DX11-B

DX-11B RESPONDER
CZDXICO

AH-8777C-MC
FICHE 1 OF 1

SEP 1982

MADE IN USA

```
        .REM    %
```

IDENTIFICATION

PRODUCT NAME: CZDXIC0 DX11-B RESPONDER

PRODUCT CODE: AC-8776C-MC

RELEASE DATE: JULY 1976

REVISED MARCH 1982

MAINTAINER: DIAGNOSTIC ENGINEERING

PROGRAM HISTORY

02-FEB-82 REV. C
THE CONTROL P COMMAND HAS BEEN CHANGED TO CONTROL R. THERE IS
A CONFLICT BETWEEN RDC AND THIS PROGRAM, WHEN USING CTL P.

1.0 GENERAL DESCRIPTION

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

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

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

2.0 REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11 COMPUTER WITH A MINIMUM OF 8K OF MEMORY.

B. DX11-B 360/370 INTERFACE OPTION.

C. ONE CONSOLE TELETYPE OR EQUIVALENT.

2.2 STORAGE

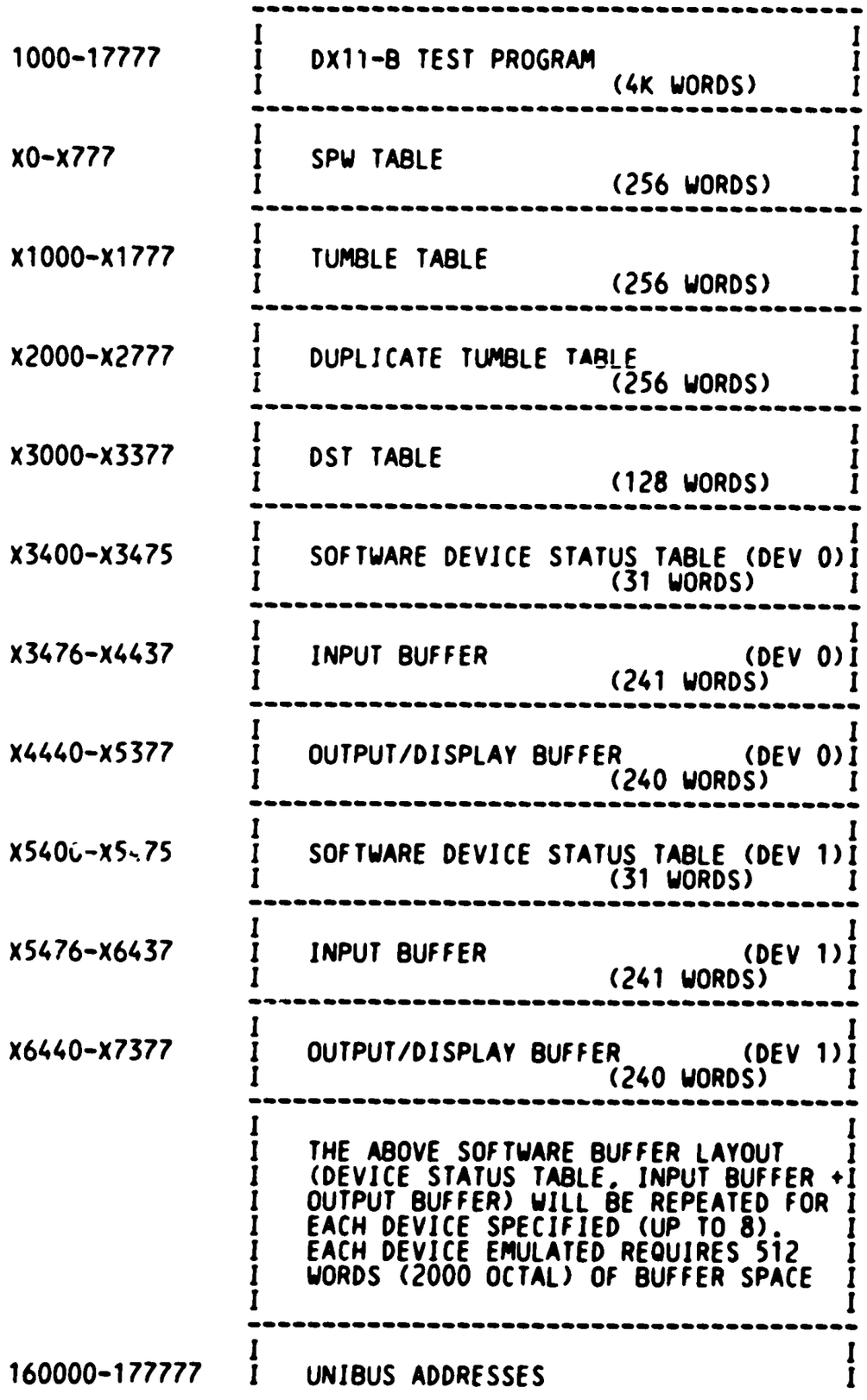
THE TEST PROGRAM LOADS INTO 4K OF MEMORY AND REQUIRES AT LEAST ANOTHER 4K FOR DATA BUFFERS. WITH 4K OF MEMORY, FOR DATA BUFFERS, UP TO SIX DEVICES (6) MAY BE EMULATED. TO EMULATE EIGHT 2260/2848 DEVICES 8K OF MEMORY FOR DATA BUFFERS IS REQUIRED.

2.3 STORAGE MAP

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

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 R (^R)
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 ''CZDXI-C 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-R (^R) 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.

5.0 ERRORS

6.1 ERROR HALTS

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

6.2 DX ERRORS

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

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

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

6.3 ERROR MESSAGES AND SUGGESTED CORRECTIVE ACTIONS

"MEMORY TIME OUT"

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

THE SYSTEM SHOULD PROBABLY BE RECONFIGURED.

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

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

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

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

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

7.0 RESTRICTIONS

SEE MEMORY REQUIREMENTS (SECTION 2.2)

7.1 MULTIPLE DEVICE ADDRESSES

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

```
%
.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.

^R - JUMP TO RESTART TO RESELECT PARAMETERS.

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

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

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

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

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

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

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

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

            ----GETS DX VECTOR - DITTO

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

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

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

            ----GETS CHANNEL TYPE  M OR S

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

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

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

                *  CHECKS FOR BOUNDARY 20000 OR GREATER

                *  CHECKS FOR MULTIPLE OF 2000

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

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

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

            ----FILL CHARACTER IN HEX
                SAVE IN FILLCH

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

NOMM:   NO MEMORY MANAGEMENT AVAILABLE.

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

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

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

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

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

-----PROGRAM INITIALIZATION----

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

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

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

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

SPW TABLE = 400(8) WORDS.

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

TT = 400(8) WORDS

DTT = 400(8) WORDS

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

```
        ----SET UP FILL CHARACTER


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


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

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


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

            - REMEMBER MEMORY MANAGEMENT HAS BEEN TURNED ON-

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



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



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


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

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

RUN DX COMMAND

```
RUN:  CHECK IF DX IS ENABLED -
        IF YES, TYPE ? AND (BELL)--RETURN TO EXEC AND
                                   WAIT FOR ANOTHER TTY COMMAND.
        IF NO,CONTINUE
        RETURN TO EXEC.

        CLR DXCS
        INC DXCS - GO

        CLR DEVICE STATUS BUFFER TABLE
                (SCMD
                 SLCMD
                 SSENSE  (NOT SCURS, SINTB, SOUBF, SONLF)
                 SSTAT
                 SBUFA
                 SRBYTC
                 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 13 DX REGISTERS CONTENTS.
```

CLR ABORT FLAG (DXABFL), CLR DONE
RESET DX, SET GO, RETURN TO EXEC.

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

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

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

ANY OTHER CHARACTER = AN ILLEGAL CHARACTER

DUMP COMMAND

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

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

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

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

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

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

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

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

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

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

FILL COMMAND

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

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

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

ACCESS COMMAND

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

ENABLE DEVICE

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

KILL DEVICE

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

INPUT COMMAND

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

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

            PUSH STACK (CREATE PHONEY INTERRUPT)
            JUMP DXEXEC



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


    SYSTEM RESET

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


    SELECTIVE RESET

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


    INTERFACE DISCONNECT:

    PINDSC:  IF DEVICE WAS ACTIVE, ITS DEVICE STATUS TABLE WILL BE
            CLEARED - IF NOT ACTIVE, IGNORE CMD.

            IF ACTIVE - QUEUE CE! DE IN SCMD

            (TYPICALLY IBM WILL INTERFACE DISCONNECT A DEVICE EVEN
            THO THE DEVICE WAS NOT ACTIVE)

            IF ACTIVE - CHECK FOR CMDCHF: & DXACT: FOR THAT PARTICULAR
            DEVICF - 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 W  N 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
```

```
        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 STAR' 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

INIT: (INT140:)

USED IN

DUMP:
DISCTL:
MWRITE: & SWRITE:

SOUTB (10)

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

HOW USED

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

SET UP IN

INIT: (INT140:)

USED IN

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

SBUFA (12)

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

HOW USED

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

SET UP IN	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:

%

```
  1950                                  .TITLE  MAINDEC-11-CZDXI-C  NEW DX11-B RESPONDER
  1951                                  .ENABL  ABS
  1952                                  .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
000021                          CTL.Q   =       21              ;CONTROL Q
000022                          CTL.R   =       22              ;CONTROL R
000023                          CTL.S   =       23              ;CONTROL S
000025                          CTL.U   =       25              ;CONTROL U
000177                          RUBOUT  =       177             ;RUBOUT
```

