPLICATE TUMBLE TABLE
2497                                     :
2498 002340 010137 013062      MOV      R1,TTADDR     ;TUMBLE TABLE ADDRESS
2499 002344 010137 013060      MOV      R1,TTPTR     ;TUMBLE TABLE FETCH POINTER
2500 002350 012702 001000      MOV      #TTSIZE,R2    ;CLEAR T/T + DUPLICATE T/T (WORD POINTER
2501 002354 005021      80$:      CLR      (R1)+      ;CLEAR NEXT WORD
2502 002356 005302      DEC      R2                  ;DONE?
2503 002360 001375      BNE      80$          ;NO, CLEAR NEXT WORD
2504                                     :
2505                                     : SET UP DST TABLE
2506                                     : THE DST TABLE IS USED TO VERIFY COMMANDS FROM THE
2507                                     : 360, THIS IS DONE BY THE HARDWARE
2508                                     : THE DST TABLE IS A BYTE TABLE, 1 BYTE PER POSSIBLE

```

```

2508      :      COMMAND 0-255. THE ENTRY IN THE DST TABLE IS
2509      :      SENT TO THE 360.
2510      :      THE FOLLOWING ARE A LIST OF VALID COMMANDS AND RESPONSES
2511      :      COMMAND      RESPONSE      DESCRIPTION
2512      :      0            0            TEST I/O
2513      :      1            0            WRITE BUFFER
2514      :      2            0            READ MANUAL INPUT
2515      :      3            CE!DE        NOP
2516      :      4            0            SENSE COMMAND
2517      :      5            0            WRITE LINE ADDRESS
2518      :      6            0            READ FULL BUFFER
2519      :      7            0            ERASE COMMAND
2520      :      12           0            SHORT READ MANUAL INPUT
2521      :
2522      :      ALL OTHER COMMANDS ARE RESPONDED WITH UNIT CHECK
2523      :
2524      :      MOV      #13,R2      ;NUMBER OF VALID 360 COMMANDS
2525      :      MOV      #VCMDTB,R3  ;VALID COMMAND TABLE
2526      :      INIT10: MOV      (R3)+,(R1)+ ;TO DST TABLE
2527      :      DEC      R2           ;DONE?
2528      :      BNE     INIT10      ;NO, MOVE IN NEXT RESPONSE
2529      :      MOV      #245,R2     ;MOVE UNIT CHECK TO INVALID ENTRIES
2530      :      100$: MOV      #UCHK,(R1)+
2531      :      DEC      R2
2532      :      BNE     100$
2533      :
2534      :      COMPUTE MAX NUMBER OF DEVICES
2535      :
2536      :      MOV      EDEV,R3
2537      :      SUB      SDEV,R3
2538      :      INC      R3           ;START AT DEVICE NUMERO UNO
2539      :      MOV      R3,MAXDEV
2540      :      MOV      SDEV,DEVCON  ;SET UP DEVICE NUMBER -1
2541      :      DEC      DEVCON
2542      :      MOV      #1,SELDEV   ;INIT DEVICE NUMBER FOR MUX AND SEL EXECUTORS
2543      :      .....
2544      :
2545      :      NOTE -- THE DEVICE BUFFERS ARE USED BY THE SOFTWARE ONLY TO CONTAIN
2546      :      POINTERS AND INPUT AND OUTPUT DATA FOR EACH DEVICE;
2547      :
2548      :      .....
2549      :
2550      :      START SETTING UP DEVICE BUFFERS
2551      :
2552      :      MOV      R1,SDEVTB    ;SAVE START OF DEVICE BUFFERS
2553      :      CLR      DEV         ;DEV # 0
2554      :      120$: MOV      R1,D TAB ;SAVE ADDR OF DEVICE STATUS TABLE
2555      :      MOV      #272,R2     ;CLEAR DEVICE STATUS TABLE + INPUT BUFFER
2556      :      122$: CLR      (R1)+
2557      :      DEC      R2           ;DONE?
2558      :      BNE     122$        ;NO, CLEAR NEXT WORD
2559      :      MOV      DTAB,SINBF(DTAB)
2560      :      ADD      #62,SINBF(DTAB) ;COMPUTE ADDRESS OF INPUT BUFFER
2561      :      MOV      DTAB,SOUTB(DTAB)
2562      :      ADD      #544,SOUTB(DTAB) ;COMPUTE ADDRESS OF OUTPUT BUFFER
2563      :      MOV      #DISP$Z,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	PBUF A, 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
2578							
2579							
2580							
2581							
2582							
2583							
2584							
2585							
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
2589							
2590							
2591							
2592	002610	000					
2593	002611	000					
2594	002612	000					
2595	002613	014					
2596	002614	000					
2597	002615	000					
2598	002616	000					
2599	002617	000					
2600	002620	002					
2601	002621	002					
2602	002622	000					
2603		002624					

```

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

JSR     R1, MMSG          ; TELL OPERATOR WE ARE READY TO GO
.WORD   RNMMSG
JMP     EXEC              ; GET THE SHOW ON THE ROAD

VALID COMMAND TABLE
VCMDTB: .BYTE 0           ; 0 = TEST I/O
        .BYTE 0           ; 1 = WRITE BUFFER
        .BYTE 0           ; 2 = READ MANUAL INPUT
        .BYTE 0           ; 3 = NOP
        .BYTE CE!DE       ; 4 = SENSE COMMAND
        .BYTE 0           ; 5 = WRITE LINE ADDRESS
        .BYTE 0           ; 6 = READ FULL BUFFER
        .BYTE 0           ; 7 = ERASE COMMAND
        .BYTE UCHK        ; 10 = INVALID
        .BYTE UCHK        ; 11 = INVALID
        .BYTE 0           ; 12 = SHORT READ MANUAL INPUT
        .EVEN

```

```

2604 .SBTTL  INITIALIZATION PARAMETER INPUT AND CONVERSION ROUTINES
2605
2606 INHEX -- PRINT MESSAGE, WAIT FOR INPUT, GET IT AND CONVERT THE HEX TO BINARY
2607
2608 CALLING SEQUENCE
2609 JSR  R1, INHEX
2610 .WORD ADDRESS OF MESSAGE TO BE PRINTED
2611 .WORD ADDRESS OF DEFAULT ROUTINE
2612 .....RETURN
2613 R2 = NEXT CHAR POINTER
2614 R3 = BINARY RESULT
2615 R4 = (BITS 0-7) FIRST NON-OCTAL CHARACTER
2616 R5 = NUMBER OF CHARACTERS CONVERTED
2617
2618 002624 012705 011604 INHEX: MOV  #CHTB,R5 ;MOVE ADDRESS OF CONVERSION ROUTINE TO R5
2619 002630 000402 BR   INRS
2620
2621 INOCT -- PRINT MESSAGE, WAIT FOR INPUT, + GET IT AND CONVERT OCTAL TO BINARY
2622
2623 CALLING SEQUENCE
2624 JSR  R1, INOCT
2625 .WORD ADDRESS OF MESSAGE TO BE PRINTED
2626 .WORD ADDRESS OF THE DEFAULT ROUTINE
2627 .....RETURN
2628 R2 = NEXT CHAR PTR
2629 R3 = BINARY RESULT
2630 R4 = (BITS 0-7) FIRST NON-OCTAL CHARACTER
2631 R5 = NUMBER OF CHARS CONVERTED
2632
2633 002632 012705 011540 INOCT: MOV  #COTB,R5 ;SET UP ADDRESS OF THE CONVERSION ROUTINE
2634 002636 012102 INRS: MOV  (R1)+,R2 ;GET ADDRESS OF THE MESSAGE
2635 002640 004737 011404 10$: JSR  PC,FRAMESG ;PRINT THE DESIRED MESSAGE
2636 002644 105037 013053 CLRB  TCMACT ;RESET ACTIVE FLAG
2637 002650 105037 013054 CLRB  TCMDB ;RESET ABORT FLAG
2638 002654 105737 013054 30$: TSTB TCMDB ;COMMAND ABORT?
2639 002660 001367 BNE  10$ ;YES, REASK QUESTION
2640 002662 105737 013053 TSTB TCMACT ;WAS ENTRY COMPLETED?
2641 002666 001772 BEQ  30$ ;NO, WAIT
2642 002670 012702 012640 MOV  #TBUF,R2 ;SET UP ADDRESS OF BEG OF INPUT BUFFER
2643 002674 004715 JSR  PC,IR5 ;CONVERT INPUT TO BINARY
2644 002676 005705 TST  R5 ;LOOK FOR DEFAULT RESP -- C/R
2645 002700 001005 BNE  40$ ;NOT DEFAULT TAKE NORMAL RETURN
2646 002702 120427 000015 CMPB  R4,#CR ;ILLEGAL CHAR MUST BE A C/R
2647 002706 001002 BNE  40$ ;ITS NOT A DEFAULT
2648 002710 011101 MOV  (R1),R1 ;---TAKE THE DEFAULT RETURN
2649 002712 000201 RTS  R1
2650 002714 005721 40$: TST  (R1)+ ;INCR FOR NORMAL RETURN
2651 002716 000201 RTS  R1

```


2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707

002720 004737 011330
002724 012702 000052
002730 004737 011346
002734 012706 012636
002740 105037 013053
002744 105037 013054
002750 105037 013055
002754 105737 013072
002760 001402
002762 000137 003340
002766 105737 013053
002772 001001
002774 000767

002776 012702 012640
003002 112203
003004 042703 177400
003010 012704 003044
003014 020324
003016 001411
003020 022404
003022 005714
003024 001373

003026 012702 137607
003032 004737 011346
003036 000137 002720

003042 000134

003044 000101
003046 004050
003050 000104
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 #SSSTACK, SP ;RE-ESTABLISH PUSH STACK
CLRB TCMAC1 ;CLEAR TELE CMD ACT
CLRB TCMDB ;CLEAR TELE CMD ABORT
CLRB LINECT ;RESET LINE COUNTER
10\$: TSTB DXABFL ;DID THE DX ABORT AN OPERATION ?
BEQ 20\$;NO, CONTINUE
JMP STOPDX ;YES IT DID, PRINT THE DX REGISTERS
20\$: TSTB TCMAC1 ;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 #TCMDB, R4 ;SET UP PTR TO COMMAND TABLE
40\$: CMP R3, (R4)+ ;DOES COMMAND MATCH TABLE ENTRY ?
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
TCMDB: .WORD 'A ;A = ACCESS
.WORD ACCESS
.WORD 'D ;D = DUMP
.WORD DUMP

D05

MAINDEC-11-DZDXI-B NEW DXI1-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
CZDX18.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 R = RUN COMMAND
2726
2727 THE RUN COMMAND READIES THE DX AND SPECIFIED DEVICE
2728 BUFFERS TO BEGIN OPERATION. THE RUN COMMAND MUST
2729 BE EXECUTED BEFORE ANY ACTION WILL BE PERFORMED
2730 OVER THE DX.
2731
2732 003112 032777 001000 007350 RUN: BIT #DXONLN,DXCS ;IS DX ENABLED?
2733 003120 001342 BNE CERR ;YES, ERROR
2734 003122 005077 007342 CLR DXCS ;INITIALIZE THE DX
2735 003126 005277 007336 INC DXCS ;SET GO
2736 003132 012700 000001 MOV #1,DEV ;START CLEARING DEVICE TABLES
2737 003136 004737 010256 10$: JSR PC,CDEVST ;CLEAR DEV STATUS TABLE
2738 003142 004737 010310 JSR PC,CSPWST ;RESET THE APPR 3 W STATUS ENTRY FOR THE DEVICE
2739 003146 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
2740 003152 105063 000017 CLRB SRDRQ(DTAB) ;CLEAR THE READ REQUEST
2741 003156 005063 000020 CLR SMINS(DTAB) ;CLEAR THE START OF MANUAL INPUT
2742 003162 005200 INC DEV ;INCR TO NEXT DEVICE
2743 003164 120037 013067 CMPB DEV,MAXDEV ;ARE WE DONE
2744 003170 003762 BLE 10$ ;NO, DO NEXT DEVICE
2745 003172 105037 013070 CLRB DXACT ;CLEAR DX ACTIVE FLAG
2746 003176 105037 013074 CLRB CMDCHF ;CLEAR COMMAND CHAINING FLAG
2747 003202 105037 013072 CLRB DXABFL ;CLEAR DX ABORT FLAG
2748 003206 013701 013062 MOV TTADDR,R1 ;GET THE TUMBLE TABLE ADDRESS
2749 003212 010137 013060 MOV R1,TTPT ;RESET THE SOFTWARE T/T POINTER
2750 003216 012702 001000 MOV #TTSIZE,R2 ;SET UP CLEAR CONSTANT (WORD COUNTER)
2751 003222 005021 20$: CLR (R1)+ ;CLEAR T/T AND DUP T/T
2752 003224 005302 DEC R2 ;ARE WE DONE?
2753 003226 001375 BNE 20$ ;NO, KEEP ON CLEARING
2754 003230 012737 000001 013076 MOV #1,MDEV ;INIT THE DEVICE NUMBER FOR MUX
2755 ; AND SEL EXECUTOR ROUTINES
2756 003236 053777 013112 007224 BIS XADDR,DXCS ;SET UP THE EXTENDED ADDRESS BITS
2757 003244 052777 004000 007216 BIS #BSYEN,DXCS ;SEL CHANNEL - SET BUSY ENABLE
2758 003252 052777 001100 007210 BIS #DXENB,DXONLN,DXCS ;ENABLE THE DX
2759 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:  MOV B (R2), R4      ;GET THE TYPE OF STOP INDICATED
        CMP B R4, #CR    ;IMMEDIATELY?
        BEQ STOPDX      ;YES, DISABLE DX AND PRINT REGISTERS
        CMP B R4, #D     ;D = AFTER NEXT DATA TRANSFER?
        BEQ 10$          ;YES, SET STOP FLAG
        CMP B R4, #E     ;E = AFTER THE NEXT ENDING SEQUENCE
        BEQ 10$          ;YES, SET STOP FLAG
        CMP B R4, #I     ;I = AFTER THE CHIS SEQUENCE
        BEQ 10$          ;YES, SET STOP FLAG
        CMP B R4, #P     ;P = STOP ON PARITY ERROR??
        BEQ 10$          ;YES, SET STOP FLAG
        JMP CERR         ;ILLEGAL FORMAT -- GIVE ERROR
10$:   MOV B 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 STPMES
                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

```

G05

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
DZDXIB.P11 30-JAN-78 14:19MACY11 30A(1052) 01-FEB-78 09:46 PAGE 58
BACKGROUND -- STOP COMMANDSEC 0058
SEC 0058

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

105:	MOV	DXDS.R1
	MOV	2(R1)+R2
	JSR	PC.OCTOMP
	DEC	R3
	BNE	105
	CLAB	DXABFL
	CLR	DXCLS
	INC	DXYCS
	JMP	EXEC

: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 EXEC

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
2880003432 004737 005050
003436 112204
003440 012705 004636
003444 120427 000101
003450 001421
003452 012705 004572
003456 120427 000105
003462 001414
003464 012705 004754
003470 120427 000110
003474 001407
003476 012705 004670
003502 120427 000117
003506 001402
003510 000137 003026
003514 010537 013124
003520 005700
003522 001043
003524 112204
003526 120427 000054
003532 001014
003534 004737 005302

003540 005004
003542 005303
003544 001403
003546 062704 002000
003552 000773
003554 060437 013120
003560 060437 013122
003564 017702 007330
003570 004777 007330
003574 105737 013071
003600 001375

.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 ; ILLEGAL 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 HEX

 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\$

MAINDEC-11-D2DNI-B NEW DNI-B RESPONDER
 D2DNI-B.P11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 60
 BACKGROUND -- DUMP COMMAND

3ES 0060
 22 0060

