

DX11-B

RESPONDER
MD-11-DZDXI-A

EP-DZDXI-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A

The microfiche card displays a grid of 100 frames of data, arranged in 10 rows and 10 columns. Each frame contains a different view of data, likely from a computer system, including text, tables, and diagrams. The data is too small to read clearly but appears to be organized in a structured manner.

.REM x

IDENTIFICATION

PRODUCT NAME: NEW DX11-B RESPONDER
PRODUCT CODE: MAINDEC-11-DZDXI-A-D
RELEASE DATE: JULY 1976
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 COPYRIGHT 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) 1976, BY DIGITAL EQUIPMENT CORPORATION

1.0 GENERAL DESCRIPTION

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 OR 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 OBSOLETES 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.

CORE MAP

0-777	INTERRUPT VECTORS (500 WORDS)
1000-1777	DX11-B TEST PROGRAM (4K WORDS)
X0-X777	SPW TABLE (256 WORDS)
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
4K	017500
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 SUCESSIVE 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 (1P) 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 "DX11-B 2848 SYSTEM TEST PROGRAM".
5. THE SYSTEM NOW REQUESTS THE OPERATOR TO ENTER THE PARAMETERS NECESSARY TO RUN THE TEST.

4.3.2 ENTERING TEST PARAMETERS

BEFORE ANY TESTS MAY BE RUN OR WHENEVER A CHANGE IN PARAMETERS IS DESIRED, THE OPERATOR WILL BE REQUIRED TO ENTER ALL THE TEST PARAMETERS. THE ENTERING OF THE PARAMETERS IS DONE VIA THE CONSOLE TELETYPE IN RESPONSE TO A SERIES OF QUESTIONS.

4.3.2.1 GENERAL RULES FOR ENTERING PARAMETERS

- A. ALL PARAMETERS MUST BE DELIMITED BY A CARRIAGE RETURN "(C/R)".
- B. IF A TYPING ERROR IS DETECTED BEFORE ENTERING THE C/R, IT MAY BE CORRECTED BY:
 1. USING RUBOUT(S) TO DELETE THE LAST CHARACTER(S)
 2. HITTING CONTROL-U (↑U) TO DELETE THE ENTIRE ENTRY
- 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 20000 BOUNDARY OR TO THE I/O PAGE. THE DX IS NOT CAPABLE OF HAVING THESE BUFFERS CROSS A 20000 BOUNDARY.
IT IS POSSIBLE TO OVERLAY THE ABSOLUTE LOADER WHICH RESIDES IN THE HIGHEST AVAILABLE 4K(10) OF THE FIRST 28K OF MEMORY.

DEFAULT BUFFER ADDRESS = 20 (20000)

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

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

DEFAULT OPTION = F -- FRIEND

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

4.4.2 DESCRIPTION OF MONITOR COMMANDS

R -- ENABLE THE DX FOR TESTING - RUN COMMAND

THE RUN COMMAND DOES THE FOLLOWING:

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

S -- DISABLE THE DX - STOP COMMAND

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

THE FORMS OF THE STOP COMMAND ARE:

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

AFTER THE CONDITIONS OF STOP ARE MET, THE DX WILL BE DIS-
ABLED. 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 IS 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

D -- 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,0 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 OUT-
PUT 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:

EXX 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 ODT. 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.
- B. 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

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 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 BEFORE COMMUNICATIONS TO THE 360 CAN RESUME.

6.3 ERROR MESSAGES AND SUGGESTED CORRECTIVE ACTIONS

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

THE SYSTEM SHOULD PROBABLY BE RECONFIGURED.

"MEMORY MANAGEMENT ERROR"

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

"ILLEGAL DEVICE NUMBER"

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

"INVALID DX COMMAND"

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

"NON EX-MEM ERROR"

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

"PARITY ERROR"

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

7.0 RESTRICTIONS
SEE MEMORY REQUIREMENTS (SECTION 2.2)

7.1 MULTIPLE DEVICE ADDRESSES

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

X
REM X
B.O PROGRAM DESCRIPTION

PURPOSE THE PURPOSE OF THIS PROGRAM IS TO GIVE INSIGHT
----- ON FUNCTIONALITY OF THE HARDWARE AND TO GIVE CREDENCE
TO THE FIRMWARE DESIGN. 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.

TKIN: THIS ROUTINE ACCEPTS CHARACTERS FROM KEYBOARD AND STUFFS
THEM AWAY IN TBUF, BUT FIRST, IT CHECKS FOR CERTAIN CON-
TROL CHARACTERS.

↑P - JUMP TO RESTART TO RESELECT PARAMETERS.

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

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

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

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

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

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

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

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

----GETS DX VECTOR - DITTO

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

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

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

----GETS CHANNEL TYPE M OR S

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

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

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

* CHECKS FOR BOUNDARY 20000 OR GREATER

* CHECKS FOR MULTIPLE OF 2000

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

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

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

----FILL CHARACTER IN HEX
SAVE IN FILLCH

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

- NOMM: NO MEMORY MANAGEMENT AVAILABLE.
- MMERR: MEMORY MANAGEMENT TRAP OUT ROUTINE
CLEAR WORLD
TYPE OUT ERROR MESSAGE
HALT
- INITRT: PRINTS MESSAGE - WAITS FOR INPUT - GETS IT OR IF IT IS
A C/R - DEFAULTS.
- COTB: 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.
- CHTB: 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 SDEVTB
MAKE THE FIRST DEVICE = 0 IN THIS TABLE
CLEAR DEVICE STATUS BUFFER TABLE & INPUT BUFFER

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

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

- REMEMBER MEMORY MANAGEMENT HAS BEEN TURNED ON-

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

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

EXEC: CLR SYSTEM FLAGS

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

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

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

RUN DX COMMAND

RUN: 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
SRORQ
SMINS)

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

CLR DXACT, CHDCHF, 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/R = CRUNCH DX, CONVERT AND PRINT CURRENT DEVICE #
IN HEX, PRINT 13 DX REGISTERS CONTENTS.
CLR ABORT FLAG (DXABFL), CLR DONE
RESET DX, SET GO, RETURN TO EXEC.

D = 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:))

I = 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

KILL DEVICE

KILL: 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)

PARITY ERROR

PPARER: 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

PCUEND: 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 PINOSC:
- * 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(OCUE=10) TO SCMD OF PROPER DEVICE
STATUS BUFFER TABLE)

-YOU WILL STAY IN THIS SECTION OF CODE UNTIL ALL TT ENTRIES
HAVE BEEN PROCESSED. WHEN THERE ARE NO MORE TT ENTRIES TO
PROCESSJUMP TO DXEXEC..

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

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

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

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

TYPICAL SELECTOR COMMANDS (FOR EACH DEVICE #)

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

TYPICAL MULTIPLEXER COMMANDS (FOR EACH DEVICE #)

WRITE FULL BUFFER	(SCMD = 1)
READ MANUAL INPUT	(SCMD = 2)
ENDING SEQUENCE	(SCMD = 3)
SENSE COMMAND	(SCMD = 4)
WRITE LINE ADDRESS	(SCMD = 5)
READ FULL BUFFER	(SCMD = 6)
ERASE COMMAND	(SCMD = 7)

CONTROL UNIT END (SCMD =10)
SEND ATTENTION (SCMD =11)
READ SHORT MANUAL INPUT (SCMD =12)

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

----- SELECTOR CHANNEL -----

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

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

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

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

WRITE LINE ADDRESS
WRITE FULL BUFFER

SMRITE: 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

CONLINE: 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 -----

MEX: IS COMMAND CHAINING SPECIFIED? (CMDCHF) IF YES, EXIT FROM
DXISR TO ALLOW PSM = 0 IF NO, PICK UP LAST DEVICE ADDRESS
THAT HAS A COMMAND EXECUTED - WAS 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 (SRDRQ=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 DXBC
CLR SSENSE
SAVE COMMAND (SLCMD <----- SCMD)
SET DEVICE ACTIVE (DXACT)
SET INPUT FUNCTION & GO IN DXCS
EXIT FROM DXISR AND WAIT FOR NEW TT ENTRIES

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

SRBYTC IS DECREASED BY THE FOLLOWING 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 TT ENTRY (PCUEND:)
SRBYTC WILL BE DECREMENTED BY 4 AND SBUFA WILL BE
INCREMENTED BY 4-- THIS APPLIES TO ALL THE SELECTOR OR
MULTIPLEXER READ OR WRITES IF THE DX HARDWARE IS
FUNCTIONING CORRECTLY.

---MISCELLANEOUS ROUTINES ---

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

DISPLAY CONTROL ROUTINE

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

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

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

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

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

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

DX ABORT

DXAB:

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

-----DEVICE STATUS TABLE FLAGS-----

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

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

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

SCMD (0)

```

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

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

HOW USED

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

SET UP IN

TISSRJ:
 PINDSC:
 PVISS:

USED IN

PESEND:
 SEX: & MEX:
 SWRITE: & MWRITE:

CLEARED IN

CDEVST: (RUN:, ENABLE:,
 KILL:, PSYSRT:, PINDSC:,
 PSEND:)

PCHEND: SSRMI: & MSRMI:
ERASCH:

SSENSE (2)

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

HOW USED

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

SET UP IN

PHIS:
PCHEND:

USED IN

SENSCH:

CLEARED IN

RUN:
ENABLE:
PSYSRT:
PSELRT:
SMRITE:
SREAD:
ERASCH:
MWRITE:
MREAD:

SSTAT (3)

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

HOW USED

USED BY 360 WHEN REQUESTING STATUS ;WITH EXECEPTION
OF THE ASYNCHRONOUS PRESENTING OF STATUS (ATTN) TO
THE 360.

IE. STOUT: MOV SSTAT,DXOS

SET UP IN

PPARER:
ESEQ:
CONLINE:
SATTN:

USED IN

STOUT:

CLEARED IN

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: (INT140:)

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)
ERASCH: USED TO CLEAR OUT BUFFER AREA (WITH EBCDIC SPACE = 100)

SET UP IN

INIT: (INT140:)

USED IN

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

SBUFA (12)

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

HOW USED

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

SET UP IN

PCUEND:
MWRITE:
MREAD:
MSRMI:

USED IN

MWRITE:
MREAD:
MSRMI:

CLEARED IN

CDEVST:(RUN:
ENABLE:,KILL:,PSYSRT:
PINDSC:,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:

SRRQ (17)

SRRQ <---- 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 UNSOLICITATED
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:

M03

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB PROGRAM DESCRIPTION

MACY11 27(732) 27-OCT-76 14:48 PAGE 65-6

7499
7500
7501

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

PROGRAM EQUATES AND DEVICE ASSIGNMENTS

.SBTTL PROGRAM EQUATES AND DEVICE ASSIGNMENTS

SYSTEM EQUATES

7503		
7504		RO	= %0
7505		R1	= %1
7506		R2	= %2
7507	000000	R3	= %3
7508	000001	R4	= %4
7509	000002	R5	= %5
7510	000003	R6	= %6
7511	000004	SP	= %6
7512	000005	PC	= %7
7513	000006	PSW	= 177776
7514	000006	KISAR0	= 172340
7515	000007	KISAR7	= 172356
7516	177776	KISDR0	= 172300
7517	172340	MMSR0	= 177572
7518	172356		
7519	172300		
7520	177572		

TELETYPE CHARACTER EQUATES

7521		
7522		CR	= 15 ; CARRIAGE RETURN
7523		LF	= 12 ; LINE FEED
7524		SPACE	= 40 ; SPACE CHARACTER
7525		CTL.C	= 3 ; CONTROL C
7526	000015	CTL.P	= 20 ; CONTROL P
7527	000012	CTL.Q	= 21 ; CONTROL Q
7528	000040	CTL.S	= 23 ; CONTROL S
7529	000003	CTL.U	= 25 ; CONTROL U
7530	000020	RUBOUT	= 177 ; RUBOUT
7531	000021		
7532	000023		
7533	000025		
7534	000177		


```

7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591

```

```

*****
*
*      DEVICE BUFFER LAYOUT      (1 PER DEVICE)
*
*****
*      LOC      0-61 = DEVICE STATUS TABLE
*      LOC      62-543 = DEVICE INPUT BUFFER
*      LOC      554-1023 = DEVICE OUTPUT/DISPLAY BUFFER
*****
      LAYOUT OF DEVICE STATUS TABLE
SCMD      =      0      ; CURRENT DEVICE COMMAND
SLCMD     =      1      ; LAST COMMAND, IF WRITE
SSENSE    =      2      ; DEVICE SENSE BYTE (NOTE -- MUST BE EVEN BYTE LOCATION)
SSTAT     =      3      ; DEVICE STATUS
SCURS     =      4      ; CURSOR POSITION
SINBF     =      6      ; ADDRESS OF DEVICE INPUT BUFFER
SOUTB     =      10     ; ADDRESS OF DEVICE OUTPUT BUFFER
SBUFA     =      12     ; CURRENT BUFFER PTR
SRBYTC    =      14     ; REMAINING BYTE COUNT
SONLF     =      16     ; DEVICE ONLINE - INDICATOR 0=ON-LINE 1=OFF-LINE
SRDRQ     =      17     ; READ MANUAL INPUT REQUEST -- IF NON-ZERO
SMINS     =      20     ; START OF MANUAL INPUT DATA
:
:      LOCATIONS 22-77 ARE AVAILABLE FOR EXPANSION PURPOSES
:
:
:      OTHER      DX EQUATES
:
DEV        =      R0      ; CURRENT DEVICE NUMBER
DTAB       =      R3      ; ADDRESS OF CUR DEV STATUS TABLE
TT1        =      R4      ; TUMBLE TABLE ENTRY 1
TT2        =      R5      ; TUMBLE TABLE ENTRY 2
CEDE       =      3      ; CHAN END & DEV END
OCUE       =      10     ; CODE TO QUE CONTROL UNIT END
NOP        =      3      ; NOP COMMAND
NEWLINE    =      25     ; NEW LINE CHARACTER
EBCDSP     =      100    ; EBCDIC SPACE CODE
SMI        =      112    ; START OF MESSAGE INDICATOR
EOM        =      152    ; END OF MESSAGE INDICATOR
LINSZ      =      40     ; NUMBER OF CHARACTERS PER LINE OF 2260 DISPLAY
NOLIN     =      12     ; NUMBER OF LINES PER 2260 DISPLAY
DISPSZ     =      NOLIN*LINSZ ; NUMBER OF CHARACTERS ON THE DISPLAY
TTSIZE     =      512    ; NUMBER OF ENTRIES IN TUMBLE TABLE
:
:      360 COMMAND EQUATES
:
CMWRT      =      1      ; WRITE DATA (FROM 360 TO PDP-11)
CMRMI      =      2      ; READ MANUAL INPUT (PDP-11 TO 360)
CMWTLA     =      5      ; WRITE LINE ADDRESS (360 TO PDP-11)
CMREAD     =      6      ; READ FULL BUFFER (PDP-11 TO 360)
CMSAMI     =      12     ; SHORT READ MANUAL INPUT (PDP-11 TO 360)

```

```

7593
7594
7595
7596
7597
7598      100000
7599      040000
7600      020000
7601      010000
7602      004000
7603      002000
7604      001000
7605      000400
7606      000200
7607      000100
7608      000040
7609      000020
7610      000010
7611      000004
7612      000002
7613      000001
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625      004000
7626      001000
7627      000400
7628      000200
7629      000100
7630
7631
7632
7633
7634      000001
7635      000003
7636      000005
7637      000007
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647      000200
7648      000100
    
```

DX REGISTER ASSIGNMENTS & LAYOUTS

DXDS OR TUMBLE TABLE ENTRY 1 (TT1)

```

PARER = 100000 ; PARITY ERROR DETECTED
NXM   = 40000  ; NON EXISTENT MEMORY CONDITION
SELRST = 20000 ; IBM SELECTIVE RE-SET
SYSRST = 10000 ; IBM SYSTEM RESET
INFDSC = 4000  ; IBM PROGRAMMED INTERFACE DISCONNECT
UCHKS  = 2000  ; UNIT CHECK WAS PRESENTED TO THE CHANNEL
CHENDS = 1000  ; CHANNEL END WAS PRESENTED TO THE CHANNEL
BYSS   = 400   ; BUSY WAS PRESENTED TO THE CHANNEL
CHIS   = 200   ; CHANNEL INIT SELECTION SEQ WAS COMPLETED
ESEND  = 100   ; CHANNEL ACCEPTED LAST STATUS
CHEND  = 40    ; CHANNEL DATA TRANSFER END
CUEND  = 20    ; DX DATA TRANSFER END
ISSREJ = 10    ; INIT SELECTION SEQ WAS REJECTED
CMDCHN = 4     ; CHANNEL SPECIFIED COMMAND CHAING
STKSTB = 2     ; CHANNEL COULD NOT ACCEPT LAST STATUS
CMDREJ = 1     ; CHANNEL COMMAND WAS REJECTED
    
```

DXCA OR TUMBLE TABLE ENTRY 2 (TT2)

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

DXCS CONTROL UNIT STATUS REGISTER

```

BSYEN = 4000 ; BUSY RENABLE - FOR SELECTOR CHANNELS
DXONLN = 1000 ; ON-LINE INDICATION
CUBUSY = 400  ; CONTROL UNIT BUSY
DONE   = 200  ; DONE FLAG
DXENB  = 100  ; INTERRUPT ENABLE
    
```

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

```

DXRST = 1 ; DX RESET COMMAND
DXWR  = 3 ; WRITE DATA TO THE 360
DXRD  = 5 ; READ DATA FROM THE 360
DXST  = 7 ; SEND STATUS TO THE 360
    
```

DXOS OFFSET AND STATUS REGISTER

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