VRG-02-FEB-82

```
1985                                ;*****************************************************************
1986                                ;*                                                               *
1987                                ;*      DEVICE BUFFER LAYOUT    (1 PER DEVICE)                   *
1988                                ;*                                                               *
1989                                ;*****************************************************************
1990                                ;*      LOC     0-61 = DEVICE STATUS TABLE                       *
1991                                ;*      LOC     62-543 = DEVICE INPUT BUFFER                     *
1992                                ;*      LOC     554-1023 = DEVICE OUTPUT/DISPLAY BUFFER          *
1993                                ;*****************************************************************
1994                                ;       LAYOUT OF DEVICE STATUS TABLE
1995                                ;
1996            000000              SCMD    =       0       ;CURRENT DEVICE COMMAND
1997            000001              SLCMD   =       1       ;LAST COMMAND, IF WRITE
1998            000002              SSENSE  =       2       ;DEVICE SENSE BYTE (NOTE -- MUST BE EVEN BYTE LOCATION)
1999            000003              SSTAT   =       3       ;DEVICE STATUS
2000            000004              SCURS   =       4       ;CURSOR POSITION
2001            000006              SINBF   =       6       ;ADDRESS OF DEVICE INPUT BUFFER
2002            000010              SOUTB   =       10      ;ADDRESS OF DEVICE OUTPUT BUFFER
2003            000012              SB FA   =       12      ;CURRENT BUFFER PTR
2004            000014              SRBYTC  =       14      ;REMAINING BYTE COUNT
2005            000016              SONLF   =       16      ;DEVICE ONLINE - INDICATOR 0=ON-LINE 1=OFF-LINE
2006            000017              SRDRQ   =       17      ;READ MANUAL INPUT REQUEST -- IF NON-ZERO
2007            000020              SMINS   =       20      ;START OF MANUAL INPUT DATA
2008                                ;
2009                                ;       LOCATIONS 22-77 ARE AVAILABLE FOR EXPANSION PURPOSES
2010                                ;
2011
2012
2013                                ;
2014                                ;       OTHER   DX EQUATES
2015                                ;
2016            000000              DEV     =       R0              ;CURRENT DEVICE NUMBER
2017            000003              DTAB    =       R3              ;ADDRESS OF CUR DEV STATUS TABLE
2018            000004              TT1     =       R4              ;TUMBLE TABLE ENTRY 1
2019            000005              TT2     =       R5              ;TUMBLE TABLE ENTRY 2
2020            000003              CEDE    =       3               ;CHAN END & DEV END
2021            000010              QCUE    =       10              ;CODE TO QUE CONTROL UNIT END
2022            000003              NOP     =       3               ;NOP COMMAND
2023            000025              NEWLNE  =       25              ;NEW LINE CHARACTER
2024            000100              EBCDSP  =       100             ;EBCDIC SPACE CODE
2025            000112              SMI     =       112             ;START OF MESSAGE INDICATOR
2026            000152              EOM     =       152             ;END OF MESSAGE INDICATOR
2027            000050              LINSZ   =       40.             ;NUMBER OF CHARACTERS PER LINE OF 2260 DISPLAY
2028            000014              NOLIN   =       12.             ;NUMBER OF LINES PER 2260 DISPLAY
2029            000740              DISPSZ  =       NOLIN*LINSZ     ;NUMBER OF CHARACTERS ON THE DISPLAY
2030            001000              TTSIZE  =       512.            ;NUMBER OF ENTRIES IN TUMBLE TABLE
2031
2032
2033                                ;
2034                                ;       360 COMMAND EQUATES
2035                                ;
2036            000001              CMWRT   =       1               ;WRITE DATA (FROM 360 TO PDP-11)
2037            000002              CMRMI   =       2               ;READ MANUAL INPUT (PDP-11 TO 360)
2038            000005              CMWTLA  =       5               ;WRITE LINE ADDRESS (360 TO PDP-11)
2039            000006              CMREAD  =       6               ;READ FULL BUFFER (PDP-11 TO 360)
2040            000012              CMSRMI  =       12              ;SHORT READ MANUAL INPUT (PDP-11 TO 360)
```

```
2041                                    ;
2042                                    ;       DX REGISTER ASSIGNMENTS & LAYOUTS
2043                                    ;
2044                                    ;               DXDS OR TUMBLE TABLE ENTRY 1 (TT1)
2045                                    ;
2046            100000                  PARER   =       100000          ;PARITY ERROR DETECTED
2047            040000                  NXM     =       40000           ;NON EXISTENT MEMORY CONDITION
2048            020000                  SELRST  =       20000           ;IBM SELECTIVE RE-SET
2049            010000                  SYSRST  =       10000           ;IBM SYSTEM RESET
2050            004000                  INFDSC  =       4000            ;IBM PROGRAMMED INTERFACE DISCONNECT
2051            002000                  UCHKS   =       2000            ;UNIT CHECK WAS PRESENTED TO THE CHANNEL
2052            001000                  CHENDS  =       1000            ;CHANNEL END WAS PRESENTED TO THE CHANNEL
2053            000400                  BYSS    =       400             ;BUSY WAS PRESENTED TO THE CHANNEL
2054            000200                  CHIS    =       200             ;CHANNEL INIT SELECTION SEQ WAS COMPLETED
2055            000100                  ESEND   =       100             ;CHANNEL ACCEPTED LAST STATUS
2056            000040                  CHEND   =       40              ;CHANNEL DATA TRANSFER END
2057            000020                  CUEND   =       20              ;DX DATA TRANSFER END
2058            000010                  ISSREJ  =       10              ;INIT SELECTION SEQ WAS REJECTED
2059            000004                  CMDCHN  =       4               ;CHANNEL SPECIFIED COMMAND CHAING
2060            000002                  STKSTB  =       2               ;CHANNEL COULD NOT ACCEPT LAST STATUS
2061            000001                  CMDREJ  =       1               ;CHANNEL COMMAND WAS REJECTED
2062
2063
2064                                    ;
2065                                    ;               DXCA OR TUMBLE TABLE ENTRY 2 (TT2)
2066                                    ;
2067                                    ;                       BITS 15-8 = COMMAND (IF ANY)
2068                                    ;                       BITS 7-0 = DEVICE ADDRESS
2069                                    ;
2070                                    ;
2071                                    ;               DXCS CONTROL UNIT STATUS REGISTER
2072                                    ;
2073            004000                  BSYEN   =       4000            ;BUSY RENABLE - FOR SELECTOR CHANNELS
2074            001000                  DXONLN  =       1000            ;ON-LINE INDICATION
2075            000400                  CUBUSY  =       400             ;CONTROL UNIT BUSY
2076            000200                  DONE    =       200             ;DONE FLAG
2077            000100                  DXENB   =       100             ;INTERRUPT ENABLE
2078                                    ;
2079                                    ;                       BITS 4+3 ARE SET IF EXTENDED ADDRESS IS USED > 32K
2080                                    ;                       BITS 2-0 ARE THE FUNCTION TO BE PERFORMED
2081                                    ;
2082            000001                  DXRST   =       1               ;DX RESET COMMAND
2083            000003                  DXWR    =       3               ;WRITE DATA TO THE 360
2084            000005                  DXRD    =       5               ;READ DATA FROM THE 360
2085            000007                  DXST    =       7               ;SEND STATUS TO THE 360
2086
2087
2088                                    ;
2089                                    ;
2090                                    ;               DXOS OFFSET AND STATUS REGISTER
2091                                    ;
2092                                    ;                       BITS 15-10 OFFSET OF SPW TABLE
2093                                    ;                       STATUS REGISTER DEF (SSTAT) - STATUS BYTE
2094                                    ;
2095            000200                  ATTN    =       200             ;ATTENTION
2096            000100                  STAMOD  =       100             ;STATUS MODIFIER
```

```
2097    000040          CUE     =       40              ;CONTROL UNIT END
2098    000020          BSY     =       20              ;BUSY
2099    000010          CE      =       10              ;CHANNEL END
2100    000004          DE      =       4               ;DEVICE END
2101    000002          UCHK    =       2               ;UNIT CHECK
2102    000001          UEXP    =       1               ;UNIT EXCEPTION
2103                    ;
2104                    ;
2105                    ;       2848 SENSE BYTE (SSENSE) DEFINITION
2106                    ;
2107    000200          SCMDRJ  =       200             ;COMMAND REJECT
2108    000100          INTREQ  =       100             ;DEVICE OFF-LINE - INTERVENTION REQ
2109    000040          BUSOUT  =       40              ;BUS OUT -- PARITY ERROR DURING CHIS
2110    000020          EQPCHK  =       20              ;EQUIPMENT CHECK - PARITY ERROR DUR DATA TRANS
```

```
                                        .SBTTL  TRAP CATCHER
                                ;
                                ;       THE TRAP CATCHER IS LOADED INTO LOW CORE


                000000                  .=0
000000  000000  000000                  HALT                    ;FOR MEMORY MANAGEMENT
000002  000000  000000                  HALT
                000176                  .REPT   176             ;TRAP CATCHER

                                        .WORD   .+2
                                        HALT
                                        .ENDR
                000200                  .=200
000200  000137  001000                  JMP     START           ;ESTABLISH LOC 200 STARTING ADDRESS
```

```
2127                                            .SBTTL  PROGRAM START-UP SEQUENCES
2128                                    ;
2129                                    ;       NORMAL SYSTEM START LOCATION  --1000
2130                                    ;
2131            001000                          .=1000
2132  001000    000402          START:  BR      SYSINT          ;NORMAL START UP
2133                                    ;
2134                                    ;       RESTART ADDRESS -- REENTER ALL PARAMETERS -- 1002
2135                                    ;
2136  001002    005037  013144  RSTART: CLR     FTIMFL          ;RESET FIRST TIME FLAG TO FORCE PARAMETER REENTRY
2137                                    ;
2138                                    ;       SYSTEM GENERALIZED INITIALIZATION PROCEDURE
2139                                    ;
2140                                    ;               SET-UP  STACK POINTER
2141                                    ;                       TRAP/VECTOR AREA
2142                                    ;                       SYSTEM GENERATED TRAPS
2143                                    ;                       TELETYPE (CONSOLE) VECTORS + STATUS REGISTERS
2144                                    ;                       CLEAR ALL LIVE SYSTEM VARIABLES
2145                                    ;                       SET UP TELETYPE INPUT / OUTPUT BUFFERS
2146                                    ;
2147  001006    012706  012650  SYSINT: MOV     #SSTACK,SP      ;SET UP THE STACK POINTER
2148  001012    000005                  RESET                   ;RESET ALL DEVICES
2149
2150
2151
2152                                    ;
2153                                    ;       SET UP CONSOLE VECTORS
2154                                    ;
2155  001014    012700  000060          MOV     #60,R0          ;START OF CONSOLE VECTORS
2156  001020    012720  010756          MOV     #TKIN,(R0)+     ;BEG OF TELE INPUT ISR
2157  001024    012720  000340          MOV     #340,(R0)+      ;NEW PROC STATUS
2158  001030    012720  011244          MOV     #PISR,(R0)+     ;BEG OF TELE PRINT ISR
2159  001034    012710  000340          MOV     #340,(R0)       ;NEW PROC STATUS
2160                                    ;
2161                                    ;       SET UP MISC TRAPS
2162                                    ;
2163  001040    012737  011704  000004  MOV     #MTO,4          ;MEMORY TIME OUT TRAP
2164  001046    012737  000340  000006  MOV     #340,6
2165  001054    012737  011712  000250  MOV     #MMERR,250      ;MEMORY MANAGEMENT ERROR
2166  001062    012737  000340  000252  MOV     #340,252
2167                                    ;
2168                                    ;       CLEAR ALL VARIABLES
2169                                    ;
2170  001070    012700  012650          MOV     #VSTRT,R0       ;START OF VARAIBLES
2171  001074    012701  000272          MOV     #VEND-VSTRT+2,R1  ;# OF VARAIBLES TO CLEAR
2172  001100    105020          10$:    CLRB    (R0)+           ;CLEAR A BYTE
2173  001102    005301                  DEC     R1              ;DONE?
2174  001104    001375                  BNE     10$             ;NO, CONTINUE CLEARING
2175                                    ;
2176                                    ;       SET UP TELE BUFFER POINTERS
2177                                    ;
2178  001106    012737  012652  012754  MOV     #TBUF,TPTR      ;TELE INPUT POINTER
2179  001114    012737  012756  013060  MOV     #PBFS,PFPTR     ;TELE OUT FETCH PTR
2180  001122    012737  012756  013062  MOV     #PBFS,PPPTR     ;TELE OUT PUT PTR
2181                                    ;
2182                                    ;       ENABLE TTY
```

```
2183                                   ;
2184 001130 052777 000100 011330        BIS     #100,@TKS        ;ENABLE TELETYPE INPUT
2185 001136 052777 000100 011326        BIS     #100,@TPS        ;ENABLE TTY OUTPUT INTERRUPTS
2186 001144 005037 177776               CLR     PSW              ;CLEAR THE PROCESSOR STATUS WORD
2187 001150 005737 013144               TST     FTIMFL           ;FIRST TIME THROUGH? (MUST PARAMETERS BE REENTERED?)
2188 001154 001402                      BEQ     GETPRM           ;YES, FORCE USER TO ENTER ALL PARAMETERS
2189 001156 000137 002040               JMP     INIT             ;NO, RESTART TEST USING SAME PARAMETERS
```

```
2190                                            .SBTTL  TOTAL SYSTEM RESTART (REQUEST NEW RUN TIME PARAMETERS)
2191                                    ;
2192                                    ;       HERE WE START GATHERING THE TEST INFORMATION
2193                                    ;               PRINT START-UP HERALD MESSAGE
2194                                    ;
2195  001162  004737  011342           GETPRM: JSR     PC,CRLF         ;RESTORE THE CARRIAGE
2196  001166  004137  011504                   JSR     R1,MESG         ;PRINT START-UP MESG
2197  001172  013146                           .WORD   STMSG
2198  001174  004737  011342                   JSR     PC,CRLF         ;;RESTORE THE CARRIAGE
2199  001200  004137  011504                   JSR     R1,MESG         ;;
2200  001204  013207                           .WORD   CTRMSG          ;;PRINT REV UPDATE NOTE         VRG-02-FEB-82
2201
2202                                    ;
2203                                    ;       GET DX11 UNIBUSS ADDRESS (OCTAL ADDRESS INPUT)
2204                                    ;               VALID UNIBUS ADDRESSESS (176200 - 177000)
2205                                    ;               DEFAULT UNIBUS ADDRESS  176200
2206                                    ;
2207  001206  005037  013144           NEWPRM: CLR     FTIMFL          ;RESET FIRST TIME PARAMETERS (FORCE ALL PARMS TO BE ENTE
2208  001212  012737  176200  012530           MOV     #176200,UNADDR  ;SET UP DEFAULT ADDRESS
2209  001220  004137  002644                   JSR     R1,INOCT        ;GET UNIBUS ADDRESS
2210  001224  013270                           .WORD   UNMSG
2211  001226  001272                           .WORD   5$              ;ADDRESS OF DEFAULT ROUTINE
2212  001230  120427  000015                   CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
2213  001234  001364                           BNE     NEWPRM          ;NO, TELL HIM TO REEENTER
2214  001236  020327  176200                   CMP     R3,#176200      ;VALID UNIBUS ADDRESS? BETWEEN 176200 AND 177000
2215  001242  002761                           BLT     NEWPRM          ;NO, GET AGAIN
2216  001244  020327  177000                   CMP     R3,#177000      ;UNIBUS ADDRESS GT 177000?
2217  001250  003356                           BGT     NEWPRM          ;YES, ERROR -- REENTER
2218  001252  032703  000037                   BIT     #37,R3          ;MAKE SURE 40 OCTAL WORD BOUNDRY
2219  001256  001353                           BNE     NEWPRM          ;ILLEGAL, REENTER
2220  001260  012737  001206  000004           MOV     #NEWPRM,4       ;SET UP TRAP OUT TO VALIDATE ADDRESS
2221  001266  010337  012530                   MOV     R3,UNADDR       ;SAVE UNIBUS ADDRESS
2222  001272  005077  011232           5$:     CLR     @UNADDR         ;VALIDATE THE UNIBUS ADDRESS
2223                                                                   ;TRAP WILL OCCUR IF INVALID UNIBUS ADDRESS
2224
2225                                    ;
2226                                    ;       GET THE DX11 INTERRUPT VECTOR ADDRESS (OCTAL ADDRESS INPUT)
2227                                    ;               VALID VECTOR ADDRESSES (300 - 770)
2228                                    ;               DEFAULT VECTOR ADDRESS  300
2229                                    ;
2230  001276  012737  011754  000004   10$:     MOV     #UNTRP,4        ;RESTORE MEMORY TIME-OUT TRAP
2231  001304  012737  000300  012532           MOV     #300,VECTAD     ;SET UP DEFAULT VECTOR ADDRESS
2232  001312  004137  002644                   JSR     R1,INOCT        ;GET VECTOR ADDRESS
2233  001316  013324                           .WORD   VECTMS
2234  001320  001356                           .WORD   20$             ;ADDRESS OF THE DEFAULT ENTRY
2235  001322  120427  000015                   CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
2236  001326  001363                           BNE     10$             ;NO, REENTER
2237  001330  020327  000300                   CMP     R3,#300         ;CHECK VECTOR ADDRESS BETWEEN 300 AND 770
2238  001334  002760                           BLT     10$             ;TOO LOW GIVE AN ERROR AND REENTER
2239  001336  020327  000770                   CMP     R3,#770         ;LT 770?
2240  001342  003355                           BGT     10$             ;YES, REENTER
2241  001344  032703  000001                   BIT     #1,R3           ;WORD ADDRESS?
2242  001350  001352                           BNE     10$             ;NO, REENTER
2243  001352  010337  012532                   MOV     R3,VECTAD       ;SAVE IT
2244
2245                                    ;
```

```