```

2881 003602 062737 000002 013120      ADD      #2,SADDR      : INCR TO NEXT WORD
2882 003610 023737 013120 013122      CMP      SADDR,EADDR   : DUMP DONE
2883 003616 003003              BGT      70$             : YES, EXIT
2884 003620 105737 013054              TSTB     TCMDAB        : COMMAND ABORT?
2885 003624 001757              BEQ      50$             : NO, PRINT NEXT WORD
2886 003626 000137 002720      70$:    JMP      EXEC        : YES, RETURN TO EXEC
2887
2888
2889
2890
2891      :
2892      :
2893      : DUMP THE DUPLICATE TUMBLE TABLE IN REVERSE SEQUENCE
2894 003632 012700 000400      5$:    C*UMTB: MOV      #TTSIZE/2,R0      : SET UP COUNTER TO DUMP ENTIRE TUMBLE TABLE
2895 003636 017702 007256      5$:    MOV      @SADDR,R2      : GET STARTING ADDRESS
2896 003642 004777 007256              JSR      PC,@DMPADR      : PRINT THE CONTENTS
2897 003646 105737 013071      10$:   TSTB     PC+R          : IS PRINT COMPLETE?
2898 003652 001375              BNE      10$             : NO, WAIT TILL DONE
2899 003654 032737 000777 013120      BIT      #TTSIZE-1,SADDR : CHECK FOR WRAP AROUND
2900 003662 001003              BNE      20$             :
2901 003664 062737 001000 013120      ADD      #TTSIZE,SADDR   : WRAP AROUND TO TOP OF TABLE
2902 003672 162737 000002 013120      20$:   SUB      #2,SADDR      : DECREMENT TO NEXT ENTRY
2903 003700 005300              DEC      R0              : HAS ENTIRE TUMBLE TABLE BEEN DUMPED?
2904 003702 001403              BEQ      30$             : YES, EXIT TO THE EXEC
2905 003704 105737 013054              TSTB     TCMDAB        : ARE WE TO ABORT?
2906 003710 001752              BEQ      5$              : NO, KEEP ON DUMPING
2907 003712 000137 002720      30$:    JMP      EXEC        : YES, BACK TO THE EXEC

```

```

2907 .SBTTL BACKGROUND -- FILL COMMAND
2908
2909 FILL COMMAND
2910
2911 THE FILL COMMAND LOADS THE SPECIFIED BYTE
2912 INTO THE GIVEN DATA AREA.
2913
2914 THE FOLLOWING SYNTAXES ARE AVAILABLE FOR THE FILL COMMAND:
2915     FIN,XX,YY      FILL INPUT BUFFER FOR DEVICE YY WITH XX
2916     FOT,XX,YY      FILL OUTPUT BUFFER FOR DEVICE YY WITH XX
2917
2918 WHERE: XX IS THE FILL CHARACTER IN HEX
2919        YY IS THE DEVICE ADDRESS IN HEX
2920
2921 FILL: JSR    PC, GLIMIT      ;GET BUFFER LIMITS
2922       JSR    PC, CHTB       ;GET THE FILL CHARACTER
2923       MOVSB  R3, FILLCH     ;SAVE FILL CHAR
2924       JSR    PC, GDEV       ;GET THE DEVICE ADDRESS
2925
2926 COMPUTE RELOCATION FOR DEVICE
2927
2928 10$: CLR    R4
2929      DEC    R3              ;DONE?
2930      BEQ    20$            ;YES, ADD TO START AND END ADDR
2931      ADD    #2000, R4
2932      BR     10$
2933 20$: ADD    R4, SADDR       ;ADD RELOC CONST TO START
2934      ADD    R4, EADDR       ;ADD RELOC CONST TO END ADDR
2935      MOV    SADDR, R1
2936
2937 FILL BUFFER WITH SPECIFIED CHARACTER
2938
2939 30$: MOVSB  FILLCH, (R1)+   ;FILL CHARACTER
2940      CMP    R1, EADDR       ;DONE?
2941      BLOS   30$             ;NOPE, FILL NEXT CHAR
2942      JMP    EXEC            ;DONE, RETURN TO EXEC

```


K05

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 62
BACKGROUND -- HELP COMMAND

SEQ 0062
EE 0062

```

2943 .SBTTL BACKGROUND -- HELP COMMAND
2944
2945 THE HELP COMMAND PROVIDES THE OPERATOR WITH A SYNOPSIS OF
2946 COMMANDS WHICH MAY BE USED FOR OPERATING THIS SYSTEM.
2947
2948 THE SYNTAX FOR THE HELP COMMAND IS:
2949 H
2950
2951 004004 012701 014300 HELP: MOV #HELPM5,R1 ;SET UP ADDRESS OF HELP MESSAGE
2952 004010 012702 002551 MOV #HELPLN,R2 ;LENGTH OF HELP MESSAGE
2953
2954 START OUTPUTTING THE HELP MESSAGE UNDER OUR CONTROL
2955 SO THE COMMAND MAY BE ABORTED QUICKLY.
2956
2957 004014 112100 10$: MOVB (R1)+,R0 ;GET BYTE TO OUTPUT
2958 004016 123727 013071 000004 15$: CMPB PCTR,R4 ;MORE THEN FOUR CHARACTER IN OUTPUT BUFFER??
2959 004024 003374 BGT 15$ ; YES, WAIT TIL DOWN A LITTLE
2960 004026 004737 011366 JSR PC,PCHAR ;PRINT IT ON CONSOLE
2961 004032 105737 013054 TSTB TCMDAB ;HAS OPERATOR INDICATED A DESIRE TO STOP?
2962 004036 001002 BNE 20$ ; YES, ABORT HELP MESSAGE
2963 004040 005302 DEC R2 ;HAS ENTIRE MESSAGE BEEN OUTPUTTED??
2964 004042 001364 BNE 10$ ; NO, OUTPUT ANOTHER BYTE
2965 004044 000137 002720 20$: JMP EXEC ; YES, RETURN TO THE EXECUTIVE

```

```

2966 .SBTTL BACKGROUND -- ACCESS COMMAND
2967
2968 ACCESS SPECIFIED LOCATIONS AND CHANGE IF DESIRED
2969
2970 THE ACCESS COMMAND IS A QUICK LOOK AND CHANGE
2971 ROUTINE MAINLY USED FOR PROGRAM DEBUGGING.
2972
2973 BASICALLY THE FOLLOWING ACTIONS ARE PERMITTED:
2974 AXXXXX -- OPEN AND PRINT SPECIFIED OCTAL LOCATION
2975 [XXXXXX](C/R) -- CHANGE CURRENT LOCATION IF DATA
2976 SPECIFIED [XXXXXX] AND OPEN NEXT LOCATION
2977 -- RETURN TO EXEC MODE
2978
2979 ACCESS: JSR PC,COTB ;GET THE START ADDRESS
2980 TST R5 ;WAS A VALID ADDRESS ENTERED?
2981 BEQ 5$ ;NO, GIVE OPERATOR AN ERROR
2982 BIT #1,R3 ;WAS ADDRESS SPECIFIED A WORD ADDRESS?
2983 BEQ 7$ ;YES, OPEN SPECIFIED LOCATION
2984 JMP CERR ;NO, GIVE OPERATOR AN ERROR INDICATION
2985 5$: MOV R3,SADDR ;SAVE STARTING ADDRESS
2986 7$: MOV SADDR,R2 ;GET OBJECT WORD
2987 10$: MOV SADDR,R2 ;CLEAR TELE ACTIVE FLAG
2988 CLRB TCMAC ;PRINT ADDRESS IN OCTAL
2989 JSR PC,OCTDMP ;PRINT 2 SPACES
2990 MOV #",R2
2991 JSR PC,PRINT2
2992 MOV JSADDR,R2 ;GET CONTENTS OF OBJECT LOCATION
2993 JSR PC,ODMP ;PRINT CONTENTS IN OCTAL
2994 MOV #",R2 ;PRINT 2 SPACES
2995 JSR PC,PRINT2
2996 20$: TSTB TCMAC ;ACTIVE COMMAND?
2997 BEQ 20$ ;NO
2998 MOV #TBUF,R2 ;SET UP INPUT BUFFER ADDRESS
2999 JSR PC,COTB ;WAS LOCATION CHANGED?
3000 TST R5 ;ANY CHANGE?
3001 BNE 30$ ;YES, STORE IT
3002 CMPB R4,#' / ;EXIT TO EXEC
3003 BEQ 50$ ;YES, RETURN TO EXEC
3004 CMPB R4,#CR ;CR, GO TO NEXT LOCATION?
3005 BEQ 40$ ;YES, OPEN AND PRINT NEXT LOC.
3006 BR 10$ ;ERROR, PRINT CONTENTS OF CURRENT LOC.
3007 30$: MOV R3,JSADDR ;CHANGE OPEN LOCATION
3008 40$: ADD #2,SADDR ;OPEN NEXT LOCATION
3009 BR 10$
3009 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

N05

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 65
BACKGROUND -- KILL DEVICE COMMAND

SEQ 0065
SEQ 0065

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
        MOV      #1,SONLF(DTAB)
        MOV      DEV,R5      ;COMPUTE THE ADDRESS OF THE SPW TABLE
        ADD      DEVCON,R5   ;COMPENSATE 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 .SBTTL BACKGROUND -- INPUT DISPLAY DATA COMMAND
3056
3057 I = INPUT
3058
3059 THE INPUT COMMAND IS USED TO ENTER DATA ONTO A 2260
3060 SCREEN AND THEN SEND IT TO THE 360 VIA THE READ MANUAL INPUT
3061 COMMANDS
3062
3063 THE INPUT COMMAND HAS THE FOLLOWING SYNTAX:
3064 IXX,DDD....DDD -- SEND DATA DDD TO DEVICE XX
3065 THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX
3066
3067 004330 105737 012532 INPUT: TSTB TSTYP ;ILLEGAL ON FRIEND TEST
3068 004334 001005 BNE 1C$ ;FRIEND -- GIVE AN ERROR
3069 004336 004737 005302 JSR PC,GDEV ;GET THE DEVICE NUMBER
3070 004342 120427 000054 CMPB R4,#' ;THE NEXT CHAR MUST BE A COMMA
3071 004346 001402 BEQ 20$ ;IT IS, CONTINUE
3072 004350 000137 003026 10$: JMP CERR ;AN ERROR WAS FOUND GIVE INDICATION
3073 004354 004737 010230 20$: JSR PC,SUDEV ;SET UP THE DEVICE STATUS TABLE PCINTERS
3074 004360 026327 000004 000734 CMP SCURS(DTAB),#DISP$Z-4 ;ARE WE AT THE END OF THE BUFFER?
3075 004366 002370 BGE 10$ ;YES, GIVE AN ERROR
3076 004370 032763 000001 000004 BIT #1,SCURS(DTAB) ;START INPUT ON EVEN BYTE ADDRESS
3077 004376 001002 BNE 30$ ;START SOM ON ODD BYTE ADDRESS
3078 004400 005263 000004 INC SCURS(DTAB) ;INCR CURSOR TO ODD BYTE ADDRESS
3079 004404 016305 000010 30$: MOV SOUTB(DTAB),R5 ;COMPUTE STARTING ADDRESS
3080 004410 066305 000004 ADD SCURS(DTAB),R5
3081 004414 112725 000112 MOVB #SMI,(R5)+ ;START CHARACTER TO BUFFER
3082 004420 010563 000020 MOV R5,SHINS(DTAB) ;SAVE START OF DATA LOCATION
3083 004424 005263 000004 40$: INC SCURS(DTAB) ;INCREMENT CURSOR POSITION
3084 004430 026327 000004 000735 CMP SCURS(DTAB),#DISP$Z-3 ;ARE WE AT THE END OF BUFFER
3085 004436 001423 BEQ 70$ ;YES, TERMINATE INPUT
3086 004440 112204 MOVB (R2)+,R4 ;GET NEXT INPUTTED CHARACTER
3087 004442 042704 177600 BIC #177600,R4 ;SAVE L.O. 7 BITS
3088 004446 020427 000015 CMP R4,#CR ;END OF INPUT?
3089 004452 001415 BEQ 70$ ;YES, SET UP TO EXIT
3090 004454 020427 000040 CMP R4,#SPACE ;CAN CHARACTER BE CONVERTED?
3091 004460 002410 BLT 60$ ;NO, MUST BE BETWEEN 40 - 137
3092 004462 020427 000137 CMP R4,#'+ ;NO, MUST BE BETWEEN 40 - 137
3093 004466 003005 BGT 60$ ;SCALE DOWN FOR INDEXING
3094 004470 162704 000040 SUB #SPACE,R4
3095 004474 116425 012354 50$: MOVB ATOETB(R4),(R5)+ ;CONVERT CHARACTER AND MOVE TO DISPLAY BUFFER
3096 004500 000751 BR 40$ ;GET AND CONVERT NEXT CHARACTER
3097 004502 005004 60$: CLR R4 ;ILLEGAL CHARACTER -- TREAT AS SPACE
3098 004504 000773 BR 50$
3099
3100 SET UP TO EXIT
3101 SET EOM INDICATOR
3102 QUEUE READ MANUAL INPUT REQUEST
3103
3104 004506 112715 000152 70$: MOVB #EOM,(R5) ;SET EOM INDICATOR
3105 004512 005263 000004 INC SCURS(DTAB) ;INCREMENT CURSOR POINTER
3106 004516 105263 000017 INCB SDRQ(DTAB) ;QUEUE READ REQUEST
3107
3108 SEE IF THE DX IS CURRENTLY ACTIVE
3109
3110 004522 105737 013070 TSTB DXACT ;IS D+ ACTIVE?

```

C06

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
CZDXIB.P11 30-JAN-78 14:19MACY11 30A(1052) 01-FEB-78 09:46 PAGE 67
BACKGROUND -- INPUT DISPLAY DATA COMMANDSEG 0067
SEG 0067