```

ATTN = 200 ; ATTENTION
STAMOD = 100 ; STATUS MODIFIER
    
```

7649	000040	CUE	=	40	:CONTROL UNIT END
7650	000020	BSY	=	20	:BUSY
7651	000010	CE	=	10	:CHANNEL END
7652	000004	DE	=	4	:DEVICE END
7653	000002	UCHK	=	2	:UNIT CHECK
7654	000001	UEXP	=	1	:UNIT EXCEPTION
7655		:			
7656		:			
7657		:			
7658		:			
7659	000200	SCHDRJ	=	200	:COMMAND REJECT
7660	000100	INTREQ	=	100	:DEVICE OFF-LINE - INTERVENTION REQ
7661	000040	BUSOUT	=	40	:BUS OUT -- PARITY ERROR DURING CHIS
7662	000020	EQCHK	=	20	:EQUIPMENT CHECK - PARITY ERROR DUR DATA TRANS

2848 SENSE BYTE (SSENSE) DEFINITION

(1)				
(1)	000104	000106	.WORD	.+2
(1)	000106	000000	HALT	
(1)				
(1)	000110	000112	.WORD	.+2
(1)	000112	000000	HALT	
(1)				
(1)	000114	000116	.WORD	.+2
(1)	000116	000000	HALT	
(1)				
(1)	000120	000122	.WORD	.+2
(1)	000122	000000	HALT	
(1)				
(1)	000124	000126	.WORD	.+2
(1)	000126	000000	HALT	
(1)				
(1)	000130	000132	.WORD	.+2
(1)	000132	000000	HALT	
(1)				
(1)	000134	000136	.WORD	.+2
(1)	000136	000000	HALT	
(1)				
(1)	000140	000142	.WORD	.+2
(1)	000142	000000	HALT	
(1)				
(1)	000144	000146	.WORD	.+2
(1)	000146	000000	HALT	
(1)				
(1)	000150	000152	.WORD	.+2
(1)	000152	000000	HALT	
(1)				
(1)	000154	000156	.WORD	.+2
(1)	000156	000000	HALT	
(1)				
(1)	000160	000162	.WORD	.+2
(1)	000162	000000	HALT	
(1)				
(1)	000164	000166	.WORD	.+2
(1)	000166	000000	HALT	
(1)				
(1)	000170	000172	.WORD	.+2
(1)	000172	000000	HALT	
(1)				
(1)	000174	000176	.WORD	.+2
(1)	000176	000000	HALT	
(1)				
(1)	000200	000202	.WORD	.+2
(1)	000202	000000	HALT	
(1)				
(1)	000204	000206	.WORD	.+2
(1)	000206	000000	HALT	
(1)				
(1)	000210	000212	.WORD	.+2
(1)	000212	000000	HALT	
(1)				
(1)	000214	000216	.WORD	.+2

(1)	000216	000000	HALT	
(1)				
(1)	000220	000222	.WORD	.+2
(1)	000222	000000	HALT	
(1)				
(1)	000224	000226	.WORD	.+2
(1)	000226	000000	HALT	
(1)				
(1)	000230	000232	.WORD	.+2
(1)	000232	000000	HALT	
(1)				
(1)	000234	000236	.WORD	.+2
(1)	000236	000000	HALT	
(1)				
(1)	000240	000242	.WORD	.+2
(1)	000242	000000	HALT	
(1)				
(1)	000244	000246	.WORD	.+2
(1)	000246	000000	HALT	
(1)				
(1)	000250	000252	.WORD	.+2
(1)	000252	000000	HALT	
(1)				
(1)	000254	000256	.WORD	.+2
(1)	000256	000000	HALT	
(1)				
(1)	000260	000262	.WORD	.+2
(1)	000262	000000	HALT	
(1)				
(1)	000264	000266	.WORD	.+2
(1)	000266	000000	HALT	
(1)				
(1)	000270	000272	.WORD	.+2
(1)	000272	000000	HALT	
(1)				
(1)	000274	000276	.WORD	.+2
(1)	000276	000000	HALT	
(1)				
(1)	000300	000302	.WORD	.+2
(1)	000302	000000	HALT	
(1)				
(1)	000304	000306	.WORD	.+2
(1)	000306	000000	HALT	
(1)				
(1)	000310	000312	.WORD	.+2
(1)	000312	000000	HALT	
(1)				
(1)	000314	000316	.WORD	.+2
(1)	000316	000000	HALT	
(1)				
(1)	000320	000322	.WORD	.+2
(1)	000322	000000	HALT	
(1)				
(1)	000324	000326	.WORD	.+2
(1)	000326	000000	HALT	
(1)				

(1)	000330	000332	.WORD	.+2
(1)	000332	000000	HALT	
(1)				
(1)	000334	000336	.WORD	.+2
(1)	000336	000000	HALT	
(1)				
(1)	000340	000342	.WORD	.+2
(1)	000342	000000	HALT	
(1)				
(1)	000344	000346	.WORD	.+2
(1)	000346	000000	HALT	
(1)				
(1)	000350	000352	.WORD	.+2
(1)	000352	000000	HALT	
(1)				
(1)	000354	000356	.WORD	.+2
(1)	000356	000000	HALT	
(1)				
(1)	000360	000362	.WORD	.+2
(1)	000362	000000	HALT	
(1)				
(1)	000364	000366	.WORD	.+2
(1)	000366	000000	HALT	
(1)				
(1)	000370	000372	.WORD	.+2
(1)	000372	000000	HALT	
(1)				
(1)	000374	000376	.WORD	.+2
(1)	000376	000000	HALT	
(1)				
(1)	000400	000402	.WORD	.+2
(1)	000402	000000	HALT	
(1)				
(1)	000404	000406	.WORD	.+2
(1)	000406	000000	HALT	
(1)				
(1)	000410	000412	.WORD	.+2
(1)	000412	000000	HALT	
(1)				
(1)	000414	000416	.WORD	.+2
(1)	000416	000000	HALT	
(1)				
(1)	000420	000422	.WORD	.+2
(1)	000422	000000	HALT	
(1)				
(1)	000424	000426	.WORD	.+2
(1)	000426	000000	HALT	
(1)				
(1)	000430	000432	.WORD	.+2
(1)	000432	000000	HALT	
(1)				
(1)	000434	000436	.WORD	.+2
(1)	000436	000000	HALT	
(1)				
(1)	000440	000442	.WORD	.+2
(1)	000442	000000	HALT	

(1)				
(1)	000444	000446	.WORD	.+2
(1)	000446	000000	HALT	
(1)				
(1)	000450	000452	.WORD	.+2
(1)	000452	000000	HALT	
(1)				
(1)	000454	000456	.WORD	.+2
(1)	000456	000000	HALT	
(1)				
(1)	000460	000462	.WORD	.+2
(1)	000462	000000	HALT	
(1)				
(1)	000464	000466	.WORD	.+2
(1)	000466	000000	HALT	
(1)				
(1)	000470	000472	.WORD	.+2
(1)	000472	000000	HALT	
(1)				
(1)	000474	000476	.WORD	.+2
(1)	000476	000000	HALT	
(1)				
(1)	000500	000502	.WORD	.+2
(1)	000502	000000	HALT	
(1)				
(1)	000504	000506	.WORD	.+2
(1)	000506	000000	HALT	
(1)				
(1)	000510	000512	.WORD	.+2
(1)	000512	000000	HALT	
(1)				
(1)	000514	000516	.WORD	.+2
(1)	000516	000000	HALT	
(1)				
(1)	000520	000522	.WORD	.+2
(1)	000522	000000	HALT	
(1)				
(1)	000524	000526	.WORD	.+2
(1)	000526	000000	HALT	
(1)				
(1)	000530	000532	.WORD	.+2
(1)	000532	000000	HALT	
(1)				
(1)	000534	000536	.WORD	.+2
(1)	000536	000000	HALT	
(1)				
(1)	000540	000542	.WORD	.+2
(1)	000542	000000	HALT	
(1)				
(1)	000544	000546	.WORD	.+2
(1)	000546	000000	HALT	
(1)				
(1)	000550	000552	.WORD	.+2
(1)	000552	000000	HALT	
(1)				
(1)	000554	000556	.WORD	.+2

(1)	000556	000000	HALT	
(1)				
(1)	000560	000562	.WORD	.+2
(1)	000562	000000	HALT	
(1)				
(1)	000564	000566	.WORD	.+2
(1)	000566	000000	HALT	
(1)				
(1)	000570	000572	.WORD	.+2
(1)	000572	000000	HALT	
(1)				
(1)	000574	000576	.WORD	.+2
(1)	000576	000000	HALT	
(1)				
(1)	000600	000602	.WORD	.+2
(1)	000602	000000	HALT	
(1)				
(1)	000604	000606	.WORD	.+2
(1)	000606	000000	HALT	
(1)				
(1)	000610	000612	.WORD	.+2
(1)	000612	000000	HALT	
(1)				
(1)	000614	000616	.WORD	.+2
(1)	000616	000000	HALT	
(1)				
(1)	000620	000622	.WORD	.+2
(1)	000622	000000	HALT	
(1)				
(1)	000624	000626	.WORD	.+2
(1)	000626	000000	HALT	
(1)				
(1)	000630	000632	.WORD	.+2
(1)	000632	000000	HALT	
(1)				
(1)	000634	000636	.WORD	.+2
(1)	000636	000000	HALT	
(1)				
(1)	000640	000642	.WORD	.+2
(1)	000642	000000	HALT	
(1)				
(1)	000644	000646	.WORD	.+2
(1)	000646	000000	HALT	
(1)				
(1)	000650	000652	.WORD	.+2
(1)	000652	000000	HALT	
(1)				
(1)	000654	000656	.WORD	.+2
(1)	000656	000000	HALT	
(1)				
(1)	000660	000662	.WORD	.+2
(1)	000662	000000	HALT	
(1)				
(1)	000664	000666	.WORD	.+2
(1)	000666	000000	HALT	
(1)				

K04

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB TRAP CATCHER

MACY11 27(732) 27-OCT-76 14:48 PAGE 65-17

(1)	000670	000672	.WORD	.+2
(1)	000672	000000	HALT	
(1)				
(1)	000674	000676	.WORD	.+2
(1)	000676	000000	HALT	
(1)				
(1)	000700	000702	.WORD	.+2
(1)	000702	000000	HALT	
(1)				
(1)	000704	000706	.WORD	.+2
(1)	000706	000000	HALT	
(1)				
(1)	000710	000712	.WORD	.+2
(1)	000712	000000	HALT	
(1)				
(1)	000714	000716	.WORD	.+2
(1)	000716	000000	HALT	
(1)				
(1)	000720	000722	.WORD	.+2
(1)	000722	000000	HALT	
(1)				
(1)	000724	000726	.WORD	.+2
(1)	000726	000000	HALT	
(1)				
(1)	000730	000732	.WORD	.+2
(1)	000732	000000	HALT	
(1)				
(1)	000734	000736	.WORD	.+2
(1)	000736	000000	HALT	
(1)				
(1)	000740	000742	.WORD	.+2
(1)	000742	000000	HALT	
(1)				
(1)	000744	000746	.WORD	.+2
(1)	000746	000000	HALT	
(1)				
(1)	000750	000752	.WORD	.+2
(1)	000752	000000	HALT	
(1)				
(1)	000754	000756	.WORD	.+2
(1)	000756	000000	HALT	
(1)				
(1)	000760	000762	.WORD	.+2
(1)	000762	000000	HALT	
(1)				
(1)	000764	000766	.WORD	.+2
(1)	000766	000000	HALT	
(1)				
(1)	000770	000772	.WORD	.+2
(1)	000772	000000	HALT	
(1)				
(1)	000774	000776	.WORD	.+2
(1)	000776	000000	HALT	
(1)				
(1)	001000	001002	.WORD	.+2
(1)	001002	000000	HALT	

(1)				
(1)	001004	001006	.WORD	.+2
(1)	001006	000000	HALT	
(1)				
(1)	001010	001012	.WORD	.+2
(1)	001012	000000	HALT	
(1)				
(1)	001014	001016	.WORD	.+2
(1)	001016	000000	HALT	
(1)				
(1)	001020	001022	.WORD	.+2
(1)	001022	000000	HALT	
(1)				
(1)	001024	001026	.WORD	.+2
(1)	001026	000000	HALT	
(1)				
(1)	001030	001032	.WORD	.+2
(1)	001032	000000	HALT	
(1)				
(1)	001034	001036	.WORD	.+2
(1)	001036	000000	HALT	
(1)				
(1)	001040	001042	.WORD	.+2
(1)	001042	000000	HALT	
(1)				
(1)	001044	001046	.WORD	.+2
(1)	001046	000000	HALT	
(1)				
(1)	001050	001052	.WORD	.+2
(1)	001052	000000	HALT	
(1)				
(1)	001054	001056	.WORD	.+2
(1)	001056	000000	HALT	
(1)				
(1)	001060	001062	.WORD	.+2
(1)	001062	000000	HALT	
(1)				
(1)	001064	001066	.WORD	.+2
(1)	001066	000000	HALT	
(1)				
(1)	001070	001072	.WORD	.+2
(1)	001072	000000	HALT	
(1)				
(1)	001074	001076	.WORD	.+2
(1)	001076	000000	HALT	
(1)				
(1)	001100	001102	.WORD	.+2
(1)	001102	000000	HALT	
(1)				
(1)	001104	001106	.WORD	.+2
(1)	001106	000000	HALT	
(1)				
(1)	001110	001112	.WORD	.+2
(1)	001112	000000	HALT	
(1)				
(1)	001114	001116	.WORD	.+2

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB TRAP CATCHER

(1)	001116	000000	HALT	
(1)				
(1)	001120	001122	.WORD	.+2
(1)	001122	000000	HALT	
(1)				
(1)	001124	001126	.WORD	.+2
(1)	001126	000000	HALT	
(1)				
(1)	001130	001132	.WORD	.+2
(1)	001132	000000	HALT	
(1)				
(1)	001134	001136	.WORD	.+2
(1)	001136	000000	HALT	
(1)				
(1)	001140	001142	.WORD	.+2
(1)	001142	000000	HALT	
(1)				
(1)	001144	001146	.WORD	.+2
(1)	001146	000000	HALT	
(1)				
(1)	001150	001152	.WORD	.+2
(1)	001152	000000	HALT	
(1)				
(1)	001154	001156	.WORD	.+2
(1)	001156	000000	HALT	
(1)				
(1)	001160	001162	.WORD	.+2
(1)	001162	000000	HALT	
(1)				
(1)	001164	001166	.WORD	.+2
(1)	001166	000000	HALT	
(1)				
(1)	001170	001172	.WORD	.+2
(1)	001172	000000	HALT	
(1)				
(1)	001174	001176	.WORD	.+2
(1)	001176	000000	HALT	
(1)				
(1)	001200	001202	.WORD	.+2
(1)	001202	000000	HALT	
(1)				
(1)	001204	001206	.WORD	.+2
(1)	001206	000000	HALT	
(1)				
(1)	001210	001212	.WORD	.+2
(1)	001212	000000	HALT	
(1)				
(1)	001214	001216	.WORD	.+2
(1)	001216	000000	HALT	
(1)				
(1)	001220	001222	.WORD	.+2
(1)	001222	000000	HALT	
(1)				
(1)	001224	001226	.WORD	.+2
(1)	001226	000000	HALT	
(1)				

N04

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB TRAP CATCHER

MACY11 27(732) 27-OCT-76 14:48 PAGE 65-20

(1)	001230	001232	.WORD	.+2
(1)	001232	000000	HALT	
(1)				
(1)	001234	001236	.WORD	.+2
(1)	001236	000000	HALT	
(1)				
(1)	001240	001242	.WORD	.+2
(1)	001242	000000	HALT	
(1)				
(1)	001244	001246	.WORD	.+2
(1)	001246	000000	HALT	
(1)				
(1)	001250	001252	.WORD	.+2
(1)	001252	000000	HALT	
(1)				
(1)	001254	001256	.WORD	.+2
(1)	001256	000000	HALT	
(1)				
(1)	001260	001262	.WORD	.+2
(1)	001262	000000	HALT	
(1)				
(1)	001264	001266	.WORD	.+2
(1)	001266	000000	HALT	
7677		000200	.=200	
7678	000200	000137	JMP	START
7679				

;ESTABLISH LOC 200 STARTING ADDRESS

```

7681
7682
7683
7684
7685      001000      001000
7686      001000      000402
7687
7688
7689
7690      001002      005037      013122
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701      001006      012706      012626
7702      001012      000005
7703
7704
7705
7706
7707
7708
7709      001014      012700      000060
7710      001020      012720      010732
7711      001024      012720      000340
7712      001030      012720      011220
7713      001034      012710      000340
7714
7715
7716
7717      001040      012737      011666      000004
7718      001046      012737      000340      000006
7719      001054      012737      011674      000250
7720      001062      012737      000340      000252
7721
7722
7723
7724      001070      012700      012626
7725      001074      012701      000272
7726      001100      105020
7727      001102      005301
7728      001104      001375
7729
7730
7731
7732      001106      012737      012630      012732
7733      001114      012737      012734      013036
7734      001122      012737      012734      013040
7735
7736

```

```

.SBTTL PROGRAM START-UP SEQUENCES
NORMAL SYSTEM START LOCATION --1000
      =1000
START: BR      SYSINT      ;NORMAL START UP
RESTART ADDRESS -- REENTER ALL PARAMETERS -- 1002
RSTART: CLR      FTIMFL      ;RESET FIRST TIME FLAG TO FORCE PARAMETER REENTRY
SYSTEM GENERALIZED INITIZATION PROCEEDURE
      SET-UP STACK POINTER
      TRAP/VECTOR AREA
      SYSTEM GENERATED TRAPS
      TELETYPE (CONSOLE) VECTORS + STATUS REGISTERS
      CLEAR ALL LIVE SYSTEM VARIABLES
      SET UP TELETYPE INPUT / OUTPUT BUFFERS
SYSINT: MOV      #SSTACK,SP      ;SET UP THE STACK POINTER
      RESET      ;RESET ALL DEVICES

      SET UP CONSOLE VECTORS
MOV      #60,R0      ;START OF CONSOLE VECTORS
MOV      #TKIN,(R0)+ ;BEG OF TELE INPUT ISR
MOV      #340,(R0)+ ;NEW PROC STATUS
MOV      #PISR,(R0)+ ;BEG OF TELE PRINT ISR
MOV      #340,(R0)  ;NEW PROC STATUS

      SET UP MISC TRAPS
MOV      #MTO,4      ;MEMORY TIME OUT TRAP
MOV      #340,6
MOV      #MMERR,250 ;MEMORY MANAGEMENT ERROR
MOV      #340,252

      CLEAR ALL VARIABLES
MOV      #VSTRT,R0   ;START OF VARIABLES
MOV      #VEND-VSTRT+2,R1 ;# OF VARIABLES TO CLEAR
IOS: CLR      (R0)+ ;CLEAR A BYTE
      DEC      R1      ;DONE?
      BNE      IOS     ;NO, CONTINUE CLEARING

      SET UP TELE BUFFER POINTERS
MOV      #TBUF,TPTR  ;TELE INPUT POINTER
MOV      #PBFS,PFPT  ;TELE OUT FETCH PTR
MOV      #PBFS,PPPTR  ;TELE OUT PUT PTR

      ENABLE TTY

```

7737					
7738	001130	052777	000100	011306	;
7739	001136	005037	177776		
7740	001142	005737	013122		
7741	001146	001402			
7742	001150	000137	002014		

BIS	#100, #TKS
CLR	PSW
TST	FTIMFL
BEQ	GETPRM
JMP	INIT

```

;ENABLE TELETYPE INPUT
;CLEAR THE PROCESSOR STATUS WORD
;FIRST TIME THROUGH? (MUST PARAMETERS BE REENTERED?)
;YES, FORCE USER TO ENTER ALL PARAMETERS
;NO, RESTART TEST USING SAME PARAMETERS

```

```

7744 .SBTTL TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)
7745
7746     .:
7747     HERE WE START GATHERING THE TEST INFORMATION
7748     PRINT START-UP HERALD MESSAGE
7749 001154 004137 011466 GETPRM: JSR R1,MESG ;PRINT START-UP MESG
7750 001160 013124 .WORD STMSG
7751
7752     .:
7753     GET DX11 UNIBUSS ADDRESS (OCTAL ADDRESS INPUT)
7754     VALID UNIBUS ADDRESSES (176200 - 177000)
7755     DEFAULT UNIBUS ADDRESS 176200
7756
7757 001162 005037 013122 NEWPRM: CLR FTIMFL ;RESET FIRST TIME PARAMETERS (FORCE ALL PARMS TO BE ENTE
7758 001166 012737 176200 012506 MOV #176200,UNADDR ;SET UP DEFAULT ADDRESS
7759 001174 004137 002620 JSR R1,INOCT ;GET UNIBUS ADDRESS
7760 001200 013165 .WORD UNMSG
7761 001202 001246 .WORD SS ;ADDRESS OF DEFAULT ROUTINE
7762 001204 120427 000015 CMPB R4,#CR ;WAS LINE DELIMITED PROPERLY?
7763 001210 001364 BNE NEWPRM ;NO, TELL HIM TO REENTER
7764 001212 020327 176200 CMP R3,#176200 ;VALID UNIBUS ADDRESS? BETWEEN 176200 AND 177000
7765 001216 002761 BLT NEWPRM ;NO, GET AGAIN
7766 001220 020327 177000 CMP R3,#177000 ;UNIBUS ADDRESS GT 177000?
7767 001224 003356 BGT NEWPRM ;YES, ERROR -- REENTER
7768 001226 032703 000037 BIT #37,R3 ;MAKE SURE 40 OCTAL WORD BOUNDARY
7769 001232 001353 BNE NEWPRM ;ILLEGAL, REENTER
7770 001234 012737 001162 000004 MOV #NEWPRM,4 ;SET UP TRAP OUT TO VALIDATE ADDRESS
7771 001242 010337 012506 MOV R3,UNADDR ;SAVE UNIBUS ADDRESS
7772 001246 005077 011234 SS: CLR UNADDR ;VALIDATE THE UNIBUS ADDRESS
7773 ;TRAP WILL OCCUR IF INVALID UNIBUS ADDRESS
7774
7775     .:
7776     GET THE DX11 INTERRUPT VECTOR ADDRESS (OCTAL ADDRESS INPUT)
7777     VALID VECTOR ADDRESSES (300 - 770)
7778     DEFAULT VECTOR ADDRESS 300
7779
7780 001252 012737 011732 000004 10s: MOV #UNTRP,4 ;RESTORE MEMORY TIME-OUT TRAP
7781 001260 012737 000300 012510 MOV #300,VECTAD ;SET UP DEFAULT VECTOR ADDRESS
7782 001266 004137 002620 JSR R1,INOCT ;GET VECTOR ADDRESS
7783 001272 013222 .WORD VECTMS
7784 001274 001332 .WORD 20$ ;ADDRESS OF THE DEFAULT ENTRY
7785 001276 120427 000015 CMPB R4,#CR ;WAS LINE DELIMITED PROPERLY?
7786 001302 001363 BNE 10$ ;NO, REENTER
7787 001304 020327 000300 CMP R3,#300 ;CHECK VECTOR ADDRESS BETWEEN 300 AND 770
7788 001310 002760 BLT 10$ ;TOO LOW GIVE AN ERROR AND REENTER
7789 001312 020327 000770 CMP R3,#770 ;LT 770?
7790 001316 003355 BGT 10$ ;YES, REENTER
7791 001320 032703 000001 BIT #1,R3 ;WORD ADDRESS?
7792 001324 001352 BNE 10$ ;NO, REENTER
7793 001326 010337 012510 MOV R3,VECTAD ;SAVE IT
7794
7795     .:
7796     GET STARTING AND ENDING DEVICE CHANNEL ADDRESSES (HEX INPUT)
7797     VALID DEVICE CHANNEL ADDRESSES (00 - FF)
7798     DEFAULT DEVICE CHANNEL ADDRESS 10,10
7799

```


E05

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB

MACY11 27(732) 27-OCT-76 14:48 PAGE 65-24
TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)

```
7800      :      NOTE:  EITHER ONE OR TWO ADDRESSES MAY BE SPECIFIED. IF ONLY ONE
7801      :      ADDRESS IS SPECIFIED IT IS USED AS THE STARTING AND
7802      :      ENDING DEVICE ADDRESS.
7803      :
7804 001332 012737 000020 012512 20$:  MOV    #20,SDEV      ;DEFAULT TO HEX ADDRESS 10
7805 001340 012737 000020 012514      MOV    #20,EDEV      ;
7806 001346 004137 002612      JSR    R1,INHEX      ;GET DEVICE ADDRESSES IN HEX
7807 001352 013270      .WORD  DEVMS         ;
7808 001354 001446      .WORD  NEWP10        ;ADDRESS OF THE DEFAULT ROUTINE
7809 001356 010337 012512      MOV    R3,SDEV      ;SAVE START DEV ADDR
7810 001362 010337 012514      MOV    R3,EDEV      ;
7811 001366 005703      TST    R3            ;BE SURE POSITIVE
7812 001370 100760      BMI    20$          ;
7813 001372 020327 000377      CMP    R3,#377      ;AND NOT GREATER THAN 377 -- HEX FF
7814 001376 003355      BGT    20$          ;ILLEGAL ENTRY
7815 001400 120427 000054      CMPB   R4,#' ,      ;MORE THAN ONE DEV? (COMMA, PARAMETER DELIMETER)
7816 001404 001015      BNE    30$          ;
7817 001406 004737 011600      JSR    PC,CHTB      ;GET ENDING DEVICE
7818 001412 010337 012514      MOV    R3,EDEV      ;SAVE ENDING ADDRESS
7819 001416 023737 012512 012514      CMP    SDEV,EDEV    ;IS START LT END?
7820 001424 003342      BGT    20$          ;YES, ERROR
7821 001426 163703 012512      SUB    SDEV,R3      ;MORE THAN 8 DEVICES?
7822 001432 020327 000007      CMP    R3,#7        ;
7823 001436 003335      BGT    20$          ;YES, ERROR
7824 001440 120427 000015 30$:  CMPB   R4,#CR      ;WAS DEVICE ADDRESSES DELIMITED PROPERLY?
7825 001444 001332      BNE    20$          ;NO, REENTER
7826      :
7827      :
7828      :      GET TYPE OF 360/370 CHANNEL
7829      :      M = MULTIPLEXER CHANNEL
7830      :      S = SELECTOR CHANNEL
7831      :      DEFAULT IS 'S', SELECTOR CHANNEL
7832      :
7833 001446 105037 012516  NEWP10: CLRB   CHTYPE      ;0 = M, 1 = S
7834 001452 004137 002620      JSR    R1,INOCT     ;GET CHANNEL TYPE
7835 001456 013334      .WORD  CHTYS        ;
7836 001460 001476      .WORD  50$          ;DEFAULT TO SELECTOR CHANNEL
7837 001462 120427 000115      CMPB   R4,#'M      ;M? -- MULTIPLEXER CHANNEL --
7838 001466 001414      BEQ    60$          ;YES, MULTIPLEXER CHANNEL
7839 001470 120427 000123      CMPB   R4,#'S      ;S? -- SELECTOR CHANNEL --
7840 001474 001364      BNE    NEWP10      ;NOT S OR M -- ERROR
7841 001476 105237 012516 50$:  INCB   CHTYPE      ;SELECTOR CHANNEL
7842 001502 000406      BR     60$          ;GET MEMORY MANAGEMENT FACILITIES
7843      :
7844      :
7845      :
7846      :      MEMORY MANAGEMENT TIME-OUT TRAP HANDLER
7847      :      USED ONLY FOR PARAMETER ENTRY PROCESS
7848      :
7849 001504 022626 55$:  CMP    (SP)+,(SP)+  ;DUMP PC AND PSW SAVED BY INTERRUPT
7850 001506 005037 177776      CLR    PSW          ;TURN DOWN PROCESSOR STATUS
7851 001512 004137 011466      JSR    R1,MESG      ;PRINT "NO MEM MANAGEMENT AVAIL"
7852 001516 014072      .WORD  PNOHM        ;
7853      :
7854      :
7855      :      ;ASK TO HAVE QUESTION REENTERED
```

F05

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB

MACY11 27(732) 27-OCT-76 14:48 PAGE 65-25
TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)