                                        ;       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.
                                        ;
001356  012737  000020  012534  20$:    MOV     #20,SDEV        ;DEFAULT TO HEX ADDRESS 10
001364  012737  000020  012536          MOV     #20,EDEV        ;
001372  004137  002636                  JSR     R1,INHFX        ;GET DEVICE ADDRESSES IN HEX
001376  013372                          .WORD   DEVMES
001400  001472                          .WORD   NEWP10          ;ADDRESS OF THE DEFAULT ROUTINE
001402  010337  012534                  MOV     R3,SDLV         ;SAVE START DEV ADDR
001406  010337  012536                  MOV     R3,EDEV
001412  005703                          TST     R3              ;BE SURE POSITIVE
001414  100760                          BMI     20$
001416  020327  000377                  CMP     R3,#377         ;AND NOT GREATER THAN 377 -- HEX FF
001422  003355                          BGT     20$             ;ILLEGAL ENTRY
001424  120427  000054                  CMPB    R4,#',          ;MORE THAN ONE DEV? (COMMA, PARAMETER DELIMETER)
001430  001015                          BNE     30$
001432  004737  011616                  JSR     PC,CHTB         ;GET ENDING DEVICE
001436  010337  012536                  MOV     R3,EDEV         ;SAVE ENDING ADDRESS
001442  023737  012534  012536          CMP     SDEV,EDEV       ;IS START LT END?
001450  003342                          BGT     20$             ;YES, ERROR
001452  163703  012534                  SUB     SDEV,R3         ;MORE THAN 8 DEVICES?
001456  020327  000007                  CMP     R3,#7
001462  003335                          BGT     20$             ;YES, ERROR
001464  120427  000015          30$:    CMPB    R4,#CR          ;WAS DEVICE ADDRESSES DELIMITED PROPERLY?
001470  001332                          BNE     20$             ;NO, REENTER

                                        ;
                                        ;       GET TYPE OF 360/370 CHANNEL
                                        ;               M = MULTIPLEXER CHANNEL
                                        ;               S = SELECTOR CHANNEL
                                        ;               DEFAULT IS 'S', SELECTOR CHANNEL
                                        ;
001472  105037  012540          NEWP10: CLRB    CHTYPE          ;0 = M, 1 = S
001476  004137  002644                  JSR     R1,INOCT        ;GET CHANNEL TYPE
001502  013436                          .WORD   CHTYMS
001504  001522                          .WORD   50$             ;DEFAULT TO SELECTOR CHANNEL
001506  120427  000115                  CMPB    R4,#'M          ;M? -- MULIPLEXER CHANNEL --
001512  001414                          BEQ     60$             ;YES, MULTIPLEXER CHANNEL
001514  120427  000123                  CMPB    R4,#'S          ;S? -- SELECTOR CHANNEL --
001520  001364                          BNE     NEWP10          ;NOT S OR M -- ERROR
001522  105237  012540          50$:    INCB    CHTYPE          ;SELECTOR CHANNEL
001526  000406                          BR      60$             ;GET MEMORY MANAGEMENT FACILITIES


                                        ;
                                        ;       MEMORY MANAGEMENT TIME-OUT TRAP HANDLER
                                        ;               USED ONLY FOR PARAMETER ENTRY PROCESS
                                        ;
001530  022626                  55$:    CMP     (SP)+,(SP)+            ;DUMP PC AND PSW SAVED BY INTERRUPT
001532  005037  177776                  CLR     PSW             ;TURN DOWN PROCESSOR STATUS
001536  004137  011504                  JSR     R1,MESG         ;PRINT 'NO MEM MANAGEMENT AVAIL''
```

```
2302  001542  014174                        .WORD   PNOMM
2303                                                          ;ASK TO HAVE QUESTION REENTERED
2304
2305                                ;
2306                                ;       DETERMINE IF MEMORY MANAGEMENT IS TO BE USED
2307                                ;               Y = YES, MEMORY MANAGEMENT TO BE USED
2308                                ;               N = NO, DO NOT USE MEMORY MANAGEMENT
2309                                ;               DEFAULT IS 'N', DO NOT USE MEMORY MANAGEMENT
2310                                ;
2311  001544  105037  012541      60$:    CLRB    MMRESP          ;DEFAULT TO NO MEMORY MANAGEMENT
2312  001550  004137  002644              JSR     R1,INOCT        ;GET MEM MANAGEMENT
2313  001554  013470                      .WORD   MMMES
2314  001556  001612                      .WORD   70$             ;DEFAULT ROUTINE ADDRESS
2315  001560  120427  000116              CMPB    R4,#'N          ;N? --DO NOT USE MEMORY MANAGEMENT
2316  001564  001412                      BEQ     70$             ;IF EQ, NO MEMORY MANAGEMENT
2317  001566  120427  000131              CMPB    R4,#'Y          ;Y? --MEMORY MANAGEMENT TO BE USED
2318  001572  001364                      BNE     60$             ;ERROR
2319  001574  105237  012541              INCB    MMRESP          ;MEMORY MANAGEMENT SPEC
2320  001600  012737  001530  000004      MOV     #55$,4          ;SET UP TRAP TO TEST MEMORY MANAGEMENT
2321  001606  005037  177572              CLR     MMSR0           ;CHECK FOR MEMORY MANAGEMENT
2322
2323                                ;
2324                                ;       GET BUFFER RELOCATION IN OCTAL ,000'S
2325                                ;               VALID RELOCATION ADDRESSES (20 - 700)
2326                                ;                       (20000 THROUGH 734000)
2327                                ;               DEFAULT RELOCATION ADDRESS  20 ---(20000)
2328                                ;
2329  001612  004137  002644      70$:    JSR     R1,INOCT        ;GET BUFFER RELOC. IN ,000'S
2330  001616  013530                      .WORD   BFREMS
2331  001620  001716                      .WORD   NEWP20          ;ADDRESS OF DEFAULT ROUTINE
2332  001622  120427  000015              CMPB    R4,#CR          ;WAS LINE DELIMITED PROPERLY?
2333  001626  001371                      BNE     70$             ;NO, REENTER
2334  001630  032703  000001              BIT     #1,R3           ;MUST BE A MULTIPLE OF 2000
2335  001634  001366                      BNE     70$
2336  001636  020327  000020              CMP     R3,#20
2337  001642  002763                      BLT     70$             ;ILLG BUFFER CONST -- LT 20000
2338  001644  005703                      TST     R3              ;IS NUMBER NEGATIVE?
2339  001646  100761                      BMI     70$             ;YES, REENTER ADDRESS
2340  001650  020327  000734              CMP     R3,#734         ;IS ADDRESS TOO LARGE?
2341  001654  002356                      BGE     70$             ;YES, REENETER ADDRESS
2342  001656  105737  012541              TSTB    MMRESP          ;WAS MEMORY MANAGEMENT SPECIFIED?
2343  001662  001412                      BEQ     71$             ;NO, CHECK FOR 28K
2344  001664  010304                      MOV     R3,R4           ;PUT VALUE IN WORK REG
2345  001666  042704  000600              BIC     #600,R4         ;IGNORE ADDRESS EXTENSION BITS
2346  001672  020427  000154              CMP     R4,#154         ;IS IT TOO CLOSE TO 200000 BOUNDARY?
2347  001676  003407                      BLE     NEWP20          ;BRANCH IF OK
2348  001700  004137  011504              JSR     R1,MESG         ;PRINT ERROR.  CANNOT SET BUFFER SO
2349                                ;CLOSE TO A 200000 BOUNDARY THAT A CARRY WOULD BE NEEDED TO CHANGE
2350                                ;THE EXTENDED ADDRESS BITS.  THE DX CANNOT WORK ACROSS 200000
2351                                ;BOUNDARIES.
2352  001704  017143                      .WORD   TOOC            ;ADDRESS OF TOO CLOSE MESSAGE
2353  001706  000741                      BR      70$             ;ASK FOR INPUT AGAIN
2354
2355  001710  020327  000134      71$:    CMP     R3,#134         ;NO, IS IT TOO CLOSE TO I/O PAGE?
2356  001714  002336                      BGE     70$             ;YES, REENTER THE ADDRESS
2357  001716  010337  012542      NEWP20: MOV     R3,BUFREL       ;SAVE REL CONST
```

```
2358
2359                                    ;
2360                                    ;       GET TYPE  OF TEST TO BE RUN
2361                                    ;               D = 2848 RESPONDER DIAGNOSTIC
2362                                    ;               F = FRIEND
2363                                    ;
2364                                    ;               DEFAULT = 'F', FRIEND
2365                                    ;
2366  001722  105037  012544      90$:    CLRB    TSTTYP          ;RESET TEST TYPE
2367  001726  004137  002644            JSR     R1,INOCT        ;GET TEST TYPE
2368  001732  013626                    .WORD   TESTMS
2369  001734  001764                    .WORD   100$            ;DEFAULT TO FRIEND
2370  001736  112737  000100  012546    MOVB    #EBCDSP,FILLCH  ;FOR 2848 SET FILL CHAR TO EBCDIC SPACE
2371  001744  120427  000104            CMPB    R4,#'D          ;D? --2848 RESPONDER DIAGNOSTIC --
2372  001750  001433                    BEQ     INIT            ;YES, 2848 TEST
2373  001752  120427  000106            CMPB    R4,#'F          ;F? -- FRIEND TEST --
2374  001756  001361                    BNE     90$             ;ILLEGAL ENTRY
2375  001760  105037  012546            CLRB    FILLCH          ;FRIEND MODE -- DEFAULT FILL CHAR TO NULL
2376  001764  105237  012544      100$:   INCB    TSTTYP          ;SET TEST TO FRIEND
2377
2378                                    ;
2379                                    ;       FRIEND TEST ONLY
2380                                    ;       DETERMINE IF SEPARATE INPUT / OUTPUT BUFFERS ARE TO BE USED
2381                                    ;               Y = YES, MAINTAIN SEPARATE INPUT / OUTPUT BUFFERS
2382                                    ;               N = NO, USE SAME BUFFER FOR INPUT AND OUTPUT
2383                                    ;               DEFAULT IS 'N', NO, USE SAME BUFFER FOR INPUT / OUTPUT
2384                                    ;
2385  001770  105037  012545      110$:   CLRB    IOBUF           ;0 = NO, 1 = YES
2386  001774  004137  002644            JSR     R1,INOCT        ;SEPARATE I/O BUFFERS?
2387  002000  013666                    .WORD   FIOMS
2388  002002  002040                    .WORD   INIT            ;DEFAULT TO NO
2389  002004  120427  000116            CMPB    R4,#'N          ;N? -- NO, SAME I/O BUFFER --
2390  002010  001413                    BEQ     INIT            ;IF EQ, USE SAME I/O BUFFER FOR INPUT AND OUTPUT
2391  002012  120427  000131            CMPB    R4,#'Y          ;Y? --YES, SEPARATE I/O BUFFERS--
2392  002016  001364                    BNE     110$            ;ERROR, REQUEST INPUT AGAIN
2393  002020  105237  012545            INCB    IOBUF           ;SET SEPARATE I/O BUFFER INDICATOR
2394
2395                                    ;
2396                                    ;       FRIEND TEST MODE ONLY
2397                                    ;       GET BUFFER FILL CHARACTER (HEX INPUT REQUIRED)
2398                                    ;               ANY VALUE WILL BE ACCEPTED
2399                                    ;
2400  002024  004137  002636            JSR     R1,INHEX        ;FILL CHARACTER
2401  002030  013730                    .WORD   FILLMS
2402  002032  002040                     WORD   INIT
2403  002034  110337  012546            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
                                ;
002040  012737  011704  000004  INIT:  MOV     #MTO,4          ;SET UP MEMORY TIME OUT TRAP
002046  013701  012530          MOV     UNADDR,R1       ;SET UP DX UNIBUS ADDRESSES
002052  012702  012476          MOV     #DXDS,R2
002056  012703  000015          MOV     #13.,R3         ;13 ADDRESSES (REGISTERS)
002062  010122                  10$:    MOV     R1,(R2)+        ;SET UP UNIBUS ADDRESS
002064  005721                  TST     (R1)+           ;INCR TO NEXT DX REGISTER
002066  005303                  DEC     R3              ;DONE?
002070  001374                  BNE     10$             ;NO, SET UP NEXT REGISTER
                                ;
                                ;       SET UP DX VECTOR ADDRESS
                                ;
002072  013701  012532          MOV     VECTAD,R1
002076  012721  005346          MOV     #DXISR,(R1)+    ;TRAP TO DX ISR
002102  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
                                ;
002106  005737  012542          TST     BUFREL          ;WAS BUFFER RELOC SPECIFIED?
002112  001003                  BNE     20$             ;YES
002114  012737  000020  012542  MOV     #20,BUFREL      ;NO, MAKE BUFFERS START AT 20000
002122  013737  012542  013112  20$:    MOV     BUFREL,PBUFA    ;SAVE PHYSICAL ADDRESS
002130  013737  012542  013114  MOV     BUFREL,VBUFA    ;SAVE VIRTUAL ADDRESS
002136  105737  012541          TSTB    MMRESP          ;WAS MEMORY MANAGEMENT SPECIFIED?
002142  001436                  BEQ     40$             ;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
                                ;
002144  012704  172340          MOV     #KISAR0,R4      ;I-SPACE PAR
002150  012705  172300          MOV     #KISDR0,R5      ;I-SPACE PDR
002154  005024                  CLR     (R4)+           ;VA 0-17777 = PA 0-17777
002156  012725  077406          MOV     #77406,(R5)+    ;64 BLOCKS, UNLIMITED ACCESS
002162  013703  013112          MOV     PBUFA,R3        ;PHYSICAL ADDR * 2-6
002166  006303                  ASL     R3
002170  006303                  ASL     R3
002172  006303                  ASL     R3
002174  010324                  30$:    MOV     R3,(P4)+        ;SET UP PA FOR VA 20000-157777
002176  012725  077406          MOV     #77406,(R5)+    ;64 BLOCKS, UNLIMITED ACCESS
002202  062703  000200          ADD     #200,R3         ;INCREMENT TO NEXT 4K BANK
002206  020427  172356          CMP     R4,#KISAR7      ;ALL BUFFER ADDRESSES SET UP?
002212  001370                  BNE     30$             ;NO, SET UP NEXT REGISTER
002214  012714  007600          MOV     #7600,(R4)      ;SET UP UNIBUS ADDRESS REGISTER
002220  012715  077406          MOV     #77406,(R5)     ;64 BLOCKS, UNLIMITED ACCESS
```

```
2460  002224  012737  000001  177572          MOV     #1,MMSR0        ;ENABLE MEMORY MANAGEMENT
2461  002232  012737  000020  013114          MOV     #20,VBUFA       ;TO 8K BANK OR 20000 AND UP
2462
2463                                          ;       START SETTING UP SPW TABLE
2464                                          ;               1 ENTRY PER DEVICE (256 DEVICES)
2465                                          ;               ENTRY DESCRIPTION
2466                                          ;                   ----FOR VALID DEVICE NUMBERS
2467                                          ;                       BITS 15-8 = OFFSET TO DST TABLE (PHYSICAL ADDR)
2468                                          ;                             7-0 = 0
2469                                          ;                   ----FOR INVALID DEVICE NUMBERS
2470                                          ;                       BITS 15-8 = 0
2471                                          ;                             7-0 = 2 -- UNIT CHECK
2472                                          ;
2473                                          ;       THIS TABLE IS REFERENCED ON EACH 360 ACTION TO DETERMINE
2474                                          ;               IF DEVICE NUMBER IS VALID. THIS AUTOMATICALLY DONE
2475                                          ;               BY THE DX CONTROL UNIT
2476                                          ;
2477  002240  013705  013112          40$:    MOV     PBUFA,R5        ;COMPUTE OFFSET PHYSICALLY
2478  002244  000305                          SWAB    R5              ;*1000
2479  002246  105005                          CLRB    R5
2480  002250  006305                          ASL     R5
2481  002252  010577  010226                  MOV     R5,@DXOS        ;OFFSET TO SPW TABLE
2482  002256  013701  013114                  MOV     VBUFA,R1        ;COMPUTE VIRT ADDR OF SPW TABLE
2483  002262  000301                          SWAB    R1              ;*1000
2484  002264  105001                          CLRB    R1
2485  002266  006301                          ASL     R1
2486  002270  010137  013126                  MOV     R1,STSPW        ;SAVE START OF SPW TABLE
2487  002274  010137  013116                  MOV     R1,PHYOFF       ;COMPUTE THE OFFSET FOR PHYSICAL ADDRESSES
2488  002300  160537  013116                  SUB     R5,PHYOFF       ;VERSES VIRTUAL ADDRESS - FOR MEM MANAGEMENT
2489  002304  062705  003000                  ADD     #3000,R5        ;COMPUTE THE OFFSET TO THE DST TABLE
2490  002310  010537  013130                  MOV     R5,DSTOFF       ;SAVE OFFSET TO DST TABLE
2491  002314  005000                          CLR     DEV             ;START AT DEVICE 0
2492  002316  120037  012534          50$.    CMPB    DEV,SDEV        ;IS DEVICE NUMBER VALID
2493  002322  002405                          BLT     60$             ;NO
2494  002324  120037  012536                  CMPB    DEV,EDEV
2495  002330  003002                          BGT     60$             ;NO
2496  002332  010521                          MOV     R5,(R1)+        ;VALID DEVICE DST OFFSET TO ENTRY
2497  002334  000402                          BR      70$
2498  002336  012721  000002          60$:    MOV     #UCHK,(R1)+     ;INVALID DEV # UNIT CHECK TO ENTRY
2499  002342  005200                  70$:    INC     DEV             ;TO NEXT DEVICE
2500  002344  020027  000400                  CMP     DEV,#256.       ;ALL DEVICES DONE?
2501  002350  001362                          BNE     50$             ;NO, SET UP SPW FOR NEXT DEVICE
2502
2503                                          ;       NEXT SET UP TUMBLE TABLE AND DUPLICATE TUMBLE TABLE
2504                                          ;
2505  002352  010137  013074                  MOV     R1,TTADDR       ;TUMBLE TABLE ADDRESS
2506  002356  010137  013072                  MOV     R1,TTPTR        ;TUMBLE TABLE FETCH POINTER
2507  002362  012702  001000                  MOV     #TTSIZE,R2      ;CLEAR T/T + DUPLICATE T/T (WORD POINTER)
2508  002366  005021                  80$:    CLR     (R1)+           ;CLEAR NEXT WORD
2509  002370  005302                          DEC     R2              ;DONE?
2510  002372  001375                          BNE     80$             ;NO, CLEAR NEXT WORD
2511
2512                                          ;       SET UP DST TABLE
2513                                          ;               THE DST TABLE IS USED TO VERIFY COMMANDS FROM THE
2514                                          ;               360, THIS IS DONE BY THE HARDWARE
2515                                          ;               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
                                ;
002374  012702  000013                  MOV     #13,R2          ;NUMBER OF VALID 360 COMMANDS
002400  012703  002622                  MOV     #VCMDTB,R3      ;VALID COMMAND TABLE
002404  112321                  INIT10: MOVB    (R3)+,(R1)+     ;TO DST TABLE
002406  005302                          DEC     R2              ;DONE?
002410  001375                          BNE     INIT10          ;NO, MOVE IN NEXT RESPONSE
002412  012702  000365                  MOV     #245.,R2        ;MOVE UNIT CHECK TO INVALID ENTRIES
002416  112721  000002          100$:   MOVB    #UCHK,(R1)+
002422  005302                          DEC     R2
