DX11-B

RESPONDER
CZDXIBO

AH-8777B-MC

FICHE 1 OF 1

APR 1978
digital
MADE IN USA

```
        .REM    %
```

IDENTIFICATION

PRODUCT CODE: AC-8776B-MC

PRODUCT NAME: CZDXIB0 DX11-B RESPONDER

RELEASE DATE: MARCH 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

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 DIAG-NOSTICS 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.

C O R E M A P

0-777	INTERRUPT VECTORS	(256 WORDS)
1000-17777	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
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 (↑P) TYPED ON THE TTY KEYBOARD WILL ALSO REQUEST THE OPERATOR TO REENTER THE TEST PARAMETERS.

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

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

4.3.2 ENTERING TEST PARAMETERS

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

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

4.3.2.1 GENERAL RULES FOR ENTERING PARAMETERS

A. ALL PARAMETERS MUST BE DELIMITED BY A CARRIAGE RETURN "(C/R)"

B. IF A TYPING ERROR IS DETECTED BEFORE ENTERING THE C/R, IT MAY BE CORRECTED BY:

1. USING RUBOUT(S) TO DELETE THE LAST CHARACTER(S)

2. HITTING CONTROL-U (↑U) TO DELETE THE ENTIRE ENTRY

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 BOUNDRY. EG: PHYSICAL ADDRESS
100000 IS ENTERED AS 100.

NOTE: THE BUFFER CANNOT BE CLOSER THAN 24000(8)
TO ANY 200000 BOUNDARY OR TO THE I/O PAGE. THE
THE DX IS NOT CAPABLE OF HAVING THESE BUFFERS
CROSS A 200000 BOUNDARY.
IT IS POSSIBLE TO OVERLAY THE ABSOLUTE LOADER
WHICH RESIDES IN THE HIGHEST AVAILABLE 4K(10)
OF THE FIRST 28K OF MEMORY.

DEFAULT BUFFER ADDRESS = 20 (20000)

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

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

DEFAULT OPTION = F -- FRIEND

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

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

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

"SEPARATE I-O BUFFERS (Y OR N)"
REQUESTS THE USER TO INDICATE WHETHER SEPARATE INPUT
AND OUTPUT BUFFER SHOULD BE MAINTAINED FOR EACH CRT
UNIT EMULATED. SEPARATE INPUT/OUTPUT BUFFERS ALLOW
THE TRANSMISSION OF THE SAME DATA PATTERN TO THE

360/370 INDEPENDENT OF WHAT DATA IS RECEIVED.
THIS IS USEFUL IN DETERMINING THE CAUSE OF BAD
DATA BEING TRANSMITTED.
NOTE -- MOST TESTS USING 'FRIEND' WILL NOT UTILIZE
SEPARATE I/O BUFFERS. THESE ARE ONLY FOR SPECIAL
SITUATIONS AS MENTIONED ABOVE.

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

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

"OUTPUT BUFFER FILL CHARACTER -HEX-"

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

DEFAULT FILL CHARACTER = 40, AN EBCDIC BLANK

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

4.3.3 SYSTEM INITIALIZATION

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

4.4 MONITOR COMMANDS

AFTER THE TEST PARAMETERS HAVE BEEN SUCESSFULLY 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 DISABLED. TYPE "R" TO CONTINUE.
THE FOLLOWING WILL BE PRINTED ON THE CONSOLE TELETYPE:

"CURRENT DEVICE -- XX"	THE CURRENT DEVICE ADDRESS IN HEX
"XXXXXX"	THE DXDS IN OCTAL - PROBABLY ZERO
"XXXXXX"	THE DXCA IN OCTAL
"XXXXXX"	THE DXCS IN OCTAL
"XXXXXX"	THE DXOS 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,O	DUMP BETWEEN GIVEN LIMITS IN OCTAL
DSSSSSS,EEEEEE,H	DUMP BETWEEN GIVEN LIMITS IN HEX
DSSSSSS,EEEEEE,E	DUMP BETWEEN GIVEN LIMITS IN EBCDIC
DSSSSSS,EEEEEE,A	DUMP BETWEEN GIVEN LIMITS IN ASCII

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

F -- FILL COMMAND

THE FILL COMMAND ALLOWS THE USER TO FILL THE INPUT OR 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 COMMAD 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
NOTE: NO SPACE BETWEEN "A" AND LOCATION.

5.0 OPERATING PROCEDURE

REFER TO SECTION 4.4 "MONITOR COMMANDS" FOR DETAILS.

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

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

COMMAND OCTAL	HEX	DESCRIPTION
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 OUT-
SIDE 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 DE-
TECTED 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.

```
%
.REM    %
8.0     PROGRAM DESCRIPTION

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

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

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

MESG:   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 MANAGEMENT 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
                     SRDRQ
                     SMINS)

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

              CLR DXACT, CMDCHF, DXABFL

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

              CHECK FOR CHANNEL TYPE

              IF SELECTOR CHANNEL SET BUSY ENABLE IN DXCS

              SET INTERRUPT ENABLE & ONLINE IN DXCS

              RETURN TO EXEC



STOP DX COMMAND

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


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



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

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

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



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

TYPICAL SELECTOR COMMANDS (FOR EACH DEVICE #)

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

TYPICAL MULTIPLEXER COMMANDS (FOR EACH DEVICE #)

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

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

---- SELECTOR CHANNEL ----

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

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

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

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

        WRITE LINE ADDRESS
        WRITE FULL BUFFER

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

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

        READ COMMAND (READ FULL BUFFER)

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

        READ MANUAL INPUT
        READ SHORT MANUAL INPUT

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

                IF READ REQUEST WAS SET-PROCEED -
                CLR SRDRQ
                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

        CONUNE: 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 PSW = 0  IF NO, PICK UP LAST DEVICE ADDRESS
                THAT HAS A COMMAND EXECUTED - HAS IT BEEN EXECUTED? (TYPICALLY
                SYSTEM RESET, SELECTIVE RESET, INTERFACE DISCONNECT,
                STATUS ACCEPTED, CHANNEL END, OR CONTROL UNIT END WILL
                TERMINATE DXACT IN A COMMAND SEQUENCE)

                GO TO  DEVICE AND FIND OUT IF THERE IS A JOB TO DO
                IF NOT, QUEUE "ATTENTION" IFF ATTENTION IS REQUESTED (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 TT ENTERED
                    ROUTINE - (PREPARE CONTROL UNIT END (PCUEND))

                    SBUFA IS INCREASED BY THE SAME ROUTINE (PCUEND)

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

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

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

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

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

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

INCREMENTED BY 4-- THIS APPLIES TO ALL THE SELECTOR 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 *
<---- SENSCM: = 4
<---- SWRITE: & MWRITE: = 5
<---- SREAD: & MREAD: = 6
<---- ERASCM: = 7
<---- QCUE = 10 *
<---- "ATTENTION" = 11 *
<---- SSRMI: & MSRMI: = 12

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

HOW USED

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

SET UP IN	USED IN	CLEARED IN
TISSRJ:	PESEND:	CDEVST: (RUN:,ENABLE:,
PINDSC:	SEX: & MEX:	KILL:,PSYSRT:,PINDSC:,
PVISS:	SWRITE: & MWRITE:	PESEND:)
PCHEND:	SSRMI: & MSRMI:	

ERASCM:

SSENSE (2)

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

HOW USED

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

SET UP IN	USED IN	CLEARED IN
PHIS:	SENSCM:	RUN:
PCHEND:		ENABLE:
		PSYSRT:
		PSELRT:
		SWRITE:
		SREAD:
		ERASCM:
		MWRITE:
		MREAD:

SSTAT (3)

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

HOW USED

USED BY 360 WHEN REQUESTING STATUS ;WITH EXECEPTION OF THE ASYNCRONOUS PRESENTING OF STATUS (ATTN) TO THE 360.
IE. STOUT: MOV SSTAT,@DXOS

SET UP IN	USED IN	CLEARED IN

PPARER:	STOUT:	PSYSRT:
ESEQ:		CDEVST:(RUN:,ENABLE:,
CONUNE:		KILL:,PSYSRT:,PINDSC:,
SATTN:		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 @DXBC

USED IN	CLEARED IN
INPUT:	PSYSRT:
SSRMI:	DISCTL:
	ERASCM:

SINBF (6)

SINBF <---- ADDRESS OF DEVICE INPUT/DISPLAY BUFFER

HOW USED

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

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

SOUTB (10)

SOUTB <---- ADDRESS OF DEVICE OUTPUT/DISPLAY BUFFER

HOW USED

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

SET UP IN	USED IN
INIT: (INT140:)	DUMP: INPUT: DISCTL: SSRMI: & MSRMI: PSYSRT: ERASCM: MREAD:

SBUFA (12)

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

HOW USED

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

SET UP IN	USED IN	CLEARED IN
PCUEND:	MWRITE:	CDEVST:(RUN:,
MWRITE:	MREAD:	ENABLE:,KILL:,PSYSRT:,
MREAD:	MSRMI:	PINDSC:,PESEND:)
MSRMI:		

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	USED IN
ENABLE: = 0	PCHIS:
KILL: = 1	

SRDRQ (17)

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

HOW USED

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

SET UP IN	USED IN	CLEARED IN
INPUT: = 1	SEX: & MEX:	RUN:
PESEND: = 1	SSRMI:	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	USED IN	CLEARED IN
INPUT:	DISCTL:	RUN:
	SSRMI: & MSRMI:	PSYSRT:

%

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

```
                                        .SBTTL  PROGRAM EQUATES AND DEVICE ASSIGNMENTS

                                ;
                                ;       SYSTEM EQUATES
                                ;
000000                          R0      =       %0
000001                          R1      =       %1
000002                          R2      =       %2
000003                          R3      =       %3
000004                          R4      =       %4
000005                          R5      =       %5
000006                          R6      =       %6
000006                          SP      =       %6
000007                          PC      =       %7
177776                          PSW     =       177776
172340                          KISAR0  =       172340
172356                          KISAR7  =       172356
172300                          KISDR0  =       172300
177572                          MMSR0   =       177572


                                ;
                                ;       TELETYPE CHARACTER EQUATES
                                ;
000015                          CR      =       15              ;CARRIAGE RETURN
000012                          LF      =       12              ;LINE FEED
000040                          SPACE   =       40              ;SPACE CHARACTER
000003                          CTL.C   =       3               ;CONTROL C
000020                          CTL.P   =       20              ;CONTROL P
000021                          CTL.Q   =       21              ;CONTROL Q
000023                          CTL.S   =       23              ;CONTROL S
000025                          CTL.U   =       25              ;CONTROL U
000177                          RUBOUT  =       177             ;RUBOUT
```

```
                        ;*****************************************************************
                        ;*                                                               *
                        ;*      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
                        ;
000000                  SCMD    =       0       ;CURRENT DEVICE COMMAND
000001                  SLCMD   =       1       ;LAST COMMAND, IF WRITE
000002                  SSENSE  =       2       ;DEVICE SENSE BYTE (NOTE -- MUST BE EVEN BYTE LOCATION)
000003                  SSTAT   =       3       ;DEVICE STATUS
000004                  SCURS   =       4       ;CURSOR POSITION
000006                  SINBF   =       6       ;ADDRESS OF DEVICE INPUT BUFFER
000010                  SOUTB   =       10      ;ADDRESS OF DEVICE OUTPUT BUFFER
000012                  SBUFA   =       12      ;CURRENT BUFFER PTR
000014                  SRBYTC  =       14      ;REMAINING BYTE COUNT
000016                  SONLF   =       16      ;DEVICE ONLINE - INDICATOR 0=ON-LINE 1=OFF-LINE
000017                  SRDRQ   =       17      ;READ MANUAL INPUT REQUEST -- IF NON-ZERO
000020                  SMINS   =       20      ;START OF MANUAL INPUT DATA
                        ;
                        ;       LOCATIONS 22-77 ARE AVAILABLE FOR EXPANSION PURPOSES
                        ;
                        ;
                        ;       OTHER   DX EQUATES
                        ;
000000                  DEV     =       R0              ;CURRENT DEVICE NUMBER
000003                  DTAB    =       R3              ;ADDRESS OF CUR DEV STATUS TABLE
000004                  TT1     =       R4              ;TUMBLE TABLE ENTRY 1
000005                  TT2     =       R5              ;TUMBLE TABLE ENTRY 2
000003                  CEDE    =       3               ;CHAN END & DEV END
000010                  QCUE    =       10              ;CODE TO QUE CONTROL UNIT END
000003                  NOP     =       3               ;NOP COMMAND
000025                  NEWLNE  =       25              ;NEW LINE CHARACTER
000100                  EBCDSP  =       100             ;EBCDIC SPACE CODE
000112                  SMI     =       112             ;START OF MESSAGE INDICATOR
000152                  EOM     =       152             ;END OF MESSAGE INDICATOR
000050                  LINSZ   =       40.             ;NUMBER OF CHARACTERS PER LINE OF 2260 DISPLAY
000014                  NOLIN   =       12.             ;NUMBER OF LINES PER 2260 DISPLAY
000740                  DISPSZ  =       NOLIN*LINSZ     ;NUMBER OF CHARACTERS ON THE DISPLAY
001000                  TTSIZE  =       512.            ;NUMBER OF ENTRIES IN TUMBLE TABLE
                        ;
                        ;       360 COMMAND EQUATES
                        ;
000001                  CMWRT   =       1               ;WRITE DATA (FROM 360 TO PDP-11)
000002                  CMRMI   =       2               ;READ MANUAL INPUT (PDP-11 TO 360)
000005                  CMWTLA  =       5               ;WRITE LINE ADDRESS (360 TO PDP-11)
000006                  CMREAD  =       6               ;READ FULL BUFFER (PDP-11 TO 360)
000012                  CMSRMI  =       12              ;SHORT READ MANUAL INPUT (PDP-11 TO 360)
```

```
                        ;
                        ;       DX REGISTER ASSIGNMENTS & LAYOUTS
                        ;
                        ;               DXDS OR TUMBLE TABLE ENTRY 1 (TT1)
                        ;
100000                  PARER   =       100000          ;PARITY ERROR DETECTED
040000                  NXM     =       40000           ;NON EXISTENT MEMORY CONDITION
020000                  SELRST  =       20000           ;IBM SELECTIVE RE-SET
010000                  SYSRST  =       10000           ;IBM SYSTEM RESET
004000                  INFDSC  =       4000            ;IBM PROGRAMMED INTERFACE DISCONNECT
002000                  UCHKS   =       2000            ;UNIT CHECK WAS PRESENTED TO THE CHANNEL
001000                  CHENDS  =       1000            ;CHANNEL END WAS PRESENTED TO THE CHANNEL
000400                  BYSS    =       400             ;BUSY WAS PRESENTED TO THE CHANNEL
000200                  CHIS    =       200             ;CHANNEL INIT SELECTION SEQ WAS COMPLETED
000100                  ESEND   =       100             ;CHANNEL ACCEPTED LAST STATUS
000040                  CHEND   =       40              ;CHANNEL DATA TRANSFER END
000020                  CUEND   =       20              ;DX DATA TRANSFER END
000010                  ISSREJ  =       10              ;INIT SELECTION SEQ WAS REJECTED
000004                  CMDCHN  =       4               ;CHANNEL SPECIFIED COMMAND CHAING
000002                  STKSTB  =       2               ;CHANNEL COULD NOT ACCEPT LAST STATUS
000001                  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
                        ;
004000                  BSYEN   =       4000            ;BUSY RENABLE - FOR SELECTOR CHANNELS
001000                  DXONLN  =       1000            ;ON-LINE INDICATION
000400                  CUBUSY  =       400             ;CONTROL UNIT BUSY
000200                  DONE    =       200             ;DONE FLAG
000100                  DXENB   =       100             ;INTERRUPT ENABLE
                        ;
                        ;                       BITS 4+3 ARE SET IF EXTENDED ADDRESS IS USED > 32K
                        ;                       BITS 2-0 ARE THE FUNCTION TO BE PERFORMED
                        ;
000001                  DXRST   =       1               ;DX RESET COMMAND
000003                  DXWR    =       3               ;WRITE DATA TO THE 360
000005                  DXRD    =       5               ;READ DATA FROM THE 360
000007                  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
                        ;
000200                  ATTN    =       200             ;ATTENTION
000100                  STAMOD  =       100             ;STATUS MODIFIER
```

```
2092            000040                  CUE     =       40              ;CONTROL UNIT END
2093            000020                  BSY     =       20              ;BUSY
2094            000010                  CE      =       10              ;CHANNEL END
2095            000004                  DE      =       4               ;DEVICE END
2096            000002                  UCHK    =       2               ;UNIT CHECK
2097            000001                  UEXP    =       1               ;UNIT EXCEPTION
2098                                    ;
2099                                    ;
2100                                    ;       2848 SENSE BYTE (SSENSE) DEFINITION
2101                                    ;
2102            000200                  $CMDRJ  =       200             ;COMMAND REJECT
2103            000100                  INTREQ  =       100             ;DEVICE OFF-LINE - INTERVENTION REQ
2104            000040                  BUSOUT  =       40              ;BUS OUT -- PARITY ERROR DURING CHIS
2105            000020                  EQPCHK  =       20              ;EQUIPMENT CHECK - PARITY ERROR DUR DATA TRANS
```

```
2106                                    .SBTTL  TRAP CATCHER
2107                                ;
2108                                ;       THE TRAP CATCHER IS LOADED INTO LOW CORE
2109
2110
2111            000000                  .=0
2112  000000    000000                  HALT                    ;FOR MEMORY MANAGEMENT
2113  000002    000000                  HALT
2114            000176                  .REPT   176             ;TRAP CATCHER
2115
2116                                    .WORD   .+2
2117                                    HALT
2118                                    .ENDR
2119            000200                  .=200
2120  000200    000137  001000          JMP     START           ;ESTABLISH LOC 200 STARTING ADDRESS
2121
```

```
                                        .SBTTL  PROGRAM START-UP SEQUENCES
                                ;
                                ;       NORMAL SYSTEM START LOCATION  --1000
                                ;
                        001000          .=1000
001000  000402          START:  BR      SYSINT          ;NORMAL START UP
                                ;
                                ;       RESTART ADDRESS -- REENTER ALL PARAMETERS -- 1002
                                ;
001002  005037  013132  RSTART: CLR     FTIMFL          ;RESET FIRST TIME FLAG TO FORCE PARAMETER REENTRY
                                ;
                                ;       SYSTEM GENERALIZED INITIALIZATION PROCEDURE
                                ;
                                ;               SET-UP  STACK POINTER
                                ;                       TRAP/VECTOR AREA
                                ;                       SYSTEM GENERATED TRAPS
                                ;                       TELETYPE (CONSOLE) VECTORS + STATUS REGISTERS
                                ;                       CLEAR ALL LIVE SYSTEM VARIABLES
                                ;                       SET UP TELETYPE INPUT / OUTPUT BUFFERS
                                ;
001006  012706  012636  SYSINT: MOV     #SSTACK,SP      ;SET UP THE STACK POINTER
001012  000005                  RESET                   ;RESET ALL DEVICES
                                ;
                                ;       SET UP CONSOLE VECTORS
                                ;
001014  012700  000060          MOV     #60,R0          ;START OF CONSOLE VECTORS
001020  012720  010744          MOV     #TKIN,(R0)+     ;BEG OF TELE INPUT ISR
001024  012720  000340          MOV     #340,(R0)+      ;NEW PROC STATUS
001030  012720  011232          MOV     #PISR,(R0)+     ;BEG OF TELE PRINT ISR
001034  012710  000340          MOV     #340,(R0)       ;NEW PROC STATUS
                                ;
                                ;       SET UP MISC TRAPS
                                ;
001040  012737  011672  000004  MOV     #MTO,4          ;MEMORY TIME OUT TRAP
001046  012737  000340  000006  MOV     #340,6
001054  012737  011700  000250  MOV     #MMERR,250      ;MEMORY MANAGEMENT ERROR
001062  012737  000340  000252  MOV     #340,252
                                ;
                                ;       CLEAR ALL VARIABLES
                                ;
001070  012700  012636          MOV     #VSTRT,R0       ;START OF VARAIBLES
001074  012701  000272          MOV     #VEND-VSTRT+2,R1  ;# OF VARAIBLES TO CLEAR
001100  105020          10$:    CLRB    (R0)+           ;CLEAR A BYTE
001102  005301                  DEC     R1              ;DONE?
001104  001375                  BNE     10$             ;NO, CONTINUE CLEARING
                                ;
                                ;       SET UP TELE BUFFER POINTERS
                                ;
001106  012737  012640  012742  MOV     #TBUF,TPTR      ;TELE INPUT POINTER
001114  012737  012744  013046  MOV     #PBFS,PFPTR     ;TELE OUT FETCH PTR
001122  012737  012744  013050  MOV     #PBFS,PPPTR     ;TELE OUT PUT PTR
                                ;
                                ;       ENABLE TTY
```

```
 2178                                           ;
 2179  001130  052777  000100  011316           BIS     #100,@TKS       ;ENABLE TELETYPE INPUT
 2180  001136  052777  000100  011314           BIS     #100,@TPS       ;ENABLE TTY OUTPUT INTERRUPTS
 2181  001144  005037  177776                    CLR     PSW             ;CLEAR THE PROCESSOR STATUS WORD
 2182  001150  005737  013132                    TST     FTIMFL          ;FIRST TIME THROUGH? (MUST PARAMETERS BE REENTERED?)
 2183  001154  001402                           BEQ     GETPRM          ;YES, FORCE USER TO ENTER ALL PARAMETERS
 2184  001156  000137  002026                    JMP     INIT            ;NO, RESTART TEST USING SAME PARAMETERS
```

```
                                        .SBTTL  TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)
                                ;
                                ;       HERE WE START GATHERING THE TEST INFORMATION
                                ;               PRINT START-UP HERALD MESSAGE
                                ;
001162  004737  011330          GETPRM: JSR     PC,CRLF         ;RESTORE THE CARRIAGE
001166  004137  011472                  JSR     R1,MESG         ;PRINT START-UP MESG
001172  013134                          .WORD   STMSG
                                ;
                                ;       GET DX11 UNIBUSS ADDRESS (OCTAL ADDRESS INPUT)
                                ;               VALID UNIBUS ADDRESSESS (176200 - 177000)
                                ;               DEFAULT UNIBUS ADDRESS  176200
                                ;
001174  005037  013132          NEWPRM: CLR     FTIMFL          ;RESET FIRST TIME PARAMETERS (FORCE ALL PARMS TO BE ENTE
001200  012737  176200  012516          MOV     #176200,UNADDR  ;SET UP DEFAULT ADDRESS
001206  004137  002632                  JSR     R1,INOCT        ;GET UNIBUS ADDRESS
001212  013175                          .WORD   UNMSG
001214  001260                          .WORD   5$              ;ADDRESS OF DEFAULT ROUTINE
001216  120427  000015                  CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
001222  001364                          BNE     NEWPRM          ;NO, TELL HIM TO REEENTER
001224  020327  176200                  CMP     R3,#176200      ;VALID UNIBUS ADDRESS? BETWEEN 176200 AND 177000
001230  002761                          BLT     NEWPRM          ;NO, GET AGAIN
001232  020327  177000                  CMP     R3,#177000      ;UNIBUS ADDRESS GT 177000?
001236  003356                          BGT     NEWPRM          ;YES, ERROR -- REENTER
001240  032703  000037                  BIT     #37,R3          ;MAKE SURE 40 OCTAL WORD BOUNDRY
001244  001353                          BNE     NEWPRM          ;ILLEGAL, REENTER
001246  012737  001174  000004          MOV     #NEWPRM,4       ;SET UP TRAP OUT TO VALIDATE ADDRESS
001254  010337  012516                  MOV     R3,UNADDR       ;SAVE UNIBUS ADDRESS
001260  005077  011232          5$:     CLR     @UNADDR         ;VALIDATE THE UNIBUS ADDRESS
                                                                ;TRAP WILL OCCUR IF INVALID UNIBUS ADDRESS
                                ;
                                ;       GET THE DX11 INTERRUPT VECTOR ADDRESS (OCTAL ADDRESS INPUT)
                                ;               VALID VECTOR ADDRESSES (300 - 770)
                                ;               DEFAULT VECTOR ADDRESS  300
                                ;
001264  012737  011742  000004  10$:    MOV     #UNTRP,4        ;RESTORE MEMORY TIME-OUT TRAP
001272  012737  000300  012520          MOV     #300,VECTAD     ;SET UP DEFAULT VECTOR ADDRESS
001300  004137  002632                  JSR     R1,INOCT        ;GET VECTOR ADDRESS
001304  013232                          .WORD   VECTMS
001306  001344                          .WORD   20$             ;ADDRESS OF THE DEFAULT ENTRY
001310  120427  000015                  CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
001314  001363                          BNE     10$             ;NO, REENTER
001316  020327  000300                  CMP     R3,#300         ;CHECK VECTOR ADDRESS BETWEEN 300 AND 770
001322  002760                          BLT     10$             ;TOO LOW GIVE AN ERROR AND REENTER
001324  020327  000770                  CMP     R3,#770         ;LT 770?
001330  003355                          BGT     10$             ;YES, REENTER
001332  032703  000001                  BIT     #1,R3           ;WORD ADDRESS?
001336  001352                          BNE     10$             ;NO, REENTER
001340  010337  012520                  MOV     R3,VECTAD       ;SAVE IT
                                ;
                                ;       GET STARTING AND ENDING DEVICE CHANNEL ADDRESSES (HEX INPUT)
                                ;               VALID DEVICE CHANNEL ADDRESSES (00 - FF)
                                ;               DEFAULT DEVICE CHANNEL ADDRESS  10,10
```