```

7856          : DETERMINE IF MEMORY MANAGEMENT IS TO BE USED
7857          : Y = YES, MEMORY MANAGEMENT TO BE USED
7858          : N = NO, DO NOT USE MEMORY MANAGEMENT
7859          : DEFAULT IS 'N', DO NOT USE MEMORY MANAGEMENT
7860
7861 001520 105037 012517          :
7862 001524 004137 002620          :
7863 001530 013366                :
7864 001532 001566                :
7865 001534 120427 000116          :
7866 001540 001412                :
7867 001542 120427 000131          :
7868 001546 001364                :
7869 001550 105237 012517          :
7870 001554 012737 001504 000004 :
7871 001562 005037 177572          :
7872          :
7873          :
7874          :
7875          :
7876          :
7877          :
7878          :
7879 001566 004137 002620          :
7880 001572 013426                :
7881 001574 001672                :
7882 001576 120427 000015          :
7883 001602 001371                :
7884 001604 032703 000001          :
7885 001610 001366                :
7886 001612 020327 000020          :
7887 001616 002763                :
7888 001620 005703                :
7889 001622 100761                :
7890 001624 020327 000734          :
7891 001630 002356                :
7892 001632 105737 012517          :
7893 001636 001412                :
7894 001640 010304                :
7895 001642 042704 000600          :
7896 001646 020427 000154          :
7897 001652 003407                :
7898 001654 004137 011466          :
7899          :
7900          :
7901          :
7902 001660 017044                :
7903 001662 000741                :
7904          :
7905 001664 020327 000134          :
7906 001670 002336                :
7907 001672 010337 012520          :
7908          :
7909          :
7910          :
7911          :

```

60\$: CLR# 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 MMSRO ; CHECK FOR MEMORY MANAGEMENT

70\$: JSR R1, INOCT ; GET BUFFER RELOC. IN ,000'S
.WORD BFRMS
.WORD NEWP20 ; ADDRESS OF DEFAULT ROUTINE
CMPB R4, #CR ; WAS LINE DELIMITED PROPERLY?
BNE 70\$; NO, REENTER
BIT #1, R3 ; MUST BE A MULTIPLE OF 2000
BNE 70\$
CMP R3, #20
BLT 70\$; ILLG BUFFER CONST -- LT 20000
TST R3 ; IS NUMBER NEGATIVE?
BMI 70\$; YES, REENTER ADDRESS
CMP R3, #734 ; IS ADDRESS TOO LARGE?
BGE 70\$; YES, REENTER ADDRESS
TSTB MMRESP ; WAS MEMORY MANAGEMENT SPECIFIED?
BEQ 71\$; NO, CHECK FOR 28K
MOV R3, R4 ; PUT VALUE IN WORK REG
BIC #600, R4 ; IGNORE ADDRESS EXTENSION BITS
CMP R4, #154 ; IS IT TOO CLOSE TO 200000 BOUNDARY?
BLE NEWP20 ; BRANCH IF OK
JSR R1, MMSG ; PRINT ERROR. CANNOT SET BUFFER SO
; CLOSE TO A 200000 BOUNDARY THAT A CARRY WOULD BE NEEDED TO CHANGE
; THE EXTENDED ADDRESS BITS. THE DX CANNOT WORK ACROSS 200000
; BOUNDARIES.
.WORD TOOC ; ADDRESS OF TOO CLOSE MESSAGE
BR 70\$; ASK FOR INPUT AGAIN

71\$: CMP R3, #134 ; NO, IS IT TOO CLOSE TO I/O PAGE?
BGE 70\$; YES, REENTER THE ADDRESS
NEWP20: MOV R3, BUFREL ; SAVE REL CONST

GET TYPE OF TEST TO BE RUN
D = 2848 RESPONDER DIAGNOSTIC

```

7912
7913
7914
7915
7916 001676 105037 012522
7917 001702 004137 002620
7918 001706 013524
7919 001710 001740
7920 001712 112737 000100 012524
7921 001720 120427 000104
7922 001724 001433
7923 001726 120427 000106
7924 001732 001361
7925 001734 105037 012524
7926 001740 105237 012522
7927
7928
7929
7930
7931
7932
7933
7934
7935 001744 105037 012523
7936 001750 004137 002620
7937 001754 013564
7938 001756 002014
7939 001760 120427 000116
7940 001764 001413
7941 001766 120427 000131
7942 001772 001364
7943 001774 105237 012523
7944
7945
7946
7947
7948
7949
7950 002000 004137 002612
7951 002004 013626
7952 002006 002014
7953 002010 110337 012524

```

F = FRIEND

DEFAULT = 'F', FRIEND

```

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

```

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

```

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

```

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

```

      JSR R1,INHEX ;FILL CHARACTER
      .WORD FILLMS
      .WORD INIT
      MOVB R3,FILLCH

```

.SBTTL PROGRAM INITIALIZATION
INITIALIZATION

SET UP ALL DX BUFFERS, MEMORY MANAGEMENT REGISTERS
AND DX REGISTERS

7955
7956
7957
7958
7959
7960
7961
7962
7963 002014 012737 011666 000004
7964 002022 013701 012506
7965 002026 012702 012454
7966 002032 012703 000015
7967 002036 010122
7968 002040 005721
7969 002042 005303
7970 002044 001374
7971
7972
7973
7974 002046 013701 012510
7975 002052 012721 005322
7976 002056 012711 000340
7977
7978
7979
7980
7981 002062 005737 012520
7982 002066 001003
7983 002070 012737 000020 012520
7984 002076 013737 012520 013070
7985 002104 013737 012520 013072
7986 002112 105737 012517
7987 002116 001436
7988
7989
7990
7991
7992
7993
7994
7995
7996 002120 012704 172340
7997 002124 012705 172300
7998 002130 005024
7999 002132 012725 077406
8000 002136 013703 013070
8001 002142 006303
8002 002144 006303
8003 002146 006303
8004 002150 010324
8005 002152 012725 077406
8006 002156 062703 000200
8007 002162 020427 172356
8008 002166 001370
8009 002170 012714 007600
8010 002174 012715 077406

INIT:
10\$:
20\$:
30\$:

```
INIT: MOV #MTO,4 ;SET UP MEMORY TIME OUT TRAP
      MOV UNADDR,R1 ;SET UP DX UNIBUS ADDRESSES
      MOV #DXDS,R2
      MOV #13,R3 ;13 ADDRESSES (REGISTERS)
10$: MOV R1,(R2)+ ;SET UP UNIBUS ADDRESS
      TST (R1)+ ;INCR TO NEXT DX REGISTER
      DEC R3 ;DONE?
      BNE 10$ ;NO, SET UP NEXT REGISTER

      SET UP DX VECTOR ADDRESS
      MOV VECTAD,R1
      MOV #DXISR,(R1)+ ;TRAP TO DX ISR
      MOV #340,(R1) ;SET UP PROC STATUS AT INTER.

      COMPUTE ADDRESSES OF DX BUFFERS
      CURRENTLY THIS INCLUDES DATA AREA, TUMBLE TABLE, AND SPW TABLE

      TST BUFREL ;WAS BUFFER RELOC SPECIFIED?
      BNE 20$ ;YES
      MOV #20,BUFREL ;NO, MAKE BUFFERS START AT 20000
20$: MOV BUFREL,PBUFA ;SAVE PHYSICAL ADDRESS
      MOV BUFREL,VBUFA ;SAVE VIRTUAL ADDRESS
      TST MMRESP ;WAS MEMORY MANAGEMENT SPECIFIED?
      BEQ 40$ ;NO, SET UP BUFFERS

      MEMORY MANAGEMENT WAS SPECIFIED
      SET UP KERNEL REGISTERS
      0-17777 = PROGRAM
      20000-157777 = BUFFERS (VIRTUAL ADDRESSES)
      160000-177777 = UNIBUS ADDRESSES
      ONLY I SPACE REGISTERS WILL BE USED

      MOV #KISAR0,R4 ;I-SPACE PAR
      MOV #KISDR0,R5 ;I-SPACE PDR
      CLR (R4)+ ;VA 0-17777 = PA 0-17777
      MOV #77406,(R5)+ ;64 BLOCKS, UNLIMITED ACCESS
      MOV PBUFA,R3 ;PHYSICAL ADDR * 2-6
      ASL R3
      ASL R3
      ASL R3
30$: MOV R3,(R4)+ ;SET UP PA FOR VA 20000-157777
      MOV #77406,(R5)+ ;64 BLOCKS, UNLIMITED ACCESS
      ADD #200,R3 ;INCREMENT TO NEXT 4K BANK
      CMP R4,#KISAR7 ;ALL BUFFER ADDRESSES SET UP?
      BNE 30$ ;NO, SET UP NEXT REGISTER
      MOV #7600,(R4) ;SET UP UNIBUS ADDRESS REGISTER
      MOV #77406,(R5) ;64 BLOCKS, UNLIMITED ACCESS
```

```

8011 002200 012737 000001 177572
8012 002206 012737 000020 013072
8013
8014
8015
8016
8017
8018
8019
8020
8021
8022
8023
8024
8025
8026
8027
8028 002214 013705 013070
8029 002220 000305
8030 002222 105005
8031 002224 006305
8032 002226 010577 010230
8033 002232 013701 013072
8034 002236 000301
8035 002240 105001
8036 002242 006301
8037 002244 010137 013104
8038 002250 010137 013074
8039 002254 160537 013074
8040 002260 062705 003000
8041 002264 010537 013106
8042 002270 005000
8043 002272 120037 012512
8044 002276 002405
8045 002301 120037 012514
8046 002304 003002
8047 002306 010521
8048 002310 000402
8049 002312 012721 000002
8050 002316 005200
8051 002320 020027 000400
8052 002324 001362
8053
8054
8055
8056 002326 010137 013052
8057 002332 010137 013050
8058 002336 012702 001000
8059 002342 005021
8060 002344 005302
8061 002346 001375
8062
8063
8064
8065
8066
    
```

```

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

```

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

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

```

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

NEXT SET UP TUMBLE TABLE AND DUPLICATE TUMBLE TABLE

```

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

SET UP DST TABLE

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

```

8067
8068
8069
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083 002350 012702 000013
8084 002354 012703 002576
8085 002360 112321
8086 002362 005302
8087 002364 001375
8088 002366 012702 000365
8089 002372 112721 000002
8090 002376 005302
8091 002400 001374
8092
8093
8094
8095 002402 013703 012514
8096 002406 163703 012512
8097 002412 005203
8098 002414 110337 013057
8099 002420 013737 012512 013100
8100 002426 005337 013100
8101 002432 012737 000001 013066
8102
8103
8104
8105
8106
8107
8108
8109
8110
8111 002440 010137 013054
8112 002444 005000
8113 002446 010103
8114 002450 0127L2 000420
8115 002454 005021
8116 002456 005302
8117 002460 001375
8118 002462 010363 000006
8119 002466 062763 000076 000006
8120 002474 010363 000010
8121 002500 062763 001040 000010
8122 002506 012702 000740

```

```

COMMAND 0-255. THE ENTRY IN THE DST TABLE IS
SENT TO THE 360.
THE FOLLOWING ARE A LIST OF VALID COMMANDS AND RESPONSES
COMMAND RESPONSE DESCRIPTION
0 0 TEST I/O
1 0 WRITE BUFFER
2 0 READ MANUAL INPUT
3 CE!DE NOP
4 0 SENSE COMMAND
5 0 WRITE LINE ADDRESS
6 0 READ FULL BUFFER
7 0 ERASE COMMAND
12 0 SHORT READ MANUAL INPUT

```

ALL OTHER COMMANDS ARE RESPONDED WITH UNIT CHECK

```

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

```

COMPUTE MAX NUMBER OF DEVICES

```

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

```

```

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

```

START SETTING UP DEVICE BUFFERS

```

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

```

```

8123 002512 113721 012524      125$: MOVB    FILLCH,(R1)+    ;FILL OUTPUT/DISPLAY BUFFER
8124 002516 005302              DEC     R2              ;DONE?
8125 002520 001374              BNE    125$            ;NO
8126 002522 005200              INC     DEV            ;HAVE ALL DEVICE BUFFERS BEEN SET UP?
8127 002524 120037 013057      CMPB   DEV,MAXDEV
8128 002530 001346              BNE    120$            ;NO, SET UP NEXT DEVICE BUFFERS
8129 002532 013705 013070      MOV    PBUFA,R5        ;SET UP EXTENDED ADDRESS BITS
8130 002536 006205              ASR    R5
8131 002540 006205              ASR    R5
8132 002542 006205              ASR    R5
8133 002544 006205              ASR    R5
8134 002546 042705 177747      BIC    #177747,R5      ;SAVE ONLY H.O. 2 BITS
8135 002552 010537 013102      MOV    R5,XADDR        ;SAVE EXTENDED ADDRESS BITS FOR DX CONTROL REG
8136 002556 012737 000001 013122  MOV    #1,FTIMFL       ;SET FIRST TIME THROUGH FLAG
8137
8138          :
8139          :      INITIALIZATION COMPLETE
8140          :      TELL OPERATOR WE ARE ALREADY TO GO
8141          :
8142          :      NOTE:  AT THIS POINT THE DX HAS NOT BEEN STARTED
8143          :      AND THE OPERATOR MUST TYPE R (RUN COMMAND)
8144          :      TO THE SHOW UNDER WAY
8145 002564 004137 011466      JSR    R1,MESG         ;TELL OPERATOR WE ARE READY TO GO
8146 002570 014214              WORD   RNMSG
8147 002572 000137 002706      JMP    EXEC            ;GET THE SHOW ON THE ROAD
8148          :
8149          :      VALID COMMAND TABLE
8150          :
8151 002576          000          VCMDTB: .BYTE 0          ;0 = TEST I/O
8152 002577          000          .BYTE 0          ;1 = WRITE BUFFER
8153 002600          000          .BYTE 0          ;2 = READ MANUAL INPUT
8154 002601          014          .BYTE CE!DE      ;3 = NOP
8155 002602          000          .BYTE 0          ;4 = SENSE COMMAND
8156 002603          000          .BYTE 0          ;5 = WRITE LINE ADDRESS
8157 002604          000          .BYTE 0          ;6 = READ FULL BUFFER
8158 002605          000          .BYTE 0          ;7 = ERASE COMMAND
8159 002606          002          .BYTE UCHK       ;10 = INVALID
8160 002607          002          .BYTE UCHK       ;11 = INVALID
8161 002610          000          .BYTE 0          ;12 = SHORT READ MANUAL INPUT
8162          002612          .EVEN

```

```

8164
8165
8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178 002612 012705 011600
8179 002616 000402
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192
8193 002620 012705 011534
8194 002624 012102
8195 002626 004737 011400
8196 002632 105037 013043
8197 002636 105037 013044
8198 002642 105737 013044
8199 002646 001367
8200 002650 105737 013043
8201 002654 001772
8202 002656 012702 012630
8203 002662 004715
8204 002664 005705
8205 002666 001005
8206 002670 120427 000015
8207 002674 001002
8208 002676 011101
8209 002700 000201
8210 002702 005721
8211 002704 000201
    
```

```

.SBTTL  INITIALIZATION PARAMETER INPUT AND CONVERSION ROUTINES

INHEX -- PRINT MESSAGE, WAIT FOR INPUT, GET IT AND CONVERT THE HEX TO BINARY

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

INHEX:  MOV     #CHTB,R5           ;MOVE ADDRESS OF CONVERSION ROUTINE TO R5
        BR      INR5

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

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

INOCY:  MOV     #COTB,R5           ;SET UP ADDRESS OF THE CONVERSION ROUTINE
INR5:   MOV     (R1)+,R2          ;GET ADDRESS OF THE MESSAGE
10$:    JSR     PC,PRMSG          ;PRINT THE DESIRED MESSAGE
        CLRB   TCMAC             ;RESET ACTIVE FLAG
        CLRB   TCMDB            ;RESET ABORT FLAG
30$:    TSTB   TCMDB            ;COMMAND ABORT?
        BNE   10$               ;YES, REASK QUESTION
        TSTB   TCMAC            ;WAS ENTRY COMPLETED?
        BEQ   30$               ;NO, WAIT
        MOV   #TBUF,R2          ;SET UP ADDRESS OF BEG OF INPUT BUFFER
        JSR   PC,INR5           ;CONVERT INPUT TO BINARY
        TST   R5                ;LOOK FOR DEFAULT RESP -- C/R
        BNE   40$               ;NOT DEFAULT TAKE NORMAL RETURN
        CMPB  R4,#CR            ;ILLEGAL CHAR MUST BE A C/R
        BNE   40$               ;ITS NOT A DEFAULT
        MOV   (R1),R1           ;---TAKE THE DEFAULT RETURN
40$:    RTS     R1

        TST   (R1)+            ;INCR FOR NORMAL RETURN
        RTS     R1
    
```



```

8213
8214
8215
8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227 002706 004737 011324
8228 002712 012702 000052
8229 002716 004737 011342
8230 002722 012706 012626
8231 002726 105037 013043
8232 002732 105037 013044
8233 002736 105037 013045
8234 002742 105737 013062
8235 002746 001402
8236 002750 000137 003326
8237 002754 105737 013043
8238 002760 001001
8239 002762 000767
8240
8241
8242
8243 002764 012702 012630
8244 002770 112203
8245 002772 042703 177400
8246 002776 012704 003032
8247 003002 020324
8248 003004 001411
8249 003006 022404
8250 003010 005714
8251 003012 001373
8252
8253
8254
8255 003014 012702 137607
8256 003020 004737 011342
8257 003024 000137 002706
8258
8259
8260
8261 003030 000134
8262
8263
8264
8265 003032 000101
8266 003034 004036
8267 003036 000104
8268 003040 003420
    
```

.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,CRLF ;PRINT CR/LF
      MOV #R2
      JSR PC,PRINT2 ;PRINT * -- DENOTE COMMAND MODE
      MOV #SSTACK,SP ;RE-ESTABLISH PUSH STACK
      CLRB TCMACT ;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$: TSTB TCMACT ;IS THERE A COMMAND TO EXECUTE
     BNE 30$ ;YES, EXECUTE IT
     BR 10$ ;NO, WAIT AGAIN IF NOTHING TO DO
    
```

THERE IS A TELETYPE COMMAND TO BE EXECUTED

```

30$: MOV #TBUF,R2 ;SET UP PTR TO START OF TELE BUFFER
     MOV# (R2)+,R3 ;GET COMMAND IDENTIFIER
     BIC #177400,R3 ;SAVE L.O. BYTE
     MOV #TCMDB,R4 ;SET UP PTR TO COMMAND TABLE
40$: 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"

```

CERR: MOV #137607,R2 ;PRINT ? AND "BELL"
      JSR PC,PRINT2
      JMP EXEC ;RETURN TO EXEC
    
```

EXECUTE COMMAND

```

EXECMD: JMP 3(R4)+ ;EXECUTE COMMAND
    
```

TELETYPE COMMAND TABLE

```

TCMDB: .WORD 'A ;A = ACCESS
       .WORD ACCESS
       .WORD 'D ;D = DUMP
       .WORD DUMP
    
```

8269	003042	000105	.WORD	'E	;E = ENABLE DEVICE
8270	003044	004210	.WORD	ENABLE	
8271	003046	000106	.WORD	'F	;F = FILL
8272	003050	003704	.WORD	FILL	
8273	003052	000110	.WORD	'H	;H = HELP COMMAND
8274	003054	003772	.WORD	HELP	
8275	003056	000111	.WORD	'I	;I = INPUT
8276	003060	004316	.WORD	INPUT	
8277	003062	000113	.WORD	'K	;K = KILL
8278	003064	004254	.WORD	KILL	
8279	003066	000122	.WORD	'R	;R = RUN
8280	003070	003100	.WORD	RUN	
8281	003072	000123	.WORD	'S	;S = STOP
8282	003074	003252	.WORD	STOP	
8283	003076	000000	.WORD	0	;END OF TABLE

```

8285
8286
8287
8288
8289
8290
8291
8292
8293
8294 003100 032777 001000 007352
8295 003106 001342
8296 003110 005077 007344
8297 003114 005277 007340
8298 003120 012700 000001
8299 003124 004737 010244
8300 003130 004737 010276
8301 003134 105063 000002
8302 003140 105063 000017
8303 003144 005063 000020
8304 003150 005200
8305 003152 120037 013057
8306 003156 003762
8307 003160 105037 013060
8308 003164 105037 013064
8309 003170 105037 013062
8310 003174 013701 013052
8311 003200 010137 013050
8312 003204 012702 001000
8313 003210 005021
8314 003212 005302
8315 003214 001375
8316 003216 012737 000001 013066
8317
8318 003224 053777 013102 007226
8319 003232 052777 004000 007220
8320 003240 052777 001100 007212
8321 003246 000137 002706
    
```

.SBTTL BACKGROUND -- RUN COMMAND

R = RUN COMMAND

THE RUN COMMAND READIES THE DX AND SPECIFIED DEVICE BUFFERS TO BEGIN OPERATION. THE RUN COMMAND MUST BE EXECUTED BEFORE ANY ACTION WILL BE PERFORMED OVER THE DX.