002424  001374                          BNE     100$
                                ;
                                ;       COMPUTE MAX NUMBER OF DEVICES
                                ;
002426  013703  012536                  MOV     EDEV,R3
002432  163703  012534                  SUB     SDEV,R3
002436  005203                          INC     R3              ;START AT DEVICE NUMERO UNO
002440  110337  013101                  MOVB    R3,MAXDEV
002444  013737  012534  013122          MOV     SDEV,DEVCON     ;SET UP DEVICE NUMBER -1
002452  005337  013122                  DEC     DEVCON
002456  012737  000001  013110          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
                                ;
002464  010137  013076                  MOV     R1,SDEVTB       ;SAVE START OF DEVICE BUFFERS
002470  005000                          CLR     DEV             ;DEV # 0
002472  010103                  120$:   MOV     R1,DTAB         ;SAVE ADDR OF DEVICE STATUS TABLE
002474  012702  000420                  MOV     #272.,R2        ;CLEAR DEVICE STATUS TABLE + INPUT BUFFER
002500  005021                  122$:   CLR     (R1)+
002502  005302                          DEC     R2              ;DONE?
002504  001375                          BNE     122$            ;NO, CLEAR NEXT WORD
002506  010363  000006                  MOV     DTAB,SINBF(DTAB)
002512  062763  000076  000006          ADD     #62.,SINBF(DTAB) ;COMPUTE ADDRESS OF INPUT BUFFER
002520  010363  000010                  MOV     DTAB,SOUTB(DTAB)
002524  062763  001040  000010          ADD     #544.,SOUTB(DTAB) ;COMPUTE ADDRESS OF OUTPUT BUFFER
002532  012702  000740                  MOV     #DISPSZ,R2
```

```
2572  002536  113721  012546          125$:   MOVB    FILLCH,(R1)+    ;FILL OUTPUT/DISPLAY BUFFER
2573  002542  005302                          DEC     R2              ;DONE?
2574  002544  001374                          BNE     125$            ;NO
2575  002546  005200                          INC     DEV             ;HAVE ALL DEVICE BUFFERS BEEN SET UP?
2576  002550  120037  013101                  CMPB    DEV,MAXDEV
2577  002554  001346                          BNE     120$            ;NO, SET UP NEXT DEVICE BUFFERS
2578  002556  013705  013112                  MOV     PBUFA,R5        ;SET UP EXTENDED ADDRESS BITS
2579  002562  006205                          ASR     R5
2580  002564  006205                          ASR     R5
2581  002566  006205                          ASR     R5
2582  002570  006205                          ASR     R5
2583  002572  042705  177747                  BIC     #177747,R5      ;SAVE ONLY H.O. 2 BITS
2584  002576  010537  013124                  MOV     R5,XADDR        ;SAVE EXTENDED ADDRESS BITS FOR DX CONTROL REG
2585  002602  012737  000001  013144          MOV     #1,FTIMFL       ;SET FIRST TIME THROUGH FLAG
2586                                ;
2587                                ;       INITIALIZATION COMPLETE
2588                                ;               TELL OPERATOR WE ARE ALREADY TO GO
2589                                ;
2590                                ;       NOTE:   AT THIS POINT THE DX HAS NOT BEEN STARTED
2591                                ;               AND THE OPERATOR MUST TYPE R (RUN COMMAND)
2592                                ;               TO THE SHOW UNDER WAY
2593                                ;
2594  002610  004137  011504                  JSR     R1,MESG         ;TELL OPERATOR WE ARE READY TO GO
2595  002614  014316                          .WORD   RNMESG
2596  002616  000137  002732                  JMP     EXEC            ;GET THE SHOW ON THE ROAD
2597
2598                                ;       VALID COMMAND TABLE
2599                                ;
2600  002622      000               VCMDTB: .BYTE 0                 ;0 = TEST I/O
2601  002623      000                       .BYTE 0                 ;1 = WRITE BUFFER
2602  002624      000                       .BYTE 0                 ;2 = READ MANUAL INPUT
2603  002625      014                       .BYTE CE!DE             ;3 = NOP
2604  002626      000                       .BYTE 0                 ;4 = SENSE COMMAND
2605  002627      000                       .BYTE 0                 ;5 = WRITE LINE ADDRESS
2606  002630      000                       .BYTE 0                 ;6 = READ FULL BUFFER
2607  002631      000                       .BYTE 0                 ;7 = ERASE COMMAND
2608  002632      002                       .BYTE UCHK              ;10 = INVALID
2609  002633      002                       .BYTE UCHK              ;11 = INVALID
2610  002634      000                       .BYTE 0                 ;12 = SHORT READ MANUAL INPUT
2611          002636                        .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
                                ;
002636  012705  011616          INHEX:  MOV     #CHTB,R5        ;MOVE ADDRESS OF CONVERSION ROUTINE TO R5
002642  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
                                ;
002644  012705  011552          INOCT:  MOV     #COTB,R5        ;SET UP ADDRESS OF THE CONVERSION ROUTINE
002650  012102                  INR5:   MOV     (R1)+,R2        ;GET ADDRESS OF THE MESSAGE
002652  004737  011416          10$:    JSR     PC,PRMESG               ;PRINT THE DESIRED MESSAGE
002656  105037  013065                  CLRB    TCMACT          ;RESET ACTIVE FLAG
002662  105037  013066                  CLRB    TCMDAB          ;RESET ABORT FLAG
002666  105737  013066          30$:    TSTB    TCMDAB          ;COMMAND ABORT?
002672  001367                  BNE     10$             ;YES, REASK QUESTION
002674  105737  013065                  TSTB    TCMACT          ;WAS ENTRY COMPLETED?
002700  001772                  BEQ     30$             ;NO, WAIT
002702  012702  012652                  MOV     #TBUF,R2        ;SET UP ADDRESS OF BEG OF INPUT BUFFER
002706  004715                  JSR     PC,@R5          ;CONVERT INPUT TO BINARY
002710  005705                  TST     R5              ;LOOK FOR DEFAULT RESP -- C/R
002712  001005                  BNE     40$             ;NOT DEFAULT TAKE NORMAL RETURN
002714  120427  000015          CMPB    R4,#CR          ;ILLEGAL CHAR MUST BE A C/R
002720  001002                  BNE     40$             ;ITS NOT A DEFAULT
002722  011101                  MOV     (R1),R1         ;---TAKE THE DEFAULT RETURN
002724  000201                  RTS     R1
002726  005721          40$:    TST     (R1)+           ;INCR FOR NORMAL RETURN
002730  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.
                                ;
002732  004737  011342  EXEC:   JSR     PC,CRLF         ;PRINT CR/LF
002736  012702  000052          MOV     #'*,R2
002742  004737  011360          JSR     PC,PRINT2               ;PRINT * -- DENOTE COMMAND MODE
002746  012706  012650          MOV     #SSTACK,SP      ;RE-ESTABLISH PUSH STACK
002752  105037  013065          CLRB    TCMACT          ;CLEAR TELE CMD ACT
002756  105037  013066          CLRB    TCMDAB          ;CLEAR TELE CMD ABORT
002762  105037  013067          CLRB    LINECT          ;RESET LINE COUNTER
002766  105737  013104  10$:    TSTB    DXABFL          ;DID THE DX ABORT AN OPERATION ?
002772  001402                  BEQ     20$             ;NO, CONTINUE
002774  000137  003352          JMP     STOPDX          ;YES IT DID, PRINT THE DX REGISTERS
003000  105737  013065  20$:    TSTB    TCMACT          ;IS THERE A COMMAND TO EXECUTE
003004  001001                  BNE     30$             ;YES, EXECUTE IT
003006  000767                  BR      10$             ;NO, WAIT AGAIN IF NOTHING TO DO
                                ;
                                ;       THERE IS A TELETYPE COMMAND TO BE EXECUTED
                                ;
003010  012702  012652  30$:    MOV     #TBUF,R2        ;SET UP PTR TO START OF TELE BUFFER
003014  112203                  MOVB    (R2)+,R3        ;GET COMMAND IDENTIFIER
003016  042703  177400          BIC     #177400,R3      ;SAVE L.O. BYTE
003022  012704  003056          MOV     #TCMDTB,R4      ;SET UP PTR TO COMMAND TABLE
003026  020324  40$:    CMP     R3,(R4)+        ;DOES COMMAND MATCH TABLE ENTRY?
003030  001411                  BEQ     EXECMD          ;YES, WE GOT A MATCH - START EXECUTION
003032  022404                  CMP     (R4)+,R4        ;INCR TO NEXT COMMAND
003034  005714                  TST     (R4)            ;END OF TABLE?
003036  001373                  BNE     40$             ;NO, TEST NEXT ENTRY
                                ;
                                ;       COMMAND ERROR - NOTIFY OPERATOR WITH ? AND 'BELL''
                                ;
003040  012702  137607  CERR:   MOV     #137607,R2      ;PRINT ? AND 'BELL''
003044  004737  011360          JSR     PC,PRINT2
003050  000137  002732          JMP     EXEC            ;RETURN TO EXEC
                                ;
                                ;       EXECUTE COMMAND
                                ;
003054  000134  EXECMD: JMP     @(R4)+          ;EXECUTE COMMAND
                                ;
                                ;       TELETYPE COMMAND TABLE
                                ;
003056  000101  TCMDTB: .WORD   'A              ;A = ACCESS
003060  004062          .WORD   ACCESS
003062  000104          .WORD   'D              ;D = DUMP
003064  003444          .WORD   DUMP
```

```
 2716  003066  000105                    .WORD   'E              ;E = ENABLE DEVICE
 2717  003070  004234                    .WORD   ENABLE
 2718  003072  000106                    .WORD   'F              ;F = FILL
 2719  003074  003730                    .WORD   FILL
 2720  003076  000110                    .WORD   'H              ;H = HELP COMMAND
 2721  003100  004016                    .WORD   HELP
 2722  003102  000111                    .WORD   'I              ;I = INPUT
 2723  003104  004342                    .WORD   INPUT
 2724  003106  000113                    .WORD   'K              ;K = KILL
 2725  003110  004300                    .WORD   KILL
 2726  003112  000122                    .WORD   'R              ;R = RUN
 2727  003114  003124                    .WORD   RUN
 2728  003116  000123                    .WORD   'S              ;S = STOP
 2729  003120  003276                    .WORD   STOP
 2730  003122  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.
                                ;
003124  032777  001000  007350  RUN:    BIT     #DXONLN,@DXCS   ;IS DX ENABLED?
003132  001342                          BNE     CERR            ;YES, ERROR
003134  005077  007342                  CLR     @DXCS           ;INITIALIZE THE DX
003140  005277  007336                  INC     @DXCS           ;SET GO
003144  012700  000001                  MOV     #1,DEV          ;START CLEARING DEVICE TABLES
003150  004737  010270          10$:    JSR     PC,CDEVST               ;CLEAR DEV STATUS TABLE
003154  004737  010322                  JSR     PC,CSPWST               ;RESET THE APPR SPW STATUS ENTRY FOR THE DEVICE
003160  105063  000002                  CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
003164  105063  000017                  CLRB    SRDRQ(DTAB)     ;CLEAR THE READ REQUEST
003170  005063  000020                  CLR     SMINS(DTAB)     ;CLEAR THE START OF MANUAL INPUT
003174  005200                          INC     DEV             ;INCR TO NEXT DEVICE
003176  120037  013101                  CMPB    DEV,MAXDEV      ;ARE WE DONE
003202  003762                          BLE     10$             ;NO, DO NEXT DEVICE
003204  105037  013102                  CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
003210  105037  013106                  CLRB    CMDCHF          ;CLEAR COMMAND CHAINING FLAG
003214  105037  013104                  CLRB    DXABFL          ;CLEAR DX ABORT FLAG
003220  013701  013074                  MOV     TTADDR,R1       ;GET THE TUMBLE TABLE ADDRESS
003224  010137  013072                  MOV     R1,TTPTR        ;RESET THE SOFTWARE T/T POINTER
003230  012702  001000                  MOV     #TTSIZ,R2       ;SET UP CLEAR CONSTANT (WORD COUNTER)
003234  005021                  20$:    CLR     (R1)+           ;CLEAR T/T AND DUP T/T
003236  005302                          DEC     R2              ;ARE WE DONE?
003240  001375                          BNE     20$             ;NO, KEEP ON CLEARING
003242  012737  000001  013110          MOV     #1,MDEV         ;INIT THE DEVICE NUMBER FOR MUX
                                                                ; AND SEL EXECUTOR ROUTINES
003250  053777  013124  007224          BIS     XADDR,@DXCS     ;SET UP THE EXTENDED ADDRESS BITS
003256  052777  004000  007216          BIS     #BSYEN,@DXCS    ;SEL CHANNEL - SET BUSY ENABLE
003264  052777  001100  007210  30$:    BIS     #DXENB!DXONLN,@DXCS ;ENABLE THE DX
003272  000137  002732                  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.
                                ;
003276  111204                 STOP:   MOVB    (R2),R4         ;GET THE TYPE OF STOP INDICATED
003300  120427  000015                  CMPB    R4,#CR          ;IMMEDIATELY?
003304  001422                          BEQ     STOPDX          ;YES, DISABLE DX AND PRINT REGISTERS
003306  120427  000104                  CMPB    R4,#'D          ;D = AFTER NEXT DATA TRANSFER?
003312  001413                          BEQ     10$             ;YES, SET STOP FLAG
003314  120427  000105                  CMPB    R4,#'E          ;E = AFTER THE NEXT ENDING SEQUENCE
003320  001410                          BEQ     10$             ;YES, SET STOP FLAG
003322  120427  000111                  CMPB    R4,#'I          ;I = AFTER THE CHIS SEQUENCE
003326  001405                          BEQ     10$             ;YES, SET STOP FLAG
003330  120427  000120                  CMPB    R4,#'P          ;P = STOP ON PARITY ERROR??
003334  001402                          BEQ     10$             ;YES, SET STOP FLAG
003336  000137  003040                  JMP     CERR            ; ILLEGAL FORMAT -- GIVE ERROR
003342  110437  013100           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.
                                ;
003346  000137  002732                  JMP     EXEC            ;RETURN TO THE EXEC
                                ;
                                ;       STOP THE DX AND PRINT THE REGISTERS
                                ;               NOTE THE PRINT OUTS WILL BE IN OCTAL
                                ;
003352  042777  000100  007122  STOPDX: BIC     #DXENB,@DXCS    ;DISABLE THE DX
003360  004737  011342                  JSR     PC,CRLF         ;START AT NEW LINE
003364  004137  011504                  JSR     R1,MESG         ;PRINT "CURRENT DEVICE -- "
003370  014110                          .WORD   STPMES
003372  013702  013120                  MOV     CDEV,R2         ;CONVERT AND PRINT THE CURRENT
003376  004737  004772                  JSR     PC,HDMP         ;DEVICE NUMBER IN HEX
003402  012703  000015                  MOV     #13.,R3         ;PRINT THE 13 DX REGISTERS IN OCTAL
```

```
2824  003406  012701  012476                MOV     #DXDS,R1        ;STARTING POINT
2825  003412  013102                10$:    MOV     @(R1)+,R2       ;GET THE REGISTER CONTENTS
2826  003414  004737  004702                JSR     PC,OCTDMP               ;PRINT IN OCTAL
2827  003420  005303                        DEC     R3              ;ARE WE DONE
2828  003422  001373                        BNE     10$             ;NO, DUMP NEXT WORD
2829  003424  105037  013104                CLRB    DXABFL          ;YES, RESET THE ABORT FLAG
2830  003430  005077  007046                CLR     @DXCS           ;RESET THE DX
2831  003434  005277  007042                INC     @DXCS
2832  003440  000137  002732                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)
                                ;
003444  004737  005062  DUMP:   JSR     PC,GLIMIT               ;GET BUFFER LIMITS
003450  112204                  MOVB    (R2)+,R4        ;GET DUMP MODE A/E/O/H
003452  012705  004650          MOV     #ASCDMP,R5      ;SET UP FOR ASCII DUMP
003456  120427  000101          CMPB    R4,#'A          ;IS IT ASCII?
003462  001421                  BEQ     10$             ;YES, START DUMP
003464  012705  004604          MOV     #EBCDMP,R5      ;SET UP FOR EBCDIC DUMP
003470  120427  000105          CMPB    R4,#'E          ;IS IT EBCDIC?
003474  001414                  BEQ     10$             ;YES, CONTINUE DUMP
003476  012705  004766          MOV     #HEXDMP,R5      ;SET UP FOR HEX DUMP
003502  120427  000110          CMPB    R4,#'H          ; IS IT HEX?
003506  001407                  BEQ     10$             ;YES, CONTINUE DUMP
003510  012705  004702          MOV     #OCTDMP,R5      ;SET UP FOR OCTAL DUMP
003514  120427  000117          CMPB    R4,#'O          ;IS IT OCTAL?
003520  001402                  BEQ     10$             ;YES, CONTINUE DUMP
003522  000137  003040          JMP     CERR            ;ILLEGAL ENTRY -- ERROR
003526  010537  013136  10$:    MOV     R5,DMPADR       ;SAVE ADDRESS OF DUMP ROUTINE
003532  005700                  TST     R0              ;WAS THIS A TUMBLE TABLE DUMP?
003534  001043                  BNE     DTUMTB          ;YES, DUMP THE TUMBLE TABLE
003536  112204                  MOVB    (R2)+,R4        ;WAS A DEV # SPECIFIED
003540  120427  000054          CMPB    R4,#',          ;IS NEXT POSITION A COMMA
003544  001014                  BNE     50$             ;NO, DUMP GIVEN LIMITS
003546  004737  005314          JSR     PC,GDEV         ;GET THE DEVICE NUMBER -- IN HEX
                                ;
                                ;       COMPUTE RELOCATION CONSTANT FOR DEVICE
                                ;
003552  005004                  CLR     R4              ;RELOCATION CONSTANT
003554  005303          30$:    DEC     R3              ;DONE?
003556  001403                  BEQ     40$             ;YES, ADD TO START + END ADDRESSES
003560  062704  002000          ADD     #2000,R4        ;TO NEXT DEVICE TABLES
003564  000773                  BR      30$
003566  060437  013132  40$:    ADD     R4,SADDR        ;ADD RELOCAT TO START ADDRESS
003572  060437  013134          ADD     R4,EADDR        ;ADD RELOCAT TO END ADDRESS
003576  017702  007330  50$:    MOV     @SADDR,R2       ;GET WORD
003602  004777  007330          JSR     PC,@DMPADR              ;CONVERT AND DUMP IT
003606  105737  013103  60$:    TSTB    PCTR
003612  001375                  BNE     60$
```

```
2889  003614  062737  000002  013132          ADD     #2,SADDR        ;INCR TO NEXT WORD
2890  003622  023737  013132  013134          CMP     SADDR,EADDR     ;DUMP DONE
2891  003630  003003                          BGT     70$             ;YES, EXIT
2892  003632  105737  013066                  TSTB    TCMDAB          ;COMMAND ABORT?
2893  003636  001757                          BEQ     50$             ;NO, PRINT NEXT WORD
2894  003640  000137  002732          70$:    JMP     EXEC            ;YES, RETURN TO EXEC
2895
2896
2897
2898                                  ;
2899                                  ;       DUMP THE DUPLICATE TUMBLE TABLE IN REVERSE SEQUENCE
2900                                  ;
2901  003644  012700  000400          DTUMTB: MOV     #TTSIZE/2,R0    ;SET UP COUNTER TO DUMP ENTIRE TUMBLE TABLE
2902  003650  017702  007256          5$:     MOV     @SADDR,R2       ;GET STARTING ADDRESS
2903  003654  004777  007256                  JSR     PC,@DMPADR              ;PRINT THE CONTENTS
2904  003660  105737  013103          10$:    TSTB    PCTR            ;IS PRINT COMPLETE?