```
                                        ;
                                        ;       NOTE:   EITHER ONE OR TWO ADDRESSES MAY BE SPECIFIED. IF ONLY ONE
                                        ;               ADDRESS IS SPECIFIED IT IS USED AS THE STARTING AND
                                        ;               ENDING DEVICE ADDRESS.
                                        ;
001344  012737  000020  012522  20$:    MOV     #20,SDEV        ;DEFAULT TO HEX ADDRESS 10
001352  012737  000020  012524          MOV     #20,EDEV        ;
001360  004137  002624                  JSR     R1,INHEX        ;GET DEVICE ADDRESSES IN HEX
001364  013300                          .WORD   DEVMES
001366  001460                          .WORD   NEWP10          ;ADDRESS OF THE DEFAULT ROUTINE
001370  010337  012522                  MOV     R3,SDEV         ;SAVE START DEV ADDR
001374  010337  012524                  MOV     R3,EDEV
001400  005703                          TST     R3              ;BE SURE POSITIVE
001402  100760                          BMI     20$
001404  020327  000377                  CMP     R3,#377         ;AND NOT GREATER THAN 377 -- HEX FF
001410  003355                          BGT     20$             ;ILLEGAL ENTRY
001412  120427  000054                  CMPB    R4,#',          ;MORE THAN ONE DEV? (COMMA, PARAMETER DELIMETER)
001416  001015                          BNE     30$
001420  004737  011604                  JSR     PC,CHTB         ;GET ENDING DEVICE
001424  010337  012524                  MOV     R3,EDEV         ;SAVE ENDING ADDRESS
001430  023737  012522  012524          CMP     SDEV,EDEV       ;IS START LT END?
001436  003342                          BGT     20$             ;YES, ERROR
001440  163703  012522                  SUB     SDEV,R3         ;MORE THAN 8 DEVICES?
001444  020327  000007                  CMP     R3,#7
001450  003335                          BGT     20$             ;YES, ERROR
001452  120427  000015          30$:    CMPB    R4,#CR          ;WAS DEVICE ADDRESSES DELIMITED PROPERLY?
001456  001332                          BNE     20$             ;NO, REENTER

                                        ;
                                        ;       GET TYPE OF 360/370 CHANNEL
                                        ;               M = MULTIPLEXER CHANNEL
                                        ;               S = SELECTOR CHANNEL
                                        ;               DEFAULT IS 'S', SELECTOR CHANNEL
                                        ;
001460  105037  012526          NEWP10: CLRB    CHTYPE          ;0 = M, 1 = S
001464  004137  002632                  JSR     R1,INOCT        ;GET CHANNEL TYPE
001470  013344                          .WORD   CHTYMS
001472  001510                          .WORD   50$             ;DEFAULT TO SELECTOR CHANNEL
001474  120427  000115                  CMPB    R4,#'M          ;M? -- MULIPLEXER CHANNEL --
001500  001414                          BEQ     60$             ;YES, MULTIPLEXER CHANNEL
001502  120427  000123                  CMPB    R4,#'S          ;S? -- SELECTOR CHANNEL --
001506  001364                          BNE     NEWP10          ;NOT S OR M -- ERROR
001510  105237  012526          50$:    INCB    CHTYPE          ;SELECTOR CHANNEL
001514  000406                          BR      60$             ;GET MEMORY MANAGEMENT FACILITIES


                                        ;
                                        ;       MEMORY MANAGEMENT TIME-OUT TRAP HANDLER
                                        ;               USED ONLY FOR PARAMETER ENTRY PROCESS
                                        ;
001516  022626                  55$:    CMP     (SP)+,(SP)+            ;DUMP PC AND PSW SAVED BY INTERRUPT
001520  005037  177776                  CLR     PSW             ;TURN DOWN PROCESSOR STATUS
001524  004137  011472                  JSR     R1,MESG         ;PRINT "NO MEM MANAGEMENT AVAIL"
001530  014102                          .WORD   PNOMM
                                                                ;ASK TO HAVE QUESTION REENTERED
```

```
                                        ;
                                        ;       DETERMINE IF MEMORY MANAGEMENT IS TO BE USED
                                        ;               Y = YES, MEMORY MANAGEMENT TO BE USED
                                        ;               N = NO, DO NOT USE MEMORY MANAGEMENT
                                        ;               DEFAULT IS 'N', DO NOT USE MEMORY MANAGEMENT
                                        ;
001532  105037  012527        60$:    CLRB    MMRESP          ;DEFAULT TO NO MEMORY MANAGEMENT
001536  004137  002632                JSR     R1,INOCT        ;GET MEM MANAGEMENT
001542  013376                        .WORD   MMMES
001544  001600                        .WORD   70$             ;DEFAULT ROUTINE ADDRESS
001546  120427  000116                CMPB    R4,#'N          ;N? --DO NOT USE MEMORY MANAGEMENT
001552  001412                        BEQ     70$             ;IF EQ, NO MEMORY MANAGEMENT
001554  120427  000131                CMPB    R4,#'Y          ;Y? --MEMORY MANAGEMENT TO BE USED
001560  001364                        BNE     60$             ;ERROR
001562  105237  012527                INCB    MMRESP          ;MEMORY MANAGEMENT SPEC
001566  012737  001516  000004        MOV     #55$,4          ;SET UP TRAP TO TEST MEMORY MANAGEMENT
001574  005037  177572                CLR     MMSR0           ;CHECK FOR MEMORY MANAGEMENT
                                        ;
                                        ;       GET BUFFER RELOCATION IN OCTAL ,000'S
                                        ;               VALID RELOCATION ADDRESSES (20 - 700)
                                        ;                       (20000 THROUGH 734000)
                                        ;               DEFAULT RELOCATION ADDRESS  20 ---(20000)
                                        ;
001600  004137  002632        70$:    JSR     R1,INOCT        ;GET BUFFER RELOC. IN ,000'S
001604  013436                        .WORD   BFREMS
001606  001704                        .WORD   NEWP20          ;ADDRESS OF DEFAULT ROUTINE
001610  120427  000015                CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
001614  001371                        BNE     70$             ;NO, REENTER
001616  032703  000001                BIT     #1,R3           ;MUST BE A MULTIPLE OF 2000
001622  001366                        BNE     70$
001624  020327  000020                CMP     R3,#20
001630  002763                        BLT     70$             ;ILLG BUFFER CONST -- LT 20000
001632  005703                        TST     R3              ;IS NUMBER NEGATIVE?
001634  100761                        BMI     70$             ;YES, REENTER ADDRESS
001636  020327  000734                CMP     R3,#734         ;IS ADDRESS TOO LARGE?
001642  002356                        BGE     70$             ;YES, REENETER ADDRESS
001644  105737  012527                TSTB    MMRESP          ;WAS MEMORY MANAGEMENT SPECIFIED?
001650  001412                        BEQ     71$             ;NO, CHECK FOR 28K
001652  010304                        MOV     R3,R4           ;PUT VALUE IN WORK REG
001654  042704  000600                BIC     #600,R4         ;IGNORE ADDRESS EXTENSION BITS
001660  020427  000154                CMP     R4,#154         ;IS IT TOO CLOSE TO 200000 BOUNDARY?
001664  003407                        BLE     NEWP20          ;BRANCH IF OK
001666  004137  011472                JSR     R1,MESG         ;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.
001672  017051                        .WORD   TOOC            ;ADDRESS OF TOO CLOSE MESSAGE
001674  000741                        BR      70$             ;ASK FOR INPUT AGAIN

001676  020327  000134        71$:    CMP     R3,#134         ;NO, IS IT TOO CLOSE TO I/O PAGE?
001702  002336                        BGE     70$             ;YES, REENTER THE ADDRESS
001704  010337  012530        NEWP20: MOV     R3,BUFREL       ;SAVE REL CONST

                                        ;
                                        ;       GET TYPE  OF TEST TO BE RUN
```

```
2353                                    ;               D = 2848 RESPONDER DIAGNOSTIC
2354                                    ;               F = FRIEND
2355                                    ;
2356                                    ;               DEFAULT = 'F', FRIEND
2357                                    ;
2358  001710  105037  012532            90$:    CLRB    TSTTYP          ;RESET TEST TYPE
2359  001714  004137  002632                    JSR     R1,INOCT        ;GET TEST TYPE
2360  001720  013534                            .WORD   TESTMS
2361  001722  001752                            .WORD   100$            ;DEFAULT TO FRIEND
2362  001724  112737  000100  012534            MOVB    #EBCDSP,FILLCH  ;FOR 2848 SET FILL CHAR TO EBCDIC SPACE
2363  001732  120427  000104                    CMPB    R4,#'D          ;D? --2848 RESPONDER DIAGNOSTIC --
2364  001736  001433                            BEQ     INIT            ;YES, 2848 TEST
2365  001740  120427  000106                    CMPB    R4,#'F          ;F? -- FRIEND TEST --
2366  001744  001361                            BNE     90$             ;ILLEGAL ENTRY
2367  001746  105037  012534                    CLRB    FILLCH          ;FRIEND MODE -- DEFAULT FILL CHAR TO NULL
2368  001752  105237  012532            100$:   INCB    TSTTYP          ;SET TEST TO FRIEND
2369
2370                                    ;
2371                                    ;       FRIEND TEST ONLY
2372                                    ;       DETERMINE IF SEPARATE INPUT / OUTPUT BUFFERS ARE TO BE USED
2373                                    ;               Y = YES, MAINTAIN SEPARATE INPUT / OUTPUT BUFFERS
2374                                    ;               N = NO, USE SAME BUFFER FOR INPUT AND OUTPUT
2375                                    ;               DEFAULT IS 'N', NO, USE SAME BUFFER FOR INPUT / OUTPUT
2376                                    ;
2377  001756  105037  012533            110$:   CLRB    IOBUF           ;0 = NO, 1 = YES
2378  001762  004137  002632                    JSR     R1,INOCT        ;SEPARATE I/O BUFFERS?
2379  001766  013574                            .WORD   FIOMS
2380  001770  002026                            .WORD   INIT            ;DEFAULT TO NO
2381  001772  120427  000116                    CMPB    R4,#'N          ;N? -- NO, SAME I/O BUFFER --
2382  001776  001413                            BEQ     INIT            ;IF EQ, USE SAME I/O BUFFER FOR INPUT AND OUTPUT
2383  002000  120427  000131                    CMPB    R4,#'Y          ;Y? --YES, SEPARATE I/O BUFFERS--
2384  002004  001364                            BNE     110$            ;ERROR, REQUEST INPUT AGAIN
2385  002006  105237  012533                    INCB    IOBUF           ;SET SEPARATE I/O BUFFER INDICATOR
2386
2387                                    ;
2388                                    ;       FRIEND TEST MODE ONLY
2389                                    ;       GET BUFFER FILL CHARACTER (HEX INPUT REQUIRED)
2390                                    ;               ANY VALUE WILL BE ACCEPTED
2391                                    ;
2392  002012  004137  002624                    JSR     R1,INHEX        ;FILL CHARACTER
2393  002016  013636                            .WORD   FILLMS
2394  002020  002026                            .WORD   INIT
2395  002022  110337  012534                    MOVB    R3,FILLCH
```

```
                                        .SBTTL  PROGRAM INITIALIZATION
                                ;
                                ;       I N I T I A L I Z A T I O N
                                ;
                                ;
                                ;               SET UP ALL DX BUFFERS, MEMORY MANAGEMENT REGISTERS
                                ;                       AND DX REGISTERS
                                ;
002026  012737  011672  000004 INIT:   MOV     #MTO,4          ;SET UP MEMORY TIME OUT TRAP
002034  013701  012516          MOV     UNADDR,R1       ;SET UP DX UNIBUS ADDRESSES
002040  012702  012464          MOV     #DXDS,R2
002044  012703  000015          MOV     #13.,R3         ;13 ADDRESSES (REGISTERS)
002050  010122          10$:    MOV     R1,(R2)+        ;SET UP UNIBUS ADDRESS
002052  005721          TST     (R1)+           ;INCR TO NEXT DX REGISTER
002054  005303          DEC     R3              ;DONE?
002056  001374          BNE     10$             ;NO, SET UP NEXT REGISTER
                                ;
                                ;       SET UP DX VECTOR ADDRESS
                                ;
002060  013701  012520          MOV     VECTAD,R1
002064  012721  005334          MOV     #DXISR,(R1)+    ;TRAP TO DX ISR
002070  012711  000340          MOV     #340,(R1)       ;SET UP PROC STATUS AT INTER.
                                ;
                                ;       COMPUTE ADDRESSES OF DX BUFFERS
                                ;               CURRENTLY THIS INCLUDES DATA AREA, TUMBLE TABLE, AND SPW TABLE
                                ;
002074  005737  012530          TST     BUFREL          ;WAS BUFFER RELOC SPECIFIED?
002100  001003          BNE     20$             ;YES
002102  012737  000020  012530  MOV     #20,BUFREL      ;NO, MAKE BUFFERS START AT 20000
002110  013737  012530  013100 20$:    MOV     BUFREL,PBUFA    ;SAVE PHYSICAL ADDRESS
002116  013737  012530  013102          MOV     BUFREL,VBUFA    ;SAVE VIRTUAL ADDRESS
002124  105737  012527          TSTB    MMRESP          ;WAS MEMORY MANAGEMENT SPECIFIED?
002130  001436          BEQ     40$             ;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
                                ;
002132  012704  172340          MOV     #KISAR0,R4      ;I-SPACE PAR
002136  012705  172300          MOV     #KISDR0,R5      ;I-SPACE PDR
002142  005024          CLR     (R4)+           ;VA 0-17777 = PA 0-17777
002144  012725  077406          MOV     #77406,(R5)+    ;64 BLOCKS, UNLIMITED ACCESS
002150  013703  013100          MOV     PBUFA,R3        ;PHYSICAL ADDR * 2-6
002154  006303          ASL     R3
002156  006303          ASL     R3
002160  006303          ASL     R3
002162  010324          30$:    MOV     R3,(R4)+        ;SET UP PA FOR VA 20000-157777
002164  012725  077406          MOV     #77406,(R5)+    ;64 BLOCKS, UNLIMITED ACCESS
002170  062703  000200          ADD     #200,R3         ;INCREMENT TO NEXT 4K BANK
002174  020427  172356          CMP     R4,#KISAR7      ;ALL BUFFER ADDRESSES SET UP?
002200  001370          BNE     30$             ;NO, SET UP NEXT REGISTER
002202  012714  007600          MOV     #7600,(R4)      ;SET UP UNIBUS ADDRESS REGISTER
002206  012715  077406          MOV     #77406,(R5)     ;64 BLOCKS, UNLIMITED ACCESS
```

```
2452  002212  012737  000001  177572          MOV     #1,MMSR0        ;ENABLE MEMORY MANAGEMENT
2453  002220  012737  000020  013102          MOV     #20,VBUFA       ;TO 8K BANK OR 20000 AND UP
2454                                  ;
2455                                  ;       START SETTING UP SPW TABLE
2456                                  ;               1 ENTRY PER DEVICE (256 DEVICES)
2457                                  ;               ENTRY DESCRIPTION
2458                                  ;                   ----FOR VALID DEVICE NUMBERS
2459                                  ;                       BITS 15-8 = OFFSET TO DST TABLE (PHYSICAL ADDR)
2460                                  ;                             7-0 = 0
2461                                  ;                   ----FOR INVALID DEVICE NUMBERS
2462                                  ;                       BITS 15-8 = 0
2463                                  ;                             7-0 = 2 -- UNIT CHECK
2464                                  ;
2465                                  ;       THIS TABLE IS REFERENCED ON EACH 360 ACTION TO DETERMINE
2466                                  ;               IF DEVICE NUMBER IS VALID. THIS AUTOMATICALLY DONE
2467                                  ;               BY THE DX CONTROL UNIT
2468                                  ;
2469  002226  013705  013100          40$:    MOV     FBUFA,R5        ;COMPUTE OFFSET PHYSICALLY
2470  002232  000305                  SWAB    R5                      ;*1000
2471  002234  105005                  CLRB    R5
2472  002236  006305                  ASL     R5
2473  002240  010577  010226          MOV     R5,@DXOS        ;OFFSET TO SPW TABLE
2474  002244  013701  013102          MOV     VBUFA,R1        ;COMPUTE VIRT ADDR OF SPW TABLE
2475  002250  000301                  SWAB    R1                      ;*1000
2476  002252  105001                  CLRB    R1
2477  002254  006301                  ASL     R1
2478  002256  010137  013114          MOV     R1,STSPW        ;SAVE START OF SPW TABLE
2479  002262  010137  013104          MOV     R1,PHYOFF       ;COMPUTE THE OFFSET FOR PHYSICAL ADDRESSES
2480  002266  160537  013104          SUB     R5,PHYOFF       ;VERSES VIRTUAL ADDRESS - FOR MEM MANAGEMENT
2481  002272  062705  003000          ADD     #3000,R5        ;COMPUTE THE OFFSET TO THE DST TABLE
2482  002276  010537  013116          MOV     R5,DSTOFF       ;SAVE OFFSET TO DST TABLE
2483  002302  005000                  CLR     DEV             ;START AT DEVICE 0
2484  002304  120037  012522          50$:    CMPB    DEV,SDEV        ;IS DEVICE NUMBER VALID
2485  002310  002405                  BLT     60$             ;NO
2486  002312  120037  012524          CMPB    DEV,EDEV
2487  002316  003002                  BGT     60$             ;NO
2488  002320  010521                  MOV     R5,(R1)+        ;VALID DEVICE DST OFFSET TO ENTRY
2489  002322  000402                  BR      70$
2490  002324  012721  000002          60$:    MOV     #UCHK,(R1)+     ;INVALID DEV # UNIT CHECK TO ENTRY
2491  002330  005200                  70$:    INC     DEV             ;TO NEXT DEVICE
2492  002332  020027  000400          CMP     DEV,#256.       ;ALL DEVICES DONE?
2493  002336  001362                  BNE     50$             ;NO, SET UP SPW FOR NEXT DEVICE
2494                                  ;
2495                                  ;       NEXT SET UP TUMBLE TABLE AND DUPLICATE TUMBLE TABLE
2496                                  ;
2497  002340  010137  013062          MOV     R1,TTADDR       ;TUMBLE TABLE ADDRESS
2498  002344  010137  013060          MOV     R1,TTPTR        ;TUMBLE TABLE FETCH POINTER
2499  002350  012702  001000          MOV     #TTSIZE,R2      ;CLEAR T/T + DUPLICATE T/T (WORD POINTER)
2500  002354  005021                  80$:    CLR     (R1)+           ;CLEAR NEXT WORD
2501  002356  005302                  DEC     R2              ;DONE?
2502  002360  001375                  BNE     80$             ;NO, CLEAR NEXT WORD
2503                                  ;
2504                                  ;       SET UP DST TABLE
2505                                  ;               THE DST TABLE IS USED TO VERIFY COMMANDS FROM THE
2506                                  ;               360, THIS IS DONE BY THE HARDWARE
2507                                  ;               THE DST TABLE IS A BYTE TABLE, 1 BYTE PER POSSIBLE
```

```
                                         ;               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
                                         ;
002362  012702  000013                           MOV     #13,R2          ;NUMBER OF VALID 360 COMMANDS
002366  012703  002610                           MOV     #VCMDTB,R3      ;VALID COMMAND TABLE
002372  112321                           INIT10: MOVB    (R3)+,(R1)+     ;TO DST TABLE
002374  005302                                   DEC     R2              ;DONE?
002376  001375                                   BNE     INIT10          ;NO, MOVE IN NEXT RESPONSE
002400  012702  000365                           MOV     #245.,R2        ;MOVE UNIT CHECK TO INVALID ENTRIES
002404  112721  000002                   100$:   MOVB    #UCHK,(R1)+
002410  005302                                   DEC     R2
002412  001374                                   BNE     100$
                                         ;
                                         ;       COMPUTE MAX NUMBER OF DEVICES
                                         ;
002414  013703  012524                           MOV     EDEV,R3
002420  163703  012522                           SUB     SDEV,R3
002424  005203                                   INC     R3              ;START AT DEVICE NUMERO UNO
002426  110337  013067                           MOVB    R3,MAXDEV
002432  013737  012522  013110                   MOV     SDEV,DEVCON     ;SET UP DEVICE NUMBER -1
002440  005337  013110                           DEC     DEVCON
002444  012737  000001  013076                   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
                                         ;
002452  010137  013064                           MOV     R1,SDEVTB       ;SAVE START OF DEVICE BUFFERS
002456  005000                                   CLR     DEV             ;DEV # 0
002460  010103                           120$:   MOV     R1,DTAB         ;SAVE ADDR OF DEVICE STATUS TABLE
002462  012702  000420                           MOV     #272.,R2        ;CLEAR DEVICE STATUS TABLE + INPUT BUFFER
002466  005021                           122$:   CLR     (R1)+
002470  005302                                   DEC     R2              ;DONE?
002472  001375                                   BNE     122$            ;NO, CLEAR NEXT WORD
002474  010363  000006                           MOV     DTAB,SINBF(DTAB)
002500  062763  000076  000006                   ADD     #62.,SINBF(DTAB) ;COMPUTE ADDRESS OF INPUT BUFFER
002506  010363  000010                           MOV     DTAB,SOUTB(DTAB)
002512  062763  001040  000010                   ADD     #544.,SOUTB(DTAB) ;COMPUTE ADDRESS OF OUTPUT BUFFER
002520  012702  000740                           MOV     #DISPSZ,R2
```

```
                                                125$:   MOVB    FILLCH,(R1)+     ;FILL OUTPUT/DISPLAY BUFFER
                                                        DEC     R2              ;DONE?
                                                        BNE     125$            ;NO
                                                        INC     DEV             ;HAVE ALL DEVICE BUFFERS BEEN SET UP?
                                                        CMPB    DEV,MAXDEV
                                                        BNE     120$            ;NO, SET UP NEXT DEVICE BUFFERS
                                                        MOV     PBUFA,R5        ;SET UP EXTENDED ADDRESS BITS
                                                        ASR     R5
                                                        ASR     R5
                                                        ASR     R5
                                                        ASR     R5
                                                        BIC     #177747,R5      ;SAVE ONLY H.O. 2 BITS
                                                        MOV     R5,XADDR        ;SAVE EXTENDED ADDRESS BITS FOR DX CONTROL REG
                                                        MOV     #1,FTIMFL       ;SET FIRST TIME THROUGH FLAG

                                                ;
                                                ;       INITIALIZATION COMPLETE
                                                ;               TELL OPERATOR WE ARE ALREADY TO GO
                                                ;
                                                ;       NOTE:   AT THIS POINT THE DX HAS NOT BEEN STARTED
                                                ;               AND THE OPERATOR MUST TYPE R (RUN COMMAND)
                                                ;               TO THE SHOW UNDER WAY
                                                ;
                                                        JSR     R1,MESG         ;TELL OPERATOR WE ARE READY TO GO
                                                        .WORD   RNMESG
                                                        JMP     EXEC            ;GET THE SHOW ON THE ROAD