```

RUN: BIT #DXONLN,DXCS ; IS DX ENABLED?
      BNE CERR ; YES, ERROR
      CLR DXCS ; INITIALIZE THE DX
      INC DXCS ; SET GO
      MOV #1,DEV ; START CLEARING DEVICE TABLES
10S: JSR PC,CDEVST ; CLEAR DEV STATUS TABLE
      JSR PC,CSPWST ; RESET THE APPR SPW STATUS ENTRY FOR THE DEVICE
      CLRB SSENSE(DTAB) ; CLEAR SENSE BYTE
      CLRB SRDRQ(DTAB) ; CLEAR THE READ REQUEST
      CLR SMINS(DTAB) ; CLEAR THE START OF MANUAL INPUT
      INC DEV ; INCR TO NEXT DEVICE
      CMPB DEV,MAXDEV ; ARE WE DONE
      BLE 10S ; NO, DO NEXT DEVICE
      CLRB DXACT ; CLEAR DX ACTIVE FLAG
      CLRB CMDCHF ; CLEAR COMMAND CHAINING FLAG
      CLRB DXABFL ; CLEAR DX ABORT FLAG
      MOV TTADDR,R1 ; GET THE TUMBLE TABLE ADDRESS
      MOV R1,TTPTR ; RESET THE SOFTWARE T/T POINTER
      MOV #TTSIZE,R2 ; SET UP CLEAR CONSTANT (WORD COUNTER)
20S: CLR (R1)+ ; CLEAR T/T AND DUP T/T
      DEC R2 ; ARE WE DONE?
      BNE 20S ; NO, KEEP ON CLEARING
      MOV #1,MDEV ; INIT THE DEVICE NUMBER FOR MUX
      ; AND SEL EXECUTOR ROUTINES
      BIS XADDR,DXCS ; SET UP THE EXTENDED ADDRESS BITS
      BIS #BSYEN,DXCS ; SEL CHANNEL - SET BUSY ENABLE
30S: BIS #DXENB!DXONLN,DXCS ; ENABLE THE DX
      JMP EXEC
    
```

```

8323
8324
8325
8326
8327
8328
8329
8330
8331
8332
8333
8334
8335
8336
8337
8338
8339
8340
8341
8342
8343
8344 003252 111204
8345 003254 120427 000015
8346 003260 001422
8347 003262 120427 000104
8348 003266 001413
8349 003270 120427 000105
8350 003274 001410
8351 003276 120427 000111
8352 003302 001405
8353 003304 120427 000120
8354 003310 001402
8355 003312 000137 003014
8356 003316 110437 013056
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367 003322 000137 002706
8368
8369
8370
8371
8372 003326 042777 000100 007124
8373 003334 004737 011324
8374 003340 004137 011466
8375 003344 014006
8376 003346 013702 013076
8377 003352 004737 004746
8378 003356 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
S(C/R) -- STOP DX IMMEDIATELY
SI(C/R) -- STOP DX AFTER NEXT INITIAL SELECTION SEQUENCE
SD(C/R) -- STOP DX AFTER NEXT DATA TRANSFER IS COMPLETED
SE(C/R) -- STOP DX AFTER NEXT ENDING SEQUENCE IS RECEIVED
SP(C/R) -- STOP ON NEXT PARITY ERROR RECEIVED FROM CHANNEL

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

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

```

STOP:  MOVB    (R2),R4      ;GET THE TYPE OF STOP INDICATED
        CMPB    R4,8CR   ;IMMEDIATELY?
        BEQ     STOPDX   ;YES, DISABLE DX AND PRINT REGISTERS
        CMPB    R4,8'D   ;D = AFTER NEXT DATA TRANSFER?
        BEQ     10$      ;YES, SET STOP FLAG
        CMPB    R4,8'E   ;E = AFTER THE NEXT ENDING SEQUENCE
        BEQ     10$      ;YES, SET STOP FLAG
        CMPB    R4,8'I   ;I = AFTER THE CHIS SEQUENCE
        BEQ     10$      ;YES, SET STOP FLAG
        CMPB    R4,8'P   ;P = STOP ON PARITY ERROR??
        BEQ     10$      ;YES, SET STOP FLAG
        JMP     CERR     ;ILLEGAL FORMAT -- GIVE ERROR
10$:   MOVB    R4,DXSTPF ;SET THE STOP FLAG

```

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

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

JMP EXEC ;RETURN TO THE EXEC

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

```

007124 STOPDX: BIC     8DXENB,8DXCS ;DISABLE THE DX
               JSR     PC,CRLF ;START AT NEW LINE
               JSR     R1,MSG  ;PRINT "CURRENT DEVICE -- "
               .WORD  STPMES
               MOV     COEV,R2 ;CONVERT AND PRINT THE CURRENT
               JSR     PC,HDP  ;DEVICE NUMBER IN HEX
               MOV     813.,R3 ;PRINT THE 13 DX REGISTERS IN OCTAL

```

8379	003362	012701	012454
8380	003366	013102	
8381	003370	004737	004656
8382	003374	005303	
8383	003376	001373	
8384	003400	105037	013062
8385	003404	005077	007050
8386	003410	005277	007044
8387	003414	000137	002706

10S:

MOV	#DXDS,R1
MOV	3(R1)+,R2
JSR	PC,OCTDMP
DEC	R3
BNE	10S
CLRB	DXABFL
CLR	DXCS
INC	DXCS
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

```

```

8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409 003420 004737 005036
8410 003424 112204
8411 003426 012705 004624
8412 003432 120427 000101
8413 003436 001421
8414 003440 012705 004560
8415 003444 120427 000105
8416 003450 001414
8417 003452 012705 004742
8418 003456 120427 000110
8419 003462 001407
8420 003464 012705 004656
8421 003470 120427 000117
8422 003474 001402
8423 003476 000137 003014
8424 003502 010537 013114
8425 003506 005700
8426 003510 001043
8427 003512 112204
8428 003514 120427 000054
8429 003520 001014
8430 003522 004737 005270
8431
8432
8433

```

.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
      MOVB   (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$:  MOV    R5, DMPADR          ;SAVE ADDRESS OF DUMP ROUTINE
      TST   R0                  ;WAS THIS A TUMBLE TABLE DUMP?
      BNE  DTUMTB               ;YES, DUMP THE TUMBLE TABLE
      MOVB  (R2)+, R4           ;WAS A DEV # SPECIFIED
      CMPB  R4, #',             ;IS NEXT POSITION A COMMA
      BNE  50$                  ;NO, DUMP GIVEN LIMITS
      JSR   PC, GDEV            ;GET THE DEVICE NUMBER -- IN HEX

```

COMPUTE RELOCATION CONSTANT FOR DEVICE

```

8435 003526 005004
8436 003530 005303          30$: CLR R4 ;RELOCATION CONSTANT
8437 003532 001403          DEC R3 ;DONE?
8438 003534 062704 002000  ADD #2000,R4 ;YES, ADD TO START + END ADDRESSES
8439 003540 000773          BR 30$ ;TO NEXT DEVICE TABLES
8440 003542 060437 013110  40$: ADD R4,SADDR ;ADD RELOCAT TO START ADDRESS
8441 003546 060437 013112  ADD R4,EADDR ;ADD RELOCAT TO END ADDRESS
8442 003552 017702 007332  50$: MOV #SADDR,R2 ;GET WORD
8443 003556 004777 007332  JSR PC,JDMPADR ;CONVERT AND DUMP IT
8444 003562 105737 013061  60$: TSTB PCTR
8445 003566 001375          BNE 60$
8446 003570 062737 000002 013110  ADD #2,SADDR ;INCR TO NEXT WORD
8447 003576 023737 013110 013112  CMP SADDR,EADDR ;DUMP DONE
8448 003604 003003          BGT 70$ ;YES, EXIT
8449 003606 105737 013044  TSTB TCMDB ;COMMAND ABORT?
8450 003612 001757          BEQ 50$ ;NO, PRINT NEXT WORD
8451 003614 000137 002706  70$: JMP EXEC ;YES, RETURN TO EXEC
8452
8453
8454
8455
8456 ;
8457 ;
8458 003620 012700 000400  DTUMTB: MOV #TTSIZE/2,R0 ;SET UP COUNTER TO DUMP ENTIRE TUMBLE TABLE
8459 003624 017702 007260  5$: MOV #SADDR,R2 ;GET STARTING ADDRESS
8460 003630 004777 007260  JSR PC,JDMPADR ;PRINT THE CONTENTS
8461 003634 105737 013061  10$: TSTB PCTR ;IS PRINT COMPLETE?
8462 003640 001375          BNE 10$ ;NO, WAIT TILL DONE
8463 003642 032737 000777 013110  BIT #TTSIZE-1,SADDR ;CHECK FOR WRAP AROUND
8464 003650 001003          BNE 20$
8465 003652 062737 001000 013110  ADD #TTSIZE,SADDR ;WRAP AROUND TO TOP OF TABLE
8466 003660 162737 000002 013110  20$: SUB #2,SADDR ;DECREMENT TO NEXT ENTRY
8467 003666 005300          DEC R0 ;HAS ENTIRE TUMBLE TABLE BEEN DUMPED?
8468 003670 001403          BEQ 30$ ;YES, EXIT TO THE EXEC
8469 003672 105737 013044  TSTB TCMDB ;ARE WE TO ABORT?
8470 003676 001752          BEQ 5$ ;NO, KEEP ON DUMPING
8471 003700 000137 002706  30$: JMP EXEC ;YUP, BACK TO THE EXEC

```

```

8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487 003704 004737 005036
8488 003710 004737 011600
8489 003714 110337 012524
8490 003720 004737 005270
8491
8492
8493
8494 003724 005004
8495 003726 005303
8496 003730 001403
8497 003732 062704 002000
8498 003736 000773
8499 003740 060437 013110
8500 003744 060437 013112
8501 003750 013701 013110
8502
8503
8504
8505 003754 113721 012524
8506 003760 020137 013112
8507 003764 101773
8508 003766 000137 002706
    
```

.SBTTL BACKGROUND -- FILL COMMAND

FILL COMMAND

THE FILL COMMAND LOADS THE SPECIFIED BYTE INTO THE GIVEN DATA AREA.

THE FOLLOWING SYNTAXES ARE AVAILABLE FOR THE FILL COMMAND:

FIN,XX,YY FILL INPUT BUFFER FOR DEVICE YY WITH XX
FOT,XX,YY FILL OUTPUT BUFFER FOR DEVICE YY WITH XX

WHERE: XX IS THE FILL CHARACTER IN HEX
YY IS THE DEVICE ADDRESS IN HEX

```

FILL: JSR PC, GLIMIT ;GET BUFFER LIMITS
      JSR PC, CHTB ;GET THE FILL CHARACTER
      MOV R3, FILLCH ;SAVE FILL CHAR
      JSR PC, GDEV ;GET THE DEVICE ADDRESS
    
```

COMPUTE RELOCATION FOR DEVICE

```

10$: CLR R4
      DEC R3 ;DONE?
      BEQ 20$ ;YES, ADD TO START AND END ADDR
      ADD #2000, R4
      BR 10$
20$: ADD R4, SADDR ;ADD RELOC CONST TO START
      ADD R4, EADDR ;ADD RELOC CONST TO END ADDR
      MOV SADDR, R1
    
```

FILL BUFFER WITH SPECIFIED CHARACTER

```

30$: MOV R1, FILLCH, (R1)+ ;FILL CHARACTER
      CMP R1, EADDR ;DONE?
      BLOS 30$ ;NOPE, FILL NEXT CHAR
      JMP EXEC ;DONE, RETURN TO EXEC
    
```



```

8510
8511
8512
8513
8514
8515
8516
8517
8518 003772 012701 014270
8519 003776 012702 002554
8520
8521
8522
8523
8524 004002 112100
8525 004004 123727 013061 000004
8526 004012 003374
8527 004014 004737 011362
8528 004020 105737 013044
8529 004024 001002
8530 004026 005302
8531 004030 001364
8532 004032 000137 002706
    
```

.SBTTL BACKGROUND -- HELP COMMAND

THE HELP COMMAND PROVIDES THE OPERATOR WITH A SYNOPSIS OF
COMMANDS WHICH MAY BE USED FOR OPERATING THIS SYSTEM.

THE SYNTAX FOR THE HELP COMMAND IS:

H

```

HELP:  MOV    #HELPMS,R1      ;SET UP ADDRESS OF HELP MESSAGE
        MOV    #HELPLN,R2    ;LENGTH OF HELP MESSAGE
    
```

START OUTPUTTING THE HELP MESSAGE UNDER OUR CONTROL
SO THE COMMAND MAY BE ABORTED QUICKLY.

```

10$:  MOVB    (R1)+,R0        ;GET BYTE TO OUTPUT
15$:  CMPB    PCTR,#4        ;MORE THEN FOUR CHARACTERS IN OUTPUT BUFFER??
        BGT    15$          ; YES, WAIT TIL DOWN A LITTLE
        JSR    PC,PCHAR      ;PRINT IT ON CONSOLE
        TSTB   TCADAB       ;HAS OPERATOR INDICATED A DESIRE TO STOP?
        BNE    20$          ; YES, ABORT HELP MESSAGE
        DEC    R2           ;HAS ENTIRE MESSAGE BEEN OUTPUTTED??
        BNE    10$         ; NO, OUTPUT ANOTHER BYTE
        JMP    EXEC        ; YES, RETURN TO THE EXECUTIVE
20$:
    
```

```

8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547 004036 004737 011534
8548 004042 005705
8549 004044 001403
8550 004046 032703 000001
8551 004052 001402
8552 004054 000137 003014
8553 004060 010337 013110
8554 004064 013702 013110
8555 004070 105037 013043
8556 004074 004737 004656
8557 004100 012702 020040
8558 004104 004737 011342
8559 004110 017702 006774
8560 004114 004737 004662
8561 004120 012702 020040
8562 004124 004737 011342
8563 004130 105737 013043
8564 004134 001775
8565 004136 012702 012630
8566 004142 004737 011534
8567 004146 005705
8568 004150 001007
8569 004152 120427 000057
8570 004156 001412
8571 004160 120427 000015
8572 004164 001403
8573 004166 000736
8574 004170 010377 006714
8575 004174 062737 000002 013110
8576 004202 000730
8577 004204 000137 002706
    
```

```

.SBTTL BACKGROUND -- ACCESS COMMAND

ACCESS SPECIFIED LOCATIONS AND CHANGE IF DESIRED

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

BASICALLY THE FOLLOWING ACTIONS ARE PERMITTED:
    AXXXXX -- OPEN AND PRINT SPECIFIED OCTAL LOCATION
    [XXXXXX](C/R) -- CHANGE CURRENT LOCATION IF DATA
    SPECIFIED [XXXXXX] AND OPEN NEXT LOCATION
    / -- RETURN TO EXEC MODE

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

```

8579
8580
8581
8582
8583
8584
8585
8586
8587
8588
8589
8590
8591 004210 004737 005270
8592 004214 004737 010244
8593 004220 010005
8594 004222 063705 013100
8595 004226 060505
8596 004230 063705 013104
8597 004234 013715 013106
8598 004240 105063 000002
8599 004244 105063 000016
8600 004250 000137 002706

```

.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

```

```

8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616 004254 004737 005270
8617 004260 004737 010244
8618 004264 112763 000001 000016
8619 004272 010005
8620 004274 063705 013100
8621 004300 060505
8622 004302 063705 013104
8623 004306 012715 000002
8624 004312 000137 002706

```

.SBTTL BACKGROUND -- KILL DEVICE COMMAND

K = KILL A DEVICE

THE KILL COMMAND DISABLES THE SPECIFIED DEVICE ADDRESS FROM PERFORMING TRANSFERS OVER THE DX. IT PUTS THE SPECIFIED DEVICE ADDRESS INTO AN OFF-LINE STATE. AN ENABLE COMMAND MUST BE ISSUED BEFORE DATA TRANSFERS MAY BE PERFORMED WITH THE DX FOR THE SPECIFIED DEVICE ADDRESS.

THE KILL COMMAND HAS THE FOLLOWING SYNTAX:

KXX -- KILL DEVICE ADDRESS XX
THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX

```

KILL: JSR PC,GDEV ;GET THE DEVICE NUMBER
      JSR PC,CDEVST
      MOVB #1,SONLF(DTAB)
      MOV DEV,R5 ;COMPUTE THE ADDRESS OF THE SPW TABLE
      ADD DEVCON,R5 ;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

```

```

8626
8627
8628
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638 004316 105737 012522
8639 004322 001005
8640 004324 004737 005270
8641 004330 120427 000054
8642 004334 001402
8643 004336 000137 003014
8644 004342 004737 010216
8645 004346 026327 000004 000734
8646 004354 002370
8647 004356 032763 000001 000004
8648 004364 001002
8649 004366 005263 000004
8650 004372 016305 000010
8651 004376 066305 000004
8652 004402 112725 000112
8653 004406 010563 000020
8654 004412 005263 000004
8655 004416 026327 000004 000735
8656 004424 001423
8657 004426 112204
8658 004430 042704 177600
8659 004434 020427 000015
8660 004440 001415
8661 004442 020427 000040
8662 004446 002410
8663 004450 020427 000137
8664 004454 003005
8665 004456 162704 000040
8666 004462 116425 012344
8667 004466 000751
8668 004470 005004
8669 004472 000773
8670
8671
8672
8673
8674
8675 004474 112715 000152
8676 004500 005263 000004
8677 004504 105263 000017
8678
8679
8680
8681 004510 105737 013060
    
```

```

.SBTTL BACKGROUND -- INPUT DISPLAY DATA COMMAND

I = INPUT

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

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

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

      SET UP TO EXIT
      SET EOM INDICATOR
      QUEUE READ MANUAL INPUT REQUEST
70$:   MOVB     #EOM,(R5) ;SET EOM INDICATOR
      INC      SCURS(DTAB) ;INCREMENT CURSOR POINTER
      INCB     SRDRQ(DTAB) ;QUEUE READ REQUEST

      SEE IF THE DX IS CURRENTLY ACTIVE
      TSTB     DXACT      ;IS DX ACTIVE?
    
```

```

8682 004514 001402          BEQ      BOS      ;NO, START ASYNCHRONOUS PROCESSING TO SEND ATTENTION
8683 004516 000137 002706  JMP      EXEC     ;YES, ATTENTION WILL BE TAKEN CARE OF BY DX
8684                                     :
8685                                     :
8686                                     :
8687                                     :
8688 004522 013746 177776  BOS:  MOV     PSW, -(SP) ;PSW TO PUSH STACK
8689 004526 012746 002706  MOV     %EXEC, -(SP) ;RETURN ADDRESS TO PUSH STACK
8690 004532 012737 000340 177776 MOV     #340, PSW    ;INHIBIT INTERRUPTS
8691 004540 010046          MOV     R0, -(SP)   ;SET UP PUSH STACK FOR FAKE INTERRUPT
8692 004542 010146          MOV     R1, -(SP)
8693 004544 010246          MOV     R2, -(SP)
8694 004546 010346          MOV     R3, -(SP)
8695 004550 010446          MOV     R4, -(SP)
8696 004552 010546          MOV     R5, -(SP)
8697 004554 000137 006542  JMP     DXEXEC     ;START PROCESSING THE ATTENTION
    
```

8699
8700
8701
8702
8703
8704
8705
8706
8707
8708
8709
8710
8711 004560 010237 013046
8712 004564 113703 013046
8713 004570 042703 177400
8714 004574 116337 011744 013046
8715 004602 113703 013047
8716 004606 042703 177400
8717 004612 116337 011744 013047
8718 004620 013702 013046
8719
8720
8721
8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733 004624 105737 013045
8734 004630 001005
8735 004632 004737 011324
8736 004636 112737 000036 013045
8737 004644 105337 013045
8738 004650 004737 011342
8739 004654 000207
8740
8741
8742
8743
8744
8745
8746
8747
8748
8749
8750
8751
8752
8753
8754 004656 004737 011324

.SBTTL BACKGROUND SUBROUTINES -- PRINT FORMATTING
DUMP WORD IN EBCDIC ON TTY
CALLING SEQUENCE
.....R2 CONTAINS WORD TO BE PRINTED
JSR PC,EBCDMP
.....RETURN

REGISTERS 2 + 3 ARE DESTROYED BY THIS SUBROUTINE

```
EBCDMP: MOV R2,WK ;SAVE WORD TO BE PRINTED
        MOVB WK,R3 ;GET LO BYTE
        BIC #177400,R3
        MOVB EBCDTB(R3),WK ;CONVERT EBCDIC TO ASCII
        MOVB WK1,R3 ;GET HI BYTE AND CONVERT
        BIC #177400,R3
        MOVB EBCDTB(R3),WK1 ;CONVERT CHAR TO ASCII
        MOV WK,R2
        FALL THROUGH TO ASCII PRINT ROUTINE
```

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

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

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

DUMP WORD IN OCTAL ON TTY
CALLING SEQUENCE
.....R2 CONTAINS WORD TO BE PRINTED
JSR PC,OCTDMP OR ODMP
.....RETURN
OCTDMP PERFORMS A CR/LF BEFORE PRINTING OCTAL DATA
NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

```
OCTDMP: JSR PC,CRLF ;GIVE A CRLF
```

```

8755 004662 010046          ODMP:  MOV    R0,-(SP)      ;SAVE IMPORTANT REGISTERS
8756 004664 010246          MOV    R2,-(SP)
8757 004666 010446          MOV    R4,-(SP)
8758 004670 012704 000006   MOV    #6,R4          ;EXTRACT 6 OCTAL DIGITS
8759 004674 005000          CLR    R0             ;CLEAR THE WORKING REGISTER
8760 004676 006102          ROL   R2             ;MOVE HIGH ORDER BIT TO C-BIT
8761 004700 006100          10S:  ROL   R0             ;GET THE REMAINING BIT STILL IN LINK
8762 004702 042700 177770   BIC   #177770,R0     ;ONLY 3 LOW ORDER BITS
8763 004706 062700 000060   ADD   #0,R0          ;MAKE ASCII
8764 004712 004737 011362   JSR   PC,PCHAR       ;PRINT IT ON THE TTY
8765 004716 006102          ROL   R2             ;ROTATE THE NEXT OCTAL CHAR INTO POSITION
8766 004720 006102          ROL   R2
8767 004722 006102          ROL   R2
8768 004724 010200          MOV   R2,R0          ;DATA TO WORKING REGISTER
8769 004726 005304          DEC   R4             ;ARE WE DONE?
8770 004730 001363          BNE   10S            ;NO, PRINT ANOTHER CHARACTER
8771 004732 012604          OCTEX: MOV   (SP)+,R4    ;RESTORE USED REGISTERS
8772 004734 012602          MOV   (SP)+,R2
8773 004736 012600          MOV   (SP)+,R0
8774 004740 000207          RTS                    ;RETURN TO THE CALLER
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785
8786
8787
8788
8789 004742 004737 011324   HEXDMP: JSR   PC,CRLF    ;DO A CR LF
8790 004746 010046          HDMP:  MOV   R0,-(SP)    ;SAVE THE WORKING REGISTERS
8791 004750 010246          MOV   R2,-(SP)
8792 004752 010446          MOV   R4,-(SP)
8793 004754 012704 000004   10S:  MOV   #4,R4          ;4 CHARACTERS PER WORD
8794 004760 006102          ROL   R2             ;ROTATE HIGH ORDER 4 BITS TO LOW ORDER 4 BITS
8795 004762 006102          ROL   R2
8796 004764 006102          ROL   R2
8797 004766 006102          ROL   R2
8798 004770 010200          MOV   R2,R0          ;TO WORKING REG
8799 004772 006100          ROL   R0             ;GET THE LINK BIT TOO
8800 004774 042700 177760   BIC   #177760,R0     ;ONLY LOW ORDER 4 BITS
8801 005000 062700 000060   ADD   #0,R0          ;MAKE ASCII IF NUMBER
8802 005004 020027 000071   CMP   R0,#9          ;SHOULD IT BE A-F?
8803 005010 003402          BLE   20S            ;NO, SHIP IT
8804 005012 062700 000007   ADD   #7,R0          ;YES, MAKE ALPHA
8805 005016 004737 011362   20S:  JSR   PC,PCHAR       ;PRINT THE HEX CHARACTER
8806 005022 005304          DEC   R4             ;ARE WE DONE?
8807 005024 001355          BNE   10S            ;NO, CONVERT AND PRINT NEXT CHARCATER
8808 005026 012604          MOV   (SP)+,R4
8809 005030 012602          MOV   (SP)+,R2
8810 005032 012600          MOV   (SP)+,R0

```

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

C07

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB BACKGROUND SUBROUTINES -- PRINT

MACY11 27(732) 27-OCT-76 14:48 PAGE 66-10
FORMATTING

8811 005034 000207

RTS PC

;RETURN TO THE CALLER

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:

RD = 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
	BEG	GLMTT		:YES, SET UP T/T LIMITS
	CMPB	R4, #'I		:I = INPUT BUFFER
	BEG	GLMIN		:YES, SET UP INPUT BUFFER LIMITS
	CMPB	R4, #'O		:O = OUTPUT BUFFER
	BEG	GLAOT		: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?
	BEG	GLERR		:NO, ERROR
	MOV	R3, EADDR		:SAVE END ADDR
	BR	GLX		:PREPARE TO EXIT