2905  003664  001375                          BNE     10$             ;NO, WAIT TILL DONE
2906  003666  032737  000777  013132          BIT     #TTSIZE-1,SADDR ;CHECK FOR WRAP AROUND
2907  003674  001003                          BNE     20$
2908  003676  062737  001000  013132          ADD     #TTSIZE,SADDR   ;WRAP AROUND TO TOP OF TABLE
2909  003704  162737  000002  013132  20$:    SUB     #2,SADDR        ;DECREMENT TO NEXT ENTRY
2910  003712  005300                          DEC     R0              ;HAS ENTIRE TUMBLE TABLE BEEN DUMPED?
2911  003714  001403                          BEQ     30$             ;YES, EXIT TO THE EXEC
2912  003716  105737  013066                  TSTB    TCMDAB          ;ARE WE TO ABORT?
2913  003722  001752                          BEQ     5$              ;NO, KEEP ON DUMPING
2914  003724  000137  002732          30$:    JMP     EXEC            ;YUP, BACK TO THE EXEC
```

```
2915                                            .SBTTL  BACKGROUND -- FILL COMMAND
2916                                    ;
2917                                    ;       FILL COMMAND
2918                                    ;
2919                                    ;       THE FILL COMMAND LOADS THE SPECIFIED BYTE
2920                                    ;       INTO THE GIVEN DATA AREA.
2921                                    ;
2922                                    ;       THE FOLLOWING SYNTAXES ARE AVAILABLE FOR THE FILL COMMAND:
2923                                    ;               FIN,XX,YY       FILL INPUT BUFFER FOR DEVICE YY WITH XX
2924                                    ;               FOT,XX,YY       FILL OUTPUT BUFFER FOR DEVICE YY WITH XX
2925                                    ;
2926                                    ;       WHERE:  XX IS THE FILL CHARACTER IN HEX
2927                                    ;               YY IS THE DEVICE ADDRESS IN HEX
2928                                    ;
2929  003730  004737  005062       FILL:   JSR     PC,GLIMIT               ;GET BUFFER LIMITS
2930  003734  004737  011616               JSR     PC,CHTB         ;GET THE FILL CHARCTER
2931  003740  110337  012546               MOVB    R3,FILLCH       ;SAVE FILL CHAR
2932  003744  004737  005314               JSR     PC,GDEV         ;GET THE DEVICE ADDRESS
2933                                    ;
2934                                    ;       COMPUTE RELOCATION FOR DEVICE
2935                                    ;
2936  003750  005004                       CLR     R4
2937  003752  005303               10$:    DEC     R3              ;DONE?
2938  003754  001403                       BEQ     20$             ;YES, ADD TO START AND END ADDR
2939  003756  062704  002000               ADD     #2000,R4
2940  003762  000773                       BR      10$
2941  003764  060437  013132       20$:    ADD     R4,SADDR        ;ADD RELOC CONST TO START
2942  003770  060437  013134               ADD     R4,EADDR        ;ADD RELOC CONST TO END ADDR
2943  003774  013701  013132               MOV     SADDR,R1
2944                                    ;
2945                                    ;       FILL BUFFER WITH SPECIFIED CAHARCTER
2946                                    ;
2947  004000  113721  012546       30$:    MOVB    FILLCH,(R1)+    ;FILL CHARACTER
2948  004004  020137  013134               CMP     R1,EADDR        ;DONE?
2949  004010  101773                       BLOS    30$             ;NOPE, FILL NEXT CHAR
2950  004012  000137  002732               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
                                        ;
004016  012701  014372          HELP:   MOV     #HELPMS,R1      ;SET UP ADDRESS OF HELP MESSAGE
004022  012702  002551                  MOV     #HELPLN,R2      ;LENGTH OF HELP MESSAGE
                                        ;
                                        ;       START OUTPUTTING THE HELP MESSAGE UNDER OUR CONTROL
                                        ;       SO THE COMMAND MAY BE ABORTED QUICKLY.
                                        ;
004026  112100                  10$:    MOVB    (R1)+,R0        ;GET BYTE TO OUTPUT
004030  123727  013103  000004  15$:    CMPB    PCTR,#4         ;MORE THEN FOUR CHARACTERS IN OUTPUT BUFFER??
004036  003374                          BGT     15$             ;  YES, WAIT TIL DOWN A LITTLE
004040  004737  011400                  JSR     PC,PCHAR                ;PRINT IT ON CONSOLE
004044  105737  013066                  TSTB    TCMDAB          ;HAS OPERATOR INDICATED A DESIRE TO STOP?
004050  001002                          BNE     20$             ;  YES, ABORT HELP MESSAGE
004052  005302                          DEC     R2              ;HAS ENTIRE MESSAGE BEEN OUTPUTTED??
004054  001364                          BNE     10$             ;  NO, OUTPUT ANOTHER BYTE
004056  000137  002732          20$:    JMP     EXEC            ;  YES, RETURN TO THE EXECUTIVE
```

```
2974                                            .SBTTL  BACKGROUND -- ACCESS COMMAND
2975                                    ;
2976                                    ;       ACCESS SPECIFIED LOCATIONS AND CHANGE IF DESIRED
2977                                    ;
2978                                    ;       THE ACCESS COMMAND IS A QUICK LOOK AND CHANGE
2979                                    ;       ROUTINE MAINLY USED FOR PROGRAM DEBUGGING.
2980                                    ;
2981                                    ;       BASICALLY THE FOLLOWING ACTIONS ARE PERMITTED:
2982                                    ;               AXXXXX  -- OPEN AND PRINT SPECIFIED OCTAL LOCATION
2983                                    ;                       [XXXXXX](C/R) -- CHANGE CURRENT LOCATION IF DATA
2984                                    ;                      SPECIFIED [XXXXXX] AND OPEN NEXT LOCATION
2985                                    ;               /       -- RETURN TO EXEC MODE
2986                                    ;
2987  004062  004737  011552    ACCESS: JSR     PC,COTB         ;GET THE START ADDRESS
2988  004066  005705                    TST     R5              ;WAS A VALID ADDRESS ENTERED?
2989  004070  001403                    BEQ     5$              ;NO, GIVE OPERATOR AN ERROR
2990  004072  032703  000001            BIT     #1,R3           ;WAS ADDRESS SPECIFIED A WORD ADDRESS?
2991  004076  001402                    BEQ     7$              ;YES, OPEN SPECIFIED LOCATION
2992  004100  000137  003040    5$:     JMP     CERR            ;NO, GIVE OPERATOR AN ERROR INDICATION
2993  004104  010337  013132    7$:     MOV     R3,SADDR        ;SAVE STARTING ADDRESS
2994  004110  013702  013132    10$:    MOV     SADDR,R2        ;GET OBJECT WORD
2995  004114  105037  013065            CLRB    TCMACT          ;CLEAR TELE ACTIVE FLAG
2996  004120  004737  004702            JSR     PC,OCTDMP               ;PRINT ADDRESS IN OCTAL
2997  004124  012702  020040            MOV     #'  ,R2          ;PRINT 2 SPACES
2998  004130  004737  011360            JSR     PC,PRINT2
2999  004134  017702  006772            MOV     @SADDR,R2       ;GET CONTENTS OF OBJECT LOCATION
3000  004140  004737  004706            JSR     PC,ODMP         ;PRINT CONTENTS IN OCTAL
3001  004144  012702  020040            MOV     #'  ,R2          ;PRINT 2 SPACES
3002  004150  004737  011360            JSR     PC,PRINT2
3003  004154  105737  013065    20$:    TSTB    TCMACT          ;ACTIVE COMMAND?
3004  004160  001775                    BEQ     20$             ;NO
3005  004162  012702  012652            MOV     #TBUF,R2        ;SET UP INPUT BUFFER ADDRESS
3006  004166  004737  011552            JSR     PC,COTB         ;WAS LOCATION CHANGED?
3007  004172  005705                    TST     R5              ;ANY CHANGE?
3008  004174  001007                    BNE     30$             ;YES, STORE IT
3009  004176  120427  000057            CMPB    R4,#'/          ;/, EXIT TO EXEC
3010  004202  001412                    BEQ     50$             ;YES, RETURN TO EXEC
3011  004204  120427  000015            CMPB    R4,#CR          ;CR, GO TO NEXT LOCATION?
3012  004210  001403                    BEQ     40$             ;YES, OPEN AND PRINT NEXT LOC.
3013  004212  000736                    BR      10$             ;ERROR, PRINT CONTENTS OF CURRENT LOC.
3014  004214  010377  006712    30$:    MOV     R3,@SADDR       ;CHANGE OPEN LOCATION
3015  004220  062737  000002  013132  40$:    ADD     #2,SADDR        ;OPEN NEXT LOCATION
3016  004226  000730                    BR      10$
3017  004230  000137  002732    50$:    JMP     EXEC            ;RETURN TO THE EXEC
```

```
3018                                            .SBTTL  BACKGROUND -- ENABLE DEVICE COMMAND
3019                                    ;
3020                                    ;       E = ENABLE DEVICE
3021                                    ;
3022                                    ;       THE ENABLE COMMAND TURNS THE DEVICE SPECIFIED INTO AN
3023                                    ;       ON-LINE MODE. THIS IS ONLY NECESSITATED BECAUSE A KILL
3024                                    ;       COMMAND WAS PERFORMED ON THE DEVICE IN QUESTION.
3025                                    ;
3026                                    ;       THE ENABLE COMMAND HAS THE FOLLOWING SYNTAX:
3027                                    ;               EXX     -- ENABLE DEVICE ADDRESS XX
3028                                    ;                          THE DEVICE ADDRESS (XX) MUST BE ENTERED IN HEX
3029                                    ;
3030  004234  004737  005314   ENABLE: JSR     PC,GDEV         ;GET THE DEVICE NUMBER
3031  004240  004737  010270           JSR     PC,CDEVST               ;CLEAR THE DEVICE STATUS TABLE
3032  004244  010005                   MOV     DEV,R5          ;COMPUTE THE ADDRESS OF THE SPW TABLE ENTRY
3033  004246  063705  013122           ADD     DEVCON,R5       ;COMPENSATE FOR OFFSET DEVICE ADDRESS
3034  004252  060505                   ADD     R5,R5
3035  004254  063705  013126           ADD     STSPW,R5
3036  004260  013715  013130           MOV     DSTOFF,(R5)     ;ENABLE THE DEVICE NUMBER
3037  004264  105063  000002           CLRB    SSENSE(DTAB)
3038  004270  105063  000016           CLRB    SONLF(DTAB)
3039  004274  000137  002732           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
                                ;
004300  004737  005314          KILL:   JSR     PC,GDEV         ;GET THE DEVICE NUMBER
004304  004737  010270                  JSR     PC,CDEVST
004310  112763  000001  000016          MOVB    #1,SONLF(DTAB)
004316  010005                          MOV     DEV,R5          ;COMPUTE THE ADDRESS OF THE SPW TABLE
004320  063705  013122                  ADD     DEVCON,R5       ;COMPONSATE FOR OFFSET DEVICE ADDRESS
004324  060505                          ADD     R5,R5
004326  063705  013126                  ADD     STSPW,R5
004332  012715  000002                  MOV     #UCHK,(R5)      ;MAKE THE DEVICE OFF-LINE SEND UNIT CHECK
004336  000137  002732          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
                                ;
004342  105737  012544          INPUT:  TSTB    TSTTYP          ;ILLEGAL ON FRIEND TEST
004346  001005                          BNE     10$             ;FRIEND -- GIVE AN ERROR
004350  004737  005314                  JSR     PC,GDEV         ;GET THE DEVICE NUMBER
004354  120427  000054                  CMPB    R4,#',          ;THE NEXT CHAR MUST BE A COMMA
004360  001402                          BEQ     20$             ;IT IS, CONTINUE
004362  000137  003040          10$:    JMP     CERR            ;AN ERROR WAS FOUND GIVE INDICATION
004366  004737  010242          20$:    JSR     PC,SUDEV        ;SET UP THE DEVICE STATUS TABLE POINTERS
004372  026327  000004  000734        CMP     SCURS(DTAB),#DISPSZ-4 ;ARE WE AT THE END OF THE BUFFER?
004400  002370                          BGE     10$             ;YES, GIVE AN ERROR
004402  032763  000001  000004        BIT     #1,SCURS(DTAB)  ;START INPUT ON EVEN BYTE ADDRESS
004410  001002                          BNE     30$             ;START SOM ON ODD BYTE ADDRESS
004412  005263  000004                  INC     SCURS(DTAB)     ;INCR CURSOR TO ODD BYTE ADDRESS
004416  016305  000010          30$:    MOV     SOUTB(DTAB),R5  ;COMPUTE STARTING ADDRESS
004422  066305  000004                  ADD     SCURS(DTAB),R5  ;
004426  112725  000112                  MOVB    #SMI,(R5)+      ;START CHARACTER TO BUFFER
004432  010563  000020                  MOV     R5,SMINS(DTAB)  ;SAVE START OF DATA LOCATION
004436  005263  000004          40$:    INC     SCURS(DTAB)     ;INCREMENT CURSOR POSITION
004442  026327  000004  000735        CMP     SCURS(DTAB),#DISPSZ-3 ;ARE WE AT THE END OF BUFFER
004450  001423                          BEQ     70$             ;YES, TERMINATE INPUT
004452  112204                          MOVB    (R2)+,R4        ;GET NEXT INPUTTED CHARACTER
004454  042704  177600                  BIC     #177600,R4      ;SAVE L.O. 7 BITS
004460  020427  000015                  CMP     R4,#CR          ;END OF INPUT?
004464  001415                          BEQ     70$             ;YES, SET UP TO EXIT
004466  020427  000040                  CMP     R4,#SPACE       ;CAN CHARACTER BE CONVERTED?
004472  002410                          BLT     60$             ;NO, MUST BE BETWEEN 40 - 137
004474  020427  000137                  CMP     R4,#'_
004500  003005                          BGT     60$             ;NO, MUST BE BETWEEN 40 - 137
004502  162704  000040                  SUB     #SPACE,R4       ;SCALE DOWN FOR INDEXING
004506  116425  012366          50$:    MOVB    ATOETB(R4),(R5)+;CONVERT CHARACATER AND MOVE TO DISPLAY BUFFER
004512  000751                          BR      40$             ;GET AND CONVERT NEXT CHARACATER
004514  005004                  60$:    CLR     R4              ;ILLEGAL CHARACTER -- TREAT AS SPACE
004516  000773                          BR      50$
                                ;
                                ;       SET UP TO EXIT
                                ;               SET EOM INDICATOR
                                ;               QUEUE READ MANUAL INPUT REQUEST
                                ;
004520  112715  000152          70$:    MOVB    #EOM,(R5)       ;SET EOM INDICATOR
004524  005263  000004                  INC     SCURS(DTAB)     ;INCREMENT CURSOR POINTER
004530  105263  000017                  INCB    SRDRQ(DTAB)     ;QUEUE READ REQUEST
                                ;
                                ;       SEE IF THE DX IS CURRENTLY ACTIVE
                                ;
004534  105737  013102                  TSTB    DXACT           ;IS DX ACTIVE?
```

```
3119  004540  001402                        BEQ     80$             ;NO, START ASYNCHRONOUS PROCESSING TO SEND ATTENTION
3120  004542  000137  002732                JMP     EXEC            ;YES, ATTENTION WILL BE TAKEN CARE OF BY DX
3121                                ;
3122                                ;       DX IS NOT ACTIVE
3123                                ;               CREATE PHONY INTERRUPT TO ALLOW DX TO ACT ON ATTENTION
3124                                ;
3125  004546  013746  177776       80$:    MOV     PSW,-(SP)       ;PSW TO PUSH STACK
3126  004552  012746  002732               MOV     #EXEC,-(SP)     ;RETURN ADDRESS TO PUSH STACK
3127  004556  012737  000340  177776       MOV     #340,PSW        ;INHIBIT INTERUPTS
3128  004564  010046                       MOV     R0,-(SP)        ;SET UP PUSH STACK FOR FAKE INTERRUPT
3129  004566  010146                       MOV     R1,-(SP)
3130  004570  010246                       MOV     R2,-(SP)
3131  004572  010346                       MOV     R3,-(SP)
3132  004574  010446                       MOV     R4,-(SP)
3133  004576  010546                       MOV     R5,-(SP)
3134  004600  000137  006566               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
                                ;
004604  010237  013070          EBCDMP: MOV     R2,WK           ;SAVE WORD TO BE PRINTED
004610  113703  013070                  MOVB    WK,R3           ;GET LO BYTE
004614  042703  177400                  BIC     #177400,R3
004620  116337  011766  013070          MOVB    EBCDTB(R3),WK   ;CONVERT EBCDIC TO ASCII
004626  113703  013071                  MOVB    WK1,R3          ;GET HI BYTE AND CONVERT
004632  042703  177400                  BIC     #177400,R3
004636  116337  011766  013071          MOVB    EBCDTB(R3),WK1  ;CONVERT CHAR TO ASCII
004644  013702  013070                  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
                                ;
004650  105737  013067          ASCDMP: TSTB    LINECT          ;NEW LINE?
004654  001005                  BNE     10$             ;NO
004656  004737  011342                  JSR     PC,CRLF         ;YES, PRINT CR/LF
004662  112737  000036  013067          MOVB    #30.,LINECT     ;60 CHARACTERS PER LINE
004670  105337  013067          10$:    DECB    LINECT          ;DECR LINE COUNTER
004674  004737  011360                  JSR     PC,PRINT2               ;PRINT 2 CHARS
004700  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
                                ;
004702  004737  011342          OCTDMP: JSR     PC,CRLF         ;GIVE A CRLF
```

```
3191  004706  010046                ODMP:   MOV     R0,-(SP)        ;SAVE IMPORTANT REGISTERS
3192  004710  010246                        MOV     R2,-(SP)
3193  004712  010446                        MOV     R4,-(SP)
3194  004714  012704  000006              MOV     #6,R4           ;EXTRACT 6 OCTAL DIGITS
3195  004720  005000                      CLR     R0              ;CLEAR THE WORKING REGISTER
3196  004722  006102                      ROL     R2              ;MOVE HIGH ORDER BIT TO C-BIT
3197  004724  006100                10$:  ROL     R0              ;GET THE REMAINING BIT STILL IN LINK
3198  004726  042700  177770            BIC     #177770,R0      ;ONLY 3 LOW ORDER BITS
3199  004732  062700  000060            ADD     #'0,R0          ;MAKE ASCII
3200  004736  004737  011400            JSR     PC,PCHAR                ;PRINT IT ON THE TTY
3201  004742  006102                      ROL     R2              ;ROTATE THE NEXT OCTAL CHAR INTO POSITION
3202  004744  006102                      ROL     R2
3203  004746  006102                      ROL     R2
3204  004750  010200                      MOV     R2,R0           ;DATA TO WORKING REGISTER
3205  004752  005304                      DEC     R4              ;ARE WE DONE?
3206  004754  001363                      BNE     10$             ;NO, PRINT ANOTHER CHARACTER
3207  004756  012604                OCTEX:  MOV     (SP)+,R4        ;RESTORE USED REGISTERS
3208  004760  012602                        MOV     (SP)+,R2
3209  004762  012600                        MOV     (SP)+,R0
3210  004764  000207                        RTS     PC                      ;RETURN TO THE CALLER
3211
3212
3213                                ;
3214                                ;       DUMP WORD IN HEX ON THE TTY
3215                                ;
3216                                ;       CALLING SEQUENCE
3217                                ;       ......R2 CONTAINS THE WORD TO BE PRINTED
3218                                ;       JSR     PC,HEXDMP OR HDMP
3219                                ;       ........RETURN
3220                                ;
3221                                ;       HEXDMP PERFORMS A CRLF BEFORE OUTPUTTING THE DATA
3222                                ;