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

```
                                        .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
                                ;
002624  012705  011604          INHEX:  MOV     #CHTB,R5        ;MOVE ADDRESS OF CONVERSION ROUTINE TO R5
002630  000402                  BR      INR5
                                ;
                                ;       INOCT -- PRINT MESSAGE, WAIT FOR INPUT, + GET IT AND CONVERT OCTALL TO BINARY
                                ;
                                ;       CALLING SEQUENCE
                                ;       JSR     R1,INOCT
                                ;       .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
                                ;
002632  012705  011540          INOCT:  MOV     #COTB,R5        ;SET UP ADDRESS OF THE CONVERSION ROUTINE
002636  012102                  INR5:   MOV     (R1)+,R2        ;GET ADDRESS OF THE MESSAGE
002640  004737  011404          10$:    JSR     PC,PRMESG               ;PRINT THE DESIRED MESSAGE
002644  105037  013053                  CLRB    TCMACT          ;RESET ACTIVE FLAG
002650  105037  013054                  CLRB    TCMDAB          ;RESET ABORT FLAG
002654  105737  013054          30$:    TSTB    TCMDAB          ;COMMAND ABORT?
002660  001367                  BNE     10$             ;YES, REASK QUESTION
002662  105737  013053                  TSTB    TCMACT          ;WAS ENTRY COMPLETED?
002666  001772                  BEQ     30$             ;NO, WAIT
002670  012702  012640                  MOV     #TBUF,R2        ;SET UP ADDRESS OF BEG OF INPUT BUFFER
002674  004715                  JSR     PC,@R5          ;CONVERT INPUT TO BINARY
002676  005705                  TST     R5              ;LOOK FOR DEFAULT RESP -- C/R
002700  001005                  BNE     40$             ;NOT DEFAULT TAKE NORMAL RETURN
002702  120427  000015          CMPB    R4,#CR          ;ILLEGAL CHAR MUST BE A C/R
002706  001002                  BNE     40$             ;ITS NOT A DEFAULT
002710  011101                  MOV     (R1),R1         ;---TAKE THE DEFAULT RETURN
002712  000201                  RTS     R1
002714  005721          40$:    TST     (R1)+           ;INCR FOR NORMAL RETURN
002716  000201                  RTS     R1
```

```
                                        .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.
                                ;
002720  004737  011330          EXEC:   JSR     PC,CRLF         ;PRINT CR/LF
002724  012702  000052                  MOV     #'*,R2
002730  004737  011346                  JSR     PC,PRINT2               ;PRINT * -- DENOTE COMMAND MODE
002734  012706  012636                  MOV     #$STACK,SP      ;RE-ESTABLISH PUSH STACK
002740  105037  013053                  CLRB    TCMACT          ;CLEAR TELE CMD ACT
002744  105037  013054                  CLRB    TCMDAB          ;CLEAR TELE CMD ABORT
002750  105037  013055                  CLRB    LINECT          ;RESET LINE COUNTER
002754  105737  013072          10$:    TSTB    DXABFL          ;DID THE DX ABORT AN OPERATION ?
002760  001402                  BEQ     20$             ;NO, CONTINUE
002762  000137  003340                  JMP     STOPDX          ;YES IT DID, PRINT THE DX REGISTERS
002766  105737  013053          20$:    TSTB    TCMACT          ;IS THERE A COMMAND TO EXECUTE
002772  001001                  BNE     30$             ;YES, EXECUTE IT
002774  000767                  BR      10$             ;NO, WAIT AGAIN IF NOTHING TO DO
                                ;
                                ;       THERE IS A TELETYPE COMMAND TO BE EXECUTED
                                ;
002776  012702  012640          30$:    MOV     #TBUF,R2        ;SET UP PTR TO START OF TELE BUFFER
003002  112203                  MOVB    (R2)+,R3        ;GET COMMAND IDENTIFIER
003004  042703  177400                  BIC     #177400,R3      ;SAVE L.O. BYTE
003010  012704  003044                  MOV     #TCMDTB,R4      ;SET UP PTR TO COMMAND TABLE
003014  020324          40$:    CMP     R3,(R4)+        ;DOES COMMAND MATCH TABLE ENTRY?
003016  001411                  BEQ     EXECMD          ;YES, WE GOT A MATCH - START EXECUTION
003020  022404                  CMP     (R4)+,R4        ;INCR TO NEXT COMMAND
003022  005714                  TST     (R4)            ;END OF TABLE?
003024  001373                  BNE     40$             ;NO, TEST NEXT ENTRY
                                ;
                                ;       COMMAND ERROR - NOTIFY OPERATOR WITH ? AND "BELL"
                                ;
003026  012702  137607  CERR:   MOV     #137607,R2      ;PRINT ? AND "BELL"
003032  004737  011346          JSR     PC,PRINT2
003036  000137  002720          JMP     EXEC            ;RETURN TO EXEC
                                ;
                                ;       EXECUTE COMMAND
                                ;
003042  000134  EXECMD: JMP     @(R4)+          ;EXECUTE COMMAND
                                ;
                                ;       TELETYPE COMMAND TABLE
                                ;
003044  000101  TCMDTB: .WORD   'A              ;A = ACCESS
003046  004050          .WORD   ACCESS
003050  000104          .WORD   'D              ;D = DUMP
003052  003432          .WORD   DUMP
```

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

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

```
                                        .SBTTL  BACKGROUND -- STOP COMMAND
                                ;
                                ;       S = STOP DX COMMAND
                                ;               STOP DISBLES 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 EROR 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.
                                ;
003264  111204          STOP:  MOVB    (R2),R4          ;GET THE TYPE OF STOP INDICATED
003266  120427  000015          CMPB    R4,#CR           ;IMMEDIATELY?
003272  001422                  BEQ     STOPDX           ;YES, DISABLE DX AND PRINT REGISTERS
003274  120427  000104          CMPB    R4,#'D           ;D = AFTER NEXT DATA TRANSFER?
003300  001413                  BEQ     10$              ;YES, SET STOP FLAG
003302  120427  000105          CMPB    R4,#'E           ;E = AFTER THE NEXT ENDING SEQUENCE
003306  001410                  BEQ     10$              ;YES, SET STOP FLAG
003310  120427  000111          CMPB    R4,#'I           ;I = AFTER THE CHIS SEQUENCE
003314  001405                  BEQ     10$              ;YES, SET STOP FLAG
003316  120427  000120          CMPB    R4,#'P           ;P = STOP ON PARITY ERROR??
003322  001402                  BEQ     10$              ;YES, SET STOP FLAG
003324  000137  003026          JMP     CERR             ; ILLEGAL FORMAT -- GIVE ERROR
003330  110437  013066  10$:    MOVB    R4,DXSTPF        ;SET THE STOP FLAG
                                ;
                                ;       WHEN THE STOP CONDITION IS SATISIFIED,
                                ;       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 SATISIFIED OR ANOTHER REQUEST
                                ;       SUPERCEDES IT.
                                ;
003334  000137  002720          JMP     EXEC             ;RETURN TO THE EXEC
                                ;
                                ;       STOP THE DX AND PRINT THE REGISTERS
                                ;               NOTE THE PRINT OUTS WILL BE IN OCTAL
                                ;
003340  042777  000100  007122  STOPDX: BIC     #DXENB,@DXCS     ;DISABLE THE DX
003346  004737  011330          JSR     PC,CRLF          ;START AT NEW LINE
003352  004137  011472          JSR     R1,MESG          ;PRINT "CURRENT DEVICE -- "
003356  014016                  .WORD   STPMES
003360  013702  013106          MOV     CDEV,R2          ;CONVERT AND PRINT THE CURRENT
003364  004737  004760          JSR     PC,HDMP          ;DEVICE NUMBER IN HEX
003370  012703  000015          MOV     #13.,R3          ;PRINT THE 13 DX REGISTERS IN OCTAL
```

```
 2816  003374  012701  012464          MOV     #DXDS,R1        ;STARTING POINT
 2817  003400  013102                  10$:    MOV     @(R1)+,R2       ;GET THE REGISTER CONTENTS
 2818  003402  004737  004670                  JSR     PC,OCTDMP               ;PRINT IN OCTAL
 2819  003406  005303                          DEC     R3              ;ARE WE DONE
 2820  003410  001373                          BNE     10$             ;NO, DUMP NEXT WORD
 2821  003412  105037  013072                  CLRB    DXABFL          ;YES, RESET THE ABORT FLAG
 2822  003416  005077  007046                  CLR     @DXCS           ;RESET THE DX
 2823  003422  005277  007042                  INC     @DXCS
 2824  003426  000137  002720                  JMP     EXEC            ;AND RETURN TO THE EXEC
```

```
                                        .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)
                                ;
003432  004737  005050  DUMP:   JSR     PC,GLIMIT               ;GET BUFFER LIMITS
003436  112204                  MOVB    (R2)+,R4        ;GET DUMP MODE A/E/O/H
003440  012705  004636          MOV     #ASCDMP,R5      ;SET UP FOR ASCII DUMP
003444  120427  000101          CMPB    R4,#'A          ;IS IT ASCII?
003450  001421                  BEQ     10$             ;YES, START DUMP
003452  012705  004572          MOV     #EBCDMP,R5      ;SET UP FOR EBCDIC DUMP
003456  120427  000105          CMPB    R4,#'E          ;IS IT EBCDIC?
003462  001414                  BEQ     10$             ;YES, CONTINUE DUMP
003464  012705  004754          MOV     #HEXDMP,R5      ;SET UP FOR HEX DUMP
003470  120427  000110          CMPB    R4,#'H          ; IS IT HEX?
003474  001407                  BEQ     10$             ;YES, CONTINUE DUMP
003476  012705  004670          MOV     #OCTDMP,R5      ;SET UP FOR OCTAL DUMP
003502  120427  000117          CMPB    R4,#'O          ;IS IT OCTAL?
003506  001402                  BEQ     10$             ;YES, CONTINUE DUMP
003510  000137  003026          JMP     CERR            ;ILLEGAL ENTRY -- ERROR
003514  010537  013124  10$:    MOV     R5,DMPADR       ;SAVE ADDRESS OF DUMP ROUTINE
003520  005700                  TST     R0              ;WAS THIS A TUMBLE TABLE DUMP?
003522  001043                  BNE     DTUMTB          ;YES, DUMP THE TUMBLE TABLE
003524  112204                  MOVB    (R2)+,R4        ;WAS A DEV # SPECIFIED
003526  120427  000054          CMPB    R4,#',          ;IS NEXT POSITION A COMMA
003532  001014                  BNE     50$             ;NO, DUMP GIVEN LIMITS
003534  004737  005302          JSR     PC,GDEV         ;GET THE DEVICE NUMBER -- IN HEX
                                ;
                                ;       COMPUTE RELOCATION CONSTANT FOR DEVICE
                                ;
003540  005004                  CLR     R4              ;RELOCATION CONSTANT
003542  005303          30$:    DEC     R3              ;DONE?
003544  001403                  BEQ     40$             ;YES, ADD TO START + END ADDRESSES
003546  062704  002000          ADD     #2000,R4        ;TO NEXT DEVICE TABLES
003552  000773                  BR      30$
003554  060437  013120  40$:    ADD     R4,SADDR        ;ADD RELOCAT TO START ADDRESS
003560  060437  013122          ADD     R4,EADDR        ;ADD RELOCAT TO END ADDRESS
003564  017702  007330  50$:    MOV     @SADDR,R2       ;GET WORD
003570  004777  007330          JSR     PC,@DMPADR              ;CONVERT AND DUMP IT
003574  105737  013071  60$:    TSTB    PCTR
003600  001375                  BNE     60$
```