```

3111 004526 001402      BEQ      B0$      ;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 B0$: 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 .SBTTL BACKGROUND SUBROUTINES -- PRINT FORMATTING
3128
3129 DUMP WORD IN EBCDIC ON TTY
3130
3131 CALLING SEQUENCE
3132 .....R2 CONTAINS WORD TO BE PRINTED
3133 JSR PC,EBCDMP
3134 .....RETURN
3135
3136
3137 REGISTERS 2 + 3 ARE DESTROYED BY THIS SUBROUTINE
3138
3139 EBCDMP: MOV R2,WK ;SAVE WORD TO BE PRINTED
3140         MOVB WK,R3 ;GET LO BYTE
3141         BIC #177400,R3
3142         MOVB EBCDTB(R3),WK ;CONVERT EBCDIC TO ASCII
3143         MOVB WK1,R3 ;GET HI BYTE AND CONVERT
3144         BIC #177400,R3
3145         MOVB EBCDTB(R3),WK1 ;CONVERT CHAR TO ASCII
3146         MOV WK,R2
3147         FALL THROUGH TO ASCII PRINT ROUTINE
3148
3149
3150
3151 DUMP WORD IN ASCII ON TTY
3152
3153 CALLING SEQUENCE
3154 .....R2 CONTAINS WORD TO BE PRINTED
3155 JSR PC,ASCDMP
3156 .....RETURN
3157
3158
3159 NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
3160
3161 ASCDMP: TSTB LINECT ;NEW LINE?
3162         BNE 10$ ;NO
3163         JSR PC,CRLF ;YES, PRINT CR/LF
3164         MOVB #30,LINECT ;60 CHARACTERS PER LINE
3165         DECB LINECT ;DECR LINE COUNTER
3166         JSR PC,PRINT2 ;PRINT 2 CHARS
3167         RTS PC ;RETURN TO CALLER
3168
3169
3170
3171 DUMP WORD IN OCTAL ON TTY
3172
3173 CALLING SEQUENCE
3174 .....R2 CONTAINS WORD TO BE PRINTED
3175 JSR PC,OCTDMP OR ODMP
3176 .....RETURN
3177
3178 OCTDMP PERFORMS A CR/LF BEFORE PRINTING OCTAL DATA
3179
3180 NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
3181
3182 OCTDMP: JSR PC,CRLF ;GIVE A CRLF

```

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

E06

MAINDEC-11-DZDI-B NEW DX11-B RESPONDER
DZDI-B.P11 30-JAN-78 14:19MACY11 30A(1052) 01-FEB-78 09:46 PAGE 69
BACKGROUND SUBROUTINES -- PRINT FORMATTINGSEG 0069
EE% 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  R6,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$:  ROL  R0      ;GET THE REMAINING BIT STILL IN LTR
3190 004714 042700 177770      BIC  R0,R0,R0      ;ONLY 3 LOW ORDER BITS
3191 004720 062700 000060      ADD  R0,R0      ;MAKE ASCII
3192 004724 004737 011366      JSR  PC,PCHAR      ;PRINT IT ON THE TTY
3193 004730 006102      ROL  R2      ;ROTATE THE NEXT OCTAL CHAR INTO POSITION
3194 004732 006102      ROL  R2
3195 004734 006102      ROL  R2
3196 004736 010200      MOV  R2,R0      ;DATA TO WORKING REGISTER
3197 004740 005304      DEC  R4      ;ARE WE DONE?
3198 004742 001363      BNE  10$      ;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  R6,R4      ;4 CHARACTERS PER WORD
3222 004772 006102      10$:  ROL  R2      ;ROTATE HIGH ORDER 4 BITS TO LOW ORDER 4 BITS
3223 004774 006102      ROL  R2
3224 004776 006102      ROL  R2
3225 005000 006102      ROL  R2
3226 005002 010200      MOV  R2,R0      ;TO WORKING REG
3227 005004 006100      ROL  R0      ;GET THE LINK BIT TOO
3228 005006 042700 177760      BIC  R0,R0,R0      ;ONLY LOW ORDER 4 BITS
3229 005012 062700 000060      ADD  R0,R0      ;MAKE ASCII IF NUMBER
3230 005016 020027 000071      CMP  R0,R0      ;SHOULD IT BE A-F?
3231 005022 003402      BLE  20$      ;NO, SKIP IT
3232 005024 062700 000007      ADD  R0,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

```


F06

MAINDEC-11-DZDI-B NEW D:11-B RESPONDER
CZDI1B.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 001614
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 GLMOT ; PREPARE TO EXIT

```

```

SET UP LIMITS OF TUMBLE TABLE

```

```

GLMTT: MOVB (R2)+, R4
CMPB R4, #'T ; MUST BE TT
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
3305 005172 112204      GLMIN:  MOVB    (R2)+,R4
3306 005174 120427 000116      CMPB    R4,#'N ;MUST BE IN
3307 005200 001341      BNE     GLERR ;ILLEGAL ENTRY
3308 005202 013704 013064      MOV     SDEVTB,R4 ;GET ADDR OF DEV 0 STATUS TABLE
3309 005206 016437 000006 013120      MOV     SINBF(R4),SADDR
3310 005214 013737 013120 013122      MOV     SADDR,EADDR
3311 005222 062737 000741 013122      ADD     #DISPSZ+1,EADDR ;DISPLAY SIZE + ROOM FOR LINE ADDRESS
3312 005230 000417      BR       GLEX1
3313
3314
3315
3316
3317
3318 005232 112204      GLMOT:  MOVB    (R2)+,R4
3319 005234 120427 000124      CMPB    R4,#'T ;MUST BE OT
3320 005240 001321      BNE     GLERR ;ILLEGAL ENTRY
3321 005242 013704 013064      MOV     SDEVTB,R4 ;GET ADDR OF DEV 0 STATUS TABLE
3322 005245 016437 000010 013120      MOV     SOUTB(R4),SADDR ;COMPUTE STARTING AND ENDING ADDRESSES OF SPECIFIED BUF
3323 005254 013737 013120 013122      MOV     SADDR,EADDR
3324 005262 062737 000737 013122      ADD     #DISPSZ-1,EADDR ;DISPLAY CHAR BUFFER
3325 005270 112204      GLE>1:  MOVB    (R2)+,R4 ;GET NEXT INPUT CHARACTER AND UPDATE POINTER
3326
3327
3328
3329
3330 005272 120427 000054      GLE>:  CMPB    R4,#' ' ;CHECK FOR
3331 005276 001302      BNE     GLERR ;ENTRY NOT PROPERLY DELIMITED (ERROR)
3332 005300 000207      RTS     PC

```

```

3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345 005302 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

Register	Designation	Description
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 000200 005062 10$: BIC #DONE, 2DXCS ;CLEAR DONE
3422 005406 010203 ;SET UP PTR TO DUP T/T
3423 005410 062703 001000 ADD #TTSIZE, R3
3424 005414 011223 MOV (R2), (R3)+ ;SAVE T/T ENTRY #1
3425 005416 011204 MOV (R2), TT1
3426 005420 005022 CLR (R2)+ ;CLEAR T/T ENTRY #1
3427 005422 011223 MOV (R2), (R3)+ ;SAVE T/T ENTRY #2
3428 005424 011205 MOV (R2), TT2
3429 005426 005022 CLR (R2)+ ;CLEAR T/T ENTRY #2
3430
3431
3432
3433 005430 032702 000777
3434
3435 005434 001002
3436 005436 013702 013062
3437 005442 010237 013060 20$: MOV TTADDR, R2
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 RELEIVE 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.

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)

CHECK FOR POINTER WRAP AROUND

AT END OF BUFFER?
NOTE -- POWER OF 2 BOUNDARY
NO
YES, RESET PTR
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 120037 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 005200
3473 005524 004737 010230

```

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

```

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.

3474				:			
3475				:			
3476				:			
3477				:			
3478				:			
3479				:			
3480				:			
3481				:			
3482	005530	032704	020000	SELST: BIT	#SELST, TT1	: SELECTIVE RESET?	
3483	005534	001402		BEQ	TINDSC	: NO, TEST INTERFACE DISC.	
3484	005536	000137	005776	JMP	PSELRT	: YES, PERFORM SELECTIVE RESET	
3485	005542	032704	004000	TINDSC: BIT	#INDSC, TT1	: INTERFACE DISCONNECT?	
3486	005546	001402		BEQ	TNXM	: NO, CHECK NON-EXISTENT MEMOR	
3487	005550	000137	006026	JMP	PINDSC	: YES, PERFORM INTER DISC	
3488	005554	032704	040000	TNXM: BIT	#NXM, TT1	: NON-EXISTENT MEMORY ERROR?	
3489	005560	001402		BEQ	TESEND	: NO, ES END	
3490	005562	000137	006102	JMP	PNXM	: YES, PROCESS NON-EXISTENT MEMOR, ERROR	
3491	005566	032704	000100	TESEND: BIT	#ESEND, TT1	: WAS STATUS ACCEPTED?	
3492	005572	001402		BEQ	TPARER	: NO, CHECK FOR PARITY ERROR	
3493	005574	000137	006114	JMP	PESEND	: YES, PERFORM STATUS ACCEPT	
3494	005600	032704	100000	TPARER: BIT	#PARER, TT1	: DID A PARITY ERROR OCCUR?	
3495	005604	001402		BEQ	TCHIS	: NO, CHECK FOR CHIS	
3496	005606	000137	006244	JMP	PPARER	: YES, PROCESS PARITY ERROR	
3497	005612	032704	000200	TCHIS: BIT	#CHIS, TT1	: DID CHANNEL START A SELECTION SEQ?	
3498	005616	001402		BEQ	TCHEND	: NO, CHECK FOR CHANNEL DATA END	
3499	005620	000137	006274	JMP	PCHIS	: YES, PROCESS SELECTION SEQUENCE	
3500	005624	032704	000040	TCHEND: BIT	#CHEND, TT1	: DID CHANNEL END OCCUR?	
3501	005630	001402		BEQ	TCUEND	: NO, CHECK FOR CONTROL UNIT END	
3502	005632	000137	006456	JMP	PCHEND	: YES, PROCESS CHANNEL END	
3503	005636	032704	000020	TCUEND: BIT	#CUEND, TT1	: DID A CONTROL UNIT END OCCUR?	
3504	005642	001650		BEQ	LOOP	: NO, IGNORE ENTRY -- ASSUME STACK STATUS	
3505	005644	000137	006442	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

```

3513 005650 012700 000001 PSYSRT: MOV #1,DEV ;START AT FIRST DEVICE
3514 005654 004737 010256 10$: JSR PC,CDEVST ;CLEAR DEVICE STATUS TABLE
3515 005660 004737 010310 JSR PC,CSPWST ;RESET SPW STATUS WORD UPON SYSTEM RESET
3516 005664 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
3517 005670 105063 000017 CLRB SRDRQ(DTAB) ;CLEAR THE READ REQUEST
3518 005674 005063 000020 CLR SMINS(DTAB) ;CLEAR THE BEG OF MANUAL INPUT ADDRESS
3519 005700 005063 000004 CLR SCURS(DTAB) ;RESET THE CURSOR
3520 005704 105063 000003 CLRB SSTAT(DTAB) ;CLEAR THE STATUS REGISTER
3521 005710 016301 000010 MOV SOUTB(DTAB),R1 ;SET UP TO CLEAR THE DISPLAY BUFFER
3522 005714 012702 000740 MOV #DISPS2,R2 ;SET UP NUMBER OF CHARACTERS IN DISPLAY
3523 005720 112704 000100 MOV #EBCDSP,R4 ;ASSUME 2848 DIAGNOSTIC TEST MODE
3524 005724 105737 012532 TSTB TSTYP ;WHAT TYPE OF TEST?
3525 005730 001402 BEQ 20$ ; IF 2848, USE EBCDIC SPACE
3526 005732 113704 012534 MOV #FILLCH,R4 ; FRIEND TEST -- USE CURRENT FILL CHARACTER
3527
3528 005736 110421 20$: MOV R4,(R1)+ ;USE THE FILL CHARACTER
3529 005740 005302 DEC R2 ;ARE WE DONE?
3530 005742 001375 BNE 20$ ;NO, LOOP TILL DONE
3531 005744 005200 INC DEV ;TO NEXT DEVICE
3532 005746 120037 013067 CMPB DEV,MAXDEV ;ARE WE DONE?
3533 005752 003740 BLE 10$ ;NO, CLEAR NEXT DEV STAT TABLE
3534 005754 105037 013070 CLRB DXACT ;CLEAR DX ACTIVE FLAG
3535 005760 105037 013074 CLRB CMDCHF ;CLEAR COMMAND CHAINING FLAG
3536 005764 042777 000400 BIC #CUBUSY,DXCS ;RESET CU BUSY FLAG
3537 005772 000137 005364 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