SET UP LIMITS OF TUMBLE TABLE

GLMTT:	MOVB	(R2)+, R4		:MUST BE TT
	CMPB	R4, #'T		:ILLEGAL ENTRY
	BNE	GLERR		
	MOV	TTPTR, SADDR		

8813				
8814				
8815				
8816				
8817				
8818				
8819				
8820				
8821				
8822				
8823				
8824				
8825				
8826				
8827				
8828				
8829				
8830				
8831				
8832	005036	005000		
8833	005040	004737	011534	
8834	005044	005705		
8835	005046	001014		
8836	005050	120427	000124	
8837	005054	001425		
8838	005056	120427	000111	
8839	005062	001436		
8840	005064	120427	000117	
8841	005070	001453		
8842				
8843				
8844				
8845	005072	012601		
8846	005074	000137	003014	
8847				
8848				
8849				
8850				
8851				
8852	005100	010337	013110	
8853	005104	120427	000054	
8854	005110	001370		
8855	005112	004737	011534	
8856	005116	005705		
8857	005120	001764		
8858	005122	010337	013112	
8859	005126	000454		
8860				
8861				
8862				
8863				
8864				
8865	005130	112204		
8866	005132	120427	000124	
8867	005136	001355		
8868	005140	013737	013050 013110	

```

8869 005146 062737 000776 013110      ADD      #TTSIZE-2,SADDR ;COMPUTE ADDRESS OF APPR DUPLICATE TT ENTRY
8870 005154 005200                      INC      RD                ;INDICATE DUMP TUMBLE TABLE
8871 005156 000437                      BR       GLEX1            ;SET UP TO EXIT
8872
8873
8874
8875      ;
8876      ;
8877 005160 112204                      GLMIN:  MOVB   (R2)+,R4
8878 005162 120427 000116                CMPB   R4,#'N            ;MUST BE IN
8879 005166 001341                      BNE    GLERR            ;ILLEGAL ENTRY
8880 005170 013704 013054                MOV    SDEVTB,R4        ;GET ADDR OF DEV 0 STATUS TABLE
8881 005174 016437 000006 013110        MOV    SINGB(R4),SADDR
8882 005202 013737 013110 013112        MOV    SADDR,EADDR
8883 005210 062737 000741 013112        ADD    #DISPSZ+1,EADDR ;DISPLAY SIZE + ROOM FOR LINE ADDRESS
8884 005216 000417                      BR       GLEX1
8885
8886
8887      ;
8888      ;
8889      ;
8890 005220 112204                      GLMOT: MOVB   (R2)+,R4
8891 005222 120427 000124                CMPB   R4,#'T            ;MUST BE OT
8892 005226 001321                      BNE    GLERR            ;ILLEGAL ENTRY
8893 005230 013704 013054                MOV    SDEVTB,R4        ;GET ADDR OF DEV 0 STATUS TABLE
8894 005234 016437 000010 013110        MOV    SOUTB(R4),SADDR ;COMPUTE STARTING AND ENDING ADDRESSES OF SPECIFIED BUF
8895 005242 013737 013110 013112        MOV    SADDR,EADDR
8896 005250 062737 000737 013112        ADD    #DISPSZ-1,EADDR ;DISPLAY CHAR BUFFER
8897 005256 112204                      GLEX1: MOVB   (R2)+,R4    ;GET NEXT INPUT CHARACTER AND UPDATE POINTER
8898
8899      ;
8900      ;
8901 005260 120427 000054                      GLEX:  CMPB   R4,#'
8902 005264 001302                      BNE    GLERR            ;CHECK FOR
8903 005266 000207                      RTS     PC              ;ENTRY NOT PROPERLY DELIMITED (ERROR)

```

F07

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
 DZDXIA.CMB BACKGROUND SUBROUTINES --

MACY11 27(732) 27-OCT-76 14:48 PAGE 66-13
 COMPUTE SPECIFIED BUFFER LIMITS AND DEVICE ADDRESSES

```

8905
8906
8907
8908
8909
8910
8911
8912
8913
8914
8915
8916
8917
8918
8919 005270 004737 011600
8920 005274 163703 012512
8921 005300 100406
8922 005302 005203
8923 005304 120337 013057
8924 005310 101002
8925 005312 010300
8926 005314 000207
8927 005316 000137 003014
  
```

```

      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

GDEV: 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
  
```

8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945
8946
8947 005322 010046
8948 005324 010146
8949 005326 010246
8950 005330 010346
8951 005332 010446
8952 005334 010546
8953 005336 013702 013050
8954 005342 005712
8955 005344 001002
8956
8957
8958
8959
8960
8961 005346 000137 010200
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984

.SBTTL DX11-B ISR (INTERRUPT REQUEST LOGIC AND TUMBLE TABLE DECODE LOGIC)

DX11-B ISR

DX11 ISR AND RELATED SUBROUTINE REGISTER USAGE

R0	DEV	DEVICE NUMBER
R1		UN ASSIGNED
R2		UNASSIGNED
R3	DTAB	ADDRESS OF CURRENT DEVICE TABLE
R4	TT1	TUMBLE TABLE ENTRY 1
R5	TT2	TUMBLE TABLE ENTRY 2

THE ABOVE REGISTER DESIGNATIONS REPRESENT WHAT USUALLY WILL BE CONTAINED IN A REGISTER DURING DX ISR PROCESSING. HOWEVER, AS SITUATIONS DICTATE REGISTERS MAY BE USED FOR DIFFERENT PURPOSES.

```

DXISR: MOV R0, -(SP) ;SAVE HARDWARE REGISTERS
      MOV R1, -(SP)
      MOV R2, -(SP)
      MOV R3, -(SP)
      MOV R4, -(SP)
      MOV R5, -(SP)
      MOV TTPTR, 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
```

PROCESS TUMBLE TABLE ENTRIES
FOR CONVICIENCE THE PROCESSING IS BEING PERFORMED AT THE INTERRUPT LEVEL. IT IS SUGGESTED THAT IN NORMAL PROCESSING ENVIRONMENTS THIS PROCESSING BE DISTRIBUTED TO LESS PRIVILEDGED 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.

H07

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB DX11-B ISR (INTERRUPT REQUEST LOGIC AND TUMBLE TABLE DECODE LOGIC)

MACY11 27(732) 27-OCT-76 14:48 PAGE 67-1

```

8985
8986 005352 013702 013050      LOOP:  MOV   TTPTR,R2      ;GET T/T PTR
8987 005356 005712              TST   (R2)           ;ANY ENTRIES LEFT IN T/T?
8988 005360 001002              BNE   10$           ;
8989 005362 000137 006542      JMP   DXEXEC        ;NO, EXECUTE NEXT DX COMMAND
8990
8991      :
8992      TUMBLE TABLE ENTRY AVAILABLE FOR PROCESSING
8993      RESET THE DONE BIT (RELIEVE INTERRUPT)
8994      COPY TUMBLE TABLE ENTRY TO DUPLICATE TUMBLE TABLE (FOR SYSTEM TESTS PURP
8995      RESET TUMBLE TABLE ENTRY (2 WORDS)
8996 005366 042777 000200 005064 10$:  BIC   #DONE,DXCS    ;CLEAR DONE
8997 005374 010203              MOV   R2,R3         ;SET UP PTR TO DUP T/T
8998 005376 062703 001000      ADD   #TTSIZE,R3
8999 005402 011223              MOV   (R2),(R3)+    ;SAVE T/T ENTRY #1
9000 005404 011204              MOV   (R2),TT1
9001 005406 005022              CLR   (R2)+         ;CLEAR T/T ENTRY #1
9002 005410 011223              MOV   (R2),(R3)+    ;SAVE T/T ENTRY #2
9003 005412 011205              MOV   (R2),TT2
9004 005414 005022              CLR   (R2)+         ;CLEAR T/T ENTRY #2
9005
9006      :
9007      CHECK FOR POINTER WRAP AROUND
9008 005416 032702 000777      BIT   #TTSIZE-1,R2 ;AT END OF BUFFER?
9009              :
9010              BNE   20$           ;NOTE -- POWER OF 2 BOUNDARY
9011              :
9012              MOV   TTADDR,R2    ;YES, RESET PTR
9013              MOV   R2,TTPTR     ;SAVE T/T PTR
9014
9015      20$:  START PROCESSING TUMBLE ENTRY ENTRY
9016              SAVE DEVICE ADDRESS
9017              CHECK FOR SYSTEM RESET
9018              VALIDATE DEVICE ADDRESS
9019
9020      NOTE -- IF SYSTEM RESET OCCURRED, THERE IS NO GUARANTEE
9021      THAT THE DEVICE ADDRESS WILL BE VALID.
9022
9022 005434 010500              MOV   TT2,DEV       ;GET DEV #
9023 005436 042700 177400      BIC   #177400,DEV
9024 005442 010037 013076      MOV   DEV,CDEV     ;SAVE CURRENT DEVICE NUMBER
9025 005446 042777 000200 005004  BIC   #DONE,DXCS    ;CLEAR DONE
9026 005454 032704 010000      BIT   #SYSRST,TT1  ;SYSTEM RESET?
9027 005460 001066              BNE   PSYSRT       ;YES, PERFORM SYSTEM RESET FUNCTION
9028 005462 163700 012512      SUB   SDEV,DEV     ;GET IN 0-7 RANGE - IF VALID
9029 005466 100403              BMI   30$          ;INVALID DEVICE NUMBER
9030 005470 120037 013057      CMPB  DEV,MAXDEV   ;VALID DEVICE?
9031 005474 103405              BLO   40$          ;YES, NOT TOO BIG
9032
9033      INVALID DEVICE ADDRESS - BITCH
9034
9035      AN INVALID DEVICE ADDRESS WILL GENERALLY INDICATE
9036      A PROBLEM IN THE CONFIGURATION OF DX DEVICE
9037      ADDRESSES. BASICALLY THE DX HAS BEEN STRAPPED
9038      TO HANDLE DEVICE ADDRESSES WHICH OVERLAP WITH
9039      OTHER DEVICES ON THE CHANNEL.
9040

```

```

9041 005476 004137 011502      30S:  JSR    R1,INMES      ;PRINT "INVALID DEVICE"
9042 005502 013752              .WORD  ILLMES
9043 005504 000137 006524      JMP    DXAB           ;ABORT DX11
9044                               :
9045                               :
9046                               :
9047 005510 005200              40S:  INC    DEV           ;MAKE DEVICE NUMBER 1 -8
9048 005512 004737 010216      JSR    PC,SUDEV      ;SET UP ADDR OF DEV STAT TABLE

```

```

9050
9051
9052
9053
9054
9055
9056
9057
9058 005516 032704 020000
9059 005522 001402
9060 005524 000137 005764
9061 005530 032704 004000
9062 005534 001402
9063 005536 000137 006014
9064 005542 032704 040000
9065 005546 001402
9066 005550 000137 006070
9067 005554 032704 000100
9068 005560 001402
9069 005562 000137 006102
9070 005566 032704 100000
9071 005572 001402
9072 005574 000137 006232
9073 005600 032704 000200
9074 005604 001402
9075 005606 000137 006262
9076 005612 032704 000040
9077 005616 001402
9078 005620 000137 006444
9079 005624 032704 000020
9080 005630 001650
9081 005632 000137 006430
    
```

DECODE DX TUMBLE TABLE STATUS ENTRY

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

```

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



```

9083
9084
9085
9086
9087
9088
9089
9090 005636 012700 000001
9091 005642 004737 010244
9092 005646 004737 010276
9093 005652 105063 000002
9094 005656 105063 000017
9095 005662 005063 000020
9096 005666 005063 000004
9097 005672 105063 000003
9098 005676 016301 000010
9099 005702 012702 000740
9100 005706 112704 000100
9101 005712 105737 012522
9102 005716 001402
9103 005720 113704 012524
9104
9105 005724 110421
9106 005726 005302
9107 005730 001375
9108 005732 005200
9109 005734 120037 013057
9110 005740 003740
9111 005742 105037 013060
9112 005746 105037 013064
9113 005752 042777 000400 004500
9114 005760 000137 005352
9115
9116
9117
9118
9119
9120
9121
9122
9123
9124 005764 004737 010244
9125 005770 004737 010276
9126 005774 105063 000002
9127 006000 105037 013060
9128 006004 105037 013064
9129 006010 000137 005352
9130
9131
9132
9133
9134
9135
9136
9137
9138

```

.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

```

```

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

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

9139
9140
9141
9142 006014 004737 010276
9143 006020 105763 000000
9144 006024 001417
9145 006026 004737 010244
9146 006032 012763 000003 000000
9147 006040 120037 013060
9148 006044 001002
9149 006046 105037 013060
9150 006052 120037 013064 10$:
9151 006056 001002
9152 006060 105037 013064
9153 006064 000137 005352 20$:
9154
9155
9156
9157
9158
9159
9160 006070 004137 011502
9161 006074 013676
9162 006076 000137 006524
9163
9164
9165
9166
9167
9168
9169
9170
9171 006102 004737 010276
9172 006106 105763 000001
9173 006112 001402
9174 006114 004737 010370
9175 006120 126327 000000 000011 10$:
9176 006126 001003
9177 006130 112763 000002 000017
9178 006136 105063 000001 20$:
9179 006142 105037 013060
9180 006146 004737 010244
9181 006152 032704 000004
9182 006156 001402
9183 006160 110037 013064
9184 006164 123727 013056 000105 30$:
9185 006172 001552
9186
9187 006174 032704 000010
9188 006200 001412
9189
9190
9191
9192
9193
9194
  
```

AND CHAN END ! DEVICE END SET IN STATUS BYTE
 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
 10\$: CMPB DEV,CMDCHF ;DOES DEVICE HAVE CMD CHAIN SPEC?
 BNE 20\$;NO, GET NEXT T/T ENTRY
 CLRB CMDCHF ;YES, CLEAR FLAG
 20\$: 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 NXMMSG
 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
 10\$: CMPB SCMD(DTAB),#11 ;WAS ATTN ACCEPTED?
 BNE 20\$;NO, CONTINUE
 MOVB #2,SRDRQ(DTAB) ;YES, INDICATE 360 ACCEPTANCE
 20\$: 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
 30\$: 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 2848 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.

9195									
9196									
9197									
9198									
9199									
9200									
9201									
9202									
9203	006202	105737	012522						
9204	006206	001004							
9205	006210	012700	000001						
9206	006214	004737	010216						
9207	006220	112763	000010	000000	40\$:				
9208	006226	000137	005352		50\$:				
9209									
9210									
9211									
9212									
9213									
9214	006232	004137	011502						
9215	006236	013726							
9216	006240	123727	013056	000120					
9217	006246	001524							
9218	006250	152763	000002	000003					
9219	006256	000137	005600						
9220									
9221									
9222									
9223									
9224									
9225									
9226									
9227									
9228									
9229	006262	004737	010276						
9230									
9231	006266	123727	013056	000111					
9232	006274	001511							
9233	006276	032704	000001						
9234	006302	001022							
9235									
9236									
9237									
9238	006304	105005							
9239	006306	000305							
9240	006310	105705							
9241	006312	001437							
9242	006314	120527	000003						
9243	006320	001434							
9244	006322	020527	000012						
9245	006326	003405							
9246	006330	004137	011502						
9247	006334	014040							
9248	006336	000137	006524						
9249	006342	10563	000000		10\$:				
9250	006346	000137							

PARITY ERROR WAS DETECTED

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

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

TT2 CONTAINS THE COMMAND TO BE EXECUTED

PCHIS: JSR PC, CSPWST ;RESET THE SPW STATUS BYTE
 ;ON NEXT CHANNEL INITIATED SELECTION SEQUENCE
 CMPB DXSTPF, #'I ;WAS STOP ON ISS SPECIFIED(SI)
 BEQ STPDX ;YES, DISABLE DX
 BIT #CMDREJ, TT1 ;WAS COMMAND REJECTED BY DX?
 BNE ZOS ;YES, COMMAND REJECTED BY THE DX

VALID COMMAND, SET UP TO PROCESS IT

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

```

9251
9252
9253
9254 006350 105763 000016 20S: TSTB SONLF(DTAB) ;IS DEVICE ON LINE?
9255 006354 001404 BEQ 30S ;YES, TEST PARITY ERROR
9256
9257
9258
9259 006356 052763 000100 000002
9260 006364 000412
9261 006366 032704 100000 30S: BIS #INTREQ,SSENSE(DTAB) ;SET INTERVENTION REQUIRED IN SENSE BYTE
9262 006372 001404 BIT 50S ;FINISH UP CHANNEL INITIATED SELECTION PROCESS
9263 BEQ 40S ;PARER,TT1 ;WAS A PARITY ERROR DETECTED?
9264 ;NO, MUST BE ILLEGAL COMMAND
9265
9266
9267 006374 052763 000040 000002
9268 006402 000403
9269
9270
9271
9272
9273 006404 052763 000200 000002 40S: BIS #SCMDRJ,SSENSE(DTAB) ;SET CMD REJ FLAG
9274
9275
9276
9277
9278 006412 120037 013064 50S: CMPB DEV,CMDCHF ;DOES DEVICE HAVE COMMAND CHAINING SPECIFIED?
9279 006416 001002 BNE 60S ;NO, GET NEXT TUMBLE TABLE ENTRY
9280 006420 105037 013064 CLRB CMDCHF ;YES, CLEAR THE COMMAND CHAINING FLAG
9281 006424 000137 005352 60S: JMP LOOP ;AND GET THE NEXT T/T ENTRY
9282
9283
9284
9285
9286
9287
9288
9289
9290 006430 105037 013060 PCUEND: CLRB DXACT ;CLEAR DX ACTIVE FLAG
9291 006434 004737 010330 JSR PC,MUXEND ;HANDLE MUX DATA TRANSFER COMPLETION
9292 006440 103017 BCC PCHEX ;IF SEL CHAN OR MUX D/T NOT DONE, MERELY EXIT
9293
9294
9295
9296
9297 006442 000404 BR PCHEN1
9298
9299
9300
9301
9302
9303
9304
9305 006444 105037 013060 PCHEND: CLRB DXACT ;CLEAR DX ACTIVE FLAG
9306 006450 004737 010330 JSR PC,MUXEND ;IF MUX CHANNEL HANDLE DATA TRANSFER

```

```

COMMAND WAS REJECTED, DETERMINE WHY
DEVICE OFF-LINE -- RESPOND INTERVENTION REQUIRED SENSE CONDITION
COMMAND WAS REJECTED BECAUSE OF A PARITY ERROR
SET UP BUS OUT SENSE RESPONSE
INVALID COMMAND RECEIVED FROM 360
SET UP COMMAND REJECT SENSE RESPONSE
CHANNEL INITIATED SELECTION SEQUENCE COMMON EXIT LOGIC
IF SELECTED DEVICE HAS COMMAND CHAINING IN AFFECT -- KILL IT
CONTROL UNIT END OF DATA TRANSFER WAS DETECTED
IF TRANSFER COMPLETE PREPARE ENDING SEQ RESP
IF TRANSFER INCOMPLETE INCR BUFFER ADDRESS
DECR BYTE COUNT
MUX DATA TRANSFER COMPLETE
TREAT SAME AS SEL CHANNEL DONE
CHANNEL END WAS DETECTED
PREPARE ENDING SEQUENCE RESPONSE

```

```

9307 006454 123727 013056 000104 PCHEX: CMPB DXSTPF, #'D ;STOP ON DATA TRANSFER DONE?(SD)
9308 006462 001416 BEQ STPDX ;YES, DISABLE DX
9309 006464 112763 000003 000000 MOVB #CEDE, SCMD(DTAB); QUE END SEQ RESPONSE
9310 006472 167763 003770 000014 SUB #DXBC, SRBYTC(DTAB); SAVE REMAINING BYTE COUNT
9311 006500 032704 100000 PCHEX: BIT #PARER, TT1 ;WAS A PARITY ERROR SENSED?
9312 006504 001403 BEQ 10S ;NO, PROCESS NEXT TUMBLE TABLE ENTRY
9313 006506 152763 000020 000002 BISB #EQCHK, SSENSE(DTAB); YES, SET EQUIP CHECK IN SENSE
9314 006514 000137 005352 10S: JMP LOOP ;LOOP BACK + PROCESS NEXT TT ENTRY
9315
9316
9317
9318
9319
9320
9321 006520 105037 013056 STPDX: CLRB DXSTPF ;CLEAR STOP FLAG
9322
9323
9324
9325
9326
9327
9328 006524 042777 001100 003726 DXAB: BIC #DXONLN!DXENB, #DXCS ;DISABLE THE DX
9329 006532 105237 013062 INCB DXABFL ;SET THE DX ABORT FLAG SO THE
9330
9331 006536 000137 010200 JMP DXEXIT ;DX REGISTERS WILL BE PRINTED
;EXIT FROM INTERRUPT
    
```

9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386

006542 105737 012516
006546 001002
006550 000137 007176

006554 013700 013066
006560 004737 010216
006564 105763 000000
006570 001030
006572 105737 013064
006576 001402
006600 000137 010200
006604 126327 000017 000001 30S:
006612 001004
006614 112763 000011 000000
006622 000413
006624 005200 40S:
006626 120037 013057
006632 003402
006634 012700 000001
006640 020037 013066 50S:
006644 001345
006646 000137 010200

006652 116304 000000 60S:
006656 005304
006660 006304
006662 010037 013066
006666 000174 006672
006672 006716
006674 007006
006676 010030
006700 010126
006702 006716
006704 007106
006706 007756
006710 007736
006712 007746
006714 007006

.SBTTL DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)
EXECUTE NEXT COMMAND FOR THE DX
DXEXEC: TST9 CHTYPE ;CHANNEL TYPE 0=M, 1=5
BNE SEX ;SELECTOR CHANNEL EXEC
JMP MEX ;MULTIPLEXER EXEC

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
SEX: MOV SELDEV,DEV ;GET SEL DEV #
10S: JSR PC,SUDEV ;SET UP DEV STATUS TABLE ADDR
TSTB SCMD(DTAB) ;ANY JOB TO DO?
BNE 60S ;YES, EXECUTE IT
TSTB CMOCHF ;WAS COMMAND CHAINING SPECIFIED
BEQ 30S ;NO
JMP DXEXIT ;YES, WAIT FOR COMMAND
30S: CMPB SDRQ(DTAB),#1 ;IS ATTENTION TO BE SENT?
BNE 40S ;NO, CONTINUE
MOVB #11,SCMD(DTAB) ;YES, SET UP TO SEND THE ATTENTION
BR 60S ;FOR THE READ MANUAL INPUT
40S: INC DEV ;TO NEXT DEV
CMPB DEV,MAXDEV ;HAVE WE TRIED THE HIGHEST DEVICE?
BLE 50S ;NO
MOV #1,DEV ;YES, RESTART AT FIRST DEVICE
50S: CMP DEV,SELDEV ;IS THIS WHERE IT ALL STARTED?
BNE 10S ;NOPE, TEST THIS DEVICE
JMP DXEXIT ;EXIT -- NO TASKS PENDING

THERE IS A JOB TO DO, LETS DO IT
60S: MOVB SCMD(DTAB),R4 ;COMMAND TO INDEX
DEC R4 ;SCALE TO 0 - 11
ASL R4 ;MAKE WORD ADDRESS
MOV DEV,SELDEV ;SAVE CURRENT DEVICE ADDR
JMP @SCMDTB(R4) ;EXECUTE THE COMMAND
SCMDTB: .WORD SWRITE ;1 = WRITE FULL BUFFER
 .WORD SRMI ;2 = READ MANUAL INPUT
 .WORD ESEQ ;3 = ENDING SEQUENCE
 .WORD SENSCH ;4 = SENSE COMMAND
 .WORD SWRITE ;5 = WRITE LINE ADDRESS
 .WORD SREAD ;6 = READ FULL BUFFER
 .WORD ERASCH ;7 = ERASE COMMAND
 .WORD CONUNE ;10 = CONTROL UNIT END
 .WORD SATTN ;11 = SEND ATTENTION TO 360
 .WORD SSRMI ;12 = READ SHORT MANUAL INPUT

9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443

006716 016377 000006 003540
006724 163777 013074 003532
006732 013702 013100
006736 060002
006740 110277 003512
006744 012777 177037 003514
006752 116363 000000 000001
006760 105063 000002
006764 110037 013060
006770 005063 000014
006774 052777 000003 003456
007002 000137 010200

007006
007006 105737 012522
007012 001035
007014 105063 000002
007020 105763 000017
007024 001002
007026 000137 010030
007032 105063 000017
007036 016377 000020 003420
007044 016302 000010
007050 066302 000004
007054 005302
007056 167702 003402
007062 100002
007064 000137 010030
007070 005402
007072 010277 003370
007076 163777 013074 003360
007104 000411

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

.....
WRITE COMMAND RECEIVED FROM 360
PREPARE TO GET DATA FROM 360
BOTH WRITE AND WRITE LINE ADDRESS COME HERE
.....
SWRITE: MOV SINBF(DTAB), 2DXBA ;SET UP BUFFER ADDRESS
SUB PHYOFF, 2DXBA ;FOR VIRTUAL MEMORY -- OFFSET FOR PHYSICAL ADDRESS
MOV DEVCON, R2 ;COMPUTE DEVICE ADDRESS
ADD DEV, R2
MOVB R2, 2DXCA
MOV #-DISPSZ-1, 2DXBC ;SET UP BYTE COUNT FOR MAX, WRITE LINE ADDRESS
MOVB SCMD(DTAB), SLCMD(DTAB) ;SET WRITE FLAG
CLRB SSENSE (DTAB) ;CLEAR SENSE BYTE
MOVB DEV, DXACT ;SET DX ACTIVE FLAG
CLR SBYTC(DTAB) ;RESET REMAINING BYTE COUNT
BIS 2DXMR, 2DXCS ;START TRANSFER
JMP DXEXIT ;RETURN FROM INTERRUPT

.....
PERFORM READ MANUAL INPUT COMMANDS
.....
SRMI:
SSRMI: TSTB TSTTYP ;IS TEST FOR FRIEND?
BNE SREAD ;YES, TREAT ALL READS AS READ FULL BUFFER
CLRB SSENSE(DTAB) ;RESET THE SENSE BYTE
TSTB SRDRQ(DTAB) ;WAS A READ REQUESTED?
BNE 10S ;YES, CONTINUE
JMP ESEQ ;NO, TREAT AS A NOP -- END SEQ ONLY

10S: CLRB SRDRQ(DTAB)
MOV SMINS(DTAB), 2DXBA ;SET UP STARTING ADDRESS
MOV SOUTB(DTAB), R2 ;DETERMINE ENDING ADDRESS
ADD SCURS(DTAB), R2
DEC R2
SUB 2DXBA, R2 ;COMPUTE BYTE COUNT
BPL 20S ;INSURE VALID BYTE COUNT
JMP ESEQ ;ILLEGAL
20S: NEG R2
MOV R2, 2DXBC ;SET UP DX'S BYTE COUNT
SUB PHYOFF, 2DXBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
BR SRD10 ;START THE READ

.....
READ COMMAND RECEIVED FROM 360
PREPARE TO SEND DISP BUFFER TO 360
.....
SREAD: MOV SOUTB(DTAB), 2DXBA ;SET UP BUFFER ADDRESS
SUB PHYOFF, 2DXBA ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS

E08

MAINDEC-11-DZDXI-A NEW DX11-B RESPONDER
DZDXIA.CMB DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)

MACY11 27(732) 27-OCT-76 14:48 PAGE 68-2.