3223                                ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
3224                                ;
3225  004766  004737  011342        HEXDMP: JSR     PC,CRLF         ;DO A CR LF
3226  004772  010046                HDMP:   MOV     R0,-(SP)        ;SAVE THE WORKING REGISTERS
3227  004774  010246                        MOV     R2,-(SP)
3228  004776  010446                        MOV     R4,-(SP)
3229  005000  012704  000004              MOV     #4,R4           ;4 CHARACTERS PER WORD
3230  005004  006102                10$:  ROL     R2              ;ROTATE HIGH ORDER 4 BITS TO LOW ORDER 4 BITS
3231  005006  006102                      ROL     R2
3232  005010  006102                      ROL     R2
3233  005012  006102                      ROL     R2
3234  005014  010200                      MOV     R2,R0           ;TO WORKING REG
3235  005016  006100                      ROL     R0              ;GET THE LINK BIT TOO
3236  005020  042700  177760            BIC     #177760,R0      ;ONLY LOW ORDER 4 BITS
3237  005024  062700  000060            ADD     #'0,R0          ;MAKE ASCII IF NUMBER
3238  005030  020027  000071            CMP     R0,#'9          ;SHOULD IT BE A-F?
3239  005034  003402                      BLE     20$             ;NO,SHIP IT
3240  005036  062700  000007            ADD     #7,R0           ;YES, MAKE ALPHA
3241  005042  004737  011400        20$:  JSR     PC,PCHAR                ;PRINT THE HEX CHARACTER
3242  005046  005304                      DEC     R4              ;ARE WE DONE?
3243  005050  001355                      BNE     10$             ;NO, CONVERT AND PRINT NEXT CHARCATER
3244  005052  012604                      MOV     (SP)+,R4        ;YES, RESTORE REGISTERS AND EXIT
3245  005054  012602                      MOV     (SP)+,R2
3246  005056  012600                      MOV     (SP)+,R0
```

```
  3247  005060  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
                                ;
005062  005000          GLIMIT: CLR     R0              ;RESET BUFFER TYPE
005064  004737  011552          JSR     PC,COTB         ;GET FIRST PARAMETER
005070  005705                  TST     R5              ;WAS AN OCTAL NUMBER ENTERED?
005072  001014                  BNE     GLOCT           ;YES, OCTAL PARAMS
005074  120427  000124          CMPB    R4,#'T          ;T = TUMBLE TABLE
005100  001425                  BEQ     GLMTT           ;YES, SET UP T/T LIMITS
005102  120427  000111          CMPB    R4,#'I          ;I = INPUT BUFFER
005106  001436                  BEQ     GLMIN           ;YES, SET UP INPUT BUFFER LIMITS
005110  120427  000117          CMPB    R4,#'O          ;O = OUTPUT BUFFER
005114  001453                  BEQ     GLMOT           ;YES, SET UP OUTPUT BUFFER LIMITS
                                ;
                                ;       ERROR DETECTED - POP OFF RETURN ADDR AND GIVE ERROR
                                ;
005116  012601          GLERR:  MOV     (SP)+,R1
005120  000137  003040          JMP     CERR
                                ;
                                ;       OCTAL LIMITS SPECIFIED
                                ;
005124  010337  013132  GLOCT:  MOV     R3,SADDR        ;SAVE START ADDR
005130  120427  000054          CMPB    R4,#',          ;CHECK FOR COMMA (,)
005134  001370                  BNE     GLERR
005136  004737  011552          JSR     PC,COTB         ;GET END ADDR
005142  005705                  TST     R5              ;WAS SECOND PARAM GIVEN?
005144  001764                  BEQ     GLERR           ;NO, ERROR
005146  010337  013134          MOV     R3,EADDR        ;SAVE END ADDR
005152  000454                  BR      GLEX            ;PREPARE TO EXIT
                                ;
                                ;       SET UP LIMITS OF TUMBLE TABLE
                                ;
005154  112204          GLMTT:  MOVB    (R2)+,R4
005156  120427  000124          CMPB    R4,#'T          ;MUST BE TT
005162  001355                  BNE     GLERR           ;ILLEGAL ENTRY
005164  013737  013072  013132  MOV     TTPTR,SADDR
```

```
3304  005172  062737  000776  013132          ADD     #TTSIZE-2,SADDR ;COMPUTE ADDRESS OF APPR DUPLICATE TT ENTRY
3305  005200  005200                          INC     R0              ;INDICATE DUMP TUMBLE TABLE
3306  005202  000437                          BR      GLEX1           ;SET UP TO EXIT
3307
3308
3309                                  ;
3310                                  ;       SET UP LIMITS FOR INPUT BUFFER OF DEVICE 0
3311                                  ;
3312  005204  112204                  GLMIN:  MOVB    (R2)+,R4
3313  005206  120427  000116                  CMPB    R4,#'N          ;MUST BE IN
3314  005212  001341                          BNE     GLERR           ;ILLEGAL ENTRY
3315  005214  013704  013076                  MOV     SDEVTB,R4       ;GET ADDR OF DEV 0 STATUS TABLE
3316  005220  016437  000006  013132          MOV     SINBF(R4),SADDR
3317  005226  013737  013132  013134          MOV     SADDR,EADDR
3318  005234  062737  000741  013134          ADD     #DISPSZ+1,EADDR ;DISPLAY SIZE + ROOM FOR LINE ADDRESS
3319  005242  000417                          BR      GLEX1
3320
3321
3322                                  ;
3323                                  ;       SET UP LIMITS FOR OUTPUT BUFFER OF DEVICE 0
3324                                  ;
3325  005244  112204                  GLMOT:  MOVB    (R2)+,R4
3326  005246  120427  000124                  CMPB    R4,#'T          ;MUST BE OT
3327  005252  001321                          BNE     GLERR           ;ILLEGAL ENTRY
3328  005254  013704  013076                  MOV     SDEVTB,R4       ;GET ADDR OF DEV 0 STATUS TABLE
3329  005260  016437  000010  013132          MOV     SOUTB(R4),SADDR ;COMPUTE STARTING AND ENDING ADDRESSESS OF SPECIFIED BUF
3330  005266  013737  013132  013134          MOV     SADDR,EADDR
3331  005274  062737  000737  013134          ADD     #DISPSZ-1,EADDR ;DISPLAY CHAR BUFFER
3332  005302  112204                  GLEX1:  MOVB    (R2)+,R4        ;GET NEXT INPUT CHARACTER AND UPDATE POINTER
3333                                  ;
3334                                  ;       INSURE ENTRY PROPERLY DELIMITED AND RETURN TO CALLER
3335                                  ;
3336  005304  120427  000054          GLEX:   CMPB    R4,#',          ;CHECK FOR ,
3337  005310  001302                          BNE     GLERR           ;ENTRY NOT PROPERLY DELIMITED (ERROR)
3338  005312  000207                          RTS     PC
```

```
3339
3340                                    ;       GDEV -- GET THE THE DEVICE NUMBER FROM THE HEX INPUT
3341                                    ;
3342                                    ;       CALLING SEQUENCE
3343                                    ;       .........R2 = ADDRESS OF DEVICE ADDRESS IN HEX
3344                                    ;       JSR     PC,GDEV
3345                                    ;       ........RETURN IF NO ERRORS DETECTED
3346                                    ;               IF ERROR DETECTED, COMMAND IS ABORTED BY GOING
3347                                    ;                       TO "CERR"
3348                                    ;       UPON VALID RETURN
3349                                    ;               R3 AND DEV (R0) WILL CONTAIN THE DEVICE ADDRESS
3350                                    ;                       SCALED TO 1 - 8, NOTATION USED BY SYSTEM.
3351                                    ;               R2 WILL POINT TO THE NEXT CHARACTER FOLLOWING DEVICE ADDRESS
3352                                    ;
3353  005314  004737  011616  GDEV:    JSR     PC,CHTB         ;CONVERT THE HEX TO BINARY
3354  005320  163703  012534            SUB     SDEV,R3         ;-STARTING ADDRESS
3355  005324  100406                    BMI     10$             ;ERROR ON INPUT
3356  005326  005203                    INC     R3              ;MAKE BETWEEN 1 AND 8
3357  005330  120337  013101            CMPB    R3,MAXDEV       ;IS DEVICE NUMBER TOO BIG?
3358  005334  101002                    BHI     10$             ;YES, GIVE ERROR
3359  005336  010300                    MOV     R3,DEV          ;SET UP THE DEVICE NUMBER
3360  005340  000207                    RTS     PC
3361  005342  000137  003040  10$:      JMP     CERR            ;INPUT PARAM ERROR
3362                                    .SBTTL  DX11-B ISR (INTERRUPT REQUEST LOGIC AND TUMBLE TABLE DECODE LOGIC)
3363
3364                                    ;       D X 1 1 - B   I S R
3365                                    ;
3366                                    ;       DX11 ISR AND RELATED SUBROUTINE REGISTER USAGE
3367                                    ;
3368                                    ;       R0      DEV     DEVICE NUMBER
3369                                    ;       R1              UN ASSIGNED
3370                                    ;       R2              UNASSIGNED
3371                                    ;       R3      DTAB    ADDRESS OF CURRENT DEVICE TABLE
3372                                    ;       R4      TT1     TUMBLE TABLE ENTRY 1
3373                                    ;       R5      TT2     TUMBLE TABLE ENTRY 2
3374                                    ;
3375                                    ;       THE ABOVE REGISTER DESIGNATIONS REPRESENT WHAT USUALLY WILL
3376                                    ;       BE CONTAINED IN A REGISTER DURING DX ISR PROCESSING. HOWEVER,
3377                                    ;       AS SITUATIONS DICTATE REGISTERS MAY BE USED FOR DIFFERENT
3378                                    ;       PURPOSES.
3379                                    ;
3380  005346  010046          DXISR:    MOV     R0,-(SP)        ;SAVE HARDWARE REGISTERS
3381  005350  010146                    MOV     R1,-(SP)
3382  005352  010246                    MOV     R2,-(SP)
3383  005354  010346                    MOV     R3,-(SP)
3384  005356  010446                    MOV     R4,-(SP)
3385  005360  010546                    MOV     R5,-(SP)
3386  005362  013702  013072            MOV     TTPTR,R2        ;CHECK FOR ZERO T/T ENTRY UPON INTERRUPT
3387  005366  005712                    TST     (R2)
3388  005370  001002                    BNE     LOOP            ;NON-ZERO -- WERE OK
3389                                    ;
3390                                    ;       NOTE -- AN INTERRUPT OCCURRED WITHOUT A TUMBLE TABLE
3391                                    ;               ENTRY, THE ASSUMPTION IS THEN MADE THAT THE TUMBLE TABLE
3392                                    ;               ENTRY HAS ALREADY BEEN PROCESSED
3393                                    ;
3394  005372  000137  010224            JMP     DXEXIT
```

```
3395                                    ;
3396                                    ;       PROCESS TUMBLE TABLE ENTRIES
3397                                    ;               FOR CONVIENCE THE PROCESSING IS BEING PERFORMED
3398                                    ;               AT THE INTERRUPT LEVEL. IT IS SUGGESTED THAT IN
3399                                    ;               NORMAL PROCESSING ENVIRONMENTS THIS PROCESSING
3400                                    ;               BE DISTRIBUTED TO LESS PRIVILEDGED PRIORITY LEVELS
3401                                    ;               SUCH AS THE FORK  LEVEL IN RSX11-M.
3402                                    ;
3403                                    ;               THE INTERRUPT SERVICE LEVEL PROCESSING REQUIRED
3404                                    ;               BY THE DX11-B IS TO RELEIVE THE INTERRUPT (DONE BIT
3405                                    ;               IN DXCS) AND SCHEDULE A REQUEST FOR PROCESSING
3406                                    ;               AT ANOTHER LEVEL. THE LEVEL SCHEDULED TO PERFORM
3407                                    ;               THE PROCESSING SHOULD BE HIGH ENOUGH
3408                                    ;               TO PROTECT AGAINST TUMBLE TABLE OVERFLOW.
3409                                    ;
3410                                    ;       THE TUMBLE TABLE ENTRIES ARE PROCESSED SEQUENTIALLY FROM THE
3411                                    ;       CIRCULAR BUFFER FILLED BY THE DX. AS EACH ENTRY IS RETRIEVED
3412                                    ;       FROM THE TUMBLE TABLE IT IS ZEROED. IT IS THIS MECHANISM
3413                                    ;       THAT ALLOWS THE PROGRAMMER TO DISCERN WHEN ALL ENTRIES HAVE
3414                                    ;       BEEN PROCESSED. WHEN ALL ENTRIES HAVE BEEN
3415                                    ;       RETRIEVED FROM THE TUMBLE TABLE THEN THE NEXT ACTION
3416                                    ;       IS PERFORMED TO THE DX. THE DX11-B NEVER ENTERS A ZERO
3417                                    ;       IN TUMBLE TABLE ENTRY 1.
3418                                    ;
3419  005376  013702  013072   LOOP:   MOV     TTPTR,R2        ;GET T/T PTR
3420  005402  005712                   TST     (R2)            ;ANY ENTRIES LEFT IN T/T?
3421  005404  001002                   BNE     10$
3422  005406  000137  006566           JMP     DXEXEC          ;NO, EXECUTE NEXT DX COMMAND
3423                                    ;
3424                                    ;       TUMBLE TABLE ENTRY AVAILABLE FOR PROCESSING
3425                                    ;               RESET THE DONE BIT (RELIEVE INTERRUPT)
3426                                    ;               COPY TUMBLE TABLE ENTRY TO DUPLICATE TUMBLE TABLE (FOR SYSTEM TESTS PURP
3427                                    ;               RESET TUMBLE TABLE ENTRY (2 WORDS)
3428                                    ;
3429  005412  042777  000200  005062  10$:   BIC     #DONE,@DXCS     ;CLEAR DONE
3430  005420  010203                   MOV     R2,R3           ;SET UP PTR TO DUP T/T
3431  005422  062703  001000           ADD     #TTSIZE,R3
3432  005426  011223                   MOV     (R2),(R3)+      ;SAVE T/T ENTRY #1
3433  005430  011204                   MOV     (R2),TT1
3434  005432  005022                   CLR     (R2)+           ;CLEAR T/T ENTRY #1
3435  005434  011223                   MOV     (R2),(R3)+      ;SAVE T/T ENTRY #2
3436  005436  011205                   MOV     (R2),TT2
3437  005440  005022                   CLR     (R2)+           ;CLEAR T/T ENTRY #2
3438                                    ;
3439                                    ;       CHECK FOR POINTER WRAP AROUND
3440                                    ;
3441  005442  032702  000777           BIT     #TTSIZE-1,R2    ;AT END OF BUFFER?
3442                                                            ;NOTE -- POWER OF 2 BOUNDRY
3443  005446  001002                   BNE     20$             ;NO
3444  005450  013702  013074           MOV     TTADDR,R2       ;YES, RESET PTR
3445  005454  010237  013072   20$:    MOV     R2,TTPTR        ;SAVE T/T PTR
3446                                    ;
3447                                    ;       START PROCESSING TUMBLE ENTRY ENTRY
3448                                    ;               SAVE DEVICE ADDRESS
3449                                    ;               CHECK FOR SYSTEM RESET
3450                                    ;               VALIDATE DEVICE ADDRESS
```

```
3451                                        ;
3452                                        ;       NOTE -- IF SYSTEM RESET OCCURRED, THERE IS NO GUARANTEE
3453                                        ;               THAT THE DEVICE ADDRESS WILL BE VALID.
3454                                        ;
3455  005460  010500                       ;       MOV     TT2,DEV         ;GET DEV #
3456  005462  042700  177400               ;       BIC     #177400,DEV
3457  005466  010037  013120               ;       MOV     DEV,CDEV        ;SAVE CURRENT DEVICE NUMBER
3458  005472  042777  000200  005002       ;       BIC     #DONE,@DXCS     ;CLEAR DONE
3459  005500  032704  010000               ;       BIT     #SYSRST,TT1     ;SYSTEM RESET?
3460  005504  001066                       ;       BNE     PSYSRT          ;YES, PERFORM SYSTEM RESET FUNCTION
3461  005506  163700  012534               ;       SUB     SDEV,DEV        ;GET IN 0-7 RANGE - IF VALID
3462  005512  100403                       ;       BMI     30$             ;INVALID DEVICE NUMBER
3463  005514  120037  013101               ;       CMPB    DEV,MAXDEV      ;VALID DEVICE?
3464  005520  103405                       ;       BLO     40$             ;YES, NOT TOO BIG
3465                                        ;
3466                                        ;       INVALID DEVICE ADDRESS - BITCH
3467                                        ;
3468                                        ;       AN INVALID DEVICE ADDRESS WILL GENERALLY INDICATE
3469                                        ;       A PROBLEM IN THE CONFIGURATION OF DX DEVICE
3470                                        ;       ADDRESSES. BASICALLY THE DX HAS BEEN STRAPPED
3471                                        ;       TO HANDLE DEVICE ADDRESSES WHICH OVERLAP WITH
3472                                        ;       OTHER DEVICES ON THE CHANNEL.
3473                                        ;
3474  005522  004137  011520       30$:    JSR     R1,INMES        ;PRINT "INVALID DEVICE"
3475  005526  014054                       .WORD   ILLMES
3476  005530  000137  006550               JMP     DXAB            ;ABORT DX11
3477                                        ;
3478                                        ;       COMPUTE ADDRESS OF SPECIFIED DEVICES STATUS TABLE
3479                                        ;
3480  005534  005200               40$:    INC     DEV             ;MAKE DEVICE NUMBER 1 -8
3481  005536  004737  010242               JSR     PC,SUDEV        ;SET UP ADDR OF DEV STAT TABLE
```

```
3482                                    ;
3483                                    ;       DECODE DX TUMBLE TABLE STATUS ENTRY
3484                                    ;
3485                                    ;               THE FOLLOWING PROCESS INDICATES THE ORDER IN WHICH
3486                                    ;               THE TUMBLE TABLE STATUS ENTRY SHOULD BE DECODED.
3487                                    ;               THIS ORDER IS IMPORTANT AND SHOULD BE ADHERED
3488                                    ;               TO FOR MOST EMULATIONS.
3489                                    ;
3490  005542  032704  020000  TSELRST:BIT     #SELRST,TT1     ;SELECTIVE RESET?
3491  005546  001402                  BEQ     TINDSC          ;NO, TEST INTERFACE DISC.
3492  005550  000137  006010          JMP     PSELRT          ;YES, PERFORM SELECTIVE RESET
3493  005554  032704  004000  TINDSC: BIT     #INFDSC,TT1     ;INTERFACE DISCONNECT?
3494  005560  001402                  BEQ     TNXM            ;NO, CHECK NON-EXISTENT MEMORY
3495  005562  000137  006040          JMP     PINDSC          ;YES, PERFORM INTER DISC
3496  005566  032704  040000  TNXM:   BIT     #NXM,TT1        ;NON-EXISTENT MEMORY ERROR?
3497  005572  001402                  BEQ     TESEND          ;NO, ES END
3498  005574  000137  006114          JMP     PNXM            ;YES, PROCESS NON-EXISTENT MEMORY ERROR
3499  005600  032704  000100  TESEND: BIT     #ESEND,TT1      ;WAS STATUS ACCEPTED?
3500  005604  001402                  BEQ     TPARER          ;NO, CHECK FOR PARITY ERROR
3501  005606  000137  006126          JMP     PESEND          ;YES, PERFORM STATUS ACCEPT
3502  005612  032704  100000  TPARER: BIT     #PARER,TT1      ;DID A PARITY ERROR OCCUR?