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

```
                                        .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
                                ;
003716  004737  005050        FILL:   JSR     PC,GLIMIT               ;GET BUFFER LIMITS
003722  004737  011604                JSR     PC,CHTB         ;GET THE FILL CHARCTER
003726  110337  012534                MOVB    R3,FILLCH       ;SAVE FILL CHAR
003732  004737  005302                JSR     PC,GDEV         ;GET THE DEVICE ADDRESS
                                ;
                                ;       COMPUTE RELOCATION FOR DEVICE
                                ;
003736  005004                        CLR     R4
003740  005303                10$:    DEC     R3              ;DONE?
003742  001403                        BEQ     20$             ;YES, ADD TO START AND END ADDR
003744  062704  002000                ADD     #2000,R4
003750  000773                        BR      10$
003752  060437  013120        20$:    ADD     R4,SADDR        ;ADD RELOC CONST TO START
003756  060437  013122                ADD     R4,EADDR        ;ADD RELOC CONST TO END ADDR
003762  013701  013120                MOV     SADDR,R1
                                ;
                                ;       FILL BUFFER WITH SPECIFIED CAHARCTER
                                ;
003766  113721  012534        30$:    MOVB    FILLCH,(R1)+    ;FILL CHARACTER
003772  020137  013122                CMP     R1,EADDR        ;DONE?
003776  101773                        BLOS    30$             ;NOPE, FILL NEXT CHAR
004000  000137  002720                JMP     EXEC            ;DONE, RETURN TO EXEC
```

```
                                        .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
                                ;
004004  012701  014300          HELP:   MOV     #HELPMS,R1      ;SET UP ADDRESS OF HELP MESSAGE
004010  012702  002551                  MOV     #HELPLN,R2      ;LENGTH OF HELP MESSAGE
                                ;
                                ;       START OUTPUTTING THE HELP MESSAGE UNDER OUR CONTROL
                                ;       SO THE COMMAND MAY BE ABORTED QUICKLY.
                                ;
004014  112100                  10$:    MOVB    (R1)+,R0        ;GET BYTE TO OUTPUT
004016  123727  013071  000004  15$:    CMPB    PCTR,#4         ;MORE THEN FOUR CHARACTERS IN OUTPUT BUFFER??
004024  003374                  BGT     15$             ;  YES, WAIT TIL DOWN A LITTLE
004026  004737  011366                  JSR     PC,PCHAR                ;PRINT IT ON CONSOLE
004032  105737  013054                  TSTB    TCMDAB          ;HAS OPERATOR INDICATED A DESIRE TO STOP?
004036  001002                  BNE     20$             ;  YES, ABORT HELP MESSAGE
004040  005302                  DEC     R2              ;HAS ENTIRE MESSAGE BEEN OUTPUTTED??
004042  001364                  BNE     10$             ;  NO, OUTPUT ANOTHER BYTE
004044  000137  002720          20$:    JMP     EXEC            ;  YES, RETURN TO THE EXECUTIVE
```

```
                                        .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
                                ;
004050  004737  011540          ACCESS: JSR     PC,COTB         ;GET THE START ADDRESS
004054  005705                          TST     R5              ;WAS A VALID ADDRESS ENTERED?
004056  001403                          BEQ     5$              ;NO, GIVE OPERATOR AN ERROR
004060  032703  000001                  BIT     #1,R3           ;WAS ADDRESS SPECIFIED A WORD ADDRESS?
004064  001402                          BEQ     7$              ;YES, OPEN SPECIFIED LOCATION
004066  000137  003026          5$:     JMP     CERR            ;NO, GIVE OPERATOR AN ERROR INDICATION
004072  010337  013120          7$:     MOV     R3,SADDR        ;SAVE STARTING ADDRESS
004076  013702  013120          10$:    MOV     SADDR,R2        ;GET OBJECT WORD
004102  105037  013053                  CLRB    TCMACT          ;CLEAR TELE ACTIVE FLAG
004106  004737  004670                  JSR     PC,OCTDMP               ;PRINT ADDRESS IN OCTAL
004112  012702  020040                  MOV     #"  ,R2       ;PRINT 2 SPACES
004116  004737  011346                  JSR     PC,PRINT2
004122  017702  006772                  MOV     @SADDR,R2       ;GET CONTENTS OF OBJECT LOCATION
004126  004737  004674                  JSR     PC,ODMP         ;PRINT CONTENTS IN OCTAL
004132  012702  020040                  MOV     #"  ,R2       ;PRINT 2 SPACES
004136  004737  011346                  JSR     PC,PRINT2
004142  105737  013053          20$:    TSTB    TCMACT          ;ACTIVE COMMAND?
004146  001775                          BEQ     20$             ;NO
004150  012702  012640                  MOV     #TBUF,R2        ;SET UP INPUT BUFFER ADDRESS
004154  004737  011540                  JSR     PC,COTB         ;WAS LOCATION CHANGED?
004160  005705                          TST     R5              ;ANY CHANGE?
004162  001007                          BNE     30$             ;YES, STORE IT
004164  120427  000057                  CMPB    R4,#'/          ;/, EXIT TO EXEC
004170  001412                          BEQ     50$             ;YES, RETURN TO EXEC
004172  120427  000015                  CMPB    R4,#CR          ;CR, GO TO NEXT LOCATION?
004176  001403                          BEQ     40$             ;YES, OPEN AND PRINT NEXT LOC.
004200  000736                          BR      10$             ;ERROR, PRINT CONTENTS OF CURRENT LOC.
004202  010377  006712          30$:    MOV     R3,@SADDR       ;CHANGE OPEN LOCATION
004206  062737  000002  013120  40$:    ADD     #2,SADDR        ;OPEN NEXT LOCATION
004214  000730                          BR      10$
004216  000137  002720          50$:    JMP     EXEC            ;RETURN TO THE EXEC
```

```
                        .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
                ;
004222  004737  005302  ENABLE: JSR     PC,GDEV         ;GET THE DEVICE NUMBER
004226  004737  010256          JSR     PC,CDEVST               ;CLEAR THE DEVICE STATUS TABLE
004232  010005                  MOV     DEV,R5          ;COMPUTE THE ADDRESS OF THE SPW TABLE ENTRY
004234  063705  013110          ADC     DEVCON,R5       ;COMPENSATE FOR OFFSET DEVICE ADDRESS
004240  060505                  ADD     R5,R5
004242  063705  013114          ADD     ST$PW,R5
004246  013715  013116          MOV     DSTOFF,(R5)     ;ENABLE THE DEVICE NUMBER
004252  105063  000002          CLRB    SSENSE(DTAB)
004256  105063  000016          CLRB    SONLF(DTAB)
004262  000137  002720          JMP     EXEC            ;RETURN TO THE EXEC
```

```
                                        .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
                                ;
004266  004737  005302          KILL:   JSR     PC,GDEV         ;GET THE DEVICE NUMBER
004272  004737  010256                  JSR     PC,CDEVST
004276  112763  000001  000016          MOVB    #1,SONLF(DTAB)
004304  010005                          MOV     DEV,R5          ;COMPUTE THE ADDRESS OF THE SPW TABLE
004306  063705  013110                  ADD     DEVCON,R5       ;COMPONSATE FOR OFFSET DEVICE ADDRESS
004312  060505                          ADD     R5,R5
004314  063705  013114                  ADD     ST$PW,R5
004320  012715  000002                  MOV     #UCHK,(R5)      ;MAKE THE DEVICE OFF-LINE SEND UNIT CHECK
004324  000137  002720          KILLEX: JMP     EXEC            ;RETURN TO THE EXEC
```

```
                                        .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
                                ;
004330  105737  012532        INPUT:  TSTB    TSTTYP          ;ILLEGAL ON FRIEND TEST
004334  001005                        BNE     10$             ;FRIEND -- GIVE AN ERROR
004336  004737  005302                JSR     PC,GDEV         ;GET THE DEVICE NUMBER
004342  120427  000054                CMPB    R4,#',          ;THE NEXT CHAR MUST BE A COMMA
004346  001402                        BEQ     20$             ;IT IS, CONTINUE
004350  000137  003026        10$:    JMP     CERR            ;AN ERROR WAS FOUND GIVE INDICATION
004354  004737  010230        20$:    JSR     PC,SUDEV        ;SET UP THE DEVICE STATUS TABLE POINTERS
004360  026327  000004  000734        CMP     SCURS(DTAB),#DISPSZ-4 ;ARE WE AT THE END OF THE BUFFER?
004366  002370                        BGE     10$             ;YES, GIVE AN ERROR
004370  032763  000001  000004        BIT     #1,SCURS(DTAB)  ;START INPUT ON EVEN BYTE ADDRESS
004376  001002                        BNE     30$             ;START SOM ON ODD BYTE ADDRESS
004400  005263  000004                INC     SCURS(DTAB)     ;INCR CURSOR TO ODD BYTE ADDRESS
004404  016305  000010        30$:    MOV     SOUTB(DTAB),R5  ;COMPUTE STARTING ADDRESS
004410  066305  000004                ADD     SCURS(DTAB),R5  ;
004414  112725  000112                MOVB    #SMI,(R5)+      ;START CHARACTER TO BUFFER
004420  010563  000020                MOV     R5,SMINS(DTAB)  ;SAVE START OF DATA LOCATION
004424  005263  000004        40$:    INC     SCURS(DTAB)     ;INCREMENT CURSOR POSITION
004430  026327  000004  000735        CMP     SCURS(DTAB),#DISPSZ-3 ;ARE WE AT THE END OF BUFFER
004436  001423                        BEQ     70$             ;YES, TERMINATE INPUT
004440  112204                        MOVB    (R2)+,R4        ;GET NEXT INPUTTED CHARACTER
004442  042704  177600                BIC     #177600,R4      ;SAVE L.O. 7 BITS
004446  020427  000015                CMP     R4,#CR          ;END OF INPUT?
004452  001415                        BEQ     70$             ;YES, SET UP TO EXIT
004454  020427  000040                CMP     R4,#SPACE       ;CAN CHARACTER BE CONVERTED?
004460  002410                        BLT     60$             ;NO, MUST BE BETWEEN 40 - 137
004462  020427  000137                CMP     R4,#'~          ;
004466  003005                        BGT     60$             ;NO, MUST BE BETWEEN 40 - 137
004470  162704  000040                SUB     #SPACE,R4       ;SCALE DOWN FOR INDEXING
004474  116425  012354        50$:    MOVB    ATOETB(R4),(R5)+;CONVERT CHARACATER AND MOVE TO DISPLAY BUFFER
004500  000751                        BR      40$             ;GET AND CONVERT NEXT CHARACATER
004502  005004                60$:    CLR     R4              ;ILLEGAL CHARACTER -- TREAT AS SPACE
004504  000773                        BR      50$
                                ;
                                ;       SET UP TO EXIT
                                ;               SET EOM INDICATOR
                                ;               QUEUE READ MANUAL INPUT REQUEST
                                ;
004506  112715  000152        70$:    MOVB    #EOM,(R5)       ;SET EOM INDICATOR
004512  005263  000004                INC     SCURS(DTAB)     ;INCREMENT CURSOR POINTER
004516  105263  000017                INCB    SRDRQ(DTAB)     ;QUEUE READ REQUEST
                                ;
                                ;       SEE IF THE DX IS CURRENTLY ACTIVE
                                ;
004522  105737  013070                TSTB    DXACT           ;IS DX ACTIVE?
```

```
3111  004526  001402                        BEQ     80$             ;NO, START ASYNCHRONOUS PROCESSING TO SEND ATTENTION
3112  004530  000137  002720                JMP     EXEC            ;YES, ATTENTION WILL BE TAKEN CARE OF BY DX
3113                                ;
3114                                ;       DX IS NOT ACTIVE
3115                                ;               CREATE PHONY INTERRUPT TO ALLOW DX TO ACT ON ATTENTION
3116                                ;
3117  004534  013746  177776       80$:    MOV     PSW,-(SP)       ;PSW TO PUSH STACK
3118  004540  012746  002720               MOV     #EXEC,-(SP)     ;RETURN ADDRESS TO PUSH STACK
3119  004544  012737  000340  177776       MOV     #340,PSW        ;INHIBIT INTERUPTS
3120  004552  010046                       MOV     R0,-(SP)        ;SET UP PUSH STACK FOR FAKE INTERRUPT
3121  004554  010146                       MOV     R1,-(SP)
3122  004556  010246                       MOV     R2,-(SP)
3123  004560  010346                       MOV     R3,-(SP)
3124  004562  010446                       MOV     R4,-(SP)
3125  004564  010546                       MOV     R5,-(SP)
3126  004566  000137  006554               JMP     DXEXEC          ;START PROCESSING THE ATTENTION
```

```
                                        .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
                                ;
004572  010237  013056          EBCDMP: MOV     R2,WK           ;SAVE WORD TO BE PRINTED
004576  113703  013056                  MOVB    WK,R3           ;GET LO BYTE
004602  042703  177400                  BIC     #177400,R3
004606  116337  011754  013056          MOVB    EBCDTB(R3),WK   ;CONVERT EBCDIC TO ASCII
004614  113703  013057                  MOVB    WK1,R3          ;GET HI BYTE AND CONVERT
004620  042703  177400                  BIC     #177400,R3
004624  116337  011754  013057          MOVB    EBCDTB(R3),WK1  ;CONVERT CHAR TO ASCII
004632  013702  013056                  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
                                ;
004636  105737  013055          ASCDMP: TSTB    LINECT          ;NEW LINE?
004642  001005                  BNE     10$             ;NO
004644  004737  011330                  JSR     PC,CRLF         ;YES, PRINT CR/LF
004650  112737  000036  013055          MOVB    #30.,LINECT     ;60 CHARACTERS PER LINE
004656  105337  013055          10$:    DECB    LINECT          ;DECR LINE COUNTER
004662  004737  011346                  JSR     PC,PRINT2               ;PRINT 2 CHARS
004666  000207                  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
                                ;
004670  004737  011330          OCTDMP: JSR     PC,CRLF         ;GIVE A CRLF
```

```
3183  004674  010046                ODMP:   MOV     R0,-(SP)        ;SAVE IMPORTANT REGISTERS
3184  004676  010246                        MOV     R2,-(SP)
3185  004700  010446                        MOV     R4,-(SP)
3186  004702  012704  000006                MOV     #6,R4           ;EXTRACT 6 OCTAL DIGITS
3187  004706  005000                        CLR     R0              ;CLEAR THE WORKING REGISTER
3188  004710  006102                        ROL     R2              ;MOVE HIGH ORDER BIT TO C-BIT
3189  004712  006100                10$:    ROL     R0              ;GET THE REMAINING BIT STILL IN LINK
3190  004714  042700  177770                BIC     #177770,R0      ;ONLY 3 LOW ORDER BITS
3191  004720  062700  000060                ADD     #'0,R0          ;MAKE ASCII
3192  004724  004737  011366                JSR     PC,PCHAR                ;PRINT IT ON THE TTY
3193  004730  006102                        ROL     R2              ;ROTATE THE NEXT OCTAL CHAR INTO POSITION
3194  004732  006102                        ROL     R2
3195  004734  006102                        ROL     R2
3196  004736  010200                        MOV     R2,R0           ;DATA TO WORKING REGISTER
3197  004740  005304                        DEC     R4              ;ARE WE DONE?
3198  004742  001363                        BNE     10$             ;NO, PRINT ANOTHER CHARACTER
3199  004744  012604                OCTEX:  MOV     (SP)+,R4        ;RESTORE USED REGISTERS
3200  004746  012602                        MOV     (SP)+,R2
3201  004750  012600                        MOV     (SP)+,R0
3202  004752  000207                        RTS     PC                      ;RETURN TO THE CALLER
3203
3204
3205                                  ;
3206                                  ;       DUMP WORD IN HEX ON THE TTY
3207                                  ;
3208                                  ;       CALLING SEQUENCE
3209                                  ;       ......R2 CONTAINS THE WORD TO BE PRINTED
3210                                  ;       JSR     PC,HEXDMP OR HDMP
3211                                  ;       ........RETURN
3212                                  ;
3213                                  ;       HEXDMP PERFORMS A CRLF BEFORE OUTPUTTING THE DATA
3214                                  ;
3215                                  ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
3216                                  ;
3217  004754  004737  011330        HEXDMP: JSR     PC,CRLF         ;DO A CR LF
3218  004760  010046                HDMP:   MOV     R0,-(SP)        ;SAVE THE WORKING REGISTERS
3219  004762  010246                        MOV     R2,-(SP)
3220  004764  010446                        MOV     R4,-(SP)
3221  004766  012704  000004                MOV     #4,R4           ;4 CHARACTERS PER WORD
3222  004772  006102                10$:    ROL     R2              ;ROTATE HIGH ORDER 4 BITS TO LOW ORDER 4 BITS
3223  004774  006102                        ROL     R2
3224  004776  006102                        ROL     R2
3225  005000  006102                        ROL     R2
3226  005002  010200                        MOV     R2,R0           ;TO WORKING REG
3227  005004  006100                        ROL     R0              ;GET THE LINK BIT TOO
3228  005006  042700  177760                BIC     #177760,R0      ;ONLY LOW ORDER 4 BITS
3229  005012  062700  000060                ADD     #'0,R0          ;MAKE ASCII IF NUMBER
3230  005016  020027  000071                CMP     R0,#'9          ;SHOULD IT BE A-F?
3231  005022  003402                        BLE     20$             ;NO,SHIP IT
3232  005024  062700  000007                ADD     #7,R0           ;YES, MAKE ALPHA
3233  005030  004737  011366        20$:    JSR     PC,PCHAR                ;PRINT THE HEX CHARACTER
3234  005034  005304                        DEC     R4              ;ARE WE DONE?
3235  005036  001355                        BNE     10$             ;NO, CONVERT AND PRINT NEXT CHARCATER
3236  005040  012604                        MOV     (SP)+,R4        ;YES, RESTORE REGISTERS AND EXIT
3237  005042  012602                        MOV     (SP)+,R2
3238  005044  012600                        MOV     (SP)+,R0
```

```
  3239  005046  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:
                                ;               R0 = 0 = NOT T/T, 1 = T/T
                                ;               R2 = NEXT CHAR POSITION IN COMMAND STRING
                                ;               SADDR = BEG ADDR TO BE DUMPED
                                ;               EADDR = END ADDR TO BE DUMPED
                                ;
                                ;       REGISTERS R5, R4, R3 WILL BE DESTROYED.
                                ;
                                ;       IF AN ERROR IS FOUND CONTROL IS PASSED TO CERR
                                ;
005050  005000          GLIMIT: CLR     R0              ;RESET BUFFER TYPE
005052  004737  011540          JSR     PC,COTB         ;GET FIRST PARAMETER
005056  005705                  TST     R5              ;WAS AN OCTAL NUMBER ENTERED?
005060  001014                  BNE     GLOCT           ;YES, OCTAL PARAMS
005062  120427  000124          CMPB    R4,#'T          ;T = TUMBLE TABLE
005066  001425                  BEQ     GLMTT           ;YES, SET UP T/T LIMITS
005070  120427  000111          CMPB    R4,#'I          ;I = INPUT BUFFER
005074  001436                  BEQ     GLMIN           ;YES, SET UP INPUT BUFFER LIMITS
005076  120427  000117          CMPB    R4,#'O          ;O = OUTPUT BUFFER
005102  001453                  BEQ     GLMOT           ;YES, SET UP OUTPUT BUFFER LIMITS
                                ;
                                ;       ERROR DETECTED - POP OFF RETURN ADDR AND GIVE ERROR
                                ;
005104  012601          GLERR:  MOV     (SP)+,R1
005106  000137  003026          JMP     CERR
                                ;
                                ;
                                ;       OCTAL LIMITS SPECIFIED
                                ;
005112  010337  013120  GLOCT:  MOV     R3,SADDR        ;SAVE START ADDR
005116  120427  000054          CMPB    R4,#',          ;CHECK FOR COMMA (,)
005122  001370                  BNE     GLERR
005124  004737  011540          JSR     PC,COTB         ;GET END ADDR
005130  005705                  TST     R5              ;WAS SECOND PARAM GIVEN?
005132  001764                  BEQ     GLERR           ;NO, ERROR
005134  010337  013122          MOV     R3,EADDR        ;SAVE END ADDR
005140  000454                  BR      GLEX            ;PREPARE TO EXIT
                                ;
                                ;       SET UP LIMITS OF TUMBLE TABLE
                                ;
005142  112204          GLMTT:  MOVB    (R2)+,R4
005144  120427  000124          CMPB    R4,#'T          ;MUST BE TT
005150  001355                  BNE     GLERR           ;ILLEGAL ENTRY
005152  013737  013060  013120  MOV     TTPTR,SADDR
```

```
3296  005160  062737  000776  013120          ADD     #TTSIZE-2,SADDR ;COMPUTE ADDRESS OF APPR DUPLICATE TT ENTRY
3297  005166  005200                          INC     R0              ;INDICATE DUMP TUMBLE TABLE
3298  005170  000437                          BR      GLEX1           ;SET UP TO EXIT
3299
3300
3301                                  ;
3302                                  ;       SET UP LIMITS FOR INPUT BUFFER OF DEVICE 0
3303                                  ;
3304  005172  112204                  GLMIN:  MOVB    (R2)+,R4
3305  005174  120427  000116                  CMPB    R4,#'N          ;MUST BE IN
3306  005200  001341                          BNE     GLERR           ;ILLEGAL ENTRY
3307  005202  013704  013064                  MOV     SDEVTB,R4       ;GET ADDR OF DEV 0 STATUS TABLE
3308  005206  016437  000006  013120          MOV     SINBF(R4),SADDR
3309  005214  013737  013120  013122          MOV     SADDR,EADDR
3310  005222  062737  000741  013122          ADD     #DISPSZ+1,EADDR ;DISPLAY SIZE + ROOM FOR LINE ADDRESS
3311  005230  000417                          BR      GLEX1
3312
3313
3314                                  ;
3315                                  ;       SET UP LIMITS FOR OUTPUT BUFFER OF DEVICE 0
3316                                  ;
3317  005232  112204                  GLMOT:  MOVB    (R2)+,R4
3318  005234  120427  000124                  CMPB    R4,#'T          ;MUST BE OT
3319  005240  001321                          BNE     GLERR           ;ILLEGAL ENTRY
3320  005242  013704  013064                  MOV     SDEVTB,R4       ;GET ADDR OF DEV 0 STATUS TABLE
3321  005246  016437  000010  013120          MOV     SOUTB(R4),SADDR ;COMPUTE STARTING AND ENDING ADDRESSESS OF SPECIFIED BUF
3322  005254  013737  013120  013122          MOV     SADDR,EADDR
3323  005262  062737  000737  013122          ADD     #DISPSZ-1,EADDR ;DISPLAY CHAR BUFFER
3324  005270  112204                  GLEX1:  MOVB    (R2)+,R4        ;GET NEXT INPUT CHARACTER AND UPDATE POINTER
3325                                  ;
3326                                  ;       INSURE ENTRY PROPERLY DELIMITED AND RETURN TO CALLER
3327                                  ;
3328  005272  120427  000054          GLEX:   CMPB    R4,#',          ;CHECK FOR
3329  005276  001302                          BNE     GLERR           ;ENTRY NOT PROPERLY DELIMITED (ERROR)
3330  005300  000207                          RTS     PC
```

```
                                        ;
                                        ;       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
                                        ;
005302  004737  011604          GDEV:   JSR     PC,CHTB         ;CONVERT THE HEX TO BINARY
005306  163703  012522                  SUB     SDEV,R3         ;-STARTING ADDRESS
005312  100406                          BMI     10$             ;ERROR ON INPUT
005314  005203                          INC     R3              ;MAKE BETWEEN 1 AND 8
005316  120337  013067                  CMPB    R3,MAXDEV       ;IS DEVICE NUMBER TOO BIG?
005322  101002                          BHI     10$             ;YES, GIVE ERROR
005324  010300                          MOV     R3,DEV          ;SET UP THE DEVICE NUMBER
005326  000207                          RTS     PC
005330  000137  003026          10$:    JMP     CERR            ;INPUT PARAM ERROR
                                        .SBTTL  DX11-B ISR (INTERRUPT REQUEST LOGIC AND TUMBLE TABLE DECODE LOGIC)
                                        ;
                                        ;       D X 1 1 - B   I S R
                                        ;
                                        ;       DX11 ISR AND RELATED SUBROUTINE REGISTER USAGE
                                        ;
                                        ;       R0      DEV     DEVICE NUMBER
                                        ;       R1              UN ASSIGNED
                                        ;       R2              UNASSIGNED
                                        ;       R3      DTAB    ADDRESS OF CURRENT DEVICE TABLE
                                        ;       R4      TT1     TUMBLE TABLE ENTRY 1
                                        ;       R5      TT2     TUMBLE TABLE ENTRY 2
                                        ;
                                        ;       THE ABOVE REGISTER DESIGNATIONS REPRESENT WHAT USUALLY WILL
                                        ;       BE CONTAINED IN A REGISTER DURING DX ISR PROCESSING. HOWEVER,
                                        ;       AS SITUATIONS DICTATE REGISTERS MAY BE USED FOR DIFFERENT
                                        ;       PURPOSES.
                                        ;
005334  010046                  DXISR:  MOV     R0,-(SP)        ;SAVE HARDWARE REGISTERS
005336  010146                          MOV     R1,-(SP)
005340  010246                          MOV     R2,-(SP)
005342  010346                          MOV     R3,-(SP)
005344  010446                          MOV     R4,-(SP)
005346  010546                          MOV     R5,-(SP)
005350  013702  013060                  MOV     TTPTR,R2        ;CHECK FOR ZERO T/T ENTRY UPON INTERRUPT
005354  005712                          TST     (R2)
005356  001002                          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
                                        ;
005360  000137  010212                  JMP     DXEXIT
```

```
                                        ;
                                        ;       PROCESS TUMBLE TABLE ENTRIES
                                        ;               FOR CONVIENCE THE PROCESSING IS BEING PERFORMED
                                        ;               AT THE INTERRUPT LEVEL. IT IS SUGGESTED THAT IN
                                        ;               NORMAL PROCESSING ENVIRONMENTS THIS PROCESSING
                                        ;               BE DISTRIBUTED TO LESS 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 SEQUENTIAL_Y 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.
                                        ;
005364  013702  013060          LOOP:   MOV     TTPTR,R2        ;GET T/T PTR
005370  005712                         TST     (R2)            ;ANY ENTRIES LEFT IN T/T?
005372  001002                         BNE     10$
005374  000137  006554                 JMP     DXEXEC          ;NO, EXECUTE NEXT DX COMMAND
                                        ;
                                        ;       TUMBLE TABLE ENTRY AVAILABLE FOR PROCESSING
                                        ;               RESET THE DONE BIT (RELIEVE INTERRUPT)
                                        ;               COPY TUMBLE TABLE ENTRY TO DUPLICATE TUMBLE TABLE (FOR SYSTEM TESTS PURP
                                        ;               RESET TUMBLE TABLE ENTRY (2 WORDS)
                                        ;
005400  042777  000200  005062   10$:    BIC     #DONE,@DXCS     ;CLEAR DONE
005406  010203                         MOV     R2,R3           ;SET UP PTR TO DUP T/T
005410  062703  001000                 ADD     #TTSIZE,R3
005414  011223                         MOV     (R2),(R3)+      ;SAVE T/T ENTRY #1
005416  011204                         MOV     (R2),TT1
005420  005022                         CLR     (R2)+           ;CLEAR T/T ENTRY #1
005422  011223                         MOV     (R2),(R3)+      ;SAVE T/T ENTRY #2
005424  011205                         MOV     (R2),TT2
005426  005022                         CLR     (R2)+           ;CLEAR T/T ENTRY #2
                                        ;
                                        ;       CHECK FOR POINTER WRAP AROUND
                                        ;
005430  032702  000777                 BIT     #TTSIZE-1,R2    ;AT END OF BUFFER?
                                                                ;NOTE -- POWER OF 2 BOUNDRY
005434  001002                         BNE     20$             ;NO
005436  013702  013062                 MOV     TTADDR,R2       ;YES, RESET PTR
005442  010237  013060           20$:  MOV     R2,TTPTR        ;SAVE T/T PTR
                                        ;
                                        ;       START PROCESSING TUMBLE ENTRY ENTRY
                                        ;               SAVE DEVICE ADDRESS
                                        ;               CHECK FOR SYSTEM RESET
                                        ;               VALIDATE DEVICE ADDRESS
```

```
                                        ;
                                        ;       NOTE -- IF SYSTEM RESET OCCURRED, THERE IS NO GUARANTEE
                                        ;               THAT THE DEVICE ADDRESS WILL BE VALID.
                                        ;
005446  010500                          ;       MOV     TT2,DEV         ;GET DEV #
005450  042700  177400                  ;       BIC     #177400,DEV
005454  010037  013106                  ;       MOV     DEV,CDEV        ;SAVE CURRENT DEVICE NUMBER
005460  042777  000200  005002          ;       BIC     #DONE,@DXCS     ;CLEAR DONE
005466  032704  010000                  ;       BIT     #SYSRST,TT1     ;SYSTEM RESET?
005472  001066                          ;       BNE     PSYSRT          ;YES, PERFORM SYSTEM RESET FUNCTION
005474  163700  012522                  ;       SUB     SDEV,DEV        ;GET IN 0-7 RANGE - IF VALID
005500  100403                          ;       BMI     30$             ;INVALID DEVICE NUMBER
005502  120037  013067                  ;       CMPB    DEV,MAXDEV      ;VALID DEVICE?
005506  103405                          ;       BLO     40$             ;YES, NOT TOO BIG
                                        ;
                                        ;       INVALID DEVICE ADDRESS - BITCH
                                        ;
                                        ;       AN INVALID DEVICE ADDRESS WILL GENERALLY INDICATE
                                        ;       A PROBLEM IN THE CONFIGURATION OF DX DEVICE
                                        ;       ADDRESSES. BASICALLY THE DX HAS BEEN STRAPPED
                                        ;       TO HANDLE DEVICE ADDRESSES WHICH OVERLAP WITH
                                        ;       OTHER DEVICES ON THE CHANNEL.
                                        ;
005510  004137  011506          30$:    ;       JSR     R1,INMES        ;PRINT "INVALID DEVICE"
005514  013762                          ;       .WORD   ILLMES
005516  000137  006536                  ;       JMP     DXAB            ;ABORT DX11
                                        ;
                                        ;       COMPUTE ADDRESS OF SPECIFIED DEVICES STATUS TABLE
                                        ;
005522  005200                  40$:    ;       INC     DEV             ;MAKE DEVICE NUMBER 1 -8
005524  004737  010230                  ;       JSR     PC,SUDEV        ;SET UP ADDR OF DEV STAT TABLE
```

```
                                        ;
                                        ;       DECODE DX TUMBLE TABLE STATUS ENTRY
                                        ;
                                        ;               THE FOLLOWING PROCESS INDICATES THE ORDER IN WHICH
                                        ;               THE TUMBLE TABLE STATUS ENTRY SHOULD BE DECODED.
                                        ;               THIS ORDER IS IMPORTANT AND SHOULD BE ADHERED
                                        ;               TO FOR MOST EMULATIONS.
                                        ;
005530  032704  020000        TSELRST:  BIT     #SELRST,TT1     ;SELECTIVE RESET?
005534  001402                          BEQ     TINDSC          ;NO, TEST INTERFACE DISC.
005536  000137  005776                  JMP     PSELRT          ;YES, PERFORM SELECTIVE RESET
005542  032704  004000        TINDSC:   BIT     #INFDSC,TT1     ;INTERFACE DISCONNECT?
005546  001402                          BEQ     TNXM            ;NO, CHECK NON-EXISTENT MEMORY
005550  000137  006026                  JMP     PINDSC          ;YES, PERFORM INTER DISC
005554  032704  040000        TNXM:     BIT     #NXM,TT1        ;NON-EXISTENT MEMORY ERROR?
005560  001402                          BEQ     TESEND          ;NO, ES END
005562  000137  006102                  JMP     PNXM            ;YES, PROCESS NON-EXISTENT MEMORY ERROR
005566  032704  000100        TESEND:   BIT     #ESEND,TT1      ;WAS STATUS ACCEPTED?
005572  001402                          BEQ     TPARER          ;NO, CHECK FOR PARITY ERROR
005574  000137  006114                  JMP     PESEND          ;YES, PERFORM STATUS ACCEPT
005600  032704  100000        TPARER:   BIT     #PARER,TT1      ;DID A PARITY ERROR OCCUR?
005604  001402                          BEQ     TCHIS           ;NO, CHECK FOR CHIS
005606  000137  006244                  JMP     PPARER          ;YES, PROCESS PARITY ERROR
005612  032704  000200        TCHIS:    BIT     #CHIS,TT1       ;DID CHANNEL START A SELECTION SEQ?
005616  001402                          BEQ     TCHEND          ;NO, CHECK FOR CHANNEL DATA END
005620  000137  006274                  JMP     PCHIS           ;YES, PROCESS SELECTION SEQUENCE
005624  032704  000040        TCHEND:   BIT     #CHEND,TT1      ;DID CHANNEL END OCCUR?