9444	007122	012777	177040	003336		MOV	#-DISPSZ,DXBC	;SET UP BYTE COUNT
9445	007130	116363	000000	000001	SRD10:	MOVB	SCMD(DTAB),SLCMD(DTAB)	;SAVE CODE OF LAST COMMAND
9446	007136	013702	013100			MOV	DEVCON,R2	;COMPUTE DEVICE ADDRESS
9447	007142	060002				ADD	DEV,R2	
9448	007144	110277	003306			MOVB	R2,DXCA	
9449	007150	105063	000002			CLRB	SENSE(DTAB)	;CLEAR SENSE BYTE
9450	007154	110037	013060			MOVB	DEV,DXACT	;SET DX ACTIVE FLAG
9451	007160	005063	000014			CLR	SRBYTC(DTAB)	;RESET REMAINING BYTE COUNT
9452	007164	052777	000005	003266		BIS	#DXRD,DXCS	;START TRANSFER
9453	007172	000137	010200			JMP	DXEXIT	;RETURN FROM INTERRUPT
9454								
9455								


```

9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471 007176 105737 013064
9472 007202 001402
9473 007204 000137 010200
9474 007210 013700 013066
9475 007214 004737 010216
9476 007220 105763 000000
9477 007224 001023
9478 007226 126327 000017 000001
9479 007234 001004
9480 007236 112763 000011 000000
9481 007244 000413
9482
9483
9484
9485
9486 007246 005200
9487 007250 120037 013057
9488 007254 003402
9489 007256 012700 000001
9490 007262 020037 013066
9491 007266 001352
9492 007270 000137 010200
9493
9494
9495
9496 007274 116304 000000
9497 007300 005304
9498 007302 006304
9499 007304 010037 013066
9500 007310 000174 007314
9501 007314 007340
9502 007316 007470
9503 007320 010030
9504 007322 010126
9505 007324 007340
9506 007326 007566
9507 007330 007756
9508 007332 007736
9509 007334 007746
9510 007336 007470

```

```

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

```

```

9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522 007340 005763 000014
9523 007344 001011
9524 007346 016363 000006 000012
9525 007354 163763 013074 000012
9526 007362 012763 000741 000014
9527 007370 016377 000012 003066
9528 007376 013702 013100
9529 007402 060002
9530 007404 110277 003046
9531 007410 012777 177774 003050
9532 007416 026327 000014 000004
9533 007424 002005
9534 007426 016302 000014
9535 007432 005402
9536 007434 010277 003026
9537 007440 105063 000002
9538 007444 116363 000000 000001
9539 007452 110037 013060
9540 007456 152777 000003 002774
9541 007464 000137 010200
9542
9543
9544
9545
9546
9547
9548
9549
9550 007470 105737 012522
9551 007474 001034
9552 007476 005763 000014
9553 007502 001031
9554 007504 105063 000002
9555 007510 105763 000017
9556 007514 001002
9557 007516 000137 010030
9558 007522 105063 000017
9559 007526 016363 000020 000012
9560 007534 016302 000010
9561 007540 066302 000004
9562 007544 005302
9563 007546 166302 000012
9564 007552 100761
9565 007554 010263 000014
9566 007560 163763 013074 000012
9567

```

```

.....
COMMANDS SPECIFICALLY FOR THE MULTIPLEXER CHANNEL
.....
WRITE COMMAND RECEIVED FROM 360
PREPARE TO GET 4 BYTES OF DATA FROM 360
NOTE--BOTH WRITE AND WRITE LINE ADDR COME HERE

MWRITE: 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 MANAG - OFFSET FOR PHY ADDRESS
        MOV #DISPSZ+1,SRBYTC(DTAB) ;SET UP BUFFER FOR MAX SIZE
10$: MOV SBUFA(DTAB),DXBA ;OUTPUT BUFFER ADDR TO DX
     MOV DEVCON,R2 ;COMPUTE DEVICE ADDRESS
     ADD DEV,R2
     MOVB R2,DXCA
     MOV #-4,DXBC ;START BYTE COUNT AT 4
     CMP SRBYTC(DTAB),#4 ;IS LESS THEN 4 BYTES LEFT?
     BGE 20$ ;NO, START TRANSFER
     MOV SRBYTC(DTAB),R2 ;YES, USE REMAINING BYTE COUNT
     NEG R2
     MOV R2,DXBC
20$: CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
     MOVB SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
     MOVB DEV,DXACT ;SET ACTIVE FLAG
     BISB #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 TSTTYP ;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 20$ ;YES, CONTINUE
10$: JMP ESEQ ;NO, RETURN AN ENDING SEQ RESP DE!CE
20$: CLRB SRDRQ(DTAB) ;CLEAR THE READ REQUEST
     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

```

```

9568      : FALL THROUGH TO NORMAL READ BUFFER ROUTINE
9569      :
9570      :
9571      :
9572      :
9573      : READ COMMAND RECEIVED FROM 360
9574      : PREPARE TO SEND 4 BYTES OF DATA TO THE 360
9575      :
9576 007566 116363 000000 000001 MREAD: MOVB SCMD(DTAB),SLCMD(DTAB) ;SAVE CODE OF LAST COMMAND FOR DISPLAY CONTROL
9577 007574 005763 000014          TST  SRBYTC(DTAB) ;READ IN PROGRESS?
9578 007600 001011          BNE  10$ ;YES, SEND OUT MORE DATA
9579 007602 016363 000010 000012 MOV  SOUTB(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
9580 007610 163763 013074 000012 SUB  PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS
9581 007616 012763 000740 000014 MOV  #DISPSZ,SRBYTC(DTAB) ;SET UP TOTAL BYTE COUNT
9582 007624 016377 000012 002632 10$: MOV  SBUFA(DTAB),DXBA ;SEND BUFFER ADDR TO DX
9583 007632 013702 013100          MOV  DEVCON,R2 ;COMPUTE DEVICE ADDR
9584 007636 060002          ADD  DEV,R2
9585 007640 110277 002612          MOVB R2,DXCA ;OUTPUT THE DEVICE ADDRESS
9586 007644 012777 177774 002614 MOV  #-4,DXBC ;OUTPUT THE BYTE COUNT -4-
9587 007652 026327 000014 000004 CMP  SRBYTC(DTAB),#4 ;SEE IF REMAINING BYTE COUNT LESS THAN 4
9588 007660 002005          BGE  20$
9589 007662 016302 000014          MOV  SRBYTC(DTAB),R2 ;SET UP BYTE COUNT
9590 007666 005402          NEG  R2
9591 007670 010277 002572          MOV  R2,DXBC ;OUTPUT THE NEW BYTE COUNT -- LT 4
9592 007674 105063 000002 20$: CLRB SSENSE(DTAB) ;CLEAR SENSE AND SET DX ACTIVE FLAG
9593 007700 110037 013060          MOVB DEV,DXACT ;SET DEVICE ACTIVE FLAG FOR SOFTWARE
9594      :
9595      :
9596      :
9597      : BEFORE TRANSMIT IS STARTED SET BUSY FLAG IN DX11 STATUS
9598      : TABLE FOR DEVICE
9599      :
9600      :
9601      :
9602      :
9603      :
9604      :
9598 007704 010002          MOV  DEV,R2 ;COMPUTE ADDRESS OF SPW ENTRY
9599 007706 063702 013100          ADD  DEVCON,R2 ; X
9600 007712 060202          ADD  R2,R2 ; X
9601 007714 063702 013104          ADD  STSPW,R2 ;ADD IN SPW BASE ADDRESS
9602 007720 052712 000020          BIS  #BSY,(R2) ;SET UNIT BUSY FLAG
9603 007724 152777 000005 002526 BISB #DXRD,DXCS ;START THE DX READING
9604 007732 000137 010200          JMP  DXEXIT

```

.SBTTL DX11-B ISR (MULTIPLEXER AND SELECTOR CHANNEL COMMANDS)

COMMANDS FOR BOTH MULTIPLEXER AND SELECTOR CHANNELS

PRESENT CONTROL UNIT END TO CHANNEL

007736 152763 000040 000003 CONUNE: BISB #CUE,SSTAT(DTAB) ;PUT IN STATUS BYTE
007744 000434 BR STOUT ;OUTPUT TO CHANNEL

SEND THE ATTENTION BIT TO THE 360

007746 152763 000200 000003 SATTN: BISB #ATTN,SSTAT(DTAB) ;PUT IN STATUS BYTE
007754 000430 BR STOUT ;OUTPUT TO THE 360

ERASE THE DISPLAY

007756 016304 000010 ERASCM: MOV SOUTB(DTAB),R4 ;SET UP BEG OF DISPLAY BUFFER
007762 012705 000740 MOV #DISPSZ,R5 ;SET UP COUNTER
007766 112702 000100 MOVB #EBCDSP,R2 ;SET BUFFER FILL FOR 2848 DIAG
007772 105737 012522 TSTB TSTTYP ;IS TEST BEING RUN FOR 2848 RESPONDER
007776 001402 BEQ 10S ;YES, FILL BUFFER WITH EBCDIC SPACE
010000 113702 012524 MOVB FILLCH,R2 ;NO, USE CURRENT FILL CHARACTER
010004 110224 10S: MOVB R2,(R4)+ ;MOVE FILL CHARACTER TO BUFFER
010006 005305 DEC R5 ;DECR COUNTER
010010 001375 BNE 10S ;NOT DONE, DO NEXT CHAR
010012 005063 000004 CLR SCURS(DTAB) ;RESET THE CURSOR
010016 105063 000002 CLRB SSENSE(DTAB) ;CLEAR SENSE BYTE
010022 112763 000003 000000 MOVB #CEDE,SCMD(DTAB);CHANGE COMMAND TO PRESENT END SEQ

FALL THROUGH TO PRESENT ENDING STATUS

PRESENT ENDING STATUS TO CHANNEL

010030 152763 000014 000003 ESEQ: BISB #CE!DE,SSTAT(DTAB) ;SET CH END + DEV END

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.

```

9662 010036 013702 013100      STOUT:  MOV     DEVCON,R2      ;OUTPUT DEVICE ADDRESS
9663 010042 060002                ADD     DEV,R2
9664 010044 110277 002406          MOVVB  R2,DXCA
9665 010050 132763 000002 000003    BITB  #UCHK,SSTAT(DTAB) ;IS THE UNIT CHECK BIT SET?
9666 010056 001403                BEQ    10$                ;NO, TRANSMIT THE STATUS
9667 010060 112763 000002 000003    MOVVB  #UCHK,SSTAT(DTAB) ;YES, THEN SEND ONLY UNIT CHECK
9668
9669      ;
9670      ; IF MULTIPLEXER CHANNEL
9671      ; CLEAR ANY PENDING STATUS IN SPW STATUS ENTRY
9672      ; (PROBABLY "BUSY")
9673 010066 105737 012516      10$:  TSTB  CHTYPE                ;SELECTOR CHANNEL?
9674 010072 001004                BNE   20$                ;YES, DON'T CLEAR STATUS IN SPW TABLE
9675 010074 060202                ADD   R2,R2              ;COMPUTE ADDRESS OF SPW STATUS ENTRY
9676 010076 063702 013104      ADD   STSPW,R2          ;OFFSET BY BASE OF SPW TABLE
9677 010102 105012                CLRB  (R2)              ;CLEAR SPW STATUS ENTRY
9678
9679      ;
9680      ; OUTPUT THE STATUS TO THE CHANNEL
9681 010104 116377 000003 002350    20$:  MOVVB  SSTAT(DTAB),DXOS ;OUTPUT STATUS TO CHANNEL
9682 010112 152777 000007 002340    BISB  #DXST,DXCS       ;PRESENT TO CHANNEL
9683 010120 110037 013060      MOVVB  DEV,DXACT       ;SET DX ACTIVE FLAG
9684 010124 000425                BR    DXEXIT           ;RETURN FROM INTERRUPT
9685
9686      ;
9687      ;
9688      ; SENSE COMMAND DESIRED BY 360
9689      ;
9690 010126 012777 000002 002330    SENSEM: MOV   #SSENSE,DXBA ;SET UP ADDRESS OF SENSE BYTE
9691 010134 060377 002324          ADD   DTAB,DXBA
9692 010140 163777 013074 002316    SUB   PHYOFF,DXBA      ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
9693 010146 013702 013100          MOV   DEVCON,R2       ;COMPUTE DEVICE ADDRESS
9694 010152 060002                ADD   DEV,R2
9695 010154 110277 002276          MOVVB R2,DXCA
9696 010160 012777 177777 002300    MOV   #-1,DXBC        ;TRANSFER 1 BYTE
9697 010166 110037 013060          MOVVB DEV,DXACT       ;SET DX ACTIVE FLAG
9698 010172 052777 000005 002260    BIS   #DXRD,DXCS      ;START TRANSFER
9699
9700      ;
9701      ;
9702      ; EXIT FROM THE DX ISR
9703      ;
9704 010200 012605      DXEXIT: MOV   (SP)+,R5      ;RESTORE REGISTERS
9705 010202 012604          MOV   (SP)+,R4
9706 010204 012603          MOV   (SP)+,R3
9707 010206 012602          MOV   (SP)+,R2
9708 010210 012601          MOV   (SP)+,R1
9709 010212 012600          MOV   (SP)+,R0
9710 010214 000002          RTI

```

```

9712          .SBTTL DX11-B ISR (UTILITY SUBROUTINES)
9713          :
9714          SET UP ADDR OF DEVICE STATUS TABLE
9715          :
9716          CALLING SEQUENCE
9717          ..... RO = DEV #
9718          JSR   PC,SUDEV
9719          ..... RETURN
9720          ..... R3 = ADDRESS OF DEVICE TABLE
9721          :
9722          ONLY REGISTER R3 IS MODIFIED BY THIS SUBROUTINE
9723          :
9724          SUDEV: MOV   SDEVTB,DTAB      ;START AT DEV 1
9725                MOV   R1,-(SP)        ;SAVE R1
9726                MOV   DEV,R1
9727          SUD10: DEC   R1              ;DEC DEVICE NUMBER
9728                BEQ   SUDEX          ;DONE, EXIT
9729                ADD   #2000,DTAB      ;INCR TO NEXT DEV TABLE
9730                BR    SUD10          ;TRY AGAIN
9731          SUDEX: MOV   (SP)+,R1       ;RETURN TO CALLER
9732                RTS   PC
9733          :
9734          CLEAR DEVICE STATUS TABLE
9735          :
9736          CALLING SEQUENCE
9737          ..... RO = DEV #
9738          JSR   PC,CDEVST
9739          ..... RETURN
9740          ..... R3 = ADDRESS OF DEVICE TABLE
9741          THE FOLLOWING TABLE ENTRIES ARE CLEARED
9742          SCMD
9743          SSTAT
9744          SBUFA
9745          SRBYTC
9746          SLCMD
9747          :
9748          ONLY REGISTER R3 IS AFFECTED BY THIS SUBROUTINE
9749          :
9750          CDEVST: JSR   PC,SUDEV        ;SET UP ADDR OF DEVICE STAT TABLE
9751                CLRB  SCMD(DTAB)      ;RESET CURRENT COMMAND ENTRY
9752                CLRB  SSTAT(DTAB)     ;RESET DEVICE STATUS ENTRY
9753                CLR   SBUFA(DTAB)     ;RESET CURRENT BUFFER ADDRESS POINTER
9754                CLR   SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
9755                CLRB  SLCMD(DTAB)     ;RESET LAST COMMAND ENTRY
9756                RTS   PC              ;RETURN TO THE CALLER
9757          :
9758          CSPWST -- CLEAR SPW STATUS BYTE
9759          :
9760          CALLING SEQUENCE
9761          .....DTAB (R3) POINTS TO CURRENT DEVICE STATUS TABLE
9762          .....DEV (RO) CONTAINS CURRENT DEVICE NUMBER
9763          JSR   PC,CSPWST
9764
9765
9766
9767

```

```

9768
9769
9770
9771
9772 010276 010546
9773 010300 105763 000016
9774 010304 001007
9775 010306 010005
9776 010310 063705 013100
9777 010314 060505
9778 010316 063705 013104
9779 010322 105015
9780 010324 012605
9781 010326 000207

```

.....RETURN TO CALLER WITH DEVICE STATUS BYTE RESET
 ALL REGISTERS ARE PRESERVED ACCROSS THIS SUBROUTINE

```

CSPWST: MOV R5, -(SP) ;SAVE REGISTER FOR SUBROUTINE USAGE
        TSTB SONLF(DTAB) ;IS DEVICE ON-LINE?
        BNE IOS ;NO, JUST EXIT
        MOV DEV, R5 ;GET DEVICE NUMBER AND COMPUTE
        ADD DEVCON, R5
        ADD R5, R5 ;ADDRESS OF SPW STATUS BYTE
        ADD STSPW, R5
        CLRB (R5) ;RESET SPW STATUS BYTE
IOS: MOV (SP)+, R5 ;RESTORE REGISTER
      RTS PC

```

```

9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794 010330 105737 012516
9795 010334 001006
9796
9797
9798
9799 010336 162763 000004 000014
9800 010344 003004
9801
9802
9803
9804 010346 005063 000014
9805 010352 000261
9806 010354 000404
9807
9808
9809
9810 010356 062763 000004 000012
9811 010364 000241
9812 010366 000207
    
```

MUXEND -- HANDLE DATA TRANSFER COMPLETIONS FOR MUX

CALLING SEQUENCE
R3 (DTAB) CONTAINS THE ADDRESS OF THE DEVICE STATUS TABLE
 JSR PC, MUXEND
RETURN C-BIT SET - MUX DATA TRANS DONE
 C-BIT RESET - SEL CHAN OR DATA TRANSFER NOT DONE

NO REGISTERS ARE AFFECTED BY THIS SUBROUTINE

MUXEND: TSTB CHTYPE ;SELECTOR OR MULTIPLEXER CHANNEL??
 BNE SS ;SELECTOR CHANNEL -- EXIT

MULTIPLEXER CHANNEL

SUB #4, SRBYTC(DTAB) ;DECR REAMINING BYTE COUNT
 BGT 10\$;IF > 1, DATA TRANSFER NOT COMPLETE YET

DATA TRANSFER COMPLETE ON MUX CHANNEL

CLR SRBYTC(DTAB) ;INSURE REMAINING BYTE COUNT ZERO
 SEC ;SET MUX TRANSFER COMPLETE FLAG
 BR 30\$;GOTO COMMON EXIT

DATA TRANSFER INCOMPLETE

10\$: ADD #4, SBUFA(DTAB) ;BUMP BUFFER ADDRESS
 20\$: CLC ;RESET FLAG TO INDICATE MUX CHAN NOT DONE
 30\$: RTS PC ;RETURN TO THE CALLER

9814
9815
9816
9817
9818
9819
9820
9821
9822
9823
9824
9825
9826
9827
9828
9829
9830
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841 010370 010046
9842 010372 010146
9843 010374 010246
9844 010376 126327 000001 000002
9845 010404 001535
9846 010406 126327 000001 000012
9847 010414 001542
9848 010416 126327 000001 000006
9849 010424 001522
9850
9851
9852
9853
9854
9855 010426 105737 012522
9856 010432 001102
9857
9858
9859
9860 010434 016301 000006
9861 010440 126327 000001 000005
9862 010446 001016
9863
9864
9865
9866
9867 010450 005263 000014
9868 010454 112102
9869 010456 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 EMULATE A 2260'S DISPLAY. THIS ALLOWS THIS PROGRAM TO BE USED WITH THE 2848 RESPONDER DIAGNOSTIC.

NOTE -- THE REMAINING BYTE COUNT (SRBYTC) IS USED TO INDICATE THE NUMBER OF CHARACTERS RECEIVED FROM THE CHANNEL. IT IS SET UP AT THE COMPLETION OF AN I/O TRANSFER TO THE NUMBER OF CHARACTERS REMAINING IN THE DX BYTE COUNT REGISTER.

NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE

DISCTL: MOV R0,-(SP) ;SAVE REGSITERS USED BY SUBROUTINE
MOV R1,-(SP)
MOV R2,-(SP)
CMPB SLCMD(DTAB),#CMRMI ;WAS IT A READ MANUAL INPUT COMMAND?
BEQ DRMI ;IF YES, PERFORM READ MANUAL INPUT PROCEEDURE
CMPB SLCMD(DTAB),#CMSRMI ;WAS IT A SHORT READ MANAUL INPUT?
BEQ DSRMI ;IF YES, EXIT
CMPB SLCMD(DTAB),#CMREAD ;WAS IT A READ FULL BUFFER COMMAND?
BEQ DREAD ;YES, RESET CURSOR ON READ FULL BUFFER

THE COMMAND MUST HAVE BEEN A 360 WRITE
DETERMINE TYPE OF TEST BEING RUN

TSTB TSTTYP ;TYPE OF TEST 0 = 2848 1 = FRIEND
BNE DISFRN ;FRIEND

FORMAT DISPLAY ALA 2260

MOV SINBF(DTAB),R1 ;GET ADDR OF START OF INPUT
CMPB SLCMD(DTAB),#CMWTLA ;WAS LAST CMD A WRITE LINE ADDRESS?
BNE 20\$;NO, NORMAL WRITE

WRITE LINE ADDRESS COMMAND

FIRST BYTE OF DATA BLOCK IS CURSOR LINE ADDRESS

INC SRBYTC(DTAB) ;INCR BYTE COUNT
MOVB (R1)+,R2 ;GET LINE NUMBER
BIC #177760,R2 ;GET ONLY LINE NUMBER

```

9870 010462 005063 000004          CLR      SCURS(DTAB)      ;
9871                                     :
9872                                     :
9873                                     :
9874 010466 005702          10S:   TST      R2              ;DONE?
9875 010470 001405          BEQ      20S              ;YES, MORE DATA INTO DISPLAY BUF
9876 010472 062763 000050 000004   ADD      #LINSZ,SCURS(DTAB) ;INCR TO NEXT LINE
9877 010500 005302          DEC      R2              ;DECR LINE COUNT
9878 010502 000771          BR       10S
9879                                     :
9880                                     :
9881                                     :
9882                                     :
9883                                     :
9884                                     :
9885                                     :
9886                                     :
9887                                     :
9888 010504 016302 000010          20S:   MOV      SOUTB(DTAB),R2 ;COMPUTE DISPLAY ADDR
9889 010510 066302 000004          ADD      SCURS(DTAB),R2
9890 010514 026327 000014 000741   CMP      SBYTC(DTAB),#DISPSZ+1 ;ALL CHARS PROCESSED?
9891 010522 103077          BHS     DISCEX           ;YES, EXIT
9892 010524 005263 000014          INC      SBYTC(DTAB)     ;INCREMENT THE BYTE COUNT
9893 010530 112100          MOV     (R1)+,R0        ;GET THE NEXT BYTE RECEIVED AND BUMP POINTER
9894 010532 042700 177400          BIC     #177400,R0      ;STRIP SIGN EXTENSION BITS (IF ANY)
9895 010536 116000 011744          MOV     EBCDTB(R0),R0   ;FOLD CHARACTER INTO ASCII CHARACTER SET
9896 010542 042700 177400          BIC     #177400,R0      ;STRIP SIGN EXTENSION BITS, IF ANY
9897 010546 162700 000040          SUB     #40,R0          ;SCALE INTO ASCII TABLE RANGE
9898 010552 116012 012344          MOV     ATOETB(R0),(R2) ;COMPLETE FOLDING BY RETRANSLATING TO EBCDIC
9899 010556 005263 000004          INC     SCURS(DTAB)     ;INCR CURSOR PTR
9900 010562 121227 000025          CMP     (R2),#NEWLINE   ;WAS A NEW LINE SPECIFIED?
9901 010566 001015          BNE     60S
9902                                     :
9903                                     :
9904                                     :
9905                                     :
9906 010570 005002          40S:   CLR      R2              ;CLEAR LINE CTR
9907 010572 005202          INC     R2              ;INCR LINE CTR
9908 010574 162763 000050 000004   SUB     #LINSZ,SCURS(DTAB)
9909 010602 003373          BGT     40S              ;KEEP DIVIDING
9910 010604 005063 000004          CLR     SCURS(DTAB)     ;CLEAR CURSOR
9911 010610 062763 000050 000004   ADD     #LINSZ,SCURS(DTAB)
9912 010616 005302          DEC     R2
9913 010620 001373          BNE     50S
9914                                     :
9915                                     :
9916                                     :
9917 010622 026327 000004 000740   60S:   CMP     SCURS(DTAB),#DISPSZ ;CURSOR OVERFLOW DISPLAY BUFFER?
9918 010630 002725          BLT     20S              ;CURSOR OK, PROCESS NEXT CHAR
9919 010632 005063 000004          CLR     SCURS(DTAB)     ;OVERFLOW, RESTART CURSOR AT POS 0
9920 010636 000722          BR      20S
9921                                     :
9922                                     :
9923                                     :
9924                                     :
9925                                     :

```

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

```

9926
9927 010640 105737 012523      DISFRN: TSTB  IOBUF      ;SEPARATE I/O BUFFERS?
9928 010644 001026              BNE  DISCEX      ;YES, DON'T COPY INPUT TO OUTPUT
9929 010646 016301 000006      MOV  SINBF(DTAB),R1 ;SET UP INPUT BUFFER ADDRESS
9930 010652 016302 000010      MOV  SOUTB(DTAB),R2 ;SET UP OUTPUT BUFFER ADDRESS
9931 010656 012700 000360      MOV  #DISPSZ/2,R0  ;TRANSFER THE INPUT BUFFER TO THE OUTPUT BUFFER
9932
9933      ;
9934      ; PERFORM COPY
9935 010662 012122      IOS:  MOV  (R1)+,(R2)+ ;INPUT TO OUTPUT
9936 010664 005300      DEC  R0           ;ARE WE DONE?
9937 010666 001375      BNE  IOS         ;NO, CONTINUE COPY
9938 010670 000414      BR   DISCEX      ;PREPARE TO RETURN TO CALLER
9939
9940      ;
9941      ; A READ FULL BUFFER WAS PERFORMED
9942      ; THE CURSOR MUST BE RESET TO THE BEGINNING OF THE SCREEN
9943
9944 010672 005063 000004      DREAD: CLR  SCURS(DTAB) ;RESET THE CURSOR
9945 010676 000411      BR   DISCEX      ;AND PREPARE TO EXIT
9946
9947      ;
9948      ;
9949      ;
9950      ; A READ MANUAL INPUT WAS PERFORMED
9951      ; TO EMULATE THE 2260 SCREEN THE START OF MANUAL INPUT CHARACTER
9952      ; MUST BE DELETED FROM THE SCREEN
9953
9954 010700 016301 000020      DRMI: MOV  SMINS(DTAB),R1 ;GET THE STARTING ADDRESS
9955 010704 005301      DEC  R1           ;DECREMENT TO THE SMI CHAR
9956 010706 112711 000100      MOVB #EBCDSP,(R1) ;BLANK OUT THE CHARACTER
9957 010712 166301 000010      SUB  SOUTB(DTAB),R1 ;AND COMPUTE THE CURSOR POSITION
9958 010716 010163 000004      MOV  R1,SCURS(DTAB)
9959
9960      ;
9961      ;
9962      ; A SHORT READ MANUAL INPUT WAS PERFORMED
9963      ; NO ACTION REQUIRED BY DISPLAY CONTROL ROUTINE
9964
9965 010722      DSRMI:
9966
9967      ;
9968      ;
9969      ; RESTORE REGISTERS AND RETURN TO CALLER
9970
9971 010722 012602      DISCEX: MOV  (SP)+,R2 ;RESTORE SAVED REGISTERS
9972 010724 012601      MOV  (SP)+,R1
9973 010726 012600      MOV  (SP)+,R0
9974 010730 000207      RTS   PC         ;RETURN TO THE CALLER

```

9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997
9998
9999
10000
10001
10002
10003
10004 010732 010046
10005 010734 010146
10006 010736 017700 001504
10007 010742 042700 177600
10008 010746 013701 012732
10009 010752 020027 000020
10010 010756 001002
10011 010760 000137 001002
10012 010764 020027 000023
10013 010770 001003
10014 010772 105237 013063
10015 010776 000503
10016 011000 020027 000021
10017 011004 001010
10018 011006 105037 013063
10019 011012 105737 013061
10020 011016 001473
10021 011020 004737 011240
10022 011024 000470
10023 011026 020027 000003
10024 011032 001011
10025 011034 105737 013043
10026 011040 001457
10027 011042 105237 013044
10028 011046 012701 012630
10029 011052 000137 011206
10030 011056 105737 013043
10031 011062 001051

.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 DUMPS
(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

```

TKIN:  MOV    RD, -(SP)          ;SAVE REGISTERS
        MOV    RI, -(SP)
        MOV    @TKB, RD      ;GET TELE CHARACTER
        BIC    @177600, RD   ;INSURE 7-BIT ASCII
        MOV    TPTR, RI     ;BUFFER PTR
        CMP    RD, @CTL.P   ;CONTROL -P ?
        BNE    3$          ;NO
        JMP    RSTART      ;YES, ALLOW OPERATOR TO REENTER ALL PARAMETERS
3$:     CMP    RD, @CTL.S   ;CONTROL-S, TEMPORARILY STOP CONSOLE OUTPUT?
        BNE    6$          ;NO, CONTINUE
        INCB   TTYSTP      ;YES, SET FLAG TO STOP TTY OUTPUT
        BR    100$         ;AND EXIT FROM INTERRUPT
6$:     CMP    RD, @CTL.Q   ;CONTROL-Q, RESUME CONSOLE OUTPUT?
        BNE    10$        ;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$:    CMP    RD, @CTL.C   ;COMMAND ABORT -- CTL C?
        BNE    20$        ;NO
        TSTB  TCMACT      ;IS A COMMAND ACTIVE?
        BEQ   90$         ;NO, TREAT AS A DELETE LAST LINE
        INCB  TCMDBAB     ;YES, SET ABORT FLAG
        MOV   @TBUF, RI   ;SET UP BUFFER POINTER
        JMP   100$        ;EXIT
20$:    TSTB  TCMACT      ;TELE CMD CURRENTLY ACTIVE?
        BNE   100$        ;YES, IGNORE CHARACTER
    
```

10032	011064	110021		MOV B	RO, (R1)+	: STORE CHAR INTO BUFFER - INC PTR
10033	011066	020027	000015	CMP	RO, #CR	: LINE DELIMITER -- C/R?
10034	011072	001005		BNE	30\$: NO
10035	011074	012701	012630	MOV	#TBUF, R1	: RESET BUFFER PTR
10036	011100	105237	013043	INCB	TCHAR	: YES, SET COMMAND ACTIVE FLAG
10037	011104	000440		BR	100\$: DON'T PRINT THE LINE DELIMITER
10038	011106	020027	000177	30\$:	CMP	RO, #RUBOUT
10039	011112	001002		BNE	40\$: A RUBOUT?
10040	011114	012700	000137	MOV	#'+, RO	: NOPE
10041	011120	120027	000025	40\$:	CMPB	RO, #CTL.U
10042	011124	001002		BNE	50\$: YES, TREAT AS A DELETE LAST CHARACTER
10043	011126	112700	000134	MOV B	#'\, RO	: CONTROL-U? (DELETE CURRENT INPUT LINE)
10044	011132	004737	011362	50\$:	JSR	PC, PCHAR
10045	011136	020027	000137	CMP	RO, #'+	: ECHO THE CHARACTER BACK
10046	011142	001004		BNE	60\$: DELETE LAST CHAR -- BACK ARROW?
10047	011144	124141		CMPB	-(R1), -(R1)	: NO
10048	011146	020127	012630	CMP	R1, #TBUF	: YES, DECR POINTER BY 2
10049	011152	003403		BLE	70\$: ARE WE BEYOND BEG OF THE BUFFER?
10050	011154	020027	000134	60\$:	CMP	RO, #'\'
10051	011160	001004		BNE	80\$: YES, RESET TO BEG OF BUFFER
10052	011162	012701	012630	70\$:	MOV	#TBUF, R1
10053	011166	004737	011324	JSR	PC, CRLF	: DELETE CUR LINE -- BACK SLASH?
10054	011172	020127	012730	80\$:	CMP	R1, #TBUFE
10055	011176	001003		BNE	100\$: NO
10056	011200	012700	000134	90\$:	MOV	#'\, RO
10057	011204	000740		BR	30\$: YES, RESET BUFFER PTR
10058	011206	010137	012732	100\$:	MOV	R1, TPTR
10059	011212	012601		MOV	(SP)+, R1	: NEW LINE FOR NEW COMMAND
10060	011214	012600		MOV	(SP)+, RO	: WERE LIMITS EXCEEDED?
10061	011216	000002		RTI		: NOPE, EXIT
						: THEY WERE -- TREAT AS A LINE ABORT
						: SAVE BUFFER PTR
						: RESTORE REGISTERS + EXIT

10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073

011220 105037 013042
011224 105737 013061
011230 001402
011232 004737 011240
011236 000002

.....

PISR:

IOS:

CLRB
TSTB
BEQ
JSR
RTI

PIUFL
PCTR
IOS
PC,PROUT

.SBTTL TELETYPE (CONSOLE) OUTPUT ISR
TELETYPE OUTPUT DRIVER (ISR) -- PRINT
CONTROL PASSES HERE ON A TELE OUT INTERRUPT

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

10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
10100
10101
10102
10103
10104
10105
10106
10107
10108
10109
10110
10111
10112
10113
10114
10115
10116
10117
10118
10119
10120
10121
10122
10123
10124
10125
10126
10127
10128
10129
10130

011240 105737 013042
011244 001026
011246 105737 013063
011252 001023
011254 105237 013042
011260 117777 001552 001164
011266 012777 000100 001154
011274 005237 013036
011300 023727 013036 013036
011306 001003
011310 012737 012734 013036
011316 105337 013061
011322 000207

011324 010246
011326 012702 105215
011332 004737 011342
011336 012602
011340 000207

.SBTTL TELETYPE OUTPUT HANDLING SUBROUTINES
SEND DATA TO PRINTER, IF NOT BUSY
CALLING SEQUENCE
JSR PC,PROUT
.....RETURN
IF TELETYPE OUTPUT IS CURRENTLY IN PROGRESS OR HAS BEEN SUSPENDED BY A CONTROL -
CONTROL IS RETURNED IMMEDIATELY WITH NO ACTION BEING INITIATED.
IF TELETYPE OUTPUT IS NOT CURRENTLY IN PROGRESS
THE PRINTER BUSY FLAG IS SET AND A CHARACTER IS SENT TO THE TERMINAL
NO REGSISTERS 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
MOVB @PFPTR,@TPB ; OUTPUT NEXT CHAR
MOV #100,@TPS ; ENABLE INTERRUPTS
INC PFPTR ; INCR PUNCH POINTER
CMP PFPTR,#PBFE ; TIME TO WRAP AROUND?
BNE 10\$; NO, EXIT
MOV #PBFS,PFPTR ; YES, RESTORE TO START OF BUFFER
10\$: DECB PCTR ; DECR CHAR COUNTER
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

10131			
10132	011342	010237	011356
10133	011346	004137	011466
10134	011352	011356	
10135	011354	000207	
10136	011356	000000	
10137	011360	377	377
10138			
10139			
10140			
10141			
10142			
10143			
10144			
10145			
10146			
10147			
10148			
10149			
10150	011362	110037	011376
10151	011366	004137	011466
10152	011372	011376	
10153	011374	000207	
10154	011376	000	377

```

PRINT2: MOV R2,P2BF
        JSR R1,MSG
        .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,MSG
        .WORD P1BF
        RTS PC
P1BF:   .BYTE 0,377
;RETURN TO THE CALLER

```


10156
10157
10158
10159
10160
10161
10162
10163
10164
10165
10166
10167 011400 010246
10168 011402 010346
10169 011404 013703 013040
10170 011410 121227 000377
10171 011414 001417
10172 011416 112223
10173 011420 105237 013061
10174 011424 020327 013036
10175 011430 001002
10176 011432 012703 012734
10177 011436 004737 011240
10178 011442 123737 013061 013120
10179 011450 001774
10180 011452 000756
10181 011454 010337 013040
10182 011460 012603
10183 011462 012602
10184 011464 000207
10185
10186
10187
10188
10189
10190
10191
10192
10193
10194
10195
10196
10197
10198
10199 011466 010246
10200 011470 012102
10201 011472 004737 011400
10202 011476 012602
10203 011500 000201
10204
10205
10206
10207
10208
10209
10210
10211

```

PRMSG PRINT A CHARACTER STRING

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

NOTE -- MESSAGE MUST BE TERMINATED BY A 377

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

MSG -- PRINT A CHARACTER STRING ON THE SYSTEM CONSOLE

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

NOTE -- MESSAGE MUST BE TERMINATED BY A 377

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

INMES PRINT A CHARACTER STRING

CALLING SEQUENCE
JSR R1,INMES
.WORD ADDRESS OF MESSAGE

```

```

10212
10213
10214
10215
10216
10217
10218
10219
10220
10221
10222 011502 010246
10223 011504 113746 013120
10224 011510 112737 000377 013120
10225 011516 012102
10226 011520 004737 011400
10227 011524 112637 013120
10228 011530 012602
10229 011532 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)
MOV B PMAX, -(SP) ;CHEAT, SAVE PMAX
MOV B #377, PMAX ;AND MAKE VERY LARGE
MOV (R1)+, R2
JSR PC, PRMSG ;USE STANDARD MESSAGE PROCESSOR
MOV B (SP)+, PMAX ;RESTORE PRINT MAX
MOV (SP)+, R2
RTS R1 ;RETURN TO CALLER

```

```

10231
10232
10233
10234
10235
10236
10237
10238
10239
10240
10241
10242
10243
10244
10245
10246 011534 005003
10247 011536 005005
10248 011540 112204
10249 011542 120427 000067
10250 011546 003013
10251 011550 120427 000060
10252 011554 002410
10253 011556 042704 177770
10254 011562 006303
10255 011564 006303
10256 011566 006303
10257 011570 060403
10258 011572 005205
10259 011574 000761
10260 011576 000207
    
```

```

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

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

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

```

10262
10263
10264
10265
10266
10267
10268
10269
10270
10271
10272
10273
10274
10275 011600 005003
10276 011602 005005
10277 011604 112204
10278 011606 162704 000060
10279 011612 100422
10280 011614 020427 000012
10281 011620 002410
10282 011622 162704 000007
10283 011626 020427 000020
10284 011632 002012
10285 011634 020427 000012
10286 011640 002407
10287 011642 006303
10288 011644 006303
10289 011646 006303
10290 011650 006303
10291 011652 060403
10292 011654 005205
10293 011656 000752
10294 011660 005302
10295 011662 112204
10296 011664 000207

```

```

      CHTB -- CONVERT ASCII HEX TO BINARY
      CALLING SEQUENCE
      JSR PC, CHTB
      RETURN
      UPON RETURN
      R2 = NEXT CHAR POSITION NOT CONVERTED
      R3 = BINARY RESULT
      R4 = (BITS 0-7) FIRST NON-HEX CHARACTER
      R5 = NUMBER OF CHARACTERS CONVERTED

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

```

```

10298 .SBTTL PROCESSOR ERROR TRAP HANDLERS
10299
10300 TRAP OUT ROUTINES
10301
10302
10303
10304
10305
10306
10307 011666 012702 014133 MTO: MOV #PMT0,R2 ;SET UP ADDRESS OF THE PRINT ROUTINE
10308 011672 000404 BR TOUTRT ;TO GENERALIZED TRAP OUT ROUTINE
10309
10310
10311
10312 MEMORY MANAGEMENT TRAP OUT ROUTINE
10313
10314 011674 005037 177572 MMERR: CLR MMSRD ;CLEAR THE MEMORY MANAGEMENT BIT
10315 011700 012701 014157 MOV #PMMERR,R1 ;SET UP ADDRESS OF ERROR MESSAGE
10316 011704 000005 TOUTRT: RESET ;CLEAR ALL DEVICES
10317 011706 105037 013042 CLR PIUFL ;CLEAR PRINT IN USE FLAG
10318 011712 005037 177776 CLR PSW ;LOWER PROCESSOR STATUS TO ALLOW INTERRUPTS TO CUM
10319 011716 004737 011400 JSR PC,PMESG ;PRINT THE ERROR MESSAGE
10320 011722 105737 013061 IOS: TSTB PCTR ;IS PRINTING DONE?
10321 011726 001375 BNE IOS
10322 011730 000000 HALT ;YES, HALT
10323
10324
10325
10326 INVALID UNIBUS ADDRESS TRAP
10327
10328 011732 022626 UNTRP: CMP (SP)+,(SP)+ ;POP THE PUSH STACK
10329 011734 005037 177776 CLR PSW ;CLEAR THE PROCESSOR STATUS WORD
10330 011740 000137 001162 JMP NEWPRM ;ASK OPERATOR TO REENTER THE DATA

```

10332				
10333				
10334				
10335				
10336				
10337				
10338				
10339	011744			
10340	011744	040440	041502	042504
	011752	043506	044510	027136
	011760	024074	020453	
10341	011764	045046	046113	057515
	011772	050117	051121	022137
	012000	024452	056473	
10342	012004	027455	052123	053125
	012012	054127	055131	026042
	012020	055445	037476	
10343	012024	030460	031462	032464
	012032	033466	034470	021472
	012040	023500	056075	
10344	012044	040440	041502	042504
	012052	043506	044510	027136
	012060	024074	020453	
10345	012064	045046	046113	047115
	012072	050117	051121	022137
	012100	024452	056473	
10346	012104	027455	052123	053125
	012112	054127	055131	026042
	012120	055445	037476	
10347	012124	030460	031462	032464
	012132	033466	034470	021472
	012140	023500	056075	
10348	012144	040440	041502	042504
	012152	043506	044510	027136
	012160	024074	020453	
10349	012164	045046	046113	047115
	012172	050117	051121	022137
	012200	024452	056473	
10350	012204	027455	052123	053125
	012212	054127	055131	026042
	012220	055445	037476	
10351	012224	030460	031462	032464
	012232	033466	034470	021472
	012240	023500	056075	
10352	012244	040440	041502	042504
	012252	043506	044510	027136
	012260	024074	020453	
10353	012264	045046	046113	047115
	012272	050117	051121	022137
	012300	024452	056473	
10354	012304	027455	052123	053125
	012312	054127	055131	026042
	012320	055445	037476	
10355	012324	030460	031462	032464
	012332	033466	034470	021472
	012340	023500	056075	

.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:

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

10357					...	ASCII TO EBCDIC CONVERSION TABLE
10358					ATOETB:	.BYTE 100,117,152,173,133,154,120,175 ;240-247
10359						
10360	012344	100	117	152		
	012347	173	133	154		
	012352	120	175			
10361	012354	115	135	134		.BYTE 115,135,134,116,153,140,113,141 ;250-257
	012357	116	153	140		
	012362	113	141			
10362	012364	360	361	362		.BYTE 360,361,362,363,364,365,366,367 ;260-267
	012367	363	364	365		
	012372	366	367			
10363	012374	370	371	172		.BYTE 370,371,172,136,114,176,156,157 ;270-277
	012377	136	114	176		
	012402	156	157			
10364	012404	174	301	302		.BYTE 174,301,302,303,304,305,306,307 ;300-307
	012407	303	304	305		
	012412	306	307			
10365	012414	310	311	321		.BYTE 310,311,321,322,323,324,325,326 ;310-317
	012417	322	323	324		
	012422	325	326			
10366	012424	327	330	331		.BYTE 327,330,331,342,343,344,345,346 ;320,327
	012427	342	343	344		
	012432	345	346			
10367	012434	347	350	351		.BYTE 347,350,351,155,177,137,112,025 ;330-337
	012437	155	177	137		
	012442	112	025			