```

3547 005776 004737 010256 PSELRT: JSR PC,CDEVST ;CLEAR DEVICE STATUS TABLE
3548 006002 004737 010310 JSR PC,CSPWST ;RESET SPW STATUS RESPONSE
3549 006006 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
3550 006012 105037 013070 CLRB DXACT ;CLEAR DX ACTIVE FLAG
3551 006016 105037 013074 CLRB CMDCHF ;CLEAR COMMAND CHAIN FLAG
3552 006022 000137 005364 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      :
3566      006032 105763 000000      :
3567      006036 001417      :
3568      006040 004737 010256      :
3569      006044 012763 000003 000000 :
3570      006052 120037 013070      :
3571      006056 001002      :
3572      006060 105037 013070      :
3573      006064 120037 013074      :
3574      006070 001002      :
3575      006072 105037 013074      :
3576      006076 000137 005364      :
3577      :
3578      :
3579      :
3580      :
3581      :
3582      :
3583      006102 004137 011506      :
3584      006106 013706      :
3585      006110 000137 006536      :
3586      :
3587      :
3588      :
3589      :
3590      :
3591      :
3592      :
3593      :
3594      006114 004737 010310      :
3595      006120 105763 000001      :
3596      006124 001402      :
3597      006126 004737 010402      :
3598      006132 126327 000000 000011 :
3599      006140 001003      :
3600      006142 112763 000002 000017 :
3601      006150 105063 000001      :
3602      006154 105037 013070      :
3603      006160 004737 010256      :
3604      006164 032704 000004      :
3605      006170 001402      :
3606      006172 110037 013074      :
3607      006176 123727 013066 000105 :
3608      006204 001552      :
3609      :
3610      006206 032704 000010      :
3611      006212 001412      :
3612      :
3613      :
3614      :
3615      :
3616      :
3617      :

      AND CHAN END ! DEVICE END SET IN STATUS B/TE
      IF THE DEVICE IS NOT ACTIVE THE COMMAND WILL BE IGNORED

PINDSC: JSR    PC,CSPWST      ;CLEAR THE SPW STATUS RESPONSE
        TSTB   SCMD(DTAB)    ;IS DEVICE ACTIVE?
        BEQ    20$           ;NO, IGNORE
        JSR    PC,CDEVST     ;CLEAR THE DEVICE STATUS TABLE
        MOV    #CEDE,SCMD(DTAB);QUE DEV END + CHAN END
        CMPB   DEV,DXACT     ;IS DEVICE USING DX NOW?
        BNE    10$           ;NO
        CLRB   DXACT         ;YES, RELEASE DX
        CMPB   DEV,CMDCHF    ;DOES DEVICE HAVE CMD CHAIN SPEC?
        BNE    20$           ;NO, GET NEXT T/T ENTRY
        CLRB   CMDCHF        ;YES, CLEAR FLAG
        JMP    LOOP          ;GET NEXT T/T ENTRY

      A NON-EXISTANT MEMORY CONDITION OCCURRED
      THIS WILL USUALLY TRAP OUT FIRST

PNXM:   JSR    R1,INMES      ;PRINT "NON EX MEM"
        .WORD  NXMMMSG
        JMP    DXAB         ;ABORT DX AND RETURN TO EXEC

      THE LAST STATUS SENT TO THE 360 WAS ACCEPTED, CLEAR DX
      ACTIVE FLAG
      IF LAST OP WAS A WRITE PERFORM THE DISPLAY CONTROL ROUTINE

PESEND: JSR    PC,CSPWST      ;RESET THE SPW STATUS BYTE
        TSTB   SLCMD(DTAB)   ;DOES LAST COMMAND REQUIRE 2260 DISPLAY EMULATION?
        BEQ    10$           ;NO
        JSR    PC,DISCTL     ;YES, FORMAT THE DISPLAY
        CMPB   SCMD(DTAB),#11 ;WAS ATTN ACCEPTED?
        BNE    20$           ;NO, CONTINUE
        MOVB   #2,SRDRQ(DTAB);YES, INDICATE 360 ACCEPTANCE
        CLRB   SLCMD(DTAB)
        CLRB   DXACT         ;CLEAR DX ACTIVE FLAG
        JSR    PC,CDEVST     ;CLEAR THE DEVICE STATUS TABLE
        BIT    #CMDCHN,TT1   ;WAS COMMAND CHAINING SPECIFIED?
        BEQ    30$           ;NO
        MOVB   DEV,CMDCHF    ;YES, SAVE THE DEVICE NUMBER
        CMPB   DXSTPF,#'E    ;WAS STOP ON END SEQ SPEC?(SE)
        BEQ    STPDX         ;YES, DISABLE THE DX

        BIT    #ISSREJ,TT1   ;WAS AN ISS REJ DETECTED?
        BEQ    50$           ;NO, EXIT

      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

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.

```

3618
3619
3620
3621
3622
3623
3624
3625
3626 006214 105737 012532      TSTB      TSTYP      :PROCESS SEPARATELY IF FRIEND
3627 006220 001004      BNE      40$      :FRIEND --QUEUE CU END ON BUSY UNIT
3628 006222 012700 000001      MOV      #1,DEV      :SET UP TO SEND CUE ON LOW ORDER CONTROLLER ADDR
3629 006226 004737 010230      JSR      PC,SUDEV      :X
3630 006232 112763 000010 000000 40$:      MOV      #OCUE,SCMD(DTAB) :QUEUE CONTROL UNIT END
3631 006240 000137 005364      JMP      LOOP      :LOOP BACK AND PROCESS NEXT TUMBLE TABLE ENTRY
3632
3633
3634
3635
3636

```

PARITY ERROR WAS DETECTED

```

3637 006244 004137 011506      PPARER: JSR      R1,INMES      :PRINT "PARITY ERROR"
3638 006250 013736      .WORD      PARMES
3639 006252 123727 013066 000120      CMPB      DXSTPF,#'P      :STOP ON PARITY ERROR??
3640 006260 001524      BEQ      STPOX      :YES, DISABLE THE DX
3641 006262 152763 000002 000003      BISB      #UCHK,SSTAT(DTAB) :SET UNIT CHECK IN STATUS WORD
3642 006270 000137 005612      JMP      TCHIS      :CONTINUE WITH TUMBLE TABLE INTERROGATION
3643
3644
3645
3646
3647
3648
3649
3650
3651

```

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

```

3652 006274 004737 010310      PCHIS: JSR      PC,CSPWST      :RESET THE SPW STATUS BYTE
3653                                :ON NEXT CHANNEL INITIATED SELECTION SEQUENCE
3654 006300 123727 013066 000111      CMPB      DXSTPF,#'I      :WAS STOP ON ISS SPECIFIED?
3655 006306 001511      BEQ      STPOX      :YES, DISABLE DX
3656 006310 032704 000001      BIT      #CMOREJ,TT1      :WAS COMMAND REJECTED BY DX?
3657 006314 001022      SNE      20$      :YES, COMMAND REJECTED BY THE DX
3658
3659
3660

```

VALID COMMAND, SET UP TO PROCESS IT

```

3661 006316 105005      CLR      TT2      :RESET DEVICE ADDRESS BITS
3662 006320 000305      SWAB      TT2      :COMMAND TO L.O. BYTE
3663 006322 105705      TST      TT2      :TEST I/O COMMAND?
3664 006324 001437      BEQ      50$      :YES, IGNORE
3665 006326 120527 000003      CMPB      TT2,#NOP      :WAS COMMAND A NOP?
3666 006332 001434      BEQ      50$      :YES, IGNORE IT
3667 006334 020527 000012      CMP      TT2,#12      :IS THIS A VALID COMMAND?
3668 006340 003405      BLE      10$      :YES, QUEUE TO BE EXECUTED
3669 006342 004137 011506      JSR      R1,INMES      :NO -- REPORT AN ILLEGAL COMMAND RECIEVED FROM THE DX
3670 006346 014050      .WORD      INVLDC
3671 006350 000137 006536      JMP      DXAB      :AND ABORT THE PROGRAM
3672 006354 110563 000000 10$:      MOV      TT2,SCMD(DTAB) :QUEUE COMMAND TO BE PROCESSED
3673 006360 000421      BR      50$      :EXIT + PROCESS NEXT T/T ENTRY

```



```

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, TT1 ; 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 D/T 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? SC.
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, SBYTC(DTAB) ; SAVE REMAINING BYTE COUNT
3734 006512 032704 100000 PCHEX: BIT #PARE, T11 ; WAS A PARITY ERROR SENSED?
3735 006516 001403 BEQ 10$ ; NO, PROCESS NEXT TUMBLE TABLE ENTRY
3736 006520 152763 000020 000002 BISB #EQCHK, SSENSE(DTAB) ; YES, SET EQUIP CHECK IN SENSE
3737 006526 000137 005364 10$: JMP LOOP ; LOOP BACK + PROCESS NEXT T1 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
3754 006550 000137 010212 JMP DXEXIT ; DX REGISTERS WILL BE PRINTED
3755 .SBTTL DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION) ; EXIT FROM INTERRUPT
3756
3757
3758 EXECUTE NEXT COMMAND FOR THE DX
3759
3760 006554 105737 012526 DXEXEC: TSTB CHTYPE ; CHANNEL TYPE 0=M, 1=S
3761 006560 001002 BNE SEX ; SELECTOR CHANNEL EXEC
3762 006562 000137 007210 JMP MEX ; MULTIPLEXER EXEC
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785

```

SEX -- SELECTOR CHANNEL EXECUTIVE

SEX EXECUTES COMMANDS FOR THE DX TO A SELECTOR CHANNEL

ON A SELECTOR CHANNEL A COMMAND WILL BE COMPLETED BEFORE ATTEMPTING TO EXECUTE A COMMAND ON ANOTHER DEVICE, ISS-DATA TRANSFER-ES.

DATA TRANSFERS ARE COMPLETED IN ONE BURST

```

3773 006566 013700 013076 SEX: MOV SELDEV, DEV ; GET SEL DEV #
3774 006572 004737 010230 10$: JSR PC, SUDDEV ; SET UP DEV STATUS TABLE ADDR
3775 006576 105763 000000 TSTB SCMD(DTAB) ; ANY JOB TO DO?
3776 006602 001030 BNE 60$ ; 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$: CMPB SDRQ(DTAB), #1 ; IS ATTENTION TO BE SENT?
3781 006624 001004 BNE 40$ ; 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 7367 CMPB DEV, MAXDEV ; HAVE WE TRIED THE HIGHEST DEVICE?

```

E07

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER
 C2DXIB.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	:1 = WRITE FULL BUFFER
3800	006706	007020			.WORD	:2 = READ MANUAL INPUT
3801	006710	010042			.WORD	:3 = ENDING SEQUENCE
3802	006712	010140			.WORD	:4 = SENSE COMMAND
3803	006714	006730			.WORD	:5 = WRITE LINE ADDRESS
3804	006716	007120			.WORD	:6 = READ FULL BUFFER
3805	006720	007770			.WORD	:7 = ERASE COMMAND
3806	006722	007750			.WORD	:10 = CONTROL UNIT END
3807	006724	007760			.WORD	:11 = SEND ATTENTION TO 360
3808	006726	007320			.WORD	:12 = READ SHORT MANUAL INPUT

```

3809 .....
3810 .....
3811 .....
3812 .....
3813 .....
3814 .....
3815 .....
3816 .....
3817 .....
3818 .....
3819 .....
3820 .....
3821 006730 016377 000006 003536 SWRITE: MOV SINBF(DTAB),D0XBA ;SET UP BUFFER ADDRESS
3822 006736 163777 013104 003530 SUB PHYOFF,D0XBA ;FOR VIRTUAL MEMORY -- OFFSET FOR PHYSICAL ADDRESS
3823 006744 013702 013110 MOV DEVCON,R2 ;COMPUTE DEVICE ADDRESS
3824 006750 060002 ADD DEV,R2
3825 006752 110277 003510 MOV R2,D0XCA
3826 006756 012777 003512 MOV #-DISP52-1,D0XBC ;SET UP BYTE COUNT FOR MAX. WRITE LINE ADDRESS
3827 006764 116363 000000 000001 MOVB SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
3828 006772 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
3829 006776 110037 013070 MOVB DEV,D0XACT ;SET DX ACTIVE FLAG
3830 007002 005063 000014 CLR SRBYTC(DTAB) ;RESET REMAINING BYTE COUNT
3831 007006 052777 000003 003454 BIS D0XWR,D0XCS ;START TRANSFER
3832 007014 000137 010212 JMP D0XEXIT ;RETURN FROM INTERRUPT
3833 .....
3834 .....
3835 .....
3836 .....
3837 .....
3838 .....
3839 007020 105737 012532 SRMI: TSTB TSTTYP ;IS TEST FOR FRIEND?
3840 007024 001035 SSRMI: BNE SREAD ;YES, TREAT ALL READS AS READ FULL BUFFER
3841 007026 105063 000002 CLRB SSENSE(DTAB) ;RESET THE SENSE BYTE
3842 007032 105763 000017 TSTB SRDRQ(DTAB) ;WAS A READ REQUESTED?
3843 007036 001002 BNE IOS ;YES, CONTINUE
3844 007040 000137 010042 JMP ESEQ ;NO, TREAT AS A NOP -- END SEQ ONLY
3845 007044 105063 000017 10$: CLRB SRDRQ(DTAB)
3846 007050 016377 000020 003416 MOV SMINS(DTAB),D0XBA ;SET UP STARTING ADDRESS
3847 007056 016302 000010 MOV SOUTB(DTAB),R2 ;DETERMINE ENDING ADDRESS
3848 007062 066302 000004 ADD SCURS(DTAB),R2
3849 007066 005302 DEC R2
3850 007070 167702 003400 SUB D0XBA,R2 ;COMPUTE BYTE COUNT
3851 007074 100002 010042 BPL 20$ ;INSURE VALID BYTE COUNT
3852 007076 000137 010042 JMP ESEQ ;ILLEGAL
3853 007102 005402 20$: NEG R2
3854 007104 010277 003366 MOV R2,D0XBC ;SET UP DX'S BYTE COUNT
3855 007110 163777 013104 003356 SUB PHYOFF,D0XBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
3856 007116 000411 BR SRDIO ;START THE READ
3857 .....
3858 .....
3859 .....
3860 .....
3861 .....
3862 .....
3863 007120 016377 000010 003346 SREAD: MOV SOUTB(DTAB),D0XBA ;SET UP BUFFER ADDRESS
3864 007126 163777 013104 003340 SUB PHYOFF,D0XBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS

```

MAINDEC-11-CZDXI-B NEW DX11-B RESPONDER MACY11 30A(1052) 01-FEB-78 09:46 PAGE 84
CZDXIB.F11 30-JAN-78 14:19 DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)

550 3087

ADDRESS	HEX	ASCII	OPERATION	COMMENT			
3865	007134	012777	177040	003334	MOV	8-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	8DXRD,DXCS	:START TRANSFER
3874	007204	000137	010212		JMP	DXEXIT	:RETURN FROM INTERRUPT
3875							
3876							

H07

MAINDEC-11-C2D\I-B NEW DX11-B RESPONDER
C2D\I.B.F11 30-JAN-78 14:19MACY11 30A(1052) 01-FEB-78 09:46 PAGE 85
DY11-B ISR (MULTIPLEXER CHANNEL COMMANDS)SEQ 0085
SEQ 0085

```

3877 .SBTTL DX11-B ISR (MULTIPLEXER CHANNEL COMMANDS)
3878
3879 MEX-- MULTIPLEXER CHANNEL EXECUTIVE
3880
3881 MEX EXECUTES COMMANDS FROM THE DX ON A MULTIPLEXER CHANNEL
3882
3883 ON A MULTIPLEXER MULTIPLE DEVICE REQUESTS WILL BE
3884 INTERLEAVED. THIS WILL PROHIBIT A TIME OUT TO OCCUR
3885 IF A DEVICE IS NOT SERVICED UNTIL ALL OTHER DEVICES
3886 BEFORE IT.
3887
3888 DATA TRANSFERS ARE DONE IN 4 BYTE BLOCKS, SO AS TO NOT
3889 HOG THE CHANNEL
3890
3891 007210 105737 013074 MEX: TSTB CMDCHF ; IS COMMAND CHAINING SPECIFIED?
3892 007214 001402 BEQ 10$ ; NO, CONTINUE
3893 007216 000137 010212 JMP DXEXIT ; YES, LEAVE DX FREE
3894 007222 013700 013076 10$: MOV MDEV,DEV ; GET LAST DEVICE ADDR THAT HAD A COMMAND
3895 007226 004737 010230 30$: JSR PC,SUDEV ; COMPUTE ADDR OF DEV STAT TABLE
3896 007232 105763 000000 TSTB SCMD(DTAB) ; ANY JOB TO DO?
3897 007236 001023 BNE 50$ ; YES, EXECUTE IT
3898 007240 126327 000017 000001 CMPB SRDRQ(DTAB),#1 ; IS ATTENTION REQUESTED?
3899 007246 001004 BNE 40$ ; NO, CONTINUE
3900 007250 112763 000011 000000 MOVB #11,SCMD(DTAB) ; YES, QUEUE ATTENTION
3901 007256 000413 BR 50$ ; FOR THE READ MANUAL INPUT
3902
3903 NO TASK PENDING FOR CURRENT DEVICE
3904 BUMP TO INTERROGATE NEXT DEVICE ON CONTROL UNIT
3905 THIS CODE WILL REPEAT SEQUENCES WHICH MAY HAVE RUN INTO
3906 A LOCKOUT CONDITION IN THE DX.
3907
3908 007260 005200 40$: INC DEV ; INCR TO NEXT DEVICE NUMBER
3909 007262 120037 013067 CMPB DEV,MAXDEV ; WAS DEVICE NUMBER WRAPPED AROUND?
3910 007266 003402 ELE 45$ ; NO, SEE IF ALL DEVICES HAVE BEEN INTERROGATED
3911 007270 012700 000001 MOV #1,DEV ; YES, RESET THE DEVICE NUMBER
3912 007274 020037 013076 45$: CMP DEV,MDEV ; NO JOB HERE, HAVE WE CHECKED ALL DEVICES?
3913 007300 001352 BNE 30$ ; NO, EXAMINE NEXT DEVICE
3914 007302 000137 010212 JMP DXEXIT ; YES, EXIT FROM ISR
3915
3916 THIS DEVICE HAS A JOB TO DO, EXECUTE IT
3917
3918 007306 116304 000000 50$: MOVB SCMD(DTAB),R4 ; COMMAND TO INDEX
3919 007312 005304 DEC R4 ; SCALE TO 0 - 11
3920 007314 006304 ASL R4 ; MAKE INTO WORD ADDRESS
3921 007316 010037 013076 MOV DEV,MDEV ; SAVE CURRENT DEVICE ADDRESS
3922 007322 000174 007326 JMP @MCMDBT(R4) ; EXECUTE THE COMMAND
3923 007326 007352 MCMDBT: .WORD MWRITE ; 1 = WRITE FULL BUFFER
3924 007330 007502 .WORD MRMI ; 2 = READ MANUAL INPUT
3925 007332 010042 .WORD ESEQ ; 3 = ENDING SEQUENCE
3926 007334 010140 .WORD SENSEM ; 4 = SENSE COMMAND
3927 007336 007352 .WORD MWRITE ; 5 = WRITE LINE ADDRESS
3928 007340 007600 .WORD MREAD ; 6 = READ FULL BUFFER
3929 007342 007770 .WORD ERASC ; 7 = ERASE COMMAND
3930 007344 007750 .WORD CONUNE ; 10 = CONTROL UNIT END
3931 007346 007760 .WORD SATTH ; 11 = SEND ATTENTION TO 360
3932 007350 007502 .WORD MSRMI ; 12 = READ SHORT MANUAL INPUT

```



```

3933 .....
3934
3935
3936
3937 .....
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988

```

007352	005763	000014			
007356	001011				
007360	016363	000006	000012		
007366	163763	013104	000012		
007374	012763	000741	000014		
007402	016377	000012	003064	10\$:	
007410	013702	013110			
007414	060002				
007416	110277	003044			
007422	012777	177774	003046		
007430	026327	000014	000004		
007436	002005				
007440	016302	000014			
007444	005402				
007446	010277	003024			
007452	105063	000002		20\$:	
007456	116363	000000	000001		
007464	110037	013070			
007470	152777	000003	002772		
007476	000137	010212			

```

WRITE: TST      SRBYTC(DTAB)      ;WRITE IN PROGRESS?
        BNE     10$              ;YES, SEND OUT MORE DATA
        MOV     SINBF(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
        SUB     PHYOFF,SBUFA(DTAB) ;FOR MEM MANG - OFFSET FOR PHY ADDRESS
        MOV     #DISPSZ+1,SRBYTC(DTAB) ;SET UP BUFFER FOR MAX SIZE
10$:    MOV     SBUFA(DTAB),DXBA ;OUTPUT BUFFER ADDR TO DX
        MOV     DEVCON,R2        ;COMPUTE DEVICE ADDRESS
        ADD     DEV,R2
        MOV     R2,DXCA
        MOV     #-4,DXBC        ;START BYTE COUNT AT 4
        CMP     SRBYTC(DTAB),#4 ;IS LESS THEN 4 BYTES LEFT?
        BGE     20$              ;NO, START TRANSFER
        MOV     SRBYTC(DTAB),R2 ;YES, USE REMAINING BYTE COUNT
        NEG     R2
        MOV     R2,DXBC
20$:    CLRB     SSENSE(DTAB) ;CLEAR SENSE BYTE
        MOV     SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
        MOV     DEV,DXACT        ;SET ACTIVE FLAG
        BIS     #DXWR,DXCS      ;START THE TRANSFER
        JMP     DXEXIT          ;RETURN FROM INTERRUPT

READ MANUAL INPUT COMMAND

THIS WILL TRANSFER ONLY THE DATA ENTERED ON THE SCREEN TO THE 360

MRMI =
MSRMI: TSTB     I$TYP            ;FRIEND OR 2848 DIAG?
        BNE     MREAD           ;FRIEND -- TREAT AS READ FULL BUFFER
        TST     SRBYTC(DTAB)    ;ANY DATA LEFT TO TRANSFER?
        BNE     MREAD           ;BRANCH IF YES TO CONTINUE
        CLRB     SSENSE(DTAB)    ;RESET THE SENSE BYTE
        TSTB     SRDRQ(DTAB)     ;WAS THE READ REQUESTED?
        BNE     10$             ;YES, CONTINUE
        JMP     ESEQ            ;NO, RETURN AN ENDING SEQ RESP DEICE
10$:    CLRB     SRDRQ(DTAB)     ;CLEAR THE READ REQUEST
20$:    MOV     SMINS(DTAB),SBUFA(DTAB) ;SET UP THE ADDRESS OF THE DATA
        MOV     SOUTB(DTAB),R2   ;COMPUTE THE BYTE COUNT
        ADD     SCURS(DTAB),R2   ;END - START
        DEC     R2
        SUB     SBUFA(DTAB),R2 ;COMPUTE THE BYTE COUNT
        BMI     10$             ;NEGATIVE -- SOMETHING IS WRONG
        MOV     R2,SRBYTC(DTAB) ;SAVE FOR READ DRIVER
        SUB     PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS

```

J07

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

 MACY11 30A(1052) 01-FEB-78 09:46 PAGE 87
 DX11-B ISR (MULTIPLEXER CHANNEL COMMANDS)

 SEQ 0087
 SEQ 0087

```

3989      : FALL THROUGH TO NORMAL READ BUFFER ROUTINE
3990      :
3991      :
3992      :
3993      :
3994      : READ COMMAND RECEIVED FROM 360
3995      : PREPARE 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  SSENSE(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 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  #DXRD,DXCS ;START THE DX READING
4025 007744 000137 010212      JMP  DXEXIT

```


K07

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 88
DX11-B ISR (MULTIPLEXER AND SELECTOR CHANNEL COMMANDS)

SEQ 0088
SE2 C088

```

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: BISB #CUE,SSSTAT(DTAB) ;PUT IN STATUS BYTE
4038 007756 000434 BR STOUT ;OUTPUT TO CHANNEL
4039
4040
4041
4042
4043
4044 007760 152763 000200 000003 SATTN: BISB #ATTN,SSSTAT(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 TSTTYP ;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: BISB #CE!DE,SSSTAT(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.

L07

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 89
DX11-B ISR (MULTIPLEXER AND SELECTOR CHANNEL COMMANDS)

SEQ 0089
SEQ 0089

```

4082 010050 013702 013110      STOUT:  MOV    DEVCON,R2      ;OUTPUT DEVICE ADDRESS
4083 010054 060002                ADD    DEV,R2
4084 010056 110277 002404      MOVVB   R2,DXCA
4085 010062 132763 000002 000003  BITB    #UCHK,SSTAT(DTAB) ;IS THE UNIT CHECK BIT SET?
4086 010070 001403                BEQ     10$      ;NO, TRANSMIT THE STATUS
4087 010072 112763 000002 000003  MOVVB   #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$:  MOVVB   SSTAT(DTAB),DXOS ;OUTPUT STATUS TO CHANNEL
4102 010124 152777 000007 002336  BISB    #DXST,DXCS    ;PRESENT TO CHANNEL
4103 010132 110037 013070      MOVVB   DEV,DXACT    ;SET DX ACTIVE FLAG
4104 010136 000425                BR      DXEXIT      ;RETURN FROM INTERRUPT
4105
4106      :
4107      SENSE COMMAND DESIRED BY 360
4108
4109      :
4110 010140 012777 000002 002326  SENSECM: 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      MOVVB   R2,DXCA
4116 010172 012777 177777 002276  MOV     #-1,DXBC    ;TRANSFER 1 BYTE
4117 010200 110037 013070      MOVVB   DEV,DXACT    ;SET DX ACTIVE FLAG
4118 010204 052777 000005 002256  BIS     #DXRD,DXCS    ;START TRANSFER
4119
4120      :
4121      EXIT FROM THE DX ISR
4122
4123      :
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      :..... R0 = 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      :      MOV      R1,-(SP)          ;SAVE R1
4145      :      MOV      DEV,R1
4146      SUD10:  DEC      R1                ;DEC DEVICE NUMBER
4147      :      BEQ      SUDEX              ;DONE, EXIT
4148      :      ADD      #2000,DTAB        ;INCR TO NEXT DEV TABLE
4149      :      BR       SUD10              ;TRY AGAIN
4150      :
4151      SUDEX:  MOV      ,SP)+,R1          ;RETURN TO CALLER
4152      :      RTS      PC
4153      :
4154      :
4155      CLEAR DEVICE STATUS TABLE
4156      :
4157      CALLING SEQUENCE
4158      :..... R0 = 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      :      CLRB     SCMD(DTAB)         ;RESET CURRENT COMMAND ENTRY
4173      :      CLRB     SSTAT(DTAB)        ;RESET DEVICE STATUS ENTRY
4174      :      CLR      SBUFA(DTAB)        ;RESET CURRENT BUFFER ADDRESS POINTER
4175      :      CLR      SRBYTC(DTAB)       ;RESET REMAINING BYTE COUNT
4176      :      CLRB     SLCMD(DTAB)        ;RESET LAST COMMAND ENTRY
4177      :      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 (R0) CONTAINS CURRENT DEVICE NUMBER
4186      JSR      PC,CSPWST

```

NO7

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 91
DX11-B ISR (UTILITY SUBROUTINES)

SEQ 0091
SEQ 0091