005630  001402                          BEQ     TCUEND          ;NO, CHECK FOR CONTROL UNIT END
005632  000137  006456                  JMP     PCHEND          ;YES, PROCESS CHANNEL END
005636  032704  000020        TCUEND:   BIT     #CUEND,TT1      ;DID A CONTROL UNIT END OCCUR?
005642  001650                          BEQ     LOOP            ;NO, IGNORE ENTRY -- ASSUME STACK STATUS
005644  000137  006442                  JMP     PCUEND          ;YES, PROCESS CONTROL UNIT END
```

```
                                        .SBTTL  DX11-B ISR (TUMBLE TABLE ENTRY PROCESSING LOGIC)
                                ;
                                ;       SYSTEM RESET OCCURRED FROM 360
                                ;               CLEAR ALL DEVICE STATUS TABLES AND RESPECTIVE
                                ;                       SENSE BYTES
                                ;               RESET DX ACTIVE FLAGS AND COMMAND CHAIN FLAG
                                ;
005650  012700  000001        PSYSRT: MOV     #1,DEV          ;START AT FIRST DEVICE
005654  004737  010256        10$:    JSR     PC,CDEVST       ;CLEAR DEVICE STATUS TABLE
005660  004737  010310                JSR     PC,CSPWST       ;RESET SPW STATUS WORD UPON SYSTEM RESET
005664  105063  000002                CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
005670  105063  000017                CLRB    SRDRQ(DTAB)     ;CLEAR THE READ REQUEST
005674  005063  000020                CLR     SMINS(DTAB)     ;CLEAR THE BEG OF MANUAL INPUT  ADDRESS
005700  005063  000004                CLR     SCURS(DTAB)     ;RESET THE CURSOR
005704  105063  000003                CLRB    SSTAT(DTAB)     ;CLEAR THE STATUS REGISTER
005710  016301  000010                MOV     SOUTB(DTAB),R1  ;SET UP TO CLEAR THE DISPLAY BUFFER
005714  012702  000740                MOV     #DISPSZ,R2      ;SET UP NUMBER OF CHARACTERS IN DISPLAY
005720  112704  000100                MOVB    #EBCDSP,R4      ;ASSUME 2848 DIAGNOSTIC TEST MODE
005724  105737  012532                TSTB    TSTTYP          ;WHAT TYPE OF TEST?
005730  001402                      BEQ     20$             ;       IF 2848, USE EBCDIC SPACE
005732  113704  012534                MOVB    FILLCH,R4       ;       FRIEND TEST -- USE CURRENT FILL CHARACTER
                                
005736  110421                20$:    MOVB    R4,(R1)+        ;USE THE FILL CHARACTER
005740  005302                      DEC     R2              ;ARE WE DONE?
005742  001375                      BNE     20$             ;NO, LOOP TILL DONE
005744  005200                      INC     DEV             ;TO NEXT DEVICE
005746  120037  013067                CMPB    DEV,MAXDEV      ;ARE WE DONE?
005752  003740                      BLE     10$             ;NO, CLEAR NEXT DEV STAT TABLE
005754  105037  013070                CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
005760  105037  013074                CLRB    CMDCHF          ;CLEAR COMMAND CHAINING FLAG
005764  042777  000400  004476        BIC     #CUBUSY,@DXCS   ;RESET CU BUSY FLAG
005772  000137  005364                JMP     LOOP            ;PROCESS NEXT T/T ENTRY
                                
                                
                                ;
                                ;       CHANNEL ISSUED A SELECTIVE RESET
                                ;               RESET THE DEVICE STATUS TABLE FOR THAT DEVICE + SENSE
                                ;       NOTE: THE SEL RESET IS ISSUED AGAINST THE CURRENT
                                ;               ACTIVE DEVICE
                                ;
005776  004737  010256        PSELRT: JSR     PC,CDEVST       ;CLEAR DEVICE STATUS TABLE
006002  004737  010310                JSR     PC,CSPWST       ;RESET SPW STATUS RESPONSE
006006  105063  000002                CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
006012  105037  013070                CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
006016  105037  013074                CLRB    CMDCHF          ;CLEAR COMMAND CHAIN FLAG
006022  000137  005364                JMP     LOOP
                                
                                
                                ;
                                ;       INTERFACE DISCONNECT WAS ISSUED FROM THE 360
                                ;               THIS IS DIRECTED TO A SPECIFIC DEVICE AND IS UNDER
                                ;               360 PROGRAM CONTROL
                                ;
                                ;               IF THE DEVICE WAS ACTIVE
                                ;                       ITS DEVICE STATUS TABLE WILL BE CLEARED
```

```
3562                                    ;                       AND CHAN END ! DEVICE END SET IN STATUS BYTE
3563                                    ;               IF THE DEVICE IS NOT ACTIVE THE COMMAND WILL BE IGNORED
3564                                    ;
3565  006026  004737  010310            PINDSC: JSR     PC,CSPWST       ;CLEAR THE SPW STATUS RESPONSE
3566  006032  105763  000000                    TSTB    SCMD(DTAB)      ;IS DEVICE ACTIVE?
3567  006036  001417                            BEQ     20$             ;NO, IGNORE
3568  006040  004737  010256                    JSR     PC,CDEVST       ;CLEAR THE DEVICE STATUS TABLE
3569  006044  012763  000003  000000            MOV     #CEDE,SCMD(DTAB);QUE DEV END + CHAN END
3570  006052  120037  013070                    CMPB    DEV,DXACT       ;IS DEVICE USING DX NOW?
3571  006056  001002                            BNE     10$             ;NO
3572  006060  105037  013070                    CLRB    DXACT           ;YES, RELEASE DX
3573  006064  120037  013074            10$:    CMPB    DEV,CMDCHF      ;DOES DEVICE HAVE CMD CHAIN SPEC?
3574  006070  001002                            BNE     20$             ;NO, GET NEXT T/T ENTRY
3575  006072  105037  013074                    CLRB    CMDCHF          ;YES, CLEAR FLAG
3576  006076  000137  005364            20$:    JMP     LOOP            ;GET NEXT T/T ENTRY
3577
3578
3579                                    ;
3580                                    ;       A NON-EXISTANT MEMORY CONDITION OCCURRED
3581                                    ;               THIS WILL USUALLY TRAP OUT FIRST
3582                                    ;
3583  006102  004137  011506            PNXM:   JSR     R1,INMES        ;PRINT "NON EX MEM"
3584  006106  013706                            .WORD   NXMMSG
3585  006110  000137  006536                    JMP     DXAB            ;ABORT DX AND RETURN TO EXEC
3586
3587
3588
3589                                    ;
3590                                    ;       THE LAST STATUS SENT TO THE 360 WAS ACCEPTED, CLEAR DX
3591                                    ;               ACTIVE FLAG
3592                                    ;       IF LAST OP WAS A WRITE PERFORM THE DISPLAY CONTROL ROUTINE
3593                                    ;
3594  006114  004737  010310            PESEND: JSR     PC,CSPWST       ;RESET THE SPW STATUS BYTE
3595  006120  105763  000001                    TSTB    SLCMD(DTAB)     ;DOES LAST COMMAND REQUIRE 2260 DISPLAY EMULATION?
3596  006124  001402                            BEQ     10$             ;NO
3597  006126  004737  010402                    JSR     PC,DISCTL       ;YES, FORMAT THE DISPLAY
3598  006132  126327  000000  000011    10$:    CMPB    SCMD(DTAB),#11  ;WAS ATTN ACCEPTED?
3599  006140  001003                            BNE     20$             ;NO, CONTINUE
3600  006142  112763  000002  000017            MOVB    #2,SRDRQ(DTAB)  ;YES, INDICATE 360 ACCEPTANCE
3601  006150  105063  000001            20$:    CLRB    SLCMD(DTAB)
3602  006154  105037  013070                    CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3603  006160  004737  010256                    JSR     PC,CDEVST       ;CLEAR THE DEVICE STATUS TABLE
3604  006164  032704  000004                    BIT     #CMDCHN,TT1     ;WAS COMMAND CHAINING SPECIFIED?
3605  006170  001402                            BEQ     30$             ;NO,
3606  006172  110037  013074                    MOVB    DEV,CMDCHF      ;YES, SAVE THE DEVICE NUMBER
3607  006176  123727  013066  000105    30$:    CMPB    DXSTPF,#'E      ;WAS STOP ON END SEQ SPEC?(SE)
3608  006204  001552                            BEQ     STPDX           ;YES, DISABLE THE DX
3609
3610  006206  032704  000010                    BIT     #ISSREJ,TT1     ;WAS AN ISS REJ DETECTED?
3611  006212  001412                            BEQ     50$             ;NO, EXIT
3612
3613                                    ;
3614                                    ;       INIT SELECTION SEQUENCE WAS REJECTED BY DX (FAST CU BUSY SEQUENCE)
3615                                    ;               IF FREIND TEST MODE -- QUEUE CONTROL UNIT END
3616                                    ;                       ON UNIT COMPLETING TRANSFER
3617                                    ;               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.
                                        ;
006214  105737  012532                        TSTB    TSTTYP          ;PROCESS SEPARETLY IF FRIEND
006220  001004                                BNE     40$             ;FRIEND --QUEUE CU END ON BUSY UNIT
006222  012700  000001                        MOV     #1,DEV          ;SET UP TO SEND CUE ON LOW ORDER CONTROLLER ADDR
006226  004737  010230                        JSR     PC,SUDEV        ;  X
006232  112763  000010  000000  40$:         MOVB    #QCUE,SCMD(DTAB);QUEUE CONTROL UNIT END
006240  000137  005364          50$:         JMP     LOOP            ;LOOP BACK AND PROCESS NEXT TUMBLE TABLE ENTRY
                                        ;
                                        ;       PARITY ERROR WAS DETECTED
                                        ;
006244  004137  011506          PPARER: JSR     R1,INMES        ;PRINT "PARITY ERROR"
006250  013736                        .WORD   PARMES
006252  123727  013066  000120        CMPB    DXSTPF,#'P      ;STOP ON PARITY ERROR??
006260  001524                        BEQ     STPDX           ;YES, DISABLE THE DX
006262  152763  000002  000003        BISB    #UCHK,SSTAT(DTAB) ;SET UNIT CHECK IN STATUS WORD
006270  000137  005612                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
                                        ;
006274  004737  010310          PCHIS:  JSR     PC,CSPWST       ;RESET THE SPW STATUS BYTE
                                                                ;ON NEXT CHANNEL INITIATED SELECTION SEQUENCE
006300  123727  013066  000111        CMPB    DXSTPF,#'I      ;WAS STOP ON ISS SPECIFIED(SI)
006306  001511                        BEQ     STPDX           ;YES, DISABLE DX
006310  032704  000001                BIT     #CMDREJ,TT1     ;WAS COMMAND REJECTED BY DX?
006314  001022                        BNE     20$             ;YES, COMMAND REJECTED BY THE DX
                                        ;
                                        ;       VALID COMMAND, SET UP TO PROCESS IT
                                        ;
006316  105005                        CLRB    TT2             ;RESET DEVICE ADDRESS BITS
006320  000305                        SWAB    TT2             ;COMMAND TO L.O. BYTE
006322  105705                        TSTB    TT2             ;TEST I/O COMMAND?
006324  001437                        BEQ     50$             ;YES, IGNORE
006326  120527  000003                CMPB    TT2,#NOP        ;WAS COMMAND A NOP?
006332  001434                        BEQ     50$             ;YES, IGNORE IT
006334  020527  000012                CMP     TT2,#12         ;IS THIS A VALID COMMAND?
006340  003405                        BLE     10$             ;YES, QUEUE TO BE EXECUTED
006342  004137  011506                JSR     R1,INMES        ;NO -- REPORT AN ILLEGAL COMMAND RECIEVED FROM THE DX
006346  014050                        .WORD   INVLDC
006350  000137  006536                JMP     DXAB            ;AND  ABORT THE PROGRAM
006354  110563  000000          10$:  MOVB    TT2,SCMD(DTAB)  ;QUEUE COMMAND TO BE PROCESSED
006360  000421                        BR      50$             ;EXIT + PROCESS NEXT T/T ENTRY
```

```
                                        ;
                                        ;       COMMAND WAS REJECTED, DETERMINE WHY
                                        ;
006362  105763  000016          20$:  TSTB    SONLF(DTAB)       ;IS DEVICE ON LINE?
006366  001404                        BEQ     30$               ;YES, TEST PARITY ERROR
                                        ;
                                        ;       DEVICE OFF-LINE -- RESPOND INTERVENTION REQUIRED SENSE CONDITION
                                        ;
006370  052763  000100  000002        BIS     #INTREQ,SSENSE(DTAB) ;SET INTERVENTION REQUIRED IN SENSE BYTE
006376  000412                        BR      50$               ;FINISH UP CHANNEL INITIATED SELECTION PROCESS
006400  032704  100000          30$:  BIT     #PARER,TT1        ;WAS A PARITY ERROR DETECTED?
006404  001404                        BEQ     40$               ;NO, MUST BE ILLEGAL COMMAND
                                        ;
                                        ;       COMMAND WAS REJECTED BECAUSE OF A PARITY ERROR
                                        ;               SET UP BUS OUT SENSE RESPONSE
                                        ;
006406  052763  000040  000002        BIS     #BUSOUT,SSENSE(DTAB) ;SET BUS OUT FLAG
006414  000403                        BR      50$               ;EXIT
                                        ;
                                        ;       INVALID COMMAND RECEIVED FROM 360
                                        ;               SET UP COMMAND REJECT SENSE RESPONSE
                                        ;
006416  052763  000200  000002  40$:  BIS     #SCMDRJ,SSENSE(DTAB) ;SET CMD REJ FLAG
                                        ;
                                        ;       CHANNEL INITIATED SELECTION SEQUENCE COMMON EXIT LOGIC
                                        ;               IF SELECTED DEVICE HAS COMMAND CHAINING IN AFFECT -- KILL IT
                                        ;
006424  120037  013074          50$:  CMPB    DEV,CMDCHF        ;DOES DEVICE HAVE COMMAND CHAINING SPECIFIED?
006430  001002                        BNE     60$               ;NO, GET NEXT TUMBLE TABLE ENTRY
006432  105037  013074                CLRB    CMDCHF            ;YES, CLEAR THE COMMAND CHAINING FLAG
006436  000137  005364          60$:  JMP     LOOP              ;AND GET THE NEXT T/T ENTRY
                                        ;
                                        ;       CONTROL UNIT END OF DATA TRANSFER WAS DETECTED
                                        ;               IF TRANSFER COMPLETE PREPARE ENDING SEQ RESP
                                        ;               IF TRANSFER INCOMPLETE INCR BUFFER ADDRESS
                                        ;                                      DECR BYTE COUNT
                                        ;
006442  105037  013070          PCUEND: CLRB    DXACT             ;CLEAR DX ACTIVE FLAG
006446  004737  010342                  JSR     PC,MUXEND         ;HANDLE MUX DATA TRANSFER COMPLETION
006452  103017                        BCC     PCHEX             ;IF SEL CHAN OR MUX D/T NOT DONE, MERELY EXIT
                                        ;
                                        ;       MUX DATA TRANSFER COMPLETE
                                        ;               TREAT SAME AS SEL CHANNEL DONE
                                        ;
006454  000404                        BR      PCHEN1
                                        ;
                                        ;       CHANNEL END WAS DETECTED
                                        ;               PREPARE ENDING SEQUENCE RESPONSE
                                        ;
006456  105037  013070          PCHEND: CLRB    DXACT             ;CLEAR DX ACTIVE FLAG
006462  004737  010342                  JSR     PC,MUXEND         ;IF MUX CHANNEL HANDLE DATA TRANSFER
```

```
3730  006466  123727  013066  000104  PCHEN1: CMPB    DXSTPF,#'D      ;STOP ON DATA TRANSFER DONE?(SD)
3731  006474  001416                          BEQ     STPDX           ;YES, DISABLE DX
3732  006476  112763  000003  000000          MOVB    #CEDE,SCMD(DTAB);QUE END SEQ RESPONSE
3733  006504  167763  003766  000014          SUB     @DXBC,SRBYTC(DTAB) ;SAVE REMAINING BYTE COUNT
3734  006512  032704  100000          PCHEX:  BIT     #PARER,TTI      ;WAS A PARITY ERROR SENSED?
3735  006516  001403                          BEQ     10$             ;NO, PROCESS NEXT TUMBLE TABLE ENTRY
3736  006520  152763  000020  000002          BISB    #EQICHK,SSENSE(DTAB) ;YES, SET EQUIP CHECK IN SENSE
3737  006526  000137  005364          10$:    JMP     LOOP            ;LOOP BACK + PROCESS NEXT TT ENTRY
3738
3739
3740
3741                                  ;
3742                                  ;       STOP DX, SOMEONE WANTS TO INVESTIGATE
3743                                  ;
3744  006532  105037  013066          STPDX:  CLRB    DXSTPF          ;CLEAR STOP FLAG
3745
3746
3747                                  ;
3748                                  ;       ABORT   CURRENT DX OPERATION
3749                                  ;               UNRECOVERABLE ERROR OCCURRED
3750                                  ;
3751  006536  042777  001100  003724  DXAB:   BIC     #DXONLN!DXENB,@DXCS ;DISABLE THE DX
3752  006544  105237  013072                  INCB    DXABFL          ;SET THE DX ABORT FLAG SO THE
3753                                                                  ;DX REGISTERS WILL BE PRINTED
3754  006550  000137  010212                  JMP     DXEXIT          ;EXIT FROM INTERRUPT
3755                                          .SBTTL  DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)
3756
3757                                  ;       EXECUTE NEXT COMMAND FOR THE DX
3758                                  ;
3759  006554  105737  012526          DXEXEC: TSTB    CHTYPE          ;CHANNEL TYPE 0=M, 1=S
3760  006560  001002                          BNE     SEX             ;SELECTOR CHANNEL EXEC
3761  006562  000137  007210                  JMP     MEX             ;MULTIPLEXER EXEC
3762
3763                                  ;       SEX  -- SELECTOR CHANNEL EXECUTIVE
3764                                  ;
3765                                  ;       SEX EXECUTES COMMANDS FOR THE DX TO A SELECTOR CHANNEL
3766                                  ;
3767                                  ;       ON A SELECTOR CHANNEL A COMMAND WILL BE COMPLETED
3768                                  ;               BEFORE ATTEMPTING TO EXECUTE A COMMAND ON ANOTHER
3769                                  ;               DEVICE, ISS-DATA TRANSFER-ES.
3770                                  ;
3771                                  ;       DATA TRANSFERS ARE COMPLETED IN ONE BURST
3772                                  ;
3773  006566  013700  013076          SEX:    MOV     SELDEV,DEV      ;GET SEL DEV #
3774  006572  004737  010230          10$:    JSR     PC,SUDEV        ;SET UP DEV STATUS TABLE ADDR
3775  006576  105763  000000                  TSTB    SCMD(DTAB)      ;ANY JOB TO DO?
3776  006602  001030                          BNE     60$             ;YES, EXECUTE IT
3777  006604  105737  013074                  TSTB    CMDCHF          ;WAS COMMAND CHAINING SPECIFIED
3778  006610  001402                          BEQ     30$             ;NO
3779  006612  000137  010212                  JMP     DXEXIT          ;YES, WAIT FOR COMMAND
3780  006616  126327  000017  000001  30$:    CMPB    SRDRQ(DTAB),#1  ;IS ATTENTION TO BE SENT?
3781  006624  001004                          BNE     40$             ;NO, CONTINUE
3782  006626  112763  000011  000000          MOVB    #11,SCMD(DTAB)  ;YES, SET UP TO SEND THE ATTENTION
3783  006634  000413                          BR      60$             ;FOR THE READ MANUAL INPUT
3784  006636  005200                  40$:    INC     DEV             ;TO NEXT DEV
3785  006640  120037  [illegible]067          CMPB    DEV,MAXDEV      ;HAVE WE TRIED THE HIGHEST DEVICE?
```

```
3786  006644  003402                        BLE     50$                ;NO
3787  006646  012700  000001                MOV     #1,DEV             ;YES, RESTART AT FIRST DEVICE
3788  006652  020037  013076        50$:    CMP     DEV,SELDEV         ;IS THIS WHERE IT ALL STARTED?
3789  006656  001345                        BNE     10$                ;NOPE, TEST THIS DEVICE
3790  006660  000137  010212                JMP     DXEXIT             ;EXIT -- NO TASKS PENDING
3791                                ;
3792                                ;       THERE IS A JOB TO DO, LETS DO IT
3793                                ;
3794  006664  116304  000000        60$:    MOVB    SCMD(DTAB),R4      ;COMMAND TO INDEX
3795  006670  005304                        DEC     R4                 ;SCALE TO 0 - 11
3796  006672  006304                        ASL     R4                 ;MAKE WORD ADDRESS
3797  006674  010037  013076                MOV     DEV,SELDEV         ;SAVE CURRENT DEVICE ADDR
3798  006700  000174  006704                JMP     @SCMDTB(R4)        ;EXECUTE THE COMMAND
3799  006704  006730                SCMDTB: .WORD   SWRITE             ;1 = WRITE FULL BUFFER
3800  006706  007020                        .WORD   SRMI               ;2 = READ MANUAL INPUT
3801  006710  010042                        .WORD   ESEQ               ;3 = ENDING SEQUENCE
3802  006712  010140                        .WORD   SENSCM             ;4 = SENSE COMMAND
3803  006714  006730                        .WORD   SWRITE             ;5 = WRITE LINE ADDRESS
3804  006716  007120                        .WORD   SREAD              ;6 = READ FULL BUFFER
3805  006720  007770                        .WORD   ERASCM             ;7 = ERASE COMMAND
3806  006722  007750                        .WORD   CONUNE             ;10 = CONTROL UNIT END
3807  006724  007760                        .WORD   SATTN              ;11 = SEND ATTENTION TO 360
3808  006726  007020                        .WORD   SSRMI              ;12 = READ SHORT MANUAL INPUT
```

```
                                      ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
                                      :
                                      :            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

006730  016377  000006  003536  SWRITE: MOV    SINBF(DTAB),@DXBA ;SET UP BUFFER ADDRESS
006736  163777  013104  003530          SUB    PHYOFF,@DXBA      ;FOR VIRTUAL MEMORY -- OFFSET FOR PHYSICAL ADDRESS
006744  013702  013110                  MOV    DEVCON,R2         ;COMPUTE DEVICE ADDRESS
006750  060002                          ADD    DEV,R2
006752  110277  003510                  MOVB   R2,@DXCA
006756  012777  177037  003512          MOV    #-DISPSZ-1,@DXBC;SET UP BYTE COUNT FOR MAX. WRITE LINE ADDRESS
006764  116363  000000  000001          MOVB   SCMD(DTAB),SLCMD(DTAB)  ;SET WRITE FLAG
006772  105063  000002                  CLRB   SSENSE (DTAB)     ;CLEAR SENSE BYTE
006776  110037  013070                  MOVB   DEV,DXACT         ;SET DX ACTIVE FLAG
007002  005063  000014                  CLR    SRBYTC(DTAB)      ;RESET REMAINING BYTE COUNT
007006  052777  000003  003454          BIS    #DXWR,@DXCS       ;START TRANSFER
007014  000137  010212                  JMP    DXEXIT            ;RETURN FROM INTERRUPT



                                      :
                                      :            PERFORM READ MANUAL INPUT COMMANDS
                                      :
007020                          SRMI:
007020  105737  012532          SSRMI:  TSTB   TSTTYP            ;IS TEST FOR FRIEND?
007024  001035                          BNE    SREAD             ;YES, TREAT ALL READS AS READ FULL BUFFER
007026  105063  000002                  CLRB   SSENSE(DTAB)      ;RESET THE SENSE BYTE
007032  105763  000017                  TSTB   SRDRQ(DTAB)       ;WAS A READ REQUESTED?
007036  001002                          BNE    10$               ;YES, CONTINUE
007040  000137  010042                  JMP    ESEQ              ;NO, TREAT AS A NOP -- END SEQ ONLY
007044  105063  000017          10$:    CLRB   SRDRQ(DTAB)
007050  016377  000020  003416          MOV    SMINS(DTAB),@DXBA ;SET UP STARTING ADDRESS
007056  016302  000010                  MOV    SOUTB(DTAB),R2    ;DETERMINE ENDING ADDRESS
007062  066302  000004                  ADD    SCURS(DTAB),R2
007066  005302                          DEC    R2
007070  167702  003400                  SUB    @DXBA,R2          ;COMPUTE BYTE COUNT
007074  100002                          BPL    20$               ;INSURE VALID BYTE COUNT
007076  000137  010042                  JMP    ESEQ              ;ILLEGAL
007102  005402                  20$:    NEG    R2
007104  010277  003366                  MOV    R2,@DXBC          ;SET UP DX'S BYTE COUNT
007110  163777  013104  003356          SUB    PHYOFF,@DXBA      ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
007116  000411                          BR     SRD10             ;START THE READ



                                      :
                                      :            READ COMMAND RECEIVED FROM 360
                                      :                    PREPARE TO SEND DISP BUFFER TO 360
                                      :
007120  016377  000010  003346  SREAD:  MOV    SOUTB(DTAB),@DXBA ;SET UP BUFFER ADDRESS
007126  163777  013104  003340          SUB    PHYOFF,@DXBA      ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
```

```
3865  007134  012777  177040  003334          MOV     #-DISPSZ,@DXBC  ;SET UP BYTE COUNT
3866  007142  116363  000000  000001  SRD10:  MOVB    SCMD(DTAB),SLCMD(DTAB) ;SAVE CODE OF LAST COMMAND
3867  007150  013702  013110                  MOV     DEVCON,R2       ;COMPUTE DEVICE ADDRESS
3868  007154  060002                          ADD     DEV,R2
3869  007156  110277  003304                  MOVB    R2,@DXCA
3870  007162  105063  000002                  CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
3871  007166  110037  013070                  MOVB    DEV,DXACT       ;SET DX ACTIVE FLAG
3872  007172  005063  000014                  CLR     SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
3873  007176  052777  000005  003264          BIS     #DXRD,@DXCS     ;START TRANSFER
3874  007204  000137  010212                  JMP     DXEXIT          ;RETURN FROM INTERRUPT
3875
3876
```

```
                                        .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
                                ;
007210  105737  013074        MEX:    TSTB    CMDCHF          ;IS COMMAND CHAINING SPECIFIED?
007214  001402                        BEQ     10$             ;NO, CONTINUE
007216  000137  010212                JMP     DXEXIT          ;YES, LEAVE DX FREE
007222  013700  013076        10$:    MOV     MDEV,DEV        ;GET LAST DEVICE ADDR THAT HAD A COMMAND
007226  004737  010230        30$:    JSR     PC,SUDEV        ;COMPUTE ADDR OF DEV STAT TABLE
007232  105763  000000                TSTB    SCMD(DTAB)      ;ANY JOB TO DO?
007236  001023                        BNE     50$             ;YES, EXECUTE IT
007240  126327  000017  000001        CMPB    SRDRQ(DTAB),#1  ;IS ATTENTION REQUESTED?
007246  001004                        BNE     40$             ;NO, CONTINUE
007250  112763  000011  000000        MOVB    #11,SCMD(DTAB)  ;YES, QUEUE ATTENTION
007256  000413                        BR      50$             ;FOR THE READ MANUAL INPUT
                                ;
                                ;       NO TASK PENDING FOR CURRENT DEVICE
                                ;               BUMP TO INTERROGATE NEXT DEVICE ON CONTROL UNIT
                                ;       THIS CODE WILL REPEAT SEQUENCES WHICH MAY HAVE RUN INTO
                                ;       A LOCKOUT CONDITION IN THE DX.
                                ;
007260  005200                40$:    INC     DEV             ;INCR TO NEXT DEVICE NUMBER
007262  120037  013067                CMPB    DEV,MAXDEV      ;WAS DEVICE NUMBER WRAPPED AROUND?
007266  003402                        BLE     45$             ;NO, SEE IF ALL DEVICES HAVE BEEN INTERROGATED
007270  012700  000001                MOV     #1,DEV          ;YES, RESET THE DEVICE NUMBER
007274  020037  013076        45$:    CMP     DEV,MDEV        ;NO JOB HERE, HAVE WE CHECKED ALL DEVICES?
007300  001352                        BNE     30$             ;NO, EXAMINE NEXT DEVICE
007302  000137  010212                JMP     DXEXIT          ;YES, EXIT FROM ISR
                                ;
                                ;       THIS DEVICE HAS A JOB TO DO, EXECUTE IT
                                ;
007306  116304  000000        50$:    MOVB    SCMD(DTAB),R4   ;COMMAND TO INDEX
007312  005304                        DEC     R4              ;SCALE TO 0 - 11
007314  006304                        ASL     R4              ;MAKE INTO WORD ADDRESS
007316  010037  013076                MOV     DEV,MDEV        ;SAVE CURRENT DEVICE ADDRESS
007322  000174  007326                JMP     @MCMDTB(R4)     ;EXECUTE THE COMMAND
007326  007352                MCMDTB: .WORD   MWRITE          ;1 = WRITE FULL BUFFER
007330  007502                        .WORD   MRMI            ;2 = READ MANUAL INPUT
007332  010042                        .WORD   ESEQ            ;3 = ENDING SEQUENCE
007334  010140                        .WORD   SENSCM          ;4 = SENSE COMMAND
007336  007352                        .WORD   MWRITE          ;5 = WRITE LINE ADDRESS
007340  007600                        .WORD   MREAD           ;6 = READ FULL BUFFER
007342  007770                        .WORD   ERASCM          ;7 = ERASE COMMAND
007344  007750                        .WORD   CONUNE          ;10 = CONTROL UNIT END
007346  007760                        .WORD   SATTN           ;11 = SEND ATTENTION TO 360
007350  007502                        .WORD   MSRMI           ;12 = READ SHORT MANUAL INPUT
```

```
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;
                                        ;          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
                                        ;
007352  005763  000014                  MWRITE: TST     SRBYTC(DTAB)    ;WRITE IN PROGRESS?
007356  001011                                  BNE     10$             ;YES, SEND OUT MORE DATA
007360  016363  000006  000012                  MOV     SINBF(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
007366  163763  013104  000012                  SUB     PHYOFF,SBUFA(DTAB) ; FOR MEM MANG - OFFSET FOR PHY ADDRESS
007374  012763  000741  000014                  MOV     #DISPSZ+1,SRBYTC(DTAB) ;SET UP BUFFER FOR MAX SIZE
007402  016377  000012  003064          10$:    MOV     SBUFA(DTAB),@DXBA ;OUTPUT BUFFER ADDR TO DX
007410  013702  013110                          MOV     DEVCON,R2       ;COMPUTE DEVICE ADDRESS
007414  060002                                  ADD     DEV,R2
007416  110277  003044                          MOVB    R2,@DXCA
007422  012777  177774  003046                  MOV     #-4,@DXBC       ;START BYTE COUNT AT 4
007430  026327  000014  000004                  CMP     SRBYTC(DTAB),#4 ;IS LESS THEN 4 BYTES LEFT?
007436  002005                                  BGE     20$             ;NO, START TRANSFER
007440  016302  000014                          MOV     SRBYTC(DTAB),R2 ;YES, USE REMAINING BYTE COUNT
007444  005402                                  NEG     R2
007446  010277  003024                          MOV     R2,@DXBC
007452  105063  000002                  20$:    CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
007456  116363  000000  000001                  MOVB    SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
007464  110037  013070                          MOVB    DEV,DXACT       ;SET ACTIVE FLAG
007470  152777  000003  002772                  BISB    #DXWR,@DXCS     ;START THE TRANSFER
007476  000137  010212                          JMP     DXEXIT          ;RETURN FROM INTERRUPT



                                        ;
                                        ;          READ MANUAL INPUT COMMAND
                                        ;
                                        ;          THIS WILL TRANSFER ONLY THE DATA ENTERED ON THE SCREEN TO THE 360
                                        ;
        007502                          MRMI    =       .
007502  105737  012532                  MSRMI:  TSTB    TSTTYP          ;FRIEND OR 2848 DIAG?
007506  001034                                  BNE     MREAD           ;FRIEND -- TREAT AS READ FULL BUFFER
007510  005763  000014                          TST     SRBYTC(DTAB)    ;ANY DATA LEFT TO TRANSFER?
007514  001031                                  BNE     MREAD           ;BRANCH IF YES TO CONTINUE
007516  105063  000002                          CLRB    SSENSE(DTAB)    ;RESET THE SENSE BYTE
007522  105763  000017                          TSTB    SRDRQ(DTAB)     ;WAS THE READ REQUESTED?
007526  001002                                  BNE     20$             ;YES, CONTINUE
007530  000137  010042                  10$:    JMP     ESEQ            ;NO, RETURN AN ENDING SEQ RESP DE!CE
007534  105063  000017                  20$:    CLRB    SRDRQ(DTAB)     ;CLEAR THE READ REQUEST
007540  016363  000020  000012                  MOV     SMINS(DTAB),SBUFA(DTAB) ;SET UP THE ADDRESS OF THE DATA
007546  016302  000010                          MOV     SOUTB(DTAB),R2  ;COMPUTE THE BYTE COUNT
007552  066302  000004                          ADD     SCURS(DTAB),R2  ;END - START
007556  005302                                  DEC     R2
007560  166302  000012                          SUB     SBUFA(DTAB),R2  ;COMPUTE THE BYTE COUNT
007564  100761                                  BMI     10$             ;NEGATIVE -- SOMETHING IS WRONG
007566  010263  000014                          MOV     R2,SRBYTC(DTAB) ;SAVE FOR READ DRIVER
007572  163763  013104  000012                  SUB     PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS
                                        ;
```