3503  005616  001402                  BEQ     TCHIS           ;NO, CHECK FOR CHIS
3504  005620  000137  006256          JMP     PPARER          ;YES, PROCESS PARITY ERROR
3505  005624  032704  000200  TCHIS:  BIT     #CHIS,TT1       ;DID CHANNEL START A SELECTION SEQ?
3506  005630  001402                  BEQ     TCHEND          ;NO, CHECK FOR CHANNEL DATA END
3507  005632  000137  006306          JMP     PCHIS           ;YES, PROCESS SELECTION SEQUENCE
3508  005636  032704  000040  TCHEND: BIT     #CHEND,TT1      ;DID CHANNEL END OCCUR?
3509  005642  001402                  BEQ     TCUEND          ;NO, CHECK FOR CONTROL UNIT END
3510  005644  000137  006470          JMP     PCHEND          ;YES, PROCESS CHANNEL END
3511  005650  032704  000020  TCUEND: BIT     #CUEND,TT1      ;DID A CONTROL UNIT END OCCUR?
3512  005654  001650                  BEQ     LOOP            ;NO, IGNORE ENTRY -- ASSUME STACK STATUS
3513  005656  000137  006454          JMP     PCUEND          ;YES, PROCESS CONTROL UNIT END
```

```
3514                                            .SBTTL  DX11-B ISR (TUMBLE TABLE ENTRY PROCESSING LOGIC)
3515                                    ;
3516                                    ;       SYSTEM RESET OCCURRED FROM 360
3517                                    ;               CLEAR ALL DEVICE STATUS TABLES AND RESPECTIVE
3518                                    ;                       SENSE BYTES
3519                                    ;               RESET DX ACTIVE FLAGS AND COMMAND CHAIN FLAG
3520                                    ;
3521  005662  012700  000001           PSYSRT: MOV     #1,DEV          ;START AT FIRST DEVICE
3522  005666  004737  010270           10$:    JSR     PC,CDEVST       ;CLEAR DEVICE STATUS TABLE
3523  005672  004737  010322                   JSR     PC,CSPWST       ;RESET SPW STATUS WORD UPON SYSTEM RESET
3524  005676  105063  000002                   CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
3525  005702  105063  000017                   CLRB    SRDRQ(DTAB)     ;CLEAR THE READ REQUEST
3526  005706  005063  000020                   CLR     SMINS(DTAB)     ;CLEAR THE BEG OF MANUAL INPUT  ADDRESS
3527  005712  005063  000004                   CLR     SCURS(DTAB)     ;RESET THE CURSOR
3528  005716  105063  000003                   CLRB    SSTAT(DTAB)     ;CLEAR THE STATUS REGISTER
3529  005722  016301  000010                   MOV     SOUTB(DTAB),R1  ;SET UP TO CLEAR THE DISPLAY BUFFER
3530  005726  012702  000740                   MOV     #DISPSZ,R2      ;SET UP NUMBER OF CHARACTERS IN DISPLAY
3531  005732  112704  000100                   MOVB    #EBCDSP,R4      ;ASSUME 2848 DIAGNOSTIC TEST MODE
3532  005736  105737  012544                   TSTB    TSTTYP          ;WHAT TYPE OF TEST?
3533  005742  001402                           BEQ     20$             ;       IF 2848, USE EBCDIC SPACE
3534  005744  113704  012546                   MOVB    FILLCH,R4       ;       FRIEND TEST -- USE CURRENT FILL CHARACTER
3535
3536  005750  110421                  20$:    MOVB    R4,(R1)+        ;USE THE FILL CHARACTER
3537  005752  005302                           DEC     R2              ;ARE WE DONE?
3538  005754  001375                           BNE     20$             ;NO, LOOP TILL DONE
3539  005756  005200                           INC     DEV             ;TO NEXT DEVICE
3540  005760  120037  013101                   CMPB    DEV,MAXDEV      ;ARE WE DONE?
3541  005764  003740                           BLE     10$             ;NO, CLEAR NEXT DEV STAT TABLE
3542  005766  105037  013102                   CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3543  005772  105037  013106                   CLRB    CMDCHF          ;CLEAR COMMAND CHAINING FLAG
3544  005776  042777  000400  004476           BIC     #CUBUSY,@DXCS   ;RESET CU BUSY FLAG
3545  006004  000137  005376                   JMP     LOOP            ;PROCESS NEXT T/T ENTRY
3546
3547
3548                                    ;
3549                                    ;
3550                                    ;       CHANNEL ISSUED A SELECTIVE RESET
3551                                    ;               RESET THE DEVICE STATUS TABLE FOR THAT DEVICE + SENSE
3552                                    ;       NOTE: THE SEL RESET IS ISSUED AGAINST THE CURRENT
3553                                    ;               ACTIVE DEVICE
3554                                    ;
3555  006010  004737  010270           PSELRT: JSR     PC,CDEVST       ;CLEAR DEVICE STATUS TABLE
3556  006014  004737  010322                   JSR     PC,CSPWST       ;RESET SPW STATUS RESPONSE
3557  006020  105063  000002                   CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
3558  006024  105037  013102                   CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3559  006030  105037  013106                   CLRB    CMDCHF          ;CLEAR COMMAND CHAIN FLAG
3560  006034  000137  005376                   JMP     LOOP
3561
3562
3563                                    ;
3564                                    ;       INTERFACE DISCONNECT WAS ISSUED FROM THE 360
3565                                    ;               THIS IS DIRECTED TO A SPECIFIC DEVICE AND IS UNDER
3566                                    ;               360 PROGRAM CONTROL
3567                                    ;
3568                                    ;               IF THE DEVICE WAS ACTIVE
3569                                    ;                       ITS DEVICE STATUS TABLE WILL BE CLEARED
```

```
3570                                    ;                       AND CHAN END ! DEVICE END SET IN STATUS BYTE
3571                                    ;               IF THE DEVICE IS NOT ACTIVE THE COMMAND WILL BE IGNORED
3572                                    ;
3573  006040  004737  010322           PINDSC: JSR     PC,CSPWST       ;CLEAR THE SPW STATUS RESPONSE
3574  006044  105763  000000                   TSTB    SCMD(DTAB)      ;IS DEVICE ACTIVE?
3575  006050  001417                           BEQ     20$             ;NO, IGNORE
3576  006052  004737  010270                   JSR     PC,CDEVST       ;CLEAR THE DEVICE STATUS TABLE
3577  006056  012763  000003  000000           MOV     #CEDE,SCMD(DTAB);QUE DEV END + CHAN END
3578  006064  120037  013102                   CMPB    DEV,DXACT       ;IS DEVICE USING DX NOW?
3579  006070  001002                           BNE     10$             ;NO
3580  006072  105037  013102                   CLRB    DXACT           ;YES, RELEASE DX
3581  006076  120037  013106           10$:    CMPB    DEV,CMDCHF      ;DOES DEVICE HAVE CMD CHAIN SPEC?
3582  006102  001002                           BNE     20$             ;NO, GET NEXT T/T ENTRY
3583  006104  105037  013106                   CLRB    CMDCHF          ;YES, CLEAR FLAG
3584  006110  000137  005376           20$:    JMP     LOOP            ;GET NEXT T/T ENTRY
3585
3586
3587                                    ;
3588                                    ;       A NON-EXISTANT MEMORY CONDITION OCCURRED
3589                                    ;               THIS WILL USUALLY TRAP OUT FIRST
3590                                    ;
3591  006114  004137  011520           PNXM:   JSR     R1,INMES        ;PRINT 'NON EX MEM''
3592  006120  014000                           .WORD   NXMMSG
3593  006122  000137  006550                   JMP     DXAB            ;ABORT DX AND RETURN TO EXEC
3594
3595
3596
3597                                    ;
3598                                    ;       THE LAST STATUS SENT TO THE 360 WAS ACCEPTED, CLEAR DX
3599                                    ;               ACTIVE FLAG
3600                                    ;       IF LAST OP WAS A WRITE PERFORM THE DISPLAY CONTROL ROUTINE
3601                                    ;
3602  006126  004737  010322           PESEND: JSR     PC,CSPWST       ;RESET THE SPW STATUS BYTE
3603  006132  105763  000001                   TSTB    SLCMD(DTAB)     ;DOES LAST COMMAND REQUIRE 2260 DISPLAY EMULATION?
3604  006136  001402                           BEQ     10$             ;NO
3605  006140  004737  010414                   JSR     PC,DISCTL       ;YES, FORMAT THE DISPLAY
3606  006144  126327  000000  000011   10$:    CMPB    SCMD(DTAB),#11  ;WAS ATTN ACCEPTED?
3607  006152  001003                           BNE     20$             ;NO, CONTINUE
3608  006154  112763  000002  000017           MOVB    #2,SRDRQ(DTAB)  ;YES, INDICATE 360 ACCEPTANCE
3609  006162  105063  000001           20$:    CLRB    SLCMD(DTAB)
3610  006166  105037  013102                   CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3611  006172  004737  010270                   JSR     PC,CDEVST       ;CLEAR THE DEVICE STATUS TABLE
3612  006176  032704  000004                   BIT     #CMDCHN,TT1     ;WAS COMMAND CHAINING SPECIFIED?
3613  006202  001402                           BEQ     30$             ;NO,
3614  006204  110037  013106                   MOVB    DEV,CMDCHF      ;YES, SAVE THE DEVICE NUMBER
3615  006210  123727  013100  000105   30$:    CMPB    DXSTPF,#'E      ;WAS STOP ON END SEQ SPEC?(SE)
3616  006216  001552                           BEQ     STPDX           ;YES, DISABLE THE DX
3617
3618  006220  032704  000010                   BIT     #ISSREJ,TT1     ;WAS AN ISS REJ DETECTED?
3619  006224  001412                           BEQ     50$             ;NO, EXIT
3620
3621                                    ;
3622                                    ;       INIT SELECTION SEQUENCE WAS REJECTED BY DX (FAST CU BUSY SEQUENCE)
3623                                    ;               IF FREIND TEST MODE -- QUEUE CONTROL UNIT END
3624                                    ;                       ON UNIT COMPLETING TRANSFER
3625                                    ;               IF 2848 DIAGNOSTIC TEST MODE -- QUEUE CONTROL UNIT END
```

```
3626                                    ;                       RESPONSE ON LOW ORDER CHANNEL ADDRESS
3627                                    ;
3628                                    ;       THE 2848 DEVICE EMULATION IS EXPECTED TO ISSUE
3629                                    ;       A CONTROL UNIT END ON THE LOW ORDER DEVICE ADDRESS
3630                                    ;       OF THE CONTROL UNIT.
3631                                    ;       MOST OTHER 360/370 DEVICES ARE EXPECTED TO ISSUE
3632                                    ;       CONTROL UNIT END ON THE DEVICE COMPLETING THE OPERATION.
3633                                    ;
3634  006226  105737  012544              TSTB    TSTTYP          ;PROCESS SEPARETLY IF FRIEND
3635  006232  001004                      BNE     40$             ;FRIEND --QUEUE CU END ON BUSY UNIT
3636  006234  012700  000001              MOV     #1,DEV          ;SET UP TO SEND CUE ON LOW ORDER CONTROLLER ADDR
3637  006240  004737  010242              JSR     PC,SUDEV        ;  X
3638  006244  112763  000010  000000 40$: MOVB    #QCUE,SCMD(DTAB);QUEUE CONTROL UNIT END
3639  006252  000137  005376        50$:  JMP     LOOP            ;LOOP BACK AND PROCESS NEXT TUMBLE TABLE ENTRY
3640
3641
3642                                    ;
3643                                    ;       PARITY ERROR WAS DETECTED
3644                                    ;
3645  006256  004137  011520        PPARER: JSR     R1,INMES        ;PRINT 'PARITY ERROR'
3646  006262  014030                      .WORD   PARMES
3647  006264  123727  013100  000120      CMPB    DXSTPF,#'P      ;STOP ON PARITY ERROR??
3648  006272  001524                      BEQ     STPDX           ;YES, DISABLE THE DX
3649  006274  152763  000002  000003      BISB    #UCHK,SSTAT(DTAB) ;SET UNIT CHECK IN STATUS WORD
3650  006302  000137  005624              JMP     TCHIS           ;CONTINUE WITH TUMBLE TABLE INTERROGATION
3651
3652
3653                                    ;
3654                                    ;       CHANNEL INITIATED SELECTION SEQUENCE
3655                                    ;               THUS FAR THE DEVICE NUMBER HAS BEEN VALIDATED
3656                                    ;               AND THE COMMAND CHECKED BY THE DX
3657                                    ;
3658                                    ;               TT2 CONTAINS THE COMMAND TO BE EXECUTED
3659                                    ;
3660  006306  004737  010322        PCHIS:  JSR     PC,CSPWST       ;RESET THE SPW STATUS BYTE
3661                                                                  ;ON NEXT CHANNEL INITIATED SELECTION SEQUENCE
3662  006312  123727  013100  000111      CMPB    DXSTPF,#'I      ;WAS STOP ON ISS SPECIFIED(SI)
3663  006320  001511                      BEQ     STPDX           ;YES, DISABLE DX
3664  006322  032704  000001              BIT     #CMDREJ,TT1     ;WAS COMMAND REJECTED BY DX?
3665  006326  001022                      BNE     20$             ;YES, COMMAND REJECTED BY THE DX
3666                                    ;
3667                                    ;       VALID COMMAND, SET UP TO PROCESS IT
3668                                    ;
3669  006330  105005                      CLRB    TT2             ;RESET DEVICE ADDRESS BITS
3670  006332  000305                      SWAB    TT2             ;COMMAND TO L.O. BYTE
3671  006334  105705                      TSTB    TT2             ;TEST I/O COMMAND?
3672  006336  001437                      BEQ     50$             ;YES, IGNORE
3673  006340  120527  000003              CMPB    TT2,#NOP        ;WAS COMMAND A NOP?
3674  006344  001434                      BEQ     50$             ;YES, IGNORE IT
3675  006346  020527  000012              CMP     TT2,#12         ;IS THIS A VALID COMMAND?
3676  006352  003405                      BLE     10$             ;YES, QUEUE TO BE EXECUTED
3677  006354  004137  011520              JSR     R1,INMES        ;NO -- REPORT AN ILLEGAL COMMAND RECIEVED FROM THE DX
3678  006360  014142                      .WORD   INVLDC
3679  006362  000137  006550              JMP     DXAB            ;AND  ABORT THE PROGRAM
3680  006366  110563  000000        10$:  MOVB    TT2,SCMD(DTAB)  ;QUEUE COMMAND TO BE PROCESSED
3681  006372  000421                      BR      50$             ;EXIT + PROCESS NEXT T/T ENTRY
```

```
3682                                    ;
3683                                    ;       COMMAND WAS REJECTED, DETERMINE WHY
3684                                    ;
3685  006374  105763  000016            20$:    TSTB    SONLF(DTAB)     ;IS DEVICE ON LINE?
3686  006400  001404                            BEQ     30$             ;YES, TEST PARITY ERROR
3687                                    ;
3688                                    ;       DEVICE OFF-LINE -- RESPOND INTERVENTION REQUIRED SENSE CONDITION
3689                                    ;
3690  006402  052763  000100  000002            BIS     #INTREQ,SSENSE(DTAB) ;SET INTERVENTION REQUIRED IN SENSE BYTE
3691  006410  000412                            BR      50$             ;FINISH UP CHANNEL INITIATED SELECTION PROCESS
3692  006412  032704  100000            30$:    BIT     #PARER,TT1      ;WAS A PARITY ERROR DETECTED?
3693  006416  001404                            BEQ     40$             ;NO, MUST BE ILLEGAL COMMAND
3694                                    ;
3695                                    ;       COMMAND WAS REJECTED BECAUSE OF A PARITY ERROR
3696                                    ;               SET UP BUS OUT SENSE RESPONSE
3697                                    ;
3698  006420  052763  000040  000002            BIS     #BUSOUT,SSENSE(DTAB) ;SET BUS OUT FLAG
3699  006426  000403                            BR      50$             ;EXIT
3700                                    ;
3701                                    ;       INVALID COMMAND RECEIVED FROM 360
3702                                    ;               SET UP COMMAND REJECT SENSE RESPONSE
3703                                    ;
3704  006430  052763  000200  000002    40$:    BIS     #SCMDRJ,SSENSE(DTAB) ;SET CMD REJ FLAG
3705                                    ;
3706                                    ;       CHANNEL INITIATED SELECTION SEQUENCE COMMON EXIT LOGIC
3707                                    ;               IF SELECTED DEVICE HAS COMMAND CHAINING IN AFFECT -- KILL IT
3708                                    ;
3709  006436  120037  013106            50$:    CMPB    DEV,CMDCHF      ;DOES DEVICE HAVE COMMAND CHAINING SPECIFIED?
3710  006442  001002                            BNE     60$             ;NO, GET NEXT TUMBLE TABLE ENTRY
3711  006444  105037  013106                    CLRB    CMDCHF          ;YES, CLEAR THE COMMAND CHAINING FLAG
3712  006450  000137  005376            60$:    JMP     LOOP            ;AND GET THE NEXT T/T ENTRY
3713
3714
3715                                    ;
3716                                    ;       CONTROL UNIT END OF DATA TRANSFER WAS DETECTED
3717                                    ;               IF TRANSFER COMPLETE PREPARE ENDING SEQ RESP
3718                                    ;               IF TRANSFER INCOMPLETE INCR BUFFER ADDRESS
3719                                    ;                                       DECR BYTE COUNT
3720                                    ;
3721  006454  105037  013102            PCUEND: CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3722  006460  004737  010354                    JSR     PC,MUXEND       ;HANDLE MUX DATA TRANSFER COMPLETION
3723  006464  103017                            BCC     PCHEX           ;IF SEL CHAN OR MUX D/T NOT DONE, MERELY EXIT
3724                                    ;
3725                                    ;       MUX DATA TRANSFER COMPLETE
3726                                    ;               TREAT SAME AS SEL CHANNEL DONE
3727                                    ;
3728  006466  000404                            BR      PCHEN1
3729
3730
3731
3732                                    ;
3733                                    ;       CHANNEL END WAS DETECTED
3734                                    ;               PREPARE ENDING SEQUENCE RESPONSE
3735                                    ;
3736  006470  105037  013102            PCHEND: CLRB    DXACT           ;CLEAR DX ACTIVE FLAG
3737  006474  004737  010354                    JSR     PC,MUXEND       ;IF MUX CHANNEL HANDLE DATA TRANSFER
```

```
3738  006500  123727  013100  000104  PCHEN1: CMPB    DXSTPF,#'D      ;STOP ON DATA TRANSFER DONE?(SD)
3739  006506  001416                          BEQ     STPDX           ;YES, DISABLE DX
3740  006510  112763  000003  000000          MOVB    #CEDE,SCMD(DTAB);QUE END SEQ RESPONSE
3741  006516  167763  003766  000014          SUB     @DXBC,SRBYTC(DTAB) ;SAVE REMAINING BYTE COUNT
3742  006524  032704  100000          PCHEX:  BIT     #PARER,TT1      ;WAS A PARITY ERROR SENSED?
3743  006530  001403                          BEQ     10$             ;NO, PROCESS NEXT TUMBLE TABLE ENTRY
3744  006532  152763  000020  000002          BISB    #EQPCHK,SSENSE(DTAB) ;YES, SET EQUIP CHECK IN SENSE
3745  006540  000137  005376          10$:    JMP     LOOP            ;LOOP BACK + PROCESS NEXT TT ENTRY
3746
3747
3748
3749                                  ;
3750                                  ;       STOP DX, SOMEONE WANTS TO INVESTIGATE
3751                                  ;
3752  006544  105037  013100          STPDX:  CLRB    DXSTPF          ;CLEAR STOP FLAG
3753
3754
3755                                  ;
3756                                  ;       ABORT   CURRENT DX OPERATION
3757                                  ;               UNRECOVERABLE ERROR OCCURRED