```

4187      : .....RETURN TO CALLER WITH DEVICE STATUS BYTE RESET
4188      :
4189      : ALL REGISTERS ARE PRESERVED ACCROSS THIS SUBROUTINE
4190      :
4191      CSPWST: MOV     R5, -(SP)      ;SAVE REGISTER FOR SUBROUTINE USAGE
4192      TSTB    SONLF(DTAB)        ;IS DEVICE ON-LINE?
4193      BNE     10$                ;NO, JUST EXIT
4194      MOV     DEV, R5             ;GET DEVICE NUMBER AND COMPUTE
4195      ADD     DEVCON, R5           ;ADDRESS OF SPW STATUS BYTE
4196      ADD     R5, R5
4197      ADD     STSPW, R5
4198      CLRB    (R5)                ;RESET SPW STATUS BYTE
4199      10$:  MOV     (SP)+, R5      ;RESTORE REGISTER
4200      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      : MUXEND: TSTB      CHTYPE      ;SELECTOR OF MULTIPLEXER CHANNEL??
4213      :          BNE      SS          ;SELECTOR CHANNEL -- EXIT
4214      :
4215      : MULTIPLEXER CHANNEL
4216      :
4217      : SUB      #4,SRBYTC(DTAB) ;DECR REMAINING BYTE COUNT
4218      : BGT      10$            ;IF > 1, DATA TRANSFER NOT COMPLETE IE*
4219      :
4220      : DATA TRANSFER COMPLETE ON MUX CHANNEL
4221      :
4222      : CLR      SRBYTC(DTAB)      ;INSURE REMAINING BYTE COUNT ZERO
4223      : SEC      30$              ;SET MUX TRANSFER COMPLETE FLAG
4224      : BR       30$              ;GOTO COMMON EXIT
4225      :
4226      : DATA TRANSFER INCOMPLETE
4227      :
4228      : 10$: ADD      #4,SBUFA(DTAB) ;BUMP BUFFER ADDRESS
4229      : 20$: CLC      ;RESET FLAG TO INDICATE MUX CHAN NOT DONE
4230      : 30$: 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 010402 010046
4259 010404 010146
4260 010406 010246
4261 010410 126327 000001 000002
4262 010416 001535
4263 010420 126327 000001 000012
4264 010426 001542
4265 010430 126327 000001 000006
4266 010436 001522
4267
4268
4269
4270
4271
4272 010440 105737 012532
4273 010444 001102
4274
4275
4276
4277 010446 016301 000006
4278 010452 126327 000001 000005
4279 010460 001016
4280
4281
4282
4283
4284 010462 005263 000014
4285 010466 112102
4286 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 REGISTERS 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 MANUAL 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 TSTYP ;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
MOVB (R1)+,R2 ;GET LINE NUMBER
BIC #177760,R2 ;GET ONLY LINE NUMBER

```



```

4287 010474 005063 000004      CLR      SCURS(DTAB)      ;
4288                                     :
4289                                     :
4290                                     :
4291 010500 005702      10$: TST      R2                      ;DONE?
4292 010502 001405      BEQ      R2                      ;YES, MORE DATA INTO DISPLAY BUF
4293 010504 062763      ADD      #LINSZ,SCURS(DTAB)      ;INCR TO NEXT LINE
4294 010512 005302      DEC      R2                      ;DECR LINE COUNT
4295 010514 000771      BR       10$
4296                                     :
4297                                     :
4298                                     :
4299                                     :
4300                                     :
4301                                     :
4302                                     :
4303                                     :
4304                                     :
4305 010516 016302 000010      20$: MOV      SOUTB(DTAB),R2 ;COMPUTE DISPLAY ADDR
4306 010522 066302 000004      ADD      SCURS(DTAB),R2
4307 010526 026327 000014 000741  CMP      SRYTC(DTAB),#DISPSZ+1 ;ALL CHARS PROCESSED?
4308 010534 103077      BHIS     DISCEX                  ;YES, EXIT
4309 010536 005263 000014      INC      SRYTC(DTAB)      ;INCREMENT THE BYTE COUNT
4310 010542 112100      MOVB     (R1)+,R0                ;GET THE NEXT BYTE RECEIVED AND BUMP POINTER
4311 010544 042700 177400      BIC      #177400,R0        ;STRIP SIGN EXTENSION BITS (IF ANY)
4312 010550 116000 011754      MOVB     EBCDTB(R0),R0    ;FOLD CHARACTER INTO ASCII CHARACTER SET
4313 010554 042700 177400      BIC      #177400,R0        ;STRIP SIGN EXTENSION BITS, IF ANY
4314 010560 162700 000040      SUB      #40,R0           ;SCALE INTO ASCII TABLE RANGE
4315 010564 116012 012354      MOVB     ATOETB(R0),(R2)  ;COMPLETE FOLDING BY RETRANSLATING TO EBCDIC
4316 010570 005263 000004      INC      SCURS(DTAB)      ;INCR CURSOR PTR
4317 010574 121227 000025      CMPB     (R2),#NEWLINE    ;WAS A NEW LINE SPECIFIED?
4318 010600 001015      BNE      60$
4319                                     :
4320                                     :
4321                                     :
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                                     :
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                                     :
4342                                     :

```

FRIEND TEST, IF SEPARATE I/O BUFFERS DON'T COPY
INPUT TO OUTPUT BUFFER

```

4343
4344 010652 105737 012533
4345 010656 001026
4346 010660 016301 000006
4347 010664 016302 000010
4348 010670 012700 000360
4349
4350
4351
4352 010674 012122
4353 010676 005300
4354 010700 001375
4355 010702 000414
4356
4357
4358
4359
4360
4361 010704 005063 000004
4362 010710 000411
4363
4364
4365
4366
4367
4368
4369
4370
4371 010712 016301 000020
4372 010716 005301
4373 010720 112711 000100
4374 010724 166301 000010
4375 010730 010163 000004
4376
4377
4378
4379
4380
4381
4382 010734
4383
4384
4385
4386
4387
4388 010734 012602
4389 010736 012601
4390 010740 012600
4391 010742 000207
4392
4393
4394
4395
4396
4397
4398

DISFRN: TSTB IOBUF ;SEPARATE I/O BUFFERS?
        BNE DISCEX ;YES, DON'T COPY INPUT TO OUTPUT
        MOV SINBF(DTAB),R1 ;SET UP INPUT BUFFER ADDRESS
        MOV SOUTB(DTAB),R2 ;SET UP OUTPUT BUFFER ADDRESS
        MOV #DISPSZ/2,R0 ;TRANSFER THE INPUT BUFFER TO THE OUTPUT BUFFER

        ;
        ; PERFORM COPY
        ;
DIS:     MOV (R1)+,(R2)+ ;INPUT TO OUTPUT
        DEC R0 ;ARE WE DONE?
        BNE DIS ;NO, CONTINUE COPY
        BR DISCEX ;PREPARE TO RETURN TO CALLER

        ;
        ; A READ FULL BUFFER WAS PERFORMED
        ; THE CURSOR MUST BE RESET TO THE BEGINNING OF THE SCREEN
        ;
DREAD:   CLR SCURS(DTAB) ;RESET THE CURSOR
        BR DISCEX ;AND PREPARE TO EXIT

        ;
        ; A READ MANUAL INPUT WAS PERFORMED
        ; TO EMULATE THE 2260 SCREEN THE START OF MANUAL INPUT CHARACTER
        ; MUST BE DELETED FROM THE SCREEN
        ;
DRMI:    MOV SMINS(DTAB),R1 ;GET THE STARTING ADDRESS
        DEC R1 ;DECREMENT TO THE SMI CHAR
        MOVB #EBCDSP,(R1) ;BLANK OUT THE CHARACTER
        SUB SOUTB(DTAB),R1 ;AND COMPUTE THE CURSOR POSITION
        MOV R1,SCURS(DTAB)

        ;
        ; A SHORT READ MANUAL INPUT WAS PERFORMED
        ; NO ACTION REQUIRED BY DISPLAY CONTROL ROUTINE
        ;
DSRMI:

        ;
        ; RESTORE REGISTERS AND RETURN TO CALLER
        ;
DISCEX:  MOV (SP)+,R2 ;RESTORE SAVED REGISTERS
        MOV (SP)+,R1
        MOV (SP)+,R0
        RTS PC ;RETURN TO THE CALLER
        .SBTTL TELETYPE (CONSOLE) INPUT ISR

        ;
        ; TELETYPE INPUT HANDLER (ISR)
        ;
        ; CONTROL PASSES HERE ON A TELETYPE INPUT INTERRUPT
        ;
        ; DATA IS INPUT FROM THE CONTROL CONSOLE AND STORED INTO

```


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 CUMPS
(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

```

4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420 010744 010046
4421 010746 010146
4422 010750 017700 001502
4423 010754 042700 177600
4424 010760 013701 012742
4425 010764 020027 000020
4426 010770 001002
4427 010772 000137 001002
4428 010776 020027 000023
4429 011002 001003
4430 011004 105237 013073
4431 011010 000503
4432 011012 020027 000021
4433 011016 001010
4434 011020 105037 013073
4435 011024 105737 013071
4436 011030 001473
4437 011032 004737 011252
4438 011036 000470
4439 011040 020027 000003
4440 011044 001011
4441 011046 105737 013053
4442 011052 001457
4443 011054 105237 013054
4444 011060 012701 012640
4445 011064 000137 011220
4446 011070 105737 013053
4447 011074 001051
4448 011076 110021
4449 011100 020027 000015
4450 011104 001005
4451 011106 012701 012640
4452 011112 105237 013053
4453 011116 000440
4454 011120 020027 000177

```

```

4 IN:  MOV    RO, -(SP)      ;SAVE REGISTERS
        MOV    RI, -(SP)
        MOV    #TKB, RO     ;GET TELE CHARACTER
        BIC    #177600, RO   ;INSURE 7-BIT ASCII
        MOV    TPTR, RI     ;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
6$:     CMP    RO, #CTL.Q     ;CONTROL-Q, RESUME CONSOLE OUTPUT?
        BNE    10$          ;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   TCMAC        ;IS A COMMAND ACTIVE?
        BEQ    90$          ;NO, TREAT AS A DELETE LAST LINE
        INCB   TCMDAB        ;YES, SET ABORT FLAG
        MOV    #TBUF, RI    ;SET UP BUFFER POINTER
        JMP    100$         ;EXIT
20$:    TSTB   TCMAC        ;TELE CMD CURRENTLY ACTIVE?
        BNE    100$         ;YES, IGNORE CHARACTER
        MOVB   RO, (RI)+     ;STORE CHAR INTO BUFFER - INC PTR
        CMP    RO, #CR       ;LINE DELIMITER -- C/R?
        BNE    30$         ;NO
        MOV    #TBUF, RI    ;RESET BUFFER PTR
        INCB   TCMAC        ;YES, SET COMMAND ACTIVE FLAG
        BR     100$         ;DON'T PRINT THE LINE DELIMITER
30$:    CMP    RO, #RUBOUT   ;A RUBOUT?

```

G08

MAINDEC-11-DZDNT-B NEW DZ11-B RESPONDER
 2018.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 97
 TELETYPE (CONSOLE) INPUT ISP

SEP 0097
 SEP 0097

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 'BACK SLASH
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, PTR	:SAVE BUFFER PTR
4475	011224	012601		MOV	(SP)+, R1	:RESTORE REGISTERS + EXIT
4476	011226	012600		MOV	(SP)+, PO	
4477	011230	000002		RTI		

H08

MAINDEC-11-DDD11-B NEW D111-B RESPONDER
C2DXIB.F11 30-JAN-78 14:19

MACY11 3DA(1052) 01-FEB-78 09:46 PAGE 98
TELETYPE (CONSOLE) OUTPUT ISR

SEG C098
SEG C098

4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489

011232 105037 013052
011236 105737 013071
011242 001402
011244 004737 011252
011250 000002

.....

PISR:

CLRB
TSTB
BEQ
JSR
RTI

PIUPL
PCTR
IOS
PC,PROUT

:CLEAR PRINTER BUS, FLAG
:ANY MORE DATA TO PRINT?
:NO, EXIT
:OUTPUT ANOTHER CHAR

.SBTTL TELETYPE (CONSOLE) OUTPUT ISR
TELETYPE OUTPUT DRIVER (ISR) -- PRINT

CONTROL PASSES HERE ON A TELE OUT INTERRUPT

.SBTTL TELETYPE OUTPUT HANDLING SUBROUTINES

SEND DATA TO PRINTER, IF NOT BUSY

CALLING SEQUENCE
JSR PC,PROUT
.....RETURNIF 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	JPFPTR, JTPB		: OUTPUT NEXT CHAR
	INC	PFPTTR		: INCR PRINT FETCH POINTER
	CMP	PFPTTR, #PBFE		: TIME TO WRAP AROUND?
	BNE	20\$		: NO, EXIT
	MOV	#PBFS, PFPTTR		: YES, RESTORE TO START OF BUFFER
20\$:	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

SEQ 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,MESG
.WORD P2BF
RTS PC
P2BF: .WORD 0
.BYTE 377,377

PRINT 1 CHARACTER
CALLING SEQUENCE
.....RO 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 RO,P1BF
JSR R1,MESG
.WORD P1BF
RTS PC
P1BF: .BYTE 0,377
;RETURN TO THE CALLER

K08

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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 101
TELETYPE OUTPUT HANDLING SUBROUTINES

SEG 0:01
SEG 0:01