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

```
                                                 .SBTTL  DX11-B ISR (MULTIPLEXER AND SELECTOR CHANNEL COMMANDS)
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                                        ;
                                        ;       COMMANDS FOR BOTH MULTIPLEXER AND SELECTOR CHANNELS
                                        ;
                                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


                                        ;
                                        ;       PRESENT CONTROL UNIT END TO CHANNEL
                                        ;
007750  152763  000040  000003  CONUNE: BISB    #CUE,SSTAT(DTAB)  ;PUT IN STATUS BYTE
007756  000434                          BR      STOUT             ;OUTPUT TO CHANNEL


                                        ;
                                        ;       SEND THE ATTENTION BIT TO THE 360
                                        ;
007760  152763  000200  000003  SATTN:  BISB    #ATTN,SSTAT(DTAB) ;PUT IN STATUS BYTE
007766  000430                          BR      STOUT             ;OUTPUT TO THE 360


                                        ;
                                        ;       ERASE THE DISPLAY
                                        ;
007770  016304  000010          ERASCM: MOV     SOUTB(DTAB),R4    ;SET UP BEG OF DISPLAY BUFFER
007774  012705  000740                  MOV     #DISPSZ,R5        ;SET UP COUNTER
010000  112702  000100                  MOVB    #EBCDSP,R2        ;SET BUFFER FILL FOR 2848 DIAG
010004  105737  012532                  TSTB    TSTTYP            ;IS TEST BEING RUN FOR 2848 RESPONDER
010010  001402                          BEQ     10$               ;YES, FILL BUFFER WITH EBCDIC SPACE
010012  113702  012534                  MOVB    FILLCH,R2         ;NO, USE CURRENT FILL CHARACTER
010016  110224                  10$:    MOVB    R2,(R4)+          ;MOVE FILL CHARACTER TO BUFFER
010020  005305                          DEC     R5                ;DECR COUNTER
010022  001375                          BNE     10$               ;NOT DONE, DO NEXT CHAR
010024  005063  000004                  CLR     SCURS(DTAB)       ;RESET THE CURSOR
010030  105063  000002                  CLRB    SSENSE(DTAB)      ;CLEAR SENSE BYTE
010034  112763  000003  000000          MOVB    #CEDE,SCMD(DTAB);CHANGE COMMAND TO PRESENT END SEQ
                                        ;
                                        ;       FALL THROUGH TO PRESENT ENDING STATUS
                                        ;


                                        ;
                                        ;       PRESENT ENDING STATUS TO CHANNEL
                                        ;
010042  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.
                                        ;
```