```

10369 .SBTTL PROGRAM CONSTANTS AND VARIABLES
10370
10371
10372
10373 012444 177560 TKS: .WORD 177560 ;KEYBOARD CONTROL STATUS REGISTER
10374 012446 177562 TKB: .WORD 177562 ;KEYBOARD DATA BUFFER
10375 012450 177564 TPS: .WORD 177564 ;PRINTER STATUS/CONTROL REGISTER
10376 012452 177566 TPB: .WORD 177566 ;PRINTER DATA BUFFER
10377
10378
10379
10380
10381
10382
10383 012454 000000 DXDS: .WORD 0 ;DEVICE STATUS -- TT1
10384 012456 000000 DXCA: .WORD 0 ;COMMAND AND ADDRESS -- TT2
10385 012460 000000 DXCS: .WORD 0 ;CONTROL UNIT STATUS
10386 012462 000000 DXOS: .WORD 0 ;OFFSET AND STATUS
10387 012464 000000 DXBA: .WORD 0 ;BUS ADDRESS
10388 012466 000000 DXBC: .WORD 0 ;BYTE COUNT
10389 012470 000000 DXMO: .WORD 0 ;MAINTANCE OUT
10390 012472 000000 DXMI: .WORD 0 ;MAINTANCE IN
10391 012474 000000 DXCB: .WORD 0 ;CONTROL BITS
10392 012476 000000 DXND: .WORD 0 ;NPR DATA
10393 012500 000000 DXES1: .WORD 0 ;EXTRA SIGNALS
10394 012502 000000 DXMOB: .WORD 0 ;BUFFERED BUS OUT
10395 012504 000000 DXES2: .WORD 0 ;EXTRA SIGNALS
10396
10397
10398
10399
10400
10401 012506 000000 UNADDR: .WORD 0 ;UNIBUS ADDRESS
10402 012510 000000 VECTAD: .WORD 0 ;DX VECTOR ADDRESS
10403 012512 000000 SDEV: .WORD 0 ;STARTING DEV NUMBER
10404 012514 000000 EDEV: .WORD 0 ;ENDING DEV NUMBER
10405 012516 000 CHTYPE: .BYTE 0 ;CHANNEL TYPE 0 = MPX - 1 = SEL
10406 012517 000 MMRESP: .BYTE 0 ;MEMORY MANAGEMENT 0 = NO - 1 = YES
10407 012520 000000 BUFREL: .WORD 0 ;BUFFER RELOCATION ADDRESS
10408 012522 000 TSTTYP: .BYTE 0 ;TEST TYPE 0 = 2848 - 1 = FRIEND
10409 012523 000 IOBUF: .BYTE 0 ;SEPERATE I/O BUFFER 0 = NO - 1 = YES
10410 012524 000 FILLCH: .BYTE 0 ;FILL CHARACTER
10411 012525 000 CONEND: .BYTE 0 ;EXTRA
10412
10413
10414
10415
10416
10417 012626
10418 012626
10419
10420
10421
10422
10423
10424

```

SYSTEM PUSH STACK

```

SSTACK = .+.100

```

SYSTEM VARIABLES

THE FOLLOWING VARIABLES ARE RESET UPON START-UP

10425			:					
10426	012626	000000	VSTRT:	.WORD	0		:	DUMMY
10427		012630	TBUF	=	.		:	START OF TELETYPE INPUT BUFFER
10428		012730						
10429	012730	000000	TBUFE:	.WORD	0		:	END OF TELETYPE INPUT BUFFER
10430	012732	000000	TPTR:	.WORD	0		:	TELE IN PTR
10431	012734	000000	PBFS:	.WORD	0		:	START OF PRINT BUFFER
10432		013036						
10433		013036	PBFE	=	.		:	END OF PRINT BUFFER
10434	013036	000000	PPTR:	.WORD	0		:	PRINT FETCH PTR
10435	013040	000000	PPPTR:	.WORD	0		:	PRINT PUT PTR
10436	013042	000	PIUFL:	.BYTE	0		:	PRINTER IN USE FLAG
10437	013043	000	TCMACT:	.BYTE	0		:	TELE COMMAND ACTIVE FLAG 0 = NON-ACT
10438	013044	000	TCMDAB:	.BYTE	0		:	TEL COMMAND ABORT 1 = ABORT
10439	013045	000	LINECT:	.BYTE	0		:	LINE CTR - CHARS / LINE
10440	013046	000	WK:	.BYTE	0		:	WORK LOC
10441	013047	000	WK1:	.BYTE	0		:	WORK LOC
10442	013050	000000	TPTR:	.WORD	0		:	TUMBLE TABLE PTR
10443	013052	000000	TTADDR:	.WORD	0		:	BEG OF TUMBLE TABLE
10444	013054	000000	SDEVTB:	.WORD	0		:	START OF DEVICE TABLES
10445	013056	000	DXSTPF:	.BYTE	0		:	DX STOP FLAG
10446	013057	000	MAXDEV:	.BYTE	0		:	HIGHEST DEV # 1 - 8
10447	013060	000	DXACT:	.BYTE	0		:	DXACTIVE FLAG
10448	013061	000	PCTR:	.BYTE	0		:	PRINT BUFFER COUNTER
10449	013062	000	DXABFL:	.BYTE	0		:	DX ABORT FLAG 0 = NO ABORT, 1 = ABORT
10450	013063	000	TTYSTP:	.BYTE	0		:	CONSOLE OUTPUT STOP FLAG 0 = OUTPUT; 1 = NO OUTPUT
10451								
10452	013064	000000	CMDCHF:	.WORD	0		:	COMMAND CHAIN FLAG
10453	013066	000000	MDEV:	.WORD	0		:	DEV # IN MPXR EXEC
10454		013066	SELDEV	=	MDEV		:	DEV # IN SEL EXEC
10455	013070	000000	PBUFA:	.WORD	0		:	PHYSICAL BUFF ADDR - IN ,000'S
10456	013072	000000	VBUFA:	.WORD	0		:	VIRTUAL BUFF ADDR - IN ,000'S
10457	013074	000000	PHYOFF:	.WORD	0		:	PHY OFFSET FOR MEMORY MANAGEMENT
10458	013076	000000	CDEV:	.WORD	0		:	CURRENT DX DEVICE -- INTER SERVICE ROUTINE
10459	013100	000000	DEVCON:	.WORD	0		:	DEVICE ADDED TO THE DEVICE NUMBER = STARTING DEV NUMB -
10460	013102	000000	XADDR:	.WORD	0		:	EXTENDED ADDRESS BITS FOR THE DX CONTROL REGISTER -- IN
10461	013104	000000	STSPW:	.WORD	0		:	START OF THE PSW TABLE
10462	013106	000000	DSTOFF:	.WORD	0		:	OFFSET TO THE DST TABLE
10463	013110	000000	SADDR:	.WORD	0		:	TELETYPE COMMAND STARTING BUFFER ADDRESS
10464	013112	000000	EADDR:	.WORD	0		:	TELETYPE COMMAND ENDING BUFFER ADDRESS
10465	013114	000000	DMPADR:	.WORD	0		:	POINTER TO DUMP ROUTINE CURRENTLY BEING UTILIZED BY TEL
10466	013116	000000	VEND:	.WORD	0			
10467								
10468								
10469								
10470			:					
10471			:					
10472	013120	000102	PMAX:	.WORD	PBFE-PBFS		:	SIZE OF PRINT BUFFER
10473	013122	000000	FTIMFL:	.WORD	0		:	FIRST TIME FLAG

THE FOLLOWING VARIABLES ARE NOT RESET ON START-UP

10475
10476
10477
10478
10479
10480 013124 215 212
10481 013126 055104 054104 026511
10482 013163 377 377
10483 013165 215 212
10484 013167 125 044516 052502
10485 013220 377
10486 013222
10487 013222 215 212
10488 013224 047111 042524 051122
10489 013267 377
10490
10491 013270 215 212
10492 013272 042504 044526 042503
10493 013332 377
10494 013334
10495 013334 215 212
10496 013336 044103 047101 042516
10497 013365 377
10498
10499 013366 215 212
10500 013370 042515 047515 054522
10501 013424 377
10502 013426
10503 013426 215 212
10504 013430 052502 043106 051105
10505 013523 377
10506
10507 013524 215 212
10508 013526 051106 042511 042116
10509 013562 377
10510 013564
10511 013564 215 212
10512 013566 042523 040520 040522
10513 013625 377
10514
10515 013626 215 212
10516 013630 052517 050124 052125
10517 013675 377
10518
10519 013676 207 207 215
10520 013702 047516 020116 054105
10521 013722 215 212 377
10522 013726
10523 013726 207 215
10524 013732 040520 044522 054524
10525 013746 212 215 377
10526 013752
10527 013752 207 207 215
10528 013756 046111 042514 040507
10529 014003 212 215 377
10530

.SBTTL MESSAGES
.NLIST BEX
:
: SYSTEM MESSAGES
: STMSG: .BYTE 215,212
: .ASCII /DZDXI-A NEW DX11-B RESPONDER/
: .BYTE 377,377
: UNMSG: .BYTE 215,212
: .ASCII /UNIBUS ADDRESS -OCTAL- : /
: .BYTE 377
: .EVEN
: VECTMS: .BYTE 215,212
: .ASCII /INTERRUPT VECTOR ADDRESS -OCTAL- : /
: .BYTE 377
: .EVEN
: DEVMS: .BYTE 215,212
: .ASCII /DEVICE ADDRESSES -HEX- (XX,XX): /
: .BYTE 377
: .EVEN
: CHTYMS: .BYTE 215,212
: .ASCII /CHANNEL TYPE (M OR S): /
: .BYTE 377
: .EVEN
: MMMS: .BYTE 215,212
: .ASCII /MEMORY MANAGEMENT (Y OR N): /
: .BYTE 377
: .EVEN
: BFRMS: .BYTE 215,212
: .ASCII /BUFFER RELOCATION, IF SPECIFIED - IN EVEN ,000'S -OCTAL- : /
: .BYTE 377
: .EVEN
: TESTMS: .BYTE 215,212
: .ASCII /FRIEND (F) OR 2848 DIAG(D): /
: .BYTE 377
: .EVEN
: FIOMS: .BYTE 215,212
: .ASCII /SEPARATE I-O BUFFERS (Y OR N): /
: .BYTE 377
: .EVEN
: FILLMS: .BYTE 215,212
: .ASCII /OUTPUT BUFFER FILL CHARACTER -HEX- : /
: .BYTE 377
: .EVEN
: NXMSG: .BYTE 207,207,215,212
: .ASCII /NON EX-NEM ERROR/
: .BYTE 215,212,377
: .EVEN
: PARMES: .BYTE 207,207,215,212
: .ASCII /PARITY ERROR/
: .BYTE 212,215,377
: .EVEN
: ILLMES: .BYTE 207,207,215,212
: .ASCII /ILLEGAL DEVICE NUMBER/
: .BYTE 212,215,377
: .EVEN

Line	Device	Device	Device	Device	Device	Message
10531	014006	052503	051122	047105	STPMES:	.ASCII /CURRENT DEVICE NUMBER -- /
10532	014037	377				.BYTE 377
10533						.EVEN
10534	014040	207	207	215	INVLDC:	.BYTE 207,207,215,212
10535	014044	047111	040526	044514		.ASCII /INVALID DX COMMAND/
10536	014066	212	215	377		.BYTE 212,215,377
10537		014072				.EVEN
10538	014072	020040	047516	046440	PNOMM:	.ASCII / NO MEMORY MANAGEMENT AVAILABLE/
10539	014132	377				.BYTE 377
10540	014133	215	212	207	PMT0:	.BYTE 215,212,207,207
10541	014137	115	046505	051117		.ASCII /MEMORY TIME OUT/
10542	014156	377				.BYTE 377
10543	014157	215	212	207	PMMERR:	.BYTE 215,212,207,207
10544	014163	115	046505	051117		.ASCII /MEMORY MANAGEMENT ERROR/
10545	014212	377				.BYTE 377
10546		014214				.EVEN
10547	014214	215	212		RNMESG:	.BYTE 215,212
10548	014216	054523	052123	046505		.ASCII /SYSTEM INITIALIZED, TYPE "R" TO ENABLE DX/
10549	014267	377				.BYTE 377
10550	014270	015	012		HELPM5:	.BYTE CR,LF
10551	014272	054104	030461	041055		.ASCII /DX11-B 2848 EMULATOR TEST PACKAGE - OPERATIONAL INFORMATION/
10552	014365	015	012			.BYTE CR,LF
10553	014367	015	012			.BYTE CR,LF
10554	014371	104	026440	020055		.ASCII /D -- DUMP COMMAND/(CR)(LF)
10555	014414	020040	020040	020040		.ASCII / DTT,C DUMP TUMBLE TABLE IN CODE "C"/(CR)(LF)
10556	014474	020040	020040	020040		.ASCII / DIN,XX,C DUMP INPUT BUFFER FOR DEVICE XX IN CODE "C"/(CR)(LF)
10557	014572	020040	020040	020040		.ASCII / DOT,XX,C DUMP OUTPUT BUFFER FOR DEVICE XX IN CODE "C"/(CR)(LF)
10558	014671	105	026440	020055		.ASCII /E -- ENABLE DEVICE ON DX/(CR)(LF)
10559	014723	040	020040	020040		.ASCII / EXX ENABLE DEVICE XX/(CR)(LF)
10560	014766	020106	026455	043040		.ASCII /F -- FILL BUFFER COMMAND/(CR)(LF)
10561	015020	020040	020040	020040		.ASCII / FIN,HH,XX FILL INPUT BUFFER ON DEV XX WITH HH/(CR)(LF)
10562	015106	020040	020040	020040		.ASCII / FOT,HH,XX FILL OUTPUT BUFFER ON DEV XX WITH HH/(CR)(LF)
10563	015175	110	026440	020055		.ASCII /H -- HELP COMMAND/(CR)(LF)
10564	015220	020040	020040	020040		.ASCII / THIS TEXT/(CR)(LF)
10565	015242	020113	026455	045440		.ASCII /K -- KILL A DEVICE ON THE DX/(CR)(LF)
10566	015300	020040	020040	020040		.ASCII / KXX KILL DEVICE XX/(CR)(LF)
10567	015341	122	026440	020055		.ASCII /R -- ENABLE DX (RUN)/(CR)(LF)
10568	015367	040	020040	020040		.ASCII / R RUN TEST/(CR)(LF)
10569	015422	020123	026455	042040		.ASCII /S -- DISABLE DX (STOP)/(CR)(LF)
10570	015452	020040	020040	020040		.ASCII / S STOP IMMEDIATELY/(CR)(LF)
10571	015515	040	020040	020040		.ASCII / SD STOP AFTER NEXT DATA TRANSFER/(CR)(LF)
10572	015575	040	020040	020040		.ASCII / SE STOP AFTER NEXT ENDING SEQUENCE/(CR)(LF)
10573	015657	040	020040	020040		.ASCII / SI STOP ON NEXT SEL SEQ (ISS)/(CR)(LF)
10574	015734	020040	020040	020040		.ASCII / SP STOP ON NEXT PARITY ERROR/(CR)(LF)
10575	016010	005015	044127	051105		.ASCII /(CR)(LF)/WHERE:/(CR)(LF)
10576	016022	020040	020040	041442		.ASCII / "C" IS CODE FORMAT 0 = OCTAL/(CR)(LF)
10577	016070	020040	020040	020040		.ASCII / A = ASCII/(CR)(LF)
10578	016136	020040	020040	020040		.ASCII / E = EBCDIC/(CR)(LF)
10579	016205	040	020040	020040		.ASCII / H = HEX/(CR)(LF)
10580	016251	040	020040	021040		.ASCII / "XX" IS DX-11 DEVICE NUMBER IN HEX/(CR)(LF)
10581	016321	040	020040	021040		.ASCII / "HH" IS A HEX CHARACTER/(CR)(LF)(LF)
10582	016357	103	047117	047523		.ASCII /CONSOLE CONTROL CHARACTERS/(CR)(LF)
10583	016413	103	046124	041455		.ASCII /CTL-C (↑C) ABORT CURRENT COMMAND/(CR)(LF)
10584	016455	103	046124	050055		.ASCII /CTL-P (↑P) REQUESTS THE REENTRY OF CONTROL PARAMETERS/(CR)(LF)
10585	016544	052103	026514	020121		.ASCII /CTL-Q (↑Q) RESUME OUTPUT AFTER TEMPORARILY STOPPING BY (↑S)/(CR)(LF)
10586	016641	103	046124	051455		.ASCII /CTL-S (↑S) TEMPORARILY STOP OUTPUT TO CONSOLE/(CR)(LF)

10587	016720	052103	025514	020125
10588	016766	052522	047502	052125
10589	017040	015	012	012
10590		002554		
10591	017044	215	212	
10592	017046	047524	020117	046103
10593	017102	377		
10594				
10595		017104		
10596		000001		

HELPLN
TOOC:

```

.ASCII /CTL-U (↑) DELETE CURRENT INPUT LINE/<CR><LF>
.ASCII /RUBOUT -- DELETE LAST CHARACTER INPUTTED/<CR><LF>
.BYTE CR,LF,LF,LF
= -HELPLN
.BYTE 215,212
.ASCII /TOO CLOSE TO 200000 BOUNDARY/
.BYTE 377
.LIST BEX
.EVEN
.END

```

ACCESS	004036	DREAD	010672	GLERR	005072	NEW20	001672	R6	=%000006
ASCDMP	004624	DRMI	010700	GLEX	005260	NOLIN	= 000014	SADDR	013110
ATOETB	012344	DSRMI	010722	GLEX1	005256	NOP	= 000003	SATTN	007746
ATTN	= 000200	DSTOFF	013106	GLIMIT	005036	NXM	= 040000	SBUFA	= 000012
BFREMS	013426	DTAB	=%007003	GLMIN	005160	NXMMSG	013676	SCMD	= 000000
BSY	= 000020	DTUMTB	003620	GLMOT	005220	OCTDMP	004656	SCMDRJ	= 000200
BSYEN	= 004000	DUMP	003420	GLMTT	005130	OCTEX	004732	SCMDTB	006672
BUFREL	012520	DXAB	006524	GLOCT	005100	ODMP	004662	SCURS	= 000004
BUSOUT	= 000040	DXABFL	013062	HMMP	004746	PARER	= 100000	SDEV	012512
BYSS	= 000400	DXACT	013060	HELP	003772	PARMES	013726	SDEVTB	013054
CDEV	013076	DXBA	012464	HELPLN	= 002554	PBFE	= 013036	SELDEV	= 013066
CDEVST	010244	DXBC	012466	HELPM5	014270	PBFS	012734	SELRST	= 020000
CE	= 000010	DXCA	012456	HEXDMP	004742	PBUFA	013070	SENSCM	010126
CEDE	= 000003	DXCB	012474	ILLMES	013752	PC	=%000007	SEX	006554
CERR	003014	DXCS	012460	INFDS	= 004000	PCHAR	011362	SINBF	= 000006
CHEND	= 000040	DXCS	012460	INHDX	002612	PCHEND	006444	SLCMD	= 000001
CHENDS	= 001000	DXDS	012454	INIT	002014	PCHEN1	006454	SMT	= 000112
CHIS	= 000200	DXENB	= 000100	INIT10	002360	PCHEX	006500	SMINS	= 000020
CHTB	011600	DXES1	012500	INMES	011502	PCHIS	006262	SOMLF	= 000016
CHTYMS	013334	DXES2	012504	INOCT	002620	PCTR	013061	SOUTB	= 000010
CHTYPE	012516	DXEXEC	006542	INPUT	004316	PCUEND	006430	SP	=%000006
CMDCHF	013064	DXEXIT	010200	INR5	002624	PESEND	006102	SPACE	= 000040
CMDCHN	= 000004	DXISR	005322	INTREQ	= 000100	PFPTR	013036	SRBYTC	= 000014
CMDREJ	= 000001	DXMI	012472	INVLDC	014040	PHYOFF	013074	SRDRQ	= 000017
CMDREAD	= 000006	DXMO	012470	IOBUF	012523	PINDSC	006014	SRD10	007130
CMMI	= 000002	DXMOB	012502	ISSREJ	= 000010	PISR	011220	SREAD	007106
CMSRMI	= 000012	DXND	012476	KILL	004254	PIUFL	013042	SRMI	007006
CMTART	= 000001	DXONLN	= 001000	KILLEX	004312	PMAX	013120	SSENSE	= 000002
CMTLA	= 000005	DXOS	012462	KISAR0	= 172340	PMMERR	014157	SSRMI	007006
CONEND	012525	DXRD	= 000005	KISAR7	= 172356	PMT0	014133	SSTACK	= 012626
CONLINE	007736	DXRST	= 000001	KISDR0	= 172300	PNOMM	014072	SSTAT	= 000003
COTB	011534	DXST	= 000007	LF	= 000012	PNXM	006070	STAMOD	= 000100
CR	= 000015	DXSTPF	013056	LINECT	013045	PPARER	006232	START	001000
CRLF	011324	DXWR	= 000003	LINSZ	= 000050	PPPTR	013040	STKSTB	= 000002
CSPWST	010276	EADDR	013112	LOOP	005352	PRINT2	011342	STMSG	013124
CTL.C	= 000003	EBCDMP	004560	MAXDEV	013057	PRMSG	011400	STOP	003252
CTL.P	= 000020	EBCDSP	= 000100	MCMDB	007314	PROUT	011240	STOPDX	003326
CTL.Q	= 000021	EBCDTB	011744	MDEV	013066	PSELRT	005764	STOUT	010036
CTL.S	= 000023	EDEV	012514	MESG	011466	PSW	= 177776	STPDX	006520
CTL.U	= 000025	ENABLE	004210	MEX	007176	PSYSRT	005636	STPMES	014006
CUBUSY	= 000400	EOM	= 000152	MMERR	011674	P1BF	011376	STSPW	013104
CUE	= 000040	EQPCHK	= 000020	MMES	013366	P2BF	011356	SUDEV	010216
CUEND	= 000020	ERASCM	007756	MMESP	012517	QCUE	= 000010	SUDEX	010240
DE	= 000004	ESEND	= 000100	MMSR0	= 177572	RNMSG	014214	SUD10	010226
DEV	=%000000	ESEQ	010030	MREAD	007566	RSTART	001002	SWRITE	006716
DEVCON	013100	EXEC	002706	MAMI	= 007470	RUBOUT	= 000177	SYSINT	001006
DEVMES	013270	EXECMD	003030	MARM	= 007470	RUN	003100	SYSRST	= 010000
DISCEX	010722	FILL	003704	MTO	011666	R0	=%000000	TBUF	= 012630
DISCTL	010370	FILLCH	012524	MUXEND	010330	R1	=%000001	TBUFE	012730
DISFRN	010640	FILLMS	013626	MWRITE	007340	R2	=%000002	TCHEND	005612
DISPSZ	= 000740	FIOMS	013564	NEWLNE	= 000025	R3	=%000003	TCHIS	005600
DMPADR	013114	FTIMFL	013122	NEWPRM	001162	R4	=%000004	TCMACT	013043
DONE	= 000200	GDEV	005270	NEWP10	001446	R5	=%000005	TCMDAB	013044
		GETPRM	001154						

TCMDTB 003032
TCUEND 005624
TESEND 005554
TESTMS 013524
TINDSC 005530
TKB 012446
TKIN 010732

TNXM 005542
TOOC 017044
TOUTRT 011704
TPARER 005566
TPB 012452
TPS 012450
TPTR 012732

TSTTYP 012522
TTADDR 013052
TTPTR 013050
TTSIZE= 001000
TTYSTP 013063
TT1 =%000004
TT2 =%000005

UCHKS = 002000
UEXP = 000001
UNADDR 012506
UNMSG 013165
UNTRP 011732
VBUFA 013072
VCMDB 002576

VECTMS 013222
VEND 013116
VSTRT 012626
WK 013046
WK1 013047
XADDR 013102
. = 017104

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.NOW.SEG/PAGNUM=SYSMAC.CO,DZDXIA.CMB
RUN-TIME: 31 46 1 SECONDS
RUN-TIME RATIO: 321/79=4.0
CORE USED: 32K (63 PAGES)