```

4568      :
4569      : PRMSG PRINT A CHARACTER STRING
4570      :
4571      : CALLING SEQ
4572      : .....R2 CONTAINS THE STARTING ADDRESS OF THE MESSAGE
4573      : JSR      PC,PRMSG
4574      : .....RETURN
4575      :
4576      : NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4577      :
4578      : NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4579      : PRMSG: MOV      R2,-(SP)      ;SAVE REGS
4580      :      MOV      R3,-(SP)
4581      :      MOV      PPTR,F3      ;GET PRINT OUTPUT POINTER
4582      :      CMPB     (R2),#377    ;END OF MESSAGE?
4583      :      BEQ      40$          ;YES, EXIT
4584      :      MOV      (R2)+,(R3)+  ;NO MOVE NEXT CHAR TO PRINT BUFFER
4585      :      INCR     PCTR          ;INCR CHAR COUNTER
4586      :      CMPB     R3,#PBFE     ;AT END OF BUFFER?
4587      :      BNE      20$          ;NO
4588      :      MOV      #PBFS,R3      ;YES, WRAP AROUND TO BEG OF BUFFER
4589      :      JSR      PC,PROUT     ;CAN WE START PRINT?
4590      :      CMPB     PCTR,PMAX     ;IS PRINT BUFFER FULL?
4591      :      BEQ      30$          ;YES, WAIT TILL ROOM AVAILABLE
4592      :      BF       10$          ;GET NEXT CHAR
4593      :      MOV      R3,PPTR      ;EXIT, RESTORE PUT PTR
4594      :      MOV      (SP)+,R3      ;RESTORE REGS
4595      :      MOV      (SP)+,R2
4596      :      RTS      PC          ;RETURN TO THE CALLER
4597      :
4598      :
4599      :
4600      : MSG -- PRINT A CHARACTER STRING ON THE SYSTEM CONSOLE
4601      :
4602      : CALLING SEQUENCE
4603      : JSP      R1,MSG
4604      : .WORD    ADDRESS OF START OF MESSAGE
4605      : .....RETURN
4606      :
4607      : NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4608      :
4609      : NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4610      : MSG:  MOV      R2,-(SP)      ;SAVE REGISTER
4611      :      MOV      (R1)+,R2      ;GET ADDRESS OF MESSAGE AND BUMP FOR RETURN
4612      :      JSR      PC,PRMSG     ;MORE MESSAGE PROCESSING
4613      :      MOV      (SP)+,R2      ;RESTORE SOILED REGISTER
4614      :      RTS      R1          ;RETURN TO THE CALLER
4615      :
4616      :
4617      :
4618      :
4619      : INMES PRINT A CHARACTER STRING
4620      :
4621      : CALLING SEQUENCE
4622      : JSR      R1,INMES
4623      : .WORD    ADDRESS OF MESSAGE

```


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

MACY11 30A(1052) 01-FEB-78 09:46 PAGE 102
TELETYPE OUTPUT HANDLING SUBROUTINES

SEC 0:02
SEC 0:02

4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641

011506 010246
011510 113746 013130
011514 112737 000377 013130
011522 012102
011524 004737 011404
011530 112637 013130
011534 012602
011536 000201

```

.....RETURN

INMES IS USED FOR ROUTINES AT THE ISR LEVEL AND DOES
      NOT CHECK TO SEE IF DATA WILL BE OVERLAYED 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, PRMES3      ;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 003012
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
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


```

CHTB -- CONVERT ASCII HEX TO BINARY

CALLING SEQUENCE
JSR ...R2 = ADDRESS OF FIRST CHARACTER TO BE CONVERTED
      PC,CTHB
      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$:   MOVSB     (R2)+,R4      ;GET THE FIRST CHARACTER
        SUB      #0,R4        ;SCALE RELATIVE TO ASCII ZERO
        BMI      30$          ;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
        MOVSB     (R2)+,R4    ;AND PUT IT R4
        RTS      PC          ;ITS TIME TO RETURN TO THE CALLER

```

```

4707      .SBTTL  PROCESSOR ERROR TRAP HANDLERS
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      :

```

TRAP OUT ROUTINES

MEMORY TIME OUT ROUTINE

```

4716      011672  012702  014143      MTO:  MOV      #PMTG,R2      ;SET UP ADDRESS OF THE PRINT ROUTINE
4717      011675  000404              BR        TOUTRT      ;TO GENERALIZED TRAP OUT ROUTINE

```

MEMORY MANAGEMENT TRAP OUT ROUTINE

```

4723      011700  005037  177572      MMERR: CLR      MMSRD      ;CLEAR THE MEMORY MANAGEMENT BIT
4724      011704  012701  014167      MOV      #PMMERR,R1    ;SET UP ADDRESS OF ERROR MESSAGE
4725      011710  000005              TOUTRT: RESET      ;CLEAR ALL DEVICES
4726      011712  105037  013052      CLRB     PIUFL      ;CLEAR PRINT IN USE FLAG
4727      011716  005037  177776      CLR      PSW        ;LOWER PROCESSOR STATUS TO ALLOW INTERRUPTS TO CUM
4728      011722  004737  011404      JSR      PC,PRHESG    ;PRINT THE ERROR MESSAGE
4729      011726  105737  013071      10$:  TSTB     PCIR      ;IS PRINTING DONE?
4730      011732  001375              BNE      10$
4731      011734  000000              HALT
4732      011736  000137  001000      JMP      START      ;YES, HALT

```

INVALID UNIBUS ADDRESS TRAP

```

4738      011742  022626              UNTRP: CMP      (SP)+,(SP)+    ;POP THE PUSH STACK
4739      011744  005037  177776      CLR      PSW        ;CLEAR THE PROCESSOR STATUS WORD
4740      01175C  000137  001174      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

EBCDTB:

4741	011754	040440	041502	042504
4742	011754	043506	044510	027136
4743	011762	024074	020453	
4744	011770	045046	046113	057515
4745	011774	050117	051121	022137
4746	012002	024452	056473	
4747	012010	027455	052123	053125
4748	012014	054127	055131	026042
4749	012022	055445	037476	
4750	012030	030460	031462	032464
4751	012034	033466	034470	021472
4752	012042	023500	056075	
4753	012050	040440	041502	042504
4754	012054	043506	044510	027136
4755	012062	024074	020453	
4756	012070	045046	046113	047115
4757	012074	050117	051121	022137
4758	012102	024452	056473	
4759	012110	027455	052123	053125
4760	012114	054127	055131	026042
4761	012122	055445	037476	
4762	012130	030460	031462	032464
4763	012134	033466	034470	021472
4764	012142	023500	056075	
4765	012150	040440	041502	042504
4766	012154	043506	044510	027136
4767	012162	024074	020453	
4768	012170	045046	046113	047115
4769	012174	050117	051121	022137
4770	012202	024452	056473	
4771	012210	027455	052123	053125
4772	012214	054127	055131	026042
4773	012222	055445	037476	
4774	012230	030460	031462	032464
4775	012234	033466	034470	021472
4776	012242	023500	056075	
4777	012250	040440	041502	042504
4778	012254	043506	044510	027136
4779	012262	024074	020453	
4780	012270	045046	046113	047115
4781	012274	050117	051121	022137
4782	012302	024452	056473	
4783	012310	027455	052123	053125
4784	012314	054127	055131	026042
4785	012322	055445	037476	
4786	012330	030460	031462	032464
4787	012334	033466	034470	021472
4788	012342	023500	056075	
4789	012350			

.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

ASCII TO EBCDIC CONVERSION TABLE

```

...
ATOE*8: .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 SIGNALS

```

```

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      C12636      :
4873      C12636 SSTACK = .+.100
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      :
4883      012740      : START OF TELETYPE INPUT BUFFER
4884      012740 000000 TBUF = .+.100
4885      012742 000000 TBUFE: .WORD 0      : END OF TELETYPE INPUT BUFFER
4886      012744 000000 TPTR: .WORD 0      : TELE IN PTR
4887      013046      : START OF PRINT BUFFER
4888      013046      :
4889      013046 000000 PBFE = .+.100
4890      013050 000000 PFPTR: .WORD 0      : END OF PRINT BUFFER
4891      013052 000 PPTR: .WORD 0      : PRINT FETCH PTR
4892      013053 000 PIUFL: .BYTE 0      : PRINT PUT PTR
4893      013054 000 TCMAC: .BYTE 0      : PRINTER IN USE FLAG
4894      013055 000 TCMAB: .BYTE 0      : TELE COMMAND ACTIVE FLAG 0 = NON-ACT
4895      013056 000 LINECT: .BYTE 0      : TEL COMMAND ABORT 1 = ABORT
4896      013057 000 WK: .BYTE 0      : LINE CTR - CHARS LINE
4897      013060 000000 WK1: .BYTE 0      : WORK LOC
4898      013062 000000 TTPTR: .WORD 0      : WORK LOC
4899      013064 000000 TTADDR: .WORD 0      : TUMBLE TABLE PTR
4900      013066 000 SDEVTB: .WORD 0      : BEG OF TUMBLE TABLE
4901      013067 000 DXSTPF: .BYTE 0      : START OF DEVICE TABLES
4902      013070 000 MAXDEV: .BYTE 0      : DX STOP FLAG
4903      013071 000 DXACT: .BYTE 0      : HIGHEST DEV # 1 - 8
4904      013072 000 PCTR: .BYTE 0      : DXACTIVE FLAG
4905      013073 000 DXABFL: .BYTE 0      : PRINT BUFFER COUNTER
4906      :
4907      013074 000000 TTYSTP: .BYTE 0      : DX ABORT FLAG 0 = NO ABORT, 1 = ABORT
4908      C13076 000000 CMDCHF: .WORD 0      : CONSOLE OUTPUT STOP FLAG 0 = OUTPUT; 1 = NO OUTPUT
      MDEV: .WORD 0      : COMMAND CHAIN FLAG
      : DEV # IN MPXR EXEC

```

MAINDEC-11-DZDXI-B NEW EX-1-B RESPONDER
 C2DXIB.F11 30-JAN-78 14:19

MAC111 30A(1052) 01-FEB-78 09:46 PAGE 109
 PROGRAM CONSTANTS AND VARIABLES

SEC 0:09
 SEC 0:09

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
 FTIMFL: .WORD 0 : FIRST TIME FLAG

MAINDEC-11-02DAI-B NEW DAI-B RESPONDER
02DAIB.F11 30-JAN-78 14:19

MACY11 30A(1052) 01-FEB-78 09:45 PAGE 110
MESSAGES

353 2110
000 0110

4929
4930

.SBTTL MESSAGES
.NLIST BEX

SYSTEM MESSAGES

013134	215	212		STMSG:	.BYTE	215,212	
013136	055104	054104	026511		.ASCII	/02DAI-B NEW DAI-B RESPONDER	
013173	377	377			.BYTE	377,377	
013175	215	212		UNMSG:	.BYTE	215,212	
013177	125	044516	052502		.ASCII	/UNIBUS ADDRESS -OCTAL- :	
013230	377				.BYTE	377	
	013232				.EVEN		
013232	215	212		VECTMS:	.BYTE	215,212	
013234	047111	042524	051122		.ASCII	/INTERRUPT VECTOR ADDRESS -OCTAL- :	
013277	377				.BYTE	377	
					.EVEN		
013300	215	212		DEVMES:	.BYTE	215,212	
013302	042504	044526	042503		.ASCII	/DEVICE ADDRESSES -HEX- XX,XXX :	
013342	377				.BYTE	377	
	013344				.EVEN		
013344	215	212		CHTYMS:	.BYTE	215,212	
013346	044103	047101	042516		.ASCII	/CHANNEL TYPE (M OR S):	
013375	377				.BYTE	377	
					.EVEN		
013376	215	212		MMES:	.BYTE	215,212	
013400	042515	047515	054522		.ASCII	/MEMORY MANAGEMENT (Y OR N):	
013434	377				.BYTE	377	
	013436				.EVEN		
013436	215	212		BFREMS:	.BYTE	215,212	
013440	052502	043106	051105		.ASCII	/BUFFER RELOCATION, IF SPECIFIED - IN EVEN .000'S -OCTAL- :	
013533	377				.BYTE	377	
					.EVEN		
013534	215	212		TESTMS:	.BYTE	215,212	
013536	051106	042511	042116		.ASCII	/FRIEND (F) OR 2848 DIAG(D):	
013572	377				.BYTE	377	
	013574				.EVEN		
013574	215	212		FIOMS:	.BYTE	215,212	
013576	042523	040520	040522		.ASCII	/SEPARATE I-O BUFFERS (Y OR N):	
013635	377				.BYTE	377	
					.EVEN		
013636	215	212		FILLMS:	.BYTE	215,212	
013640	052517	050124	052125		.ASCII	/OUTPUT BUFFER FILL CHARACTER -HEX- :	
013705	377				.BYTE	377	
					.EVEN		
013706	207	207	215	NXMSG:	.BYTE	207,207,215,212	
013712	047516	020116	054105		.ASCII	/NON EX-MEM ERROR	
013732	215	212	377		.BYTE	215,212,377	
	013736				.EVEN		
013736	207	207	215	PARMES:	.BYTE	207,207,215,212	
013742	040520	044522	054524		.ASCII	/PARITY ERROR	
013756	212	215	377		.BYTE	212,215,377	
	013762				.EVEN		
013762	207	207	215	ILLMES:	.BYTE	207,207,215,212	
013766	046111	042514	040507		.ASCII	/ILLEGAL DEVICE NUMBER	
014013	212	215	377		.BYTE	212,215,377	
					.EVEN		

MESSAGES