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

```
                                        .SBTTL  DX11-B ISR (UTILITY SUBROUTINES)
                                ;
                                ;       SET UP ADDR OF DEVICE STATUS TABLE
                                ;
                                ;       CALLING SEQUENCE
                                ;       ..... R0 = DEV #
                                ;       JSR     PC,SUDEV
                                ;       ..... RETURN
                                ;             R3 = ADDRESS OF DEVICE TABLE
                                ;
                                ;       ONLY REGISTER R3 IS MODIFIED BY THIS SUBROUTINE
                                ;
010230  013703  013064          SUDEV:  MOV     SDEVTB,DTAB     ;START AT DEV 1
010234  010146                  MOV     R1,-(SP)        ;SAVE R1
010236  010001                  MOV     DEV,R1
010240  005301                  SUD10:  DEC     R1              ;DEC DEVICE NUMBER
010242  001403                  BEQ     SUDEX           ;DONE, EXIT
010244  062703  002000          ADD     #2000,DTAB      ;INCR TO NEXT DEV TABLE
010250  000773                  BR      SUD10           ;TRY AGAIN
010252  012601                  SUDEX:  MOV     (SP)+,R1        ;RETURN TO CALLER
010254  000207                  RTS     PC


                                ;       CLEAR DEVICE STATUS TABLE
                                ;
                                ;       CALLING SEQUENCE
                                ;       ..... R0 = DEV #
                                ;       JSR     PC,CDEVST
                                ;       ..... RETURN
                                ;             R3 = ADDRESS OF DEVICE TABLE
                                ;                  THE FOLLOWING TABLE ENTRIES ARE CLEARED
                                ;                       SCMD
                                ;                       SSTAT
                                ;                       SBUFA
                                ;                       SRBYTC
                                ;                       SLCMD
                                ;
                                ;       ONLY REGISTER R3 IS AFFECTED BY THIS SUBROUTINE
                                ;
010256  004737  010230          CDEVST: JSR     PC,SUDEV                ;SET UP ADDR OF DEVICE STAT TABLE
010262  105063  000000                  CLRB    SCMD(DTAB)      ;RESET CURRENT COMMAND ENTRY
010266  105063  000003                  CLRB    SSTAT(DTAB)     ;RESET DEVICE STATUS ENTRY
010272  005063  000012                  CLR     SBUFA(DTAB)     ;RESET CURRENT BUFFER ADDRESS POINTER
010276  005063  000014                  CLR     SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
010302  105063  000001                  CLRB    SLCMD(DTAB)     ;RESET LAST COMMAND ENTRY
010306  000207                  RTS     PC                      ;RETURN TO THE CALLER


                                ;       CSPWST -- CLEAR SPW STATUS BYTE
                                ;
                                ;       CALLING SEQUENCE
                                ;       .........DTAB (R3) POINTS TO CURRENT DEVICE STATUS TABLE
                                ;       .........DEV (R0) CONTAINS CURRENT DEVICE NUMBER
                                ;       JSR     PC,CSPWST
```

```
4187                                    ;       .........RETURN TO CALLER WITH DEVICE STATUS BYTE RESET
4188                                    ;
4189                                    ;       ALL REGISTERS ARE PRESERVED ACCROSS THIS SUBROUTINE
4190                                    ;
4191  010310  010546                    CSPWST: MOV     R5,-(SP)        ;SAVE REGISTER FOR SUBROUTINE USAGE
4192  010312  105763  000016                    TSTB    SONLF(DTAB)     ;IS DEVICE ON-LINE?
4193  010316  001007                            BNE     10$             ;NO, JUST EXIT
4194  010320  010005                            MOV     DEV,R5          ;GET DEVICE NUMBER AND COMPUTE
4195  010322  063705  013110                    ADD     DEVCON,R5
4196  010326  060505                            ADD     R5,R5           ;ADDRESS OF SPW STATUS BYTE
4197  010330  063705  013114                    ADD     ST$PW,R5
4198  010334  105015                            CLRB    (R5)            ;RESET SPW STATUS BYTE
4199  010336  012605                    10$:    MOV     (SP)+,R5        ;RESTORE REGISTER
4200  010340  000207                            RTS     PC
```

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

```
                        .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
                        ;
010402  010046          DISCTL: MOV     R0,-(SP)        ;SAVE REGSITERS USED BY SUBROUTINE
010404  010146                  MOV     R1,-(SP)
010406  010246                  MOV     R2,-(SP)
010410  126327  000001  000002          CMPB    SLCMD(DTAB),#CMRMI ;WAS IT A READ MANUAL INPUT COMMAND?
010416  001535                  BEQ     DRMI            ;IF YES, PERFORM READ MANUAL INPUT PROCEEDURE
010420  126327  000001  000012          CMPB    SLCMD(DTAB),#CMSRMI ;WAS IT A SHORT READ MANAUL INPUT?
010426  001542                  BEQ     DSRMI           ;IF YES, EXIT
010430  126327  000001  000006          CMPB    SLCMD(DTAB),#CMREAD ;WAS IT A READ FULL BUFFER COMMAND?
010436  001522                  BEQ     DREAD           ;YES, RESET CURSOR ON READ FULL BUFFER
                        ;
                        ;
                        ;       THE COMMAND MUST HAVE BEEN A 360 WRITE
                        ;               DETERMINE TYPE OF TEST BEING RUN
                        ;
010440  105737  012532          TSTB    TSTTYP          ;TYPE OF TEST 0 = 2848  1 = FRIEND
010444  001102                  BNE     DISFRN          ;FRIEND
                        ;
                        ;       FORMAT DISPLAY ALA 2260
                        ;
010446  016301  000006          MOV     SINBF(DTAB),R1  ;GET ADDR OF START OF INPUT
010452  126327  000001  000005          CMPB    SLCMD(DTAB),#CMWTLA ;WAS LAST CMD A WRITE LINE ADDRESS?
010460  001016                  BNE     20$             ;NO, NORMAL WRITE
                        ;
                        ;       WRITE LINE ADDRESS COMMAND
                        ;               FIRST BYTE OF DATA BLOCK IS CURSOR LINE ADDRESS
                        ;
010462  005263  000014          INC     SRBYTC(DTAB)    ;INCR BYTE COUNT
010466  112102                  MOVB    (R1)+,R2        ;GET LINE NUMBER
010470  042702  177760          BIC     #177760,R2      ;GET ONLY LINE NUMBER
```