```

014016 052503 051122 047105 STPMES: .ASCII /CURRENT DEVICE NUMBER --
014047 377 .BYTE 377
.EVEN

014050 207 207 215 215 INVLDG: .BYTE 207,207,215,215
014054 047111 040526 044514 .ASCII /INVALID DX COMMAND/
014076 212 215 377 .BYTE 212,215,377
.EVEN

014102 014102 047516 046440 PNMOM: .ASCII / NO MEMORY MANAGEMENT AVAILABLE
014142 020040 377 .BYTE 377
014143 215 212 207 207 PMTO: .BYTE 215,212,207,207
014147 115 046505 051117 .ASCII /MEMORY TIME OUT/
014166 377 .BYTE 377
014167 215 212 207 207 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 HELPM: .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 /
014504 020040 020040 020040 .ASCII / DTT,C DUMP TUMBLE TABLE IN CODE "C"/<CR><LF>
014602 020040 020040 020040 .ASCII / DIN,C,XX DUMP INPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014701 105 026440 020055 .ASCII / DOT,C,XX DUMP OUTPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014733 040 020040 020040 .ASCII /E -- ENABLE DEVICE ON DX/(CR)<LF>
014776 020106 026455 043040 .ASCII / EXX ENABLE DEVICE XX/(CR)<LF>
015030 020040 020040 020040 .ASCII /F -- FILL BUFFER COMMAND/(CR)<LF>
015116 020040 020040 020040 .ASCII / FIN,HH,XX FILL INPUT BUFFER ON DEV XX WITH HH/(CR)<LF>
015205 110 026440 020055 .ASCII / FOT,HH,XX FILL OUTPUT BUFFER ON DEV XX WITH HH/(CR)<LF>
015230 020040 020040 020040 .ASCII /H -- HELP COMMAND/(CR)<LF>
015252 020113 026455 045440 .ASCII / THIS TEXT/(CR)<LF>
015310 020040 020040 020040 .ASCII /K -- KILL A DEVICE ON THE DX/(CR)<LF>
015351 122 026440 020055 .ASCII / KXX KILL DEVICE XX/(CR)<LF>
015377 040 020040 020040 .ASCII /R -- ENABLE DX (RUN)/(CR)<LF>
015432 020123 026455 042040 .ASCII / R RUN TEST/(CR)<LF>
015462 020040 020040 020040 .ASCII /S -- DISABLE DX (STOP)/(CR)<LF>
015525 040 020040 020040 .ASCII / S STOP IMMEDIATELY/(CR)<LF>
015605 040 020040 020040 .ASCII / SD STOP AFTER NEXT DATA TRANSFER/(CR)<LF>
015667 040 020040 020040 .ASCII / SE STOP AFTER NEXT ENDING SEQUENCE/(CR)<LF>
015744 020040 020040 020040 .ASCII / SI STOP ON NEXT SEL SEQ (ISS)/(CR)<LF>
016020 005015 044127 051105 .ASCII / SP STOP ON NEXT PARITY ERROR/(CR)<LF>
016032 020040 020040 041442 .ASCII / (CR)<LF>/WHERE: (CR)<LF>
016100 020040 020040 020040 .ASCII / "C" IS CODE FORMAT 0 = OCTAL/(CR)<LF>
016146 020040 020040 020040 .ASCII / A = ASCII/(CR)<LF>
016215 040 020040 020040 .ASCII / E = EBCDIC/(CR)<LF>
016261 040 020040 021040 .ASCII / H = HEX/(CR)<LF>
016331 040 020040 021040 .ASCII /
016367 103 047117 047523 .ASCII / "XX" IS DX-11 DEVICE NUMBER IN HEX/(CR)<LF>
016423 103 046124 041455 .ASCII / "HH" IS A HEX CHARACTER/(CR)<LF>
016465 103 046124 050055 .ASCII /CONSOLE CONTROL CHARACTERS/(CR)<LF>
016554 052103 026514 020121 .ASCII /CTL-C (IC) ABORT CURRENT COMMAND/(CR)<LF>
016651 103 046124 051455 .ASCII /CTL-P (IP) REQUESTS THE REENTRY OF CONTROL PARAMETERS/(CR)<LF>
016651 103 046124 051455 .ASCII /CTL-Q (IQ) RESUME OUTPUT AFTER TEMPORARILY STOPPING BY CTL-S/(CR)<LF>
016651 103 046124 051455 .ASCII /CTL-S (IS) TEMPORARILY STOP OUTPUT TO CONSOLE/(CR)<LF>

```


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

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

353 0112
 353 0112

MESSAGES

016730	052103	026514	020125
016776	052522	047502	052125
017045	015	012	012
	002551		
017051	215	212	
017053	124	047517	041440
017107	377		

HELPLN
 TOOC:

.ASCII /CTL-U (1U) DELETE CURRENT INPUT LINE (CR LF
 .ASCII /RUBOUT -- DELETE LAST CHARACTER INPUT (CR LF
 .BYTE CR,LF,LF,LF
 = -HELPM5
 .BYTE 215,212
 .ASCII TOO CLOSE TO 200000 BOUNDAR-
 .BYTE 377
 .LIST BEY
 .EVEN
 .ENC

433:
 4332

000001

SEP 08 13
SEP 08 13

[illegible]

SEC 0114
SEC 0114

DTUMTB	003632	2862	2893	4917											
DJMP	003432	2707	2845												
DXAB	006536	3468	3585	3671	3751										
DXABFL	013072	2673	2747	2821	3752	4904									
DXACT	013070	2745	3110	3534	3550	3570	3572	3602	3713	3728	3829	3871	3960	4014	
		4103	4117	4902											
DXBA	012474	3821	3822	3846	3850	3855	3863	3864	3948	4003	4110	4111	4112	4842	
DXBC	012476	3733	3826	3854	3865	3952	3957	4007	4012	4116	4843				
DXCA	012466	3825	3869	3951	4006	4084	4115	4839							
DXCB	012504	4846													
DXCS	012470	2732	2734	2735	2756	2757	2758	2809	2822	2823	3421	3450	3536	3751	
		3831	3873	3961	4024	4102	4118	4840							
DXDS	012464	2406	2816	4838											
DXENB =	000100	2072	2758	2809	3751										
DXES1	012510	4848													
DXES2	012514	4850													
DXEXEC	006554	3126	3414	3759											
DXEXIT	010212	3386	3754	3779	3790	3832	3874	3993	3914	3962	4025	4104	4124		
DXISR	005334	2416	3372												
DXMI	012502	4845													
DXMO	012500	4844													
DXMOB	012512	4849													
DXNC	012506	4847													
DXONLN=	001000	2069	2732	2758	3751										
DXOS	012472	2473	4101	4841											
DXRD =	000005	2079	3873	4024	4118										
DXRST =	000001	2077													
DXST =	000007	2080	4102												
DXSTPF	013066	2793	3607	3639	3654	3730	3744	4900							
DXWR =	000003	2078	3831	3961											
EADDR	013122	2876	2882	2934	2940	3285	3309	3310	3322	3323	4919				
EBCDMP	004572	2850	3139												
EBCDSP=	000100	2019	2362	3523	4053	4373									
EBCDTB	011754	3142	3145	4312	4748										
EDEV	012524	2247	2252	2260	2261	2486	2536	4859							
ENABLE	004222	2709	3022												
EOM =	000152	2021	3104												
EQPCHK=	000020	2105	3736												
ERASCM	007770	3805	3929	4051											
ESEND =	000100	2350	3491												
ESEQ	010042	3801	3844	3852	3925	3978	4071								

GLMIN	005172	3266	3304#						
GLMOT	005232	3268	3317#						
GLMTT	005142	3264	3292#						
GLOCT	005112	3262	3279#						
HDMP	004760	2814	3218#						
HELP	004004	2713	2951#						
HELPLN=	002551	2952	4930#						
HELPM5	014300	2951	4930#						
HEXDMP	004754	2853	3217#						
ILLMES	013762	3467	4930#						
INFOSC=	004000	2045#	3485						
INHEX	002624	2248	2392	2618#					
INIT	002026	2184	2364	2380	2382	2394	2404#		
INIT10	002372	2526#	2528						
INMES	011506	3466	3583	3637	3669	4634#			
INOCT	002632	2201	2224	2276	2304	2321	2359	2378	2633#
INPUT	004030	2715	3067#						
INRS	002636	2619	2634#						
INTREQ=	000100	2103#	3682						
INVLDC	014050	3670	4930#						
IOBUF	012533	2377#	2385#	4344	4664#				
ISSREJ=	000010	2053#	3610						
KILL	004266	2717	3046#						
KILLEX	004324	3054#							
KISAR0=	172340	1962#	2437						
KISAR7=	172356	1963#	2448						
KISDR0=	172300	1964#	2438						
LF	= 000012	1972#	4930						
LINECT	013055	2672#	3161	3164#	3165#	4894#			
LINSZ	= 000050	2022#	2024	4293	4325	4328			
LOOP	005364	3380	3411#	3504	3537	3552	3576	3631	3704
MAXDEV	013067	2539#	2568	2743	3349	3455	3532	3785	3909
MCMDTB	007326	3922	3923#						4901#
MDEV	013076	2754#	3894	3912	3921#	4908#	4909		
MESG	011472	2191	2293	2340	2586	2811	4546	4564	4611#
MEX	007210	3761	3891#						
MMERR	011700	2160	4723#						
MMES	013376	2305	4930#						
MMRESP	012527	2303#	2311#	2334	2427	4861#			
MMSRO	= 177572	1965#	2313#	2452#	4723#				
MREAD	007600	3928	3972	3974	3997#				
MRMI	= 007502	3924	3970#						
MSRMI	007502	3932	3971#						
MTO	011672	2158	2404	4716#					
MUXEND	010342	3714	3729	4212#					
MWRITE	007352	3923	3927	3943#					
NEWLNE=	000025	2018#	4317						
NEWPRM	001174	2199#	2205	2207	2209	2211	2212	4740	
NEWPI0	001460	2250	2275#	2282					
NEWPI20	001704	2323	2339	2349#					
NOLIN	= 000014	2023#	2024						
NOP	= 000003	2017#	3665						
NXM	= 040000	2042#	3488						
NXMMSG	013706	3584	4930#						
OCTDMP	004670	2816	2856	2988	3182#				
OCTEX	004744	3199#							

SEQ 0116
SEQ 0116

[illegible]

MAINDEC-11-DZDXI-B NEW DX11-B RESPONDER MACY11 30A(1052) 01-FEB-78 09:46 PAGE 118
CZDXIB.P11 30-JAN-78 14:19 CROSS REFERENCE TABLE -- USER SYMBOLS

SLCMD = 000001	1992#	3595	3601*	3827*	3866*	3959*	3997*	4176*	4261	4263	4265	4278	
SMI = 000112	2020#	3081											
SMINS = 000020	2002#	2741*	3082*	3518*	3846	3980	4371						
SOMLF = 000016	2000#	3030*	3048*	3677	4192								
SOUTB = 000010	1997#	2561*	2562*	3079	3321	3521	3847	3863	3981	4000	4051	4305	4347
	4374												
SPACE = 000040	1973#	3090	3094										
SRBYTC= 000014	1999#	3733*	3830*	3872*	3943	3947*	3953	3955	3973	3986*	3998	4002*	4008
	4010	4175*	4217*	4222*	4284*	4307	4309*						
SRDRQ = 000017	2001#	2740*	3106*	3517*	3600*	3780	3842	3845*	3898	3976	3979*		
SRD10 007142	3856	3866#											
SREAD 007120	3804	3840	3863#										
SRMI 007020	3800	3838#											
SSENSE= 000002	1993#	2739*	3029*	3516*	3549*	3682*	3690*	3696*	3736*	3828*	3841*	3870*	3958*
	3975*	4013*	4061*	4110									
SSRMI 007020	3808	3839#											
SSTACK= 012636	2142	2669	4873#										
SSTAT = 000003	1994#	3520*	3641*	4037*	4044*	4071*	4085	4087*	4101	4173*			
STAMOD= 000100	2091#												
START 001000	2120	2127#	4732										
STKSTB= 000002	2055#												
STMSG 013134	2192	4930#											
STOP 003264	2721	2781#											
STOPDX 003340	2675	2783	2809#										
STOUT 010050	4038	4045	4082#										
STPDX 006532	3608	3640	3655	3731	3744#								
STPMES 014016	2812	4930#											
STSPW 013114	2478*	3027	3052	4022	4096	4197	4916#						
SUDEV 010230	3073	3473	3629	3774	3895	4143#	4171						
SUDEX 010252	4147	4150#											
SUD10 010240	4146#	4149											
SWRITE 006730	3799	3803	3821#										
SYSINT 001006	2127	2142#											
SYSRST= 010000	2044#	3451											
TBUF = 012640	2173	2642	2682	2997	4444	4451	4464	4468	4882#				
TBUFE 012740	4470	4884#											
TCHEND 005624	3498	3500#											
TCHIS 005612	3495	3497#	3642										
TCTACT 013053	2636*	2640	2670*	2676	2987*	2995	4441	4446	4452*	4892#			
TCMDAB 013054	2637*	2638	2671*	2884	2904	2961	4443*	4893#					
TCMDTB 003044	2685	2704#											
TCUEND 005636	3501	3503#											
TESEND 005566	3489	3491#											
TESTMS 013534	2360	4930#											

B10

MAINDEC-11-DZDXI-B NEW D-11-B RESPONDER
CZDXIB.F11 30-JAN-78 14.19

MACY11 30A 10521 01-FEB-78 09:46 PAGE 119
CROSS REFERENCE TABLE -- USER SYMBOLS

3E6 01:18
3E6 01:13

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