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

```
                                ;
                                DISFRN: TSTB    IOBUF           ;SEPARATE I/O BUFFERS?
                                        BNE     DISCEX          ;YES, DON'T COPY INPUT TO OUTPUT
                                        MOV     SINBF(DTAB),R1  ;SET UP INPUT BUFFER ADDRESS
                                        MOV     SOUTB(DTAB),R2  ;SET UP OUTPUT BUFFER ADDRESS
                                        MOV     #DISPSZ/2,R0    ;TRANSFER THE INPUT BUFFER TO THE OUTPUT BUFFER
                                ;
                                ;       PERFORM COPY
                                ;
                                10$:    MOV     (R1)+,(R2)+     ;INPUT TO OUTPUT
                                        DEC     R0              ;ARE WE DONE?
                                        BNE     10$             ;NO, CONTINUE COPY
                                        BR      DISCEX          ;PREPARE TO RETURN TO CALLER
                                ;
                                ;       A READ FULL BUFFER WAS PERFORMED
                                ;               THE CURSOR MUST BE RESET TO THE BEGINNING OF THE SCREEN
                                ;
                                DREAD:  CLR     SCURS(DTAB)     ;RESET THE CURSOR
                                        BR      DISCEX          ;AND PREPARE TO EXIT
                                ;
                                ;       A READ MANUAL INPUT WAS PERFORMED
                                ;               TO EMULATE THE 2260 SCREEN THE START OF MANUAL INPUT CHARCATER
                                ;               MUST BE DELETED FROM THE SCREEN
                                ;
                                DRMI:   MOV     SMINS(DTAB),R1  ;GET THE STARTING ADDRESS
                                        DEC     R1              ;DECREMENT TO THE SMI CHAR
                                        MOVB    #EBCDSP,(R1)    ;BLANK OUT THE CHARACTER
                                        SUB     SOUTB(DTAB),R1  ;AND COMPUTE THE CURSOR POSITION
                                        MOV     R1,SCURS(DTAB)
                                ;
                                ;       A SHORT READ MANUAL INPUT WAS PERFORMED
                                ;               NO ACTION REQUIRED BY DISPLAY CONTROL ROUTINE
                                ;
                                DSRMI:
                                ;
                                ;       RESTORE REGISTERS AND RETURN TO CALLER
                                ;
                                DISCEX: MOV     (SP)+,R2        ;RESTORE SAVED REGISTERS
                                        MOV     (SP)+,R1
                                        MOV     (SP)+,R0
                                        RTS     PC              ;RETURN TO THE CALLER
                                        .SBTTL  TELETYPE (CONSOLE) INPUT ISR
                                ;
                                ;       TELETYPE INPUT HANDLER (ISR)
                                ;
                                ;               CONTROL PASSES HERE ON A TELETYPE INPUT INTERRUPT
                                ;
                                ;       DATA IS INPUT FROM THE CONTROL CONSOLE AND STORED INTO
```

```
                        ;       THE TELETYPE INPUT BUFFER (TBUF). WHEN ALL THE DATA IS
                        ;       ENTERED, THE OPERATOR HITS A C/R TO END THE LINE, THEN
                        ;       AN ACTIVE FLAG IS SET AND THE COMMAND EXECUTED BY THE
                        ;       SYSTEM.
                        ;
                        ;               THE FOLLOWING CONTROL FUNCTIONS ARE AVAILABLE FOR
                        ;               OPERATOR CONVENIENCE.
                        ;
                        ;                       C/R  =  LINE DELIMETER
                        ;                       ←    =  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
                        ;
010744  010046          TKIN:   MOV     R0,-(SP)        ;SAVE REGISTERS
010746  010146                  MOV     R1,-(SP)
010750  017700  001502          MOV     @TKB,R0         ;GET TELE CHARACTER
010754  042700  177600          BIC     #177600,R0      ;INSURE 7-BIT ASCII
010760  013701  012742          MOV     TPTR,R1         ;BUFFER PTR
010764  020027  000020          CMP     R0,#CTL.P       ;CONTROL -P ?
010770  001002                  BNE     3$              ;NO
010772  000137  001002          JMP     RSTART          ;YES, ALLOW OPERATOR TO REENTER ALL PARAMETERS
010776  020027  000023  3$:     CMP     R0,#CTL.S       ;CONTROL-S , TEMPORALILY STOP CONSOLE OUTPUT?
011002  001003                  BNE     6$              ;NO, CONTINUE
011004  105237  013073          INCB    TTYSTP          ;YES, SET FLAG TO STOP TTY OUTPUT
011010  000503                  BR      100$            ;       AND EXIT FROM INTERRUPT
011012  020027  000021  6$:     CMP     R0,#CTL.Q       ;CONTROL-Q, RESUME CONSOLE OUTPUT?
011016  001010                  BNE     10$             ;NO, CONTINUE
011020  105037  013073          CLRB    TTYSTP          ;YES, RESET CONSOLE STOP FLAG
011024  105737  013071          TSTB    PCTR            ;       CHECK TO INSURE OUTPUT TO RESUME
011030  001473                  BEQ     100$            ;       NO OUTPUT -- EXIT
011032  004737  011252          JSR     PC,PROUT        ;               RESTART CONSOLE OUTPUT
011036  000470                  BR      100$            ;       AND EXIT FROM THE INTERRUPT
011040  020027  000003  10$:     CMP     R0,#CTL.C       ;COMMAND ABORT -- CTL C?
011044  001011                  BNE     20$             ;NO
011046  105737  013053          TSTB    TCMACT          ;IS A COMMAND ACTIVE?
011052  001457                  BEQ     90$             ;NO, TREAT AS A DELETE LAST LINE
011054  105237  013054          INCB    TCMDAB          ;YES, SET ABORT FLAG
011060  012701  012640          MOV     #TBUF,R1        ;SET UP BUFFER POINTER
011064  000137  011220          JMP     100$            ;EXIT
011070  105737  013053  20$:     TSTB    TCMACT          ;TELE CMD CURRENTLY ACTIVE?
011074  001051                  BNE     100$            ;YES, IGNORE CHARACTER
011076  110021                  MOVB    R0,(R1)+        ;STORE CHAR INTO BUFFER - INC PTR
011100  020027  000015          CMP     R0,#CR          ;LINE DELLIMETER -- C/R?
011104  001005                  BNE     30$             ;NO
011106  012701  012640          MOV     #TBUF,R1        ;RESET BUFFER PTR
011112  105237  013053          INCB    TCMACT          ;YES, SET COMMAND ACTIVE FLAG
011116  000440                  BR      100$            ;DONT PRINT THE LINE DELIMITER
011120  020027  000177  30$:     CMP     R0,#RUBOUT      ;A RUBOUT?
```

```
 4455  011124  001002                        BNE     40$             ;NOPE
 4456  011126  012700  000137                MOV     #'←,R0          ;YES, TREAT AS A DELETE LAST CHARACTER
 4457  011132  120027  000025        40$:    CMPB    R0,#CTL.U       ;CONTROL-U? (DELETE CURRENT INPUT LINE)
 4458  011136  001002                        BNE     50$             ;NOPE, CONTINUE
 4459  011140  112700  000134                MOVB    #'\,R0          ;YES, TREAT AS DELETE LAST LINE (BACKSLASH)
 4460  011144  004737  011366        50$:    JSR     PC,PCHAR        ;ECHO THE CHARCTER BACK
 4461  011150  020027  000137                CMP     R0,#'←          ;DELETE LAST CHAR -- BACK ARROW?
 4462  011154  001004                        BNE     60$             ;NO
 4463  011156  124141                        CMPB    -(R1),-(R1)     ;YES, DECR POINTER BY 2
 4464  011160  020127  012640                CMP     R1,#TBUF        ;ARE WE  BEYOND BEG OF THE BUFFER?
 4465  011164  003403                        BLE     70$             ;YES, RESET TO BEG OF BUFFER
 4466  011166  020027  000134        60$:    CMP     R0,#'\          ;DELETE CUR LINE -- BACK SLASH?
 4467  011172  001004                        BNE     80$             ;NO
 4468  011174  012701  012640        70$:    MOV     #TBUF,R1        ;YES, RESET BUFFER PTR
 4469  011200  004737  011330                JSR     PC,CRLF         ;NEW LINE FOR NEW COMMAND
 4470  011204  020127  012740        80$:    CMP     R1,#TBUFE       ;WERE LIMITS EXCEEDED?
 4471  011210  001003                        BNE     100$            ;NOPE, EXIT
 4472  011212  012700  000134        90$:    MOV     #'\,R0          ;THEY WERE -- TREAT AS A LINE ABORT
 4473  011216  000740                        BR      30$
 4474  011220  010137  012742        100$:   MOV     R1,TPTR         ;SAVE BUFFER PTR
 4475  011224  012601                        MOV     (SP)+,R1        ;RESTORE REGISTERS + EXIT
 4476  011226  012600                        MOV     (SP)+,R0
 4477  011230  000002                        RTI
```

```
                                        .SBTTL  TELETYPE (CONSOLE) OUTPUT ISR
                                ;
                                ;       TELETYPE OUTPUT DRIVER (ISR) -- PRINT
                                ;
                                ;               CONTROL PASSES HERE ON A TELE OUT INTERRUPT
                                ;
011232  105037  013052        PISR:   CLRB    PIUBL           ;CLEAR PRINTER BUSY FLAG
011236  105737  013071                TSTB    PCTR            ;ANY MORE DATA TO PRINT?
011242  001402                        BEQ     10$             ;NO, EXIT
011244  004737  011252                JSR     PC,PROUT        ;OUTPUT ANOTHER CHAR
011250  000002                10$:    RTI
```

```
                                        .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
                                ;
011252  105737  013052        PROUT:  TSTB    PIUFL           ;IS IT BUSY?
011256  001023                        BNE     20$             ;YES, EXIT
011260  105737  013073                TSTB    TTYSTP          ;HAS CONSOLE OUTPUT BEEN SUSPENDED?
011264  001020                        BNE     20$             ;YES, RETURN IMMEDIATELY TO CALLER
011266  105237  013052                INCB    PIUFL           ;NO, SET BUSY FLAG
011272  105337  013071                DECB    PCTR            ;DECR CHAR COUNTER
011276  117777  001544  001156        MOVB    @PFPTR,@TPB     ;OUTPUT NEXT CHAR
011304  005237  013046                INC     PFPTR           ;INCR PRINT FETCH POINTER
011310  023727  013046  013046        CMP     PFPTR,#PBFE     ;TIME TO WRAP AROUND?
011316  001003                        BNE     20$             ;NO, EXIT
011320  012737  012744  013046        MOV     #PBFS,PFPTR     ;YES, RESTORE TO START OF BUFFER
011326  000207                20$:    RTS     PC                      ;RETURN TO CALLER
                                ;
                                ;       PRINT A CR/LF
                                ;
                                ;       CALLING SEQUENCE
                                ;       JSR     PC,CRLF
                                ;       .........RETURN
                                ;
                                ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
                                ;
011330  010246                CRLF:   MOV     R2,-(SP)        ;SAVE THE R2 REGISTER
011332  012702  105215                MOV     #105215,R2      ;DO A CRLF
011336  004737  011346                JSR     PC,PRINT2       ;PRINT IT
011342  012602                        MOV     (SP)+,R2        ;RESTORE THE R2 REGISTER
011344  000207                        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
                                ;
```

```
 4545  011346  010237  011362          PRINT2: MOV     R2,P2BF
 4546  011352  004137  011472                  JSR     R1,MESG
 4547  011356  011362                          .WORD   P2BF
 4548  011360  000207                          RTS     PC
 4549  011362  000000                  P2BF:   .WORD   0
 4550  011364     377     377                  .BYTE   377,377
 4551
 4552
 4553                                  ;
 4554                                  ;       PRINT 1 CHARACTER
 4555                                  ;
 4556                                  ;       CALLING SEQUENCE
 4557                                  ;       .........R0 CONTAINS THE CHARACTER TO BE PRINTED
 4558                                  ;       JSR     PC,PCHAR
 4559                                  ;       .........RETURN WITH THE DATA IN THE PRINT BUFFER
 4560                                  ;
 4561                                  ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
 4562                                  ;
 4563  011366  110037  011402          PCHAR:  MOVB    R0,P1BF
 4564  011372  004137  011472                  JSR     R1,MESG
 4565  011376  011402                          .WORD   P1BF
 4566  011400  000207                          RTS     PC              ;RETURN TO THE CALLER
 4567  011402     000     377          P1BF:   .BYTE   0,377
```

```
4568                                    ;
4569                                    ;       PRMESG PRINT A CHARACTER STRING
4570                                    ;
4571                                    ;       CALLING SEQ
4572                                    ;       .....R2 CONTAINS THE STARTING ADDRESS OF THE MESSAGE
4573                                    ;       JSR     PC,PRMESG
4574                                    ;       ......RETURN
4575                                    ;
4576                                    ;       NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4577                                    ;
4578                                    ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4579  011404  010246                    PRMESG: MOV     R2,-(SP)        ;SAVE REGS
4580  011406  010346                            MOV     R3,-(SP)
4581  011410  013703  013050                    MOV     PPPTR,R3        ;GET PRINT OUTPUT POINTER
4582  011414  121227  000377            10$:    CMPB    (R2),#377       ;END OF MESSAGE?
4583  011420  001417                            BEQ     40$             ;YES, EXIT
4584  011422  112223                            MOVB    (R2)+,(R3)+     ;NO MOVE NEXT CHAR TO PRINT BUFFER
4585  011424  105237  013071                    INCB    PCTR            ;INCR CHAR COUNTER
4586  011430  020327  013046                    CMP     R3,#PBFE        ;AT END OF BUFFER?
4587  011434  001002                            BNE     20$             ;NO
4588  011436  012703  012744                    MOV     #PBFS,R3        ;YES, WRAP AROUND TO BEG OF BUFFER
4589  011442  004737  011252            20$:    JSR     PC,PROUT        ;CAN WE START PRINT?
4590  011446  123737  013071  013130    30$:    CMPB    PCTR,PMAX       ;IS PRINT BUFFER FULL?
4591  011454  001774                            BEQ     30$             ;YES, WAIT TILL ROOM AVAILABLE
4592  011456  000756                            BR      10$             ;GET NEXT CHAR
4593  011460  010337  013050            40$:    MOV     R3,PPPTR        ;EXIT, RESTORE PUT PTR
4594  011464  012603                            MOV     (SP)+,R3        ;RESTORE REGS
4595  011466  012602                            MOV     (SP)+,R2
4596  011470  000207                            RTS     PC              ;RETURN TO THE CALLER
4597
4598
4599                                    ;
4600                                    ;       MESG -- PRINT A CHARACTER STRING ON THE SYSTEM CONSOLE
4601                                    ;
4602                                    ;       CALLING SEQUENCE
4603                                    ;       JSR     R1,MESG
4604                                    ;       .WORD   ADDRESS OF START OF MESSAGE
4605                                    ;       ........RETURN
4606                                    ;
4607                                    ;       NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4608                                    ;
4609                                    ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4610
4611  011472  010246                    MESG:   MOV     R2,-(SP)        ;SAVE REGISTER
4612  011474  012102                            MOV     (R1)+,R2        ;GET ADDRESS OF MESSAGE AND BUMP FOR RETURN
4613  011476  004737  011404                    JSR     PC,PRMESG       ;MORE MESSAGE PROCESSING
4614  011502  012602                            MOV     (SP)+,R2        ;RESTORE SOILED REGISTER
4615  011504  000201                            RTS     R1              ;RETURN TO THE CALLER
4616
4617
4618                                    ;
4619                                    ;       INMES PRINT A CHARACTER STRING
4620                                    ;
4621                                    ;       CALLING SEQUENCE
4622                                    ;       JSR     R1,INMES
4623                                    ;       .WORD ADDRESS OF MESSAGE
```

```
                                        ;       .....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
                                        ;
4634  011506  010246                   INMES:  MOV     R2,-(SP)
4635  011510  113746  013130                   MOVB    PMAX,-(SP)      ;CHEAT, SAVE PMAX
4636  011514  112737  000377  013130           MOVB    #377,PMAX       ;AND MAKE VERY LARGE
4637  011522  012102                           MOV     (R1)+,R2
4638  011524  004737  011404                   JSR     PC,PRMESG       ;USE STANDARD MESSAGE PROCESSOR
4639  011530  112637  013130                   MOVB    (SP)+,PMAX      ;RESTORE PRINT MAX
4640  011534  012602                           MOV     (SP)+,R2
4641  011536  000201                           RTS     R1              ;RETURN TO CALLER
```

```
                        .SBTTL  UTILITY SUBROUTINES (CONVERT OCTAL OR HEX TO BINARY)
                ;
                ;       COTB -- CONVERT ASCII OCTAL TO BINARY (COTB)
                ;
                ;       CALLING SEQUENCE
                ;       ......R2 = CHAR ADDRESS OF FIRST CHARCTER 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
                ;
011540  005003  COTB:   CLR     R3
011542  005005         CLR     R5
011544  112204  10$:    MOVB    (R2)+,R4        ;GET NEXT CHAR
011546  120427  000067  CMPB    R4,#'7          ;CHAR GT 7?
011552  003013  BGT     20$                     ;YES, EXIT
011554  120427  000060  CMPB    R4,#'0          ;CHAR LT 0?
011560  002410  BLT     20$                     ;YES, EXIT
011562  042704  177770  BIC     #177770,R4      ;SAVE ONLY L.S. 3 BITS
011566  006303         ASL     R3              ;SHIFT OLD RESULT BY 8
011570  006303         ASL     R3
011572  006303         ASL     R3
011574  060403         ADD     R4,R3           ;ADD IN NEW NUMBER
011576  005205         INC     R5              ;INCR CHAR COUNT
011600  000761         BR      10$             ;GET NEXT CHAR
011602  000207  20$:    RTS     PC              ;RETURN TO CALLER
```

```
                                        ;
                                        ;       CHTB -- CONVERT ASCII HEX TO BINARY
                                        ;
                                        ;       CALLING SEQUENCE
                                        ;       ......R2 = ADDRESS OF FIRST CHARCTER TO BE CONVERTED
                                        ;       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
                                        ;
011604  005003                  CHTB:  CLR     R3
011606  005005                         CLR     R5
011610  112204                  10$:   MOVB    (R2)+,R4        ;GET THE FIRST CHARACTER
011612  162704  000060                 SUB     #'0,R4          ;SCALE RELATIVE TO ASCII ZERO
011616  100422                         BMI     30$             ;NOT A VALID HEX CHAR
011620  020427  000012                 CMP     R4,#10.         ;IS RESULT STILL GT 10?
011624  002410                         BLT     20$             ;YES, WE HAVE A VALID HEX DIGIT
011626  162704  000007                 SUB     #7,R4
011632  020427  000020                 CMP     R4,#16.         ;IS IT A LETTER?
011636  002012                         BGE     30$             ;NO, INVALID CHAR
011640  020427  000012                 CMP     R4,#10.         ;AND GT 10
011644  002407                         BLT     30$             ;NO,ILLEGAL CHAR
011646  006303                  20$:   ASL     R3              ;MAKE ROOM FOR NEW ENTRY
011650  006303                         ASL     R3
011652  006303                         ASL     R3
011654  006303                         ASL     R3
011656  060403                         ADD     R4,R3           ;INSERT NEW ENTRY
011660  005205                         INC     R5              ;INCR CHAR COUNT
011662  000752                         BR      10$             ;AND CONVERT NEXT CHAR
011664  005302                  30$:   DEC     R2              ;GET THE ILLEGAL CHARCTER
011666  112204                         MOVB    (R2)+,R4        ;AND PUT IT R4
011670  000207                         RTS     PC              ;ITS TIME TO RETURN TO THE CALLER
```

```
                                        .SBTTL  PROCESSOR ERROR TRAP HANDLERS

                                ;
                                ;       TRAP OUT ROUTINES
                                ;
                                ;
                                ;
                                ;       MEMORY TIME OUT ROUTINE
                                ;
011672  012702  014143          MTO:    MOV     #PMTO,R2        ;SET UP ADDRESS OF THE PRINT ROUTINE
011676  000404                          BR      TOUTRT          ;TO GENERALIZED TRAP OUT ROUTINE


                                ;
                                ;       MEMORY MANAGEMENT TRAP OUT ROUTINE
                                ;
011700  005037  177572          MMERR:  CLR     MMSR0           ;CLEAR THE MEMORY MANAGEMENT BIT
011704  012701  014167                  MOV     #PMMERR,R1      ;SET UP ADDRES OF ERROR MESSAGE
011710  000005                  TOUTRT: RESET                   ;CLEAR ALL DEVICES
011712  105037  013052                  CLRB    PIUFL           ;CLEAR PRINT IN USE FLAG
011716  005037  177776                  CLR     PSW             ;LOWER PROCESSOR STATUS TO ALLOW INTERRUPTS TO CUM
011722  004737  011404                  JSR     PC,PRMESG       ;PRINT THE ERROR MESSAGE
011726  105737  013071          10$:    TSTB    PCTR            ;IS PRINTING DONE?
011732  001375                          BNE     10$
011734  000000                          HALT                    ;YES, HALT
011736  000137  001000                  JMP     START


                                ;
                                ;       INVALID UNIBUS ADDRESS TRAP
                                ;
011742  022626                  UNTRP:  CMP     (SP)+,(SP)+     ;POP THE PUSH STACK
011744  005037  177776                  CLR     PSW             ;CLEAR THE PROCESSOR STATUS WORD
011750  000137  001174                  JMP     NEWPRM          ;ASK OPERATOR TO REENTER THE DATA
```

```
                        .SBTTL  CODE CONVERSION TABLES

                        ;
                        ;       EBCDIC TO ASCII CODE CONVERSION TABLE
                        ;               THIS TABLE FOLDS ALL INPUT INTO A 64 CHARACTER SET
                        ;
                        ;       NOTE -- BACKARROW IS USED TO DENOTE A NEWLINE
                        ;
011754                  EBCDTB:
011754  040440  041502  042504    .ASCII  / ABCDEFGHI↑.<(+!/      ;00 - 0F
011762  043506  044510  027136
011770  024074  020453
011774  045046  046113  057515    .ASCII  "&JKLM←OPQR←$*);]"      ;10 - 1F
012002  050117  051121  022137
012010  024452  056473
012014  027455  052123  053125    .ASCII  '-/STUVWXYZ",%[>?'      ;20 - 2F
012022  054127  055131  026042
012030  055445  037476
012034  030460  031462  032464    .ASCII  "0123456789:#@'=\"      ;30 - 3F
012042  033466  034470  021472
012050  023500  056075
012054  040440  041502  042504    .ASCII  / ABCDEFGHI↑.<(+!/      ;40 - 4F
012062  043506  044510  027136
012070  024074  020453
012074  045046  046113  047115    .ASCII  "&JKLMNOPQR←$*);]"      ;50 - 5F
012102  050117  051121  022137
012110  024452  056473
012114  027455  052123  053125    .ASCII  '-/STUVWXYZ",%[>?'      ;60 - 6F
012122  054127  055131  026042
012130  055445  037476
012134  030460  031462  032464    .ASCII  "0123456789:#@'=\"      ;70 - 7F
012142  033466  034470  021472
012150  023500  056075
012154  040440  041502  042504    .ASCII  / ABCDEFGHI↑.<(+!/      ;80 - 8F
012162  043506  044510  027136
012170  024074  020453
012174  045046  046113  047115    .ASCII  "&JKLMNOPQR←$*);]"      ;90 - 9F
012202  050117  051121  022137
012210  024452  056473
012214  027455  052123  053125    .ASCII  '-/STUVWXYZ",%[>?'      ;A0 - AF
012222  054127  055131  026042
012230  055445  037476
012234  030460  031462  032464    .ASCII  "0123456789:#@'=\"      ;B0 - BF
012242  033466  034470  021472
012250  023500  056075
012254  040440  041502  042504    .ASCII  / ABCDEFGHI↑.<(+!/      ;C0 - CF
012262  043506  044510  027136
012270  024074  020453
012274  045046  046113  047115    .ASCII  "&JKLMNOPQR←$*);]"      ;D0 - DF
012302  050117  051121  022137
012310  024452  056473
012314  027455  052123  053125    .ASCII  '-/STUVWXYZ",%[>?'      ;E0 - EF
012322  054127  055131  026042
012330  055445  037476
012334  030460  031462  032464    .ASCII  "0123456789:#@'=\"      ;F0 - FF
012342  033466  034470  021472
012350  023500  056075
```

```
                                        ;
                                        ;       ASCII TO EBCDIC CONVERSION TABLE
                                        ;
012354  100  117  152  ATOETB: .BYTE   100,117,152,173,133,154,120,175 ;240-247
012357  173  133  154
012362  120  175
012364  115  135  134          .BYTE   115,135,134,116,153,140,113,141 ;250-257
012367  116  153  140
012372  113  141
012374  360  361  362          .BYTE   360,361,362,363,364,365,366,367 ;260-267
012377  363  364  365
012402  366  367
012404  370  371  172          .BYTE   370,371,172,136,114,176,156,157 ;270-277
012407  136  114  176
012412  156  157
012414  174  301  302          .BYTE   174,301,302,303,304,305,306,307 ;300-307
012417  303  304  305
012422  306  307
012424  310  311  321          .BYTE   310,311,321,322,323,324,325,326 ;310-317
012427  322  323  324
012432  325  326
012434  327  330  331          .BYTE   327,330,331,342,343,344,345,346 ;320,327
012437  342  343  344
012442  345  346
012444  347  350  351          .BYTE   347,350,351,155,177,137,112,025 ;330-337
012447  155  177  137
012452  112  025
                                        .SBTTL  PROGRAM CONSTANTS AND VARIABLES
                                        ;
                                        ;       CONSOLE UNIBUS ADDRESS CONSTANTS
                                        ;
012454  177560  TKS:    .WORD   177560          ;KEYBOARD CONTROL STATUS REGISTER
012456  177562  TKB:    .WORD   177562          ;KEYBOARD DATA BUFFER
012460  177564  TPS:    .WORD   177564          ;PRINTER STATUS/CONTROL REGISTER
012462  177566  TPB:    .WORD   177566          ;PRINTER DATA BUFFER
                                        ;
                                        ;       DX REGISTERS - ADDRESS GENERATED BY INITIALIZATION
                                        ;
012464  000000  DXDS:   .WORD   0               ;DEVICE STATUS -- TT1
012466  000000  DXCA:   .WORD   0               ;COMMAND AND ADDRESS -- TT2
012470  000000  DXCS:   .WORD   0               ;CONTROL UNIT STATUS
012472  000000  DXOS:   .WORD   0               ;OFFSET AND STATUS
012474  000000  DXBA:   .WORD   0               ;BUS ADDRESS
012476  000000  DXBC:   .WORD   0               ;BYTE COUNT
012500  000000  DXMO:   .WORD   0               ;MAINTANCE OUT
012502  000000  DXMI:   .WORD   0               ;MAINTANCE IN
012504  000000  DXCB:   .WORD   0               ;CONTROL BITS
012506  000000  DXND:   .WORD   0               ;NPR DATA
012510  000000  DXES1:  .WORD   0               ;EXTRA SIGNALS
012512  000000  DXMOB:  .WORD   0               ;BUFFERED BUS OUT
012514  000000  DXES2:  .WORD   0               ;EXTRA SIGNALS
```

```
                                ;
                                ;       CONFIGURATION CONSTANTS
                                ;
012516  000000                  UNADDR: .WORD   0               ;UNIBUS ADDRESS
012520  000000                  VECTAD: .WORD   0               ;DX VECTOR ADDRESS
012522  000000                  SDEV:   .WORD   0               ;STARTING DEV NUMBER
012524  000000                  EDEV:   .WORD   0               ;ENDING DEV NUMBER
012526     000                  CHTYPE: .BYTE   0               ;CHANNEL TYPE 0 = MPX - 1 = SEL
012527     000                  MMRESP: .BYTE   0               ;MEMORY MANAGEMENT 0 = NO - 1 = YES
012530  000000                  BUFREL: .WORD   0               ;BUFFER RELOCATION ADDRESS
012532     000                  TSTTYP: .BYTE   0               ;TEST TYPE 0 = 2848 - 1 = FRIEND
012533     000                  IOBUF:  .BYTE   0               ;SEPERATE I/O BUFFER 0 = NO - 1 = YES
012534     000                  FILLCH: .BYTE   0               ;FILL CHARACTER
012535     000                  CONEND: .BYTE   0               ;EXTRA


                                ;
                                ;       SYSTEM PUSH STACK
                                ;
        012636                          .=.+100
        012636                  SSTACK  =       .


                                ;
                                ;       SYSTEM VARIABLES
                                ;
                                ;       THE FOLLOWING VARIABLES ARE RESET UPON START-UP
                                ;
012636  000000                  VSTRT:  .WORD   0               ;DUMMY
        012640                  TBUF    =       .               ;START OF TELETYPE INPUT BUFFER
        012740                          .=.+100
012740  000000                  TBUFE:  .WORD   0               ;END OF TELETYPE INPUT BUFFER
012742  000000                  TPTR:   .WORD   0               ;TELE IN PTR
012744  000000                  PBFS:   .WORD   0               ;START OF PRINT BUFFER
        013046                          .=.+100
        013046                  PBFE    =       .               ;END OF PRINT BUFFER
013046  000000                  PFPTR:  .WORD   0               ;PRINT FETCH PTR
013050  000000                  PPPTR:  .WORD   0               ;PRINT PUT PTR
013052     000                  PIUFL:  .BYTE   0               ;PRINTER IN USE FLAG
013053     000                  TCMACT: .BYTE   0               ;TELE COMMAND ACTIVE FLAG 0 = NON-ACT
013054     000                  TCMDAB: .BYTE   0               ;TEL COMMAND ABORT 1 = ABORT
013055     000                  LINECT: .BYTE   0               ;LINE CTR - CHARS / LINE
013056     000                  WK:     .BYTE   0               ;WORK LOC
013057     000                  WK1:    .BYTE   0               ;WORK LOC
013060  000000                  TTPTR:  .WORD   0               ;TUMBLE TABLE PTR
013062  000000                  TTADDR: .WORD   0               ;BEG OF TUMBLE TABLE
013064  000000                  SDEVTB: .WORD   0               ;START OF DEVICE TABLES
013066     000                  DXSTPF: .BYTE   0               ;DX STOP FLAG
013067     000                  MAXDEV: .BYTE   0               ;HIGHEST DEV # 1 - 8
013070     000                  DXACT:  .BYTE   0               ;DXACTIVE FLAG
013071     000                  PCTR:   .BYTE   0               ;PRINT BUFFER COUNTER
013072     000                  DXABFL: .BYTE   0               ;DX ABORT FLAG  0 = NO ABORT, 1 = ABORT
013073     000                  TTYSTP: .BYTE   0               ;CONSOLE OUTPUT STOP FLAG 0 = OUTPUT; 1 = NO OUTPUT
                                        .EVEN
013074  000000                  CMDCHF: .WORD   0               ;COMMAND CHAIN FLAG
013076  000000                  MDEV:   .WORD   0               ;DEV # IN MPXR EXEC
```

```
 4909            013076                  SELDEV  =       MDEV            ;DEV # IN SEL EXEC
 4910  013100    000000                  PBUFA:  .WORD   0               ;PHYSICAL BUFF ADDR - IN ,000'S
 4911  013102    000000                  VBUFA:  .WORD   0               ;VIRTUAL BUFF ADDR - IN ,000'S
 4912  013104    000000                  PHYOFF: .WORD   0               ;PHY OFFSET FOR MEMORY MANAGEMENT
 4913  013106    000000                  CDEV:   .WORD   0               ;CURRENT DX DEVICE -- INTER SERVICE ROUTINE
 4914  013110    000000                  DEVCON: .WORD   0               ;DEVICE ADDED TO THE DEVICE NUMBER = STARTING DEV NUMB -
 4915  013112    000000                  XADDR:  .WORD   0               ;EXTENDED ADDRESS BITS FOR THE DX CONTROL REGISTER -- IN
 4916  013114    000000                  STSPW:  .WORD   0               ;START OF THE PSW TABLE
 4917  013116    000000                  DSTOFF: .WORD   0               ;OFFSET TO THE DST TABLE
 4918  013120    000000                  SADDR:  .WORD   0               ;TELETYPE COMMAND STARTING BUFFER ADDRESS
 4919  013122    000000                  EADDR:  .WORD   0               ;TELETYPE COMMAND ENDING BUFFER ADDRESS
 4920  013124    000000                  DMPADR: .WORD   0               ;POINTER TO DUMP ROUTINE CURRENTLY BEING UTILIZED BY TEL
 4921  013126    000000                  VEND:   .WORD   0
 4922
 4923
 4924                                    ;
 4925                                    ;       THE FOLLOWING VARIABLES ARE NOT RESET ON START-UP
 4926                                    ;
 4927  013130    000102                  PMAX:   .WORD   PBFE-PBFS       ;SIZE OF PRINT BUFFER
 4928  013132    000000                  FTIMFL: .WORD   0               ;FIRST TIME FLAG
```

```
                                        .SBTTL  MESSAGES
                                        .NLIST  BEX
                                ;
                                ;       SYSTEM MESSAGES
                                ;
013134      215     212         STMSG:  .BYTE   215,212
013136  055104  054104  026511          .ASCII /DZDXI-B  NEW DX11-B RESPONDER/
013173      377     377                 .BYTE   377,377
013175      215     212         UNMSG:  .BYTE   215,212
013177      125  044516  052502          .ASCII /UNIBUS ADDRESS -OCTAL- : /
013230      377                         .BYTE   377
            013232                      .EVEN
013232      215     212         VECTMS: .BYTE   215,212
013234  047111  042524  051122          .ASCII /INTERRUPT VECTOR ADDRESS -OCTAL- : /
013277      377                         .BYTE   377
                                        .EVEN
013300      215     212         DEVMES: .BYTE   215,212
013302  042504  044526  042503          .ASCII /DEVICE ADDRESSES -HEX- (XX,XX): /
013342      377                         .BYTE   377
            013344                      .EVEN
013344      215     212         CHTYMS: .BYTE   215,212
013346  044103  047101  042516          .ASCII /CHANNEL TYPE (M OR S): /
013375      377                         .BYTE   377
                                        .EVEN
013376      215     212         MMMES:  .BYTE   215,212
013400  042515  047515  054522          .ASCII /MEMORY MANAGEMENT (Y OR N): /
013434      377                         .BYTE   377
            013436                      .EVEN
013436      215     212         BFREMS: .BYTE   215,212
013440  052502  043106  051105          .ASCII /BUFFER RELOCATION, IF SPECIFIED - IN EVEN ,000'S -OCTAL- : /
013533      377                         .BYTE   377
                                        .EVEN
013534      215     212         TESTMS: .BYTE   215,212
013536  051106  042511  042116          .ASCII /FRIEND (F) OR 2848 DIAG(D): /
013572      377                         .BYTE   377
            013574                      .EVEN
013574      215     212         FIOMS:  .BYTE   215,212
013576  042523  040520  040522          .ASCII /SEPARATE I-O BUFFERS (Y OR N): /
013635      377                         .BYTE   377
                                        .EVEN
013636      215     212         FILLMS: .BYTE   215,212
013640  052517  050124  052125          .ASCII /OUTPUT BUFFER FILL CHARACTER -HEX- : /
013705      377                         .BYTE   377
                                        .EVEN
013706      207     207     215 NXMMSG: .BYTE   207,207,215,212
013712  047516  020116  054105          .ASCII /NON EX-MEM ERROR/
013732      215     212     377         .BYTE   215,212,377
            013736                      .EVEN
013736      207     207     215 PARMES: .BYTE   207,207,215,212
013742  040520  044522  054524          .ASCII /PARITY ERROR/
013756      212     215     377         .BYTE   212,215,377
            013762                      .EVEN
013762      207     207     215 ILLMES: .BYTE   207,207,215,212
013766  046111  042514  040507          .ASCII  /ILLEGAL DEVICE NUMBER/
014013      212     215     377         .BYTE   212,215,377
                                        .EVEN
```

```
014016 052503 051122 047105 STPMES: .ASCII  /CURRENT DEVICE NUMBER -- /
014047    377                       .BYTE   377
                                    .EVEN
014050    207    207    215 INVLDC: .BYTE   207,207,215,212
014054 047111 040526 044514         .ASCII /INVALID DX COMMAND/
014076    212    215    377         .BYTE   212,215,377
          014102                    .EVEN
014102 020040 047516 046440 PNOMM:  .ASCII  /  NO MEMORY MANAGEMENT AVAILABLE/
014142    377                       .BYTE   377
014143    215    212    207 PMTO:    .BYTE   215,212,207,207
014147    115 046505 051117         .ASCII  /MEMORY TIME OUT/
014166    377                       .BYTE   377
014167    215    212    207 PMMERR: .BYTE   215,212,207,207
014173    115 046505 051117         .ASCII  /MEMORY MANAGEMENT ERROR/
014222    377                       .BYTE   377
          014224                    .EVEN
014224    215    212        RNMESG: .BYTE   215,212
014226 054523 052123 046505         .ASCII  /SYSTEM INITIALIZED, TYPE "R" TO ENABLE DX/
014277    377                       .BYTE   377
014300    015    012        HELPMS: .BYTE   CR,LF
014302 054104 030461 041055         .ASCII  /DX11-B 2848 EMULATOR TEST PACKAGE - OPERATIONAL INFORMATION/
014375    015    012                .BYTE   CR,LF
014377    015    012                .BYTE   CR,LF
014401    104 026440 020055         .ASCII  /D -- DUMP COMMAND/<CR><LF>
014424 020040 020040 020040         .ASCII  /       DTT,C      DUMP TUMBLE TABLE IN CODE "C"/<CR><LF>
014504 020040 020040 020040         .ASCII  /       DIN,C,XX   DUMP INPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014602 020040 020040 020040         .ASCII  /       DOT,C,XX   DUMP OUTPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014701    105 026440 020055         .ASCII  /E -- ENABLE DEVICE ON DX/<CR><LF>
014733    040 020040 020040         .ASCII  /       EXX        ENABLE DEVICE XX/<CR><LF>
014776 020106 026455 043040         .ASCII  /F -- FILL BUFFER COMMAND/<CR><LF>
015030 020040 020040 020040         .ASCII  /       FIN,HH,XX FILL INPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015116 020040 020040 020040         .ASCII  /       FOT,HH,XX FILL OUTPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015205    110 026440 020055         .ASCII  /H -- HELP COMMAND/<CR><LF>
015230 020040 020040 020040         .ASCII  /       THIS TEXT/<CR><LF>
015252 020113 026455 045440         .ASCII  /K -- KILL A DEVICE ON THE DX/<CR><LF>
015310 020040 020040 020040         .ASCII  /       KXX        KILL DEVICE XX/<CR><LF>
015351    122 026440 020055         .ASCII  /R -- ENABLE DX (RUN)/<CR><LF>
015377    040 020040 020040         .ASCII  /       R          RUN TEST/<CR><LF>
015432 020123 026455 042040         .ASCII  /S -- DISABLE DX (STOP)/<CR><LF>
015462 020040 020040 020040         .ASCII  /       S          STOP IMMEDIATELY/<CR><LF>
015525    040 020040 020040         .ASCII  /       SD         STOP AFTER NEXT DATA TRANSFER/<CR><LF>
015605    040 020040 020040         .ASCII  /       SE         STOP AFTER NEXT ENDING SEQUENCE/<CR><LF>
015667    040 020040 020040         .ASCII  /       SI         STOP ON NEXT SEL SEQ (ISS)/<CR><LF>
015744 020040 020040 020040         .ASCII  /       SP         STOP ON NEXT PARITY ERROR/<CR><LF>
016020 005015 044127 051105         .ASCII  <CR><LF>/WHERE:/<CR><LF>
016032 020040 020040 041442         .ASCII  /     "C" IS CODE FORMAT      O = OCTAL/<CR><LF>
016100 020040 020040 020040         .ASCII  /                             A = ASCII/<CR><LF>
016146 020040 020040 020040         .ASCII  /                             E = EBCDIC/<CR><LF>
016215    040 020040 020040         .ASCII  /                             H = HEX/<CR><LF>
016261    040 020040 021040         .ASCII  /     "XX" IS DX-11 DEVICE NUMBER IN HEX/<CR><LF>
016331    040 020040 021040         .ASCII  /     "HH" IS A HEX CHARACTER/<CR><LF><LF>
016367    103 047117 047523         .ASCII  /CONSOLE CONTROL CHARACTERS/<CR><LF>
016423    103 046124 041455         .ASCII  /CTL-C (↑C) ABORT CURRENT COMMAND/<CR><LF>
016465    103 046124 050055         .ASCII  /CTL-P (↑P) REQUESTS THE REENTRY OF CONTROL PARAMETERS/<CR><LF>
016554 052103 026514 020121         .ASCII  /CTL-Q (↑Q) RESUME OUTPUT AFTER TEMPORARILY STOPPING BY (↑S)/<CR><LF>
016651    103 046124 051455         .ASCII  /CTL-S (↑S) TEMPORARILY STOP OUTPUT TO CONSOLE/<CR><LF>
```

```
        016730  052103  026514  020125          .ASCII  /CTL-U (↑U) DELETE CURRENT INPUT LINE//(CR)(LF)/
        016776  052522  047502  052125          .ASCII  /RUBOUT -- DELETE LAST CHARACTER INPUT//(CR)(LF)/
        017045     015     012     012          .BYTE   CR,LF,LF,LF
                002551                  HELPLN  =       .-HELPMS
        017051     215     212          TOOC:   .BYTE   215,212
        017053     124  047517  041440          .ASCII  /TOO CLOSE TO 200000 BOUNDARY/
        017107     377                          .BYTE   377
                                                .LIST   BEX
  4931                                          .EVEN
  4932            000001                        .END
```

```
ACCESS  004050          2705    2979#
ASCDMP  004636          2847    3161#
ATOETB  012354          3095    4315    4800#
ATTN  = 000200          2090#   4044
BFREMS  013436          2322    4930#
BSY   = 000020          2093#   4023
BSYEN = 004000          2068#   2757
BUFREL  012530          2349*   2422    2424*   2425    2426    4862#
BUSOUT= 000040          2104#   3690
BYSS  = 000400          2048#
CDEV    013106          2813    3449*   4913#
CDEVST  010256          2737    3023    3047    3514    3547    3568    3603    4171#
CE    = 000010          2094#   2595    4071
CEDE  = 000003          2015#   3569    3732    4062
CERR    003026          2694#   2733    2792    2859    2984    3072    3273    3353
CHEND = 000040          2051#   3500
CHENDS= 001000          2047#
CHIS  = 000200          2049#   3497
CHTB    011604          2259    2618    2922    3345    4685#
CHTYMS  013344          2277    4930#
CHTYPE  012526          2275*   2283*   3759    4093    4212    4860#
CMDCHF  013074          2746*   3535*   3551*   3573    3575*   3606*   3701    3703*   3777    3891    4907#
CMDCHN= 000004          2054#   3604
CMDREJ= 000001          2056#   3656
CMREAD= 000006          2034#   4265
CMRMI = 000002          2032#   4261
CMSRMI= 000012          2035#   4263
CMWRT = 000001          2031#
CMWTLA= 000005          2033#   4278
CONEND  012535          4866#
CONUNE  007750          3806    3930    4037#
COTB    011540          2633    2979    2998    3260    3282    4657#
CR    = 000015          1971#   2204    2227    2266    2324    2646    2782    3003    3088    4449    4930
CRLF    011330          2190    2666    2810    3163    3182    3217    4469    4528#
CSPWST  010310          2738    3515    3548    3565    3594    3652    4191#
CTL.C = 000003          1974#   4439
CTL.P = 000020          1975#   4425
CTL.Q = 000021          1976#   4432
CTL.S = 000023          1977#   4428
CTL.U = 000025          1978#   4457
CUBUSY= 000400          2070#   3536
CUE   = 000040          2092#   4037
CUEND = 000020          2052#   3503
DE    = 000004          2095#   2595    4071
DEVCON  013110          2540*   2541*   3025    3050    3823    3867    3949    4004    4020    4082    4113    4195    4914#
DEVMES  013300          2249    4930#
DISCEX  010734          4308    4345    4355    4362    4388#
DISCTL  010402          3597    4258#
DISFRN  010652          4273    4344#
DISPSZ= 000740          2024#   2563    3074    3084    3310    3323    3522    3826    3865    3947    4002    4052    4307
                        4334    4348
DMPADR  013124          2860*   2878    2895    4920#
DONE  = 000200          2071#   3421    3450
DREAD   010704          4266    4361#
DRMI    010712          4262    4371#
DSRMI   010734          4264    4382#
```

```
DSTOFF  013116          2482*   3028    4917#
DTUMTB  003632          2862    2893#
DUMP    003432          2707    2845#
DXAB    006536          3468    3585    3671    3751#
DXABFL  013072          2673    2747*   2821*   3752*   4904#
DXACT   013070          2745*   3110    3534*   3550*   3570    3572*   3602*   3713*   3728*   3829*   3871*   3960*   4014*
                        4103*   4117*   4902#
DXBA    012474          3821*   3822*   3846*   3850    3855*   3863*   3864*   3948*   4003*   4110*   4111*   4112*   4842#
DXBC    012476          3733    3826*   3854*   3865*   3952*   3957*   4007*   4012*   4116*   4843#
DXCA    012466          3825*   3869*   3951*   4006*   4084*   4115*   4839#
DXCB    012504          4846#
DXCS    012470          2732    2734*   2735*   2756*   2757*   2758*   2809*   2822*   2823*   3421*   3450*   3536*   3751*
                        3831*   3873*   3961*   4024*   4102*   4118*   4840#
DXDS    012464          2406    2816    4838#
DXENB =  000100         2072#   2758    2809    3751
DXES1   012510          4848#
DXES2   012514          4850#
DXEXEC  006554          3126    3414    3759#
DXEXIT  010212          3386    3754    3779    3790    3832    3874    3893    3914    3962    4025    4104    4124#
DXISR   005334          2416    3372#
DXMI    012502          4845#
DXMO    012500          4844#
DXMOB   012512          4849#
DXND    012506          4847#
DXONLN=  001000         2069#   2732    2758    3751
DXOS    012472          2473*   4101*   4841#
DXRD  =  000005         2079#   3873    4024    4118
DXRST =  000001         2077#
DXST  =  000007         2080#   4102
DXSTPF  013066          2793*   3607    3639    3654    3730    3744*   4900#
DXWR  =  000003         2078#   3831    3961
EADDR   013122          2876*   2882    2934*   2940    3285*   3309*   3310*   3322*   3323*   4919#
EBCDMP  004572          2850    3139#
EBCDSP=  000100         2019#   2362    3523    4053    4373
EBCDTB  011754          3142    3145    4312    4748#
EDEV    012524          2247*   2252*   2260*   2261    2486    2536    4859#
ENABLE  004222          2709    3022#
EOM   =  000152         2021#   3104
EQPCHK=  000020         2105#   3736
ERASCM  007770          3805    3929    4051#
ESEND =  000100         2050#   3491
ESEQ    010042          3801    3844    3852    3925    3978    4071#
EXEC    002720          2588    2666#   2696    2759    2804    2824    2886    2906    2942    2965    3009    3031    3054
                        3112    3118
EXECMD  003042          2687    2700#
FILL    003716          2711    2921#
FILLCH  012534          2362*   2367*   2395*   2564    2923*   2939    3526    4056    4865#
FILLMS  013636          2393    4930#
FIOMS   013574          2379    4930#
FTIMFL  013132          2131*   2182    2199*   2577*   4928#
GDEV    005302          2866    2924    3022    3046    3069    3345#
GETPRM  001162          2183    2190#
GLERR   005104          3272#   3281    3284    3294    3306    3319    3329
GLEX    005272          3286    3328#
GLEX1   005270          3298    3311    3324#
GLIMIT  005050          2845    2921    3259#
```

```
GLMIN   005172          3266    3304#
GLMOT   005232          3268    3317#
GLMTT   005142          3264    3292#
GLOCT   005112          3262    3279#
HDMP    004760          2814    3218#
HELP    004004          2713    2951#
HELPLN= 002551          2952    4930#
HELPMS  014300          2951    4930#
HEXDMP  004754          2853    3217#
ILLMES  013762          3467    4930#
INFDSC= 004000          2045#   3485
INHEX   002624          2248    2392    2618#
INIT    002026          2184    2364    2380    2382    2394    2404#
INIT10  002372          2526#   2528
INMES   011506          3466    3583    3637    3669    4634#
INOCT   002632          2201    2224    2276    2304    2321    2359    2378    2633#
INPUT   004330          2715    3067#
INRS    002636          2619    2634#
INTREQ= 000100          2103#   3682
INVLDC  014050          3670    4930#
IOBUF   012533          2377*   2385*   4344    4864#
ISSREJ= 000010          2053#   3610
KILL    004266          2717    3046#
KILLEX  004324          3054#
KISAR0= 172340          1962#   2437
KISAR7= 172356          1963#   2448
KISDR0= 172300          1964#   2438
LF    = 000012          1972#   4930
LINECT  013055          2672#   3161    3164*   3165*   4894#
LINSZ = 000050          2022#   2024    4293    4325    4328
LOOP    005364          3380    3411#   3504    3537    3552    3576    3631    3704    3737
MAXDEV  013067          2539*   2568    2743    3349    3455    3532    3785    3909    4901#
MCMDTB  007326          3922    3923#
MDEV    013076          2754*   3894    3912    3921*   4908#   4909
MESG    011472          2191    2293    2340    2586    2811    4546    4564    4611#
MEX     007210          3761    3891#
MMERR   011700          2160    4723#
MMMES   013376          2305    4930#
MMRESP  012527          2303*   2311*   2334    2427    4861#
MMSR0 = 177572          1965#   2313*   2452*   4723*
MREAD   007600          3928    3972    3974    3997#
MRMI  = 007502          3924    3970#
MSRMI   007502          3932    3971#
MTO     011672          2158    2404    4716#
MUXEND  010342          3714    3729    4212#
MWRITE  007352          3923    3927    3943#
NEWLNE= 000025          2018#   4317
NEWPRM  001174          2199#   2205    2207    2209    2211    2212    4740
NEWP10  001460          2250    2275#   2282
NEWP20  001704          2323    2339    2349#
NOLIN = 000014          2023#   2024
NOP   = 000003          2017#   3665
NXM   = 040000          2042#   3488
NXMMSG  013706          3584    4930#
OCTDMP  004670          2818    2856    2988    3182#
OCTEX   004744          3199#
```

```
ODMP    004674          2992    3183#
PARER = 100000          2041#   3494    3684    3734
PARMES  013736          3638    4930#
PBFE  = 013046          4513    4586    4888#   4927
PBFS    012744          2174    2175    4515    4588    4886#   4927
PBUFA   013100          2425*   2441    2469    2570    4910#
PCHAR   011366          2960    3192    3233    4460    4563#
PCHEND  006456          3502    3728#
PCHEN1  006466          3720    3730#
PCHEX   006512          3715    3734#
PCHIS   006274          3499    3652#
PCTR    013071          2879    2896    2958    4435    4485    4510*   4585*   4590    4729    4903#
PCUEND  006442          3505    3713#
PESEND  006114          3493    3594#
PFPTR   013046          2174*   4511    4512*   4513    4515*   4889#
PHYOFF  013104          2479*   2480*   3822    3855    3864    3946    3987    4001    4112    4912#
PINDSC  006026          3487    3565#
PISR    011232          2153    4484#
PIUFL   013052          4484*   4505    4509*   4726*   4891#
PMAX    013130          4590    4635    4636*   4639*   4927#
PMMERR  014167          4724    4930#
PMTO    014143          4716    4930#
PNOMM   014102          2294    4930#
PNXM    006102          3490    3583#
PPARER  006244          3496    3637#
PPPTR   013050          2175*   4581    4593*   4890#
PRINT2  011346          2668    2695    2990    2994    3166    4530    4545#
PRMESG  011404          2635    4579#   4613    4638    4728
PROUT   011252          4437    4487    4505#   4589
PSELRT  005776          3484    3547#
PSW   = 177776          1961#   2181*   2292*   3117    3119*   4727*   4739*
PSYSRT  005650          3452    3513#
P1BF    011402          4563*   4565    4567#
P2BF    011362          4545*   4547    4549#
QCUE  = 000010          2016#   3630
RNMESG  014224          2587    4930#
RSTART  001002          2131#   4427
RUBOUT= 000177          1979#   4454
RUN     003112          2719    2732#
SADDR   013120          2875*   2877    2881*   2882    2894    2898    2900*   2901*   2933*   2935    2985*   2986    2991
                        3006*   3007*   3279*   3295*   3296*   3308*   3309    3321*   3322    4918#
SATTN   007760          3807    3931    4044#
SBUFA = 000012          1998#   3945*   3946*   3948    3980*   3984    3987*   4000*   4001*   4003    4174*   4228*
SCMD  = 000000          1991#   3566    3569*   3598    3630*   3672*   3732*   3775    3782*   3794    3827    3866    3896
                        3900*   3918    3959    3997    4062*   4172*
SCMDRJ= 000200          2102#   3696
SCMDTB  006704          3798    3799#
SCURS = 000004          1995#   3074    3076    3078*   3080    3083*   3084    3105*   3519*   3848    3982    4060*   4287*
                        4293*   4306    4316*   4325*   4327*   4328*   4334    4336*   4361*   4375*
SDEV    012522          2246*   2251*   2261    2263    2484    2537    2540    3346    3453    4858#
SDEVTB  013064          2552*   3307    3320    4143    4899#
SELDEV= 013076          2542*   3773    3788    3797*   4909#
SELRST= 020000          2043#   3482
SENSCM  010140          3802    3926    4110#
SEX     006566          3760    3773#
SINBF = 000006          1996#   2559*   2560*   3308    3821    3945    4277    4346
```

```
SLCMD = 000001   1992#  3595  3601*  3827*  3866*  3959*  3997*  4176*  4261  4263  4265  4278
SMI   = 000112   2020#  3081
SMINS = 000020   2002#  2741*  3082*  3518*  3846  3980  4371
SONLF = 000016   2000#  3030*  3048*  3677  4192
SOUTB = 000010   1997#  2561*  2562*  3079  3321  3521  3847  3863  3981  4000  4051  4305  4347
                 4374
SPACE = 000040   1973#  3090  3094
SRBYTC= 000014   1999#  3733*  3830*  3872*  3943  3947*  3953  3955  3973  3986*  3998  4002*  4008
                 4010  4175*  4217*  4222*  4284*  4307  4309*
SRDRQ = 000017   2001#  2740*  3106*  3517*  3600*  3780  3842  3845*  3898  3976  3979*
SRD10   007142   3856  3866#
SREAD   007120   3804  3840  3863#
SRMI    007020   3800  3838#
SSENSE= 000002   1993#  2739*  3029*  3516*  3549*  3682*  3690*  3696*  3736*  3828*  3841*  3870*  3958*
                 3975*  4013*  4061*  4110
SSRMI   007020   3808  3839#
SSTACK= 012636   2142  2669  4873#
SSTAT = 000003   1994#  3520*  3641*  4037*  4044*  4071*  4085  4087*  4101  4173*
STAMOD= 000100   2091#
START   001000   2120  2127#  4732
STKSTB= 000002   2055#
STMSG   013134   2192  4930#
STOP    003264   2721  2781#
STOPDX  003340   2675  2783  2809#
STOUT   010050   4038  4045  4082#
STPDX   006532   3608  3640  3655  3731  3744#
STPMES  014016   2812  4930#
STSPW   013114   2478*  3027  3052  4022  4096  4197  4916#
SUDEV   010230   3073  3473  3629  3774  3895  4143#  4171
SUDEX   010252   4147  4150#
SUD10   010240   4146#  4149
SWRITE  006730   3799  3803  3821#
SYSINT  001006   2127  2142#
SYSRST= 010000   2044#  3451
TBUF  = 012640   2173  2642  2682  2997  4444  4451  4464  4468  4882#
TBUFE   012740   4470  4884#
TCHEND  005624   3498  3500#
TCHIS   005612   3495  3497#  3642
TCMACT  013053   2636*  2640  2670*  2676  2987*  2995  4441  4446  4452*  4892#
TCMDAB  013054   2637*  2638  2671*  2884  2904  2961  4443*  4893#
TCMDTB  003044   2685  2704#
TCUEND  005636   3501  3503#
TESEND  005566   3489  3491#
TESTMS  013534   2360  4930#
TINDSC  005542   3483  3485#
TKB     012456   4422  4829#
TKIN    010744   2151  4420#
TKS     012454   2179*  4828#
TNXM    005554   3486  3488#
TOOC    017051   2344  4930#
TOUTRT  011710   4717  4725#
TPARER  005600   3492  3494#
TPB     012462   4511*  4831#
TPS     012460   2180*  4830#
TPTR    012742   2173*  4424  4474*  4885#
TSELRS  005530   3482#
```

```
MAINDEC-11-DZDXI-B  NEW DX11-B RESPONDER        MACY11 30A(1052)  01-FEB-78  09:46  PAGE 119                                SEQ 0118
CZDXIB.P11    30-JAN-78 14:19            CROSS REFERENCE TABLE -- USER SYMBOLS                                               SEQ 0118

TSTTYP  012532            2358*   2368*   3067    3524    3626    3839    3971    4054    4272    4863*
TTADDR  013062            2497*   2748    3436    4898*
TTPTR   013060            2498*   2749*   3295    3378    3411    3437*   4897*
TTSIZE= 001000            2025*   2499    2750    2893    2898    2900    3296    3423    3433
TTYSTP  013073            4430*   4434*   4507    4905*
UCHK  = 000002            2096*   2490    2530    2600    2601    3053    3641    4085    4087
UCHKS = 002000            2046*
UEXP  = 000001            2097*
UNADDR  012516            2200*   2213*   2214*   2405    4856*
UNMSG   013175            2202    4930*
UNTRP   011742            2222    4738*
VBUFA   013102            2426*   2453*   2474    4911*
VCMDTB  002610            2525    2592*
VECTAD  012520            2223*   2235*   2415    4857*
VECTMS  013232            2225    4930*
VEND    013126            2166    4921*
VSTRT   012636            2165    2166    4881*
WK      013056            3139*   3140    3142*   3146    4895*
WK1     013057            3143    3145*   4896*
XADDR   013112            2576*   2756    4915*
.     = 017110            2111*   2119*   2126*   2603*   3970    4872*   4873    4882    4883*   4887*   4888    4930*

. ABS.  017110      000

 ERRORS DETECTED:  0

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