3758                                  ;
3759  006550  042777  001100  003724  DXAB:   BIC     #DXONLN!DXENB,@DXCS ;DISABLE THE DX
3760  006556  105237  013104                  INCB    DXABFL          ;SET THE DX ABORT FLAG SO THE
3761                                                                  ;DX REGISTERS WILL BE PRINTED
3762  006562  000137  010224                  JMP     DXEXIT          ;EXIT FROM INTERRUPT
3763                                  .SBTTL  DX11-B ISR (SELECTOR CHANNEL COMMAND EXECUTION)
3764                                  ;
3765                                  ;       EXECUTE NEXT COMMAND FOR THE DX
3766                                  ;
3767  006566  105737  012540          DXEXEC: TSTB    CHTYPE          ;CHANNEL TYPE 0=M, 1=S
3768  006572  001002                          BNE     SEX             ;SELECTOR CHANNEL EXEC
3769  006574  000137  007222                  JMP     MEX             ;MULTIPLEXER EXEC
3770                                  ;
3771                                  ;       SEX  -- SELECTOR CHANNEL EXECUTIVE
3772                                  ;
3773                                  ;               SEX EXECUTES COMMANDS FOR THE DX TO A SELECTOR CHANNEL
3774                                  ;
3775                                  ;               ON A SELECTOR CHANNEL A COMMAND WILL BE COMPLETED
3776                                  ;                       BEFORE ATTEMPTING TO EXECUTE A COMMAND ON ANOTHER
3777                                  ;                       DEVICE, ISS-DATA TRANSFER-ES.
3778                                  ;
3779                                  ;               DATA TRANSFERS ARE COMPLETED IN ONE BURST
3780                                  ;
3781  006600  013700  013110          SEX:    MOV     SELDEV,DEV      ;GET SEL DEV #
3782  006604  004737  010242          10$:    JSR     PC,SUDEV        ;SET UP DEV STATUS TABLE ADDR
3783  006610  105763  000000                  TSTB    SCMD(DTAB)      ;ANY JOB TO DO?
3784  006614  001030                          BNE     60$             ;YES, EXECUTE IT
3785  006616  105737  013106                  TSTB    CMDCHF          ;WAS COMMAND CHAINING SPECIFIED
3786  006622  001402                          BEQ     30$             ;NO
3787  006624  000137  010224                  JMP     DXEXIT          ;YES, WAIT FOR COMMAND
3788  006630  126327  000017  000001  30$:    CMPB    SRDRQ(DTAB),#1  ;IS ATTENTION TO BE SENT?
3789  006636  001004                          BNE     40$             ;NO, CONTINUE
3790  006640  112763  000011  000000          MOVB    #11,SCMD(DTAB)  ;YES, SET UP TO SEND THE ATTENTION
3791  006646  000413                          BR      60$             ;FOR THE READ MANUAL INPUT
3792  006650  005200                  40$:    INC     DEV             ;TO NEXT DEV
3793  006652  120037  013101                  CMPB    DEV,MAXDEV      ;HAVE WE TRIED THE HIGHEST DEVICE?
```

```
3794  006656  003402                        BLE     50$             ;NO
3795  006660  012700  000001                MOV     #1,DEV          ;YES, RESTART AT FIRST DEVICE
3796  006664  020037  013110        50$:    CMP     DEV,SELDEV      ;IS THIS WHERE IT ALL STARTED?
3797  006670  001345                        BNE     10$             ;NOPE, TEST THIS DEVICE
3798  006672  000137  010224                JMP     DXEXIT          ;EXIT -- NO TASKS PENDING
3799                                ;
3800                                ;       THERE IS A JOB TO DO, LETS DO IT
3801                                ;
3802  006676  116304  000000        60$:    MOVB    SCMD(DTAB),R4   ;COMMAND TO INDEX
3803  006702  005304                        DEC     R4              ;SCALE TO 0 - 11
3804  006704  006304                        ASL     R4              ;MAKE WORD ADDRESS
3805  006706  010037  013110                MOV     DEV,SELDEV      ;SAVE CURRENT DEVICE ADDR
3806  006712  000174  006716                JMP     @SCMDTB(R4)     ;EXECUTE THE COMMAND
3807  006716  006742                SCMDTB: .WORD   SWRITE          ;1 = WRITE FULL BUFFER
3808  006720  007032                        .WORD   SRMI            ;2 = READ MANUAL INPUT
3809  006722  010054                        .WORD   ESEQ            ;3 = ENDING SEQUENCE
3810  006724  010152                        .WORD   SENSCM          ;4 = SENSE COMMAND
3811  006726  006742                        .WORD   SWRITE          ;5 = WRITE LINE ADDRESS
3812  006730  007132                        .WORD   SREAD           ;6 = READ FULL BUFFER
3813  006732  010002                        .WORD   ERASCM          ;7 = ERASE COMMAND
3814  006734  007762                        .WORD   CONUNE          ;10 = CONTROL UNIT END
3815  006736  007772                        .WORD   SATTN           ;11 = SEND ATTENTION TO 360
3816  006740  007032                        .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
                                        ;
 006742 016377 000006 003536    SWRITE: MOV     SINBF(DTAB),@DXBA ;SET UP BUFFER ADDRESS
 006750 163777 013116 003530            SUB     PHYOFF,@DXBA    ;FOR VIRTUAL MEMORY -- OFFSET FOR PHYSICAL ADDRESS
 006756 013702 013122                   MOV     DEVCON,R2       ;COMPUTE DEVICE ADDRESS
 006762 060002                          ADD     DEV,R2
 006764 110277 003510                   MOVB    R2,@DXCA
 006770 012777 177037 003512            MOV     #-DISPSZ-1,@DXBC;SET UP BYTE COUNT FOR MAX, WRITE LINE ADDRESS
 006776 116363 000000 000001            MOVB    SCMD(DTAB),SLCMD(DTAB)  ;SET WRITE FLAG
 007004 105063 000002                   CLRB    SSENSE (DTAB)   ;CLEAR SENSE BYTE
 007010 110037 013102                   MOVB    DEV,DXACT       ;SET DX ACTIVE FLAG
 007014 005063 000014                   CLR     SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
 007020 052777 000003 003454            BIS     #DXWR,@DXCS     ;START TRANSFER
 007026 000137 010224                   JMP     DXEXIT          ;RETURN FROM INTERRUPT


                                        ;
                                        ;       PERFORM READ MANUAL INPUT COMMANDS
                                        ;
 007032                                 SRMI:
 007032 105737 012544                   SSRMI:  TSTB    TSTTYP          ;IS TEST FOR FRIEND?
 007036 001035                                  BNE     SREAD           ;YES, TREAT ALL READS AS READ FULL BUFFER
 007040 105063 000002                           CLRB    SSENSE(DTAB)    ;RESET THE SENSE BYTE
 007044 105763 000017                           TSTB    SRDRQ(DTAB)     ;WAS A READ REQUESTED?
 007050 001002                                  BNE     10$             ;YES, CONTINUE
 007052 000137 010054                           JMP     ESEQ            ;NO, TREAT AS A NOP -- END SEQ ONLY
 007056 105063 000017                   10$:    CLRB    SRDRQ(DTAB)
 007062 016377 000020 003416                    MOV     SMINS(DTAB),@DXBA ;SET UP STARTING ADDRESS
 007070 016302 000010                           MOV     SOUTB(DTAB),R2  ;DETERMINE ENDING ADDRESS
 007074 066302 000004                           ADD     SCURS(DTAB),R2
 007100 005302                                  DEC     R2
 007102 167702 003400                           SUB     @DXBA,R2        ;COMPUTE BYTE COUNT
 007106 100002                                  BPL     20$             ;INSURE VALID BYTE COUNT
 007110 000137 010054                           JMP     ESEQ            ;ILLEGAL
 007114 005402                          20$:    NEG     R2
 007116 010277 003366                           MOV     R2,@DXBC        ;SET UP DX'S BYTE COUNT
 007122 163777 013116 003356                    SUB     PHYOFF,@DXBA    ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
 007130 000411                                  BR      SRD10           ;START THE READ



                                        ;
                                        ;       READ COMMAND RECEIVED FROM 360
                                        ;               PREPARE TO SEND DISP BUFFER TO 360
                                        ;
 007132 016377 000010 003346    SREAD:  MOV     SOUTB(DTAB),@DXBA ;SET UP BUFFER ADDRESS
 007140 163777 013116 003340            SUB     PHYOFF,@DXBA    ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
```

```
3873  007146  012777  177040  003334          MOV     #-DISPSZ,@DXBC  ;SET UP BYTE COUNT
3874  007154  116363  000000  000001  SRD10:  MOVB    SCMD(DTAB),SLCMD(DTAB) ;SAVE CODE OF LAST COMMAND
3875  007162  013702  013122                  MOV     DEVCON,R2       ;COMPUTE DEVICE ADDRESS
3876  007166  060002                          ADD     DEV,R2
3877  007170  110277  003304                  MOVB    R2,@DXCA
3878  007174  105063  000002                  CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
3879  007200  110037  013102                  MOVB    DEV,DXACT       ;SET DX ACTIVE FLAG
3880  007204  005063  000014                  CLR     SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
3881  007210  052777  000005  003264          BIS     #DXRD,@DXCS     ;START TRANSFER
3882  007216  000137  010224                  JMP     DXEXIT          ;RETURN FROM INTERRUPT
3683
3884
```

```
                                        .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
                                ;
007222  105737  013106        MEX:    TSTB    CMDCHF          ;IS COMMAND CHAINING SPECIFIED?
007226  001402                        BEQ     10$             ;NO, CONTINUE
007230  000137  010224                JMP     DXEXIT          ;YES, LEAVE DX FREE
007234  013700  013110        10$:    MOV     MDEV,DEV        ;GET LAST DEVICE ADDR THAT HAD A COMMAND
007240  004737  010242        30$:    JSR     PC,SUDEV        ;COMPUTE ADDR OF DEV STAT TABLE
007244  105763  000000                TSTB    SCMD(DTAB)      ;ANY JOB TO DO?
007250  001023                        BNE     50$             ;YES, EXECUTE IT
007252  126327  000017  000001        CMPB    SRDRQ(DTAB),#1  ;IS ATTENTION REQUESTED?
007260  001004                        BNE     40$             ;NO, CONTINUE
007262  112763  000011  000000        MOVB    #11,SCMD(DTAB)  ;YES, QUEUE ATTENTION
007270  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.
                                ;
007272  005200                40$:    INC     DEV             ;INCR TO NEXT DEVICE NUMBER
007274  120037  013101                CMPB    DEV,MAXDEV      ;WAS DEVICE NUMBER WRAPPED AROUND?
007300  003402                        BLE     45$             ;NO, SEE IF ALL DEVICES HAVE BEEN INTERROGATED
007302  012700  000001                MOV     #1,DEV          ;YES, RESET THE DEVICE NUMBER
007306  020037  013110        45$:    CMP     DEV,MDEV        ;NO JOB HERE, HAVE WE CHECKED ALL DEVICES?
007312  001352                        BNE     30$             ;NO, EXAMINE NEXT DEVICE
007314  000137  010224                JMP     DXEXIT          ;YES, EXIT FROM ISR
                                ;
                                ;       THIS DEVICE HAS A JOB TO DO, EXECUTE IT
                                ;
007320  116304  000000        50$:    MOVB    SCMD(DTAB),R4   ;COMMAND TO INDEX
007324  005304                        DEC     R4              ;SCALE TO 0 - 11
007326  006304                        ASL     R4              ;MAKE INTO WORD ADDRESS
007330  010037  013110                MOV     DEV,MDEV        ;SAVE CURRENT DEVICE ADDRESS
007334  000174  007340                JMP     @MCMDTB(R4)     ;EXECUTE THE COMMAND
007340  007364                MCMDTB: .WORD   MWRITE          ;1 = WRITE FULL BUFFER
007342  007514                        .WORD   MRMI            ;2 = READ MANUAL INPUT
007344  010054                        .WORD   ESEQ            ;3 = ENDING SEQUENCE
007346  010152                        .WORD   SENSCM          ;4 = SENSE COMMAND
007350  007364                        .WORD   MWRITE          ;5 = WRITE LINE ADDRESS
007352  007612                        .WORD   MREAD           ;6 = READ FULL BUFFER
007354  010002                        .WORD   ERASCM          ;7 = ERASE COMMAND
007356  007762                        .WORD   CONUNE          ;10 = CONTROL UNIT END
007360  007772                        .WORD   SATTN           ;11 = SEND ATTENTION TO 360
007362  007514                        .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
                                        ;
 007364  005763  000014                 MWRITE: TST     SRBYTC(DTAB)     ;WRITE IN PROGRESS?
 007370  001011                                 BNE     10$              ;YES, SEND OUT MORE DATA
 007372  016363  000006  000012                 MOV     SINBF(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
 007400  163763  013116  000012                 SUB     PHYOFF,SBUFA(DTAB) ; FOR MEM MANG - OFFSET FOR PHY ADDRESS
 007406  012763  000741  000014                 MOV     #DISPSZ+1,SRBYTC(DTAB) ;SET UP BUFFER FOR MAX SIZE
 007414  016377  000012  003064 10$:    MOV     SBUFA(DTAB),@DXBA ;OUTPUT BUFFER ADDR TO DX
 007422  013702  013122                         MOV     DEVCON,R2        ;COMPUTE DEVICE ADDRESS
 007426  060002                                 ADD     DEV,R2
 007430  110277  003044                         MOVB    R2,@DXCA
 007434  012777  177774  003046                 MOV     #-4,@DXBC        ;START BYTE COUNT AT 4
 007442  026327  000014  000004                 CMP     SRBYTC(DTAB),#4 ;IS LESS THEN 4 BYTES LEFT?
 007450  002005                                 BGE     20$              ;NO, START TRANSFER
 007452  016302  000014                         MOV     SRBYTC(DTAB),R2 ;YES, USE REMAINING BYTE COUNT
 007456  005402                                 NEG     R2
 007460  010277  003024                         MOV     R2,@DXBC
 007464  105063  000002                 20$:    CLRB    SSENSE(DTAB)     ;CLEAR SENSE BYTE
 007470  116363  000000  000001                 MOVB    SCMD(DTAB),SLCMD(DTAB) ;SET WRITE FLAG
 007476  110037  013102                         MOVB    DEV,DXACT        ;SET ACTIVE FLAG
 007502  152777  000003  002772                 BISB    #DXWR,@DXCS      ;START THE TRANSFER
 007510  000137  010224                         JMP     DXEXIT           ;RETURN FROM INTERRUPT



                                        ;
                                        ;       READ MANUAL INPUT COMMAND
                                        ;
                                        ;       THIS WILL TRANSFER ONLY THE DATA ENTERED ON THE SCREEN TO THE 360
                                        ;
         007514                         MRMI    =       .
 007514  105737  012544                 MSRMI:  TSTB    TSTTYP           ;FRIEND OR 2848 DIAG?
 007520  001034                                 BNE     MREAD            ;FRIEND -- TREAT AS READ FULL BUFFER
 007522  005763  000014                         TST     SRBYTC(DTAB)     ;ANY DATA LEFT TO TRANSFER?
 007526  001031                                 BNE     MREAD            ;BRANCH IF YES TO CONTINUE
 007530  105063  000002                         CLRB    SSENSE(DTAB)     ;RESET THE SENSE BYTE
 007534  105763  000017                         TSTB    SRDRQ(DTAB)      ;WAS THE READ REQUESTED?
 007540  001002                                 BNE     20$              ;YES, CONTINUE
 007542  000137  010054                 10$:    JMP     ESEQ             ;NO, RETURN AN ENDING SEQ RESP DE!CE
 007546  105063  000017                 20$:    CLRB    SRDRQ(DTAB)      ;CLEAR THE READ REQUEST
 007552  016363  000020  000012                 MOV     SMINS(DTAB),SBUFA(DTAB) ;SET UP THE ADDRESS OF THE DATA
 007560  016302  000010                         MOV     SOUTB(DTAB),R2  ;COMPUTE THE BYTE COUNT
 007564  066302  000004                         ADD     SCURS(DTAB),R2  ;END - START
 007570  005302                                 DEC     R2
 007572  166302  000012                         SUB     SBUFA(DTAB),R2  ;COMPUTE THE BYTE COUNT
 007576  100761                                 BMI     10$              ;NEGATIVE -- SOMETHING IS WRONG
 007600  010263  000014                         MOV     R2,SRBYTC(DTAB) ;SAVE FOR READ DRIVER
 007604  163763  013116  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
                                        ;
007612  116363  000000  000001  MREAD:  MOVB    SCMD(DTAB),SLCMD(DTAB) ;SAVE CODE OF LAST COMMAND FOR DISPLAY CONTROL
007620  005763  000014                  TST     SRBYTC(DTAB)    ;READ IN PROGRESS?
007624  001011                          BNE     10$             ;YES, SEND OUT MORE DATA
007626  016363  000010  000012          MOV     SOUTB(DTAB),SBUFA(DTAB) ;SET UP BUFFER ADDRESS
007634  163763  013116  000012          SUB     PHYOFF,SBUFA(DTAB) ;FOR MEM MANAG - OFFSET FOR PHY ADDRESS
007642  012763  000740  000014          MOV     #DISPSZ,SRBYTC(DTAB) ;SET UP TOTAL BYTE COUNT
007650  016377  000012  002630  10$:    MOV     SBUFA(DTAB),@DXBA ;SEND BUFFER ADDR TO DX
007656  013702  013122                  MOV     DEVCON,R2       ;COMPUTE DEVICE ADDR
007662  060002                          ADD     DEV,R2
007664  110277  002610                  MOVB    R2,@DXCA        ;OUTPUT THE DEVICE ADDRESS
007670  012777  177774  002612          MOV     #-4,@DXBC       ;OUTPUT THE BYTE COUNT -4-
007676  026327  000014  000004          CMP     SRBYTC(DTAB),#4 ;SEE IF REMAINING BYTE COUNT LESS THAN 4
007704  002005                          BGE     20$
007706  016302  000014                  MOV     SRBYTC(DTAB),R2 ;SET UP BYTE COUNT
007712  005402                          NEG     R2
007714  010277  002570                  MOV     R2,@DXBC        ;OUTPUT THE NEW BYTE COUNT -- LT 4
007720  105063  000002          20$:    CLRB    SSENSE(DTAB)    ;CLEAR SENSE AND SET DX ACTIVE FLAG
007724  110037  013102                  MOVB    DEV,DXACT       ;SET DEVICE ACTIVE FLAG FOR SOFTWARE
                                        ;
                                        ;       BEFORE TRANSMIT IS STARTED SET BUSY FLAG IN DX11 STATUS
                                        ;       TABLE FOR DEVICE
                                        ;
007730  010002                  MOV     DEV,R2          ;COMPUTE ADDRESS OF SPW ENTRY
007732  063702  013122          ADD     DEVCON,R2       ; X
007736  060202                  ADD     R2,R2           ; X
007740  063702  013126          ADD     STSPW,R2        ;ADD IN SPW BASE ADDRESS
007744  052712  000020          BIS     #BSY,(R2)       ;SET UNIT BUSY FLAG
007750  152777  000005  002524  BISB    #DXRD,@DXCS     ;START THE DX READING
007756  000137  010224          JMP     DXEXIT
```

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


                                ;
                                ;       PRESENT CONTROL UNIT END TO CHANNEL
                                ;
007762  152763  000040  000003  CONUNE: BISB    #CUE,SSTAT(DTAB) ;PUT IN STATUS BYTE
007770  000434                          BR      STOUT           ;OUTPUT TO CHANNEL


                                ;
                                ;       SEND THE ATTENTION BIT TO THE 360
                                ;
007772  152763  000200  000003  SATTN:  BISB    #ATTN,SSTAT(DTAB) ;PUT IN STATUS BYTE
010000  000430                          BR      STOUT           ;OUTPUT TO THE 360


                                ;
                                ;       ERASE THE DISPLAY
                                ;
010002  016304  000010          ERASCM: MOV     SOUTB(DTAB),R4  ;SET UP BEG OF DISPLAY BUFFER
010006  012705  000740                  MOV     #DISPSZ,R5      ;SET UP COUNTER
010012  112702  000100                  MOVB    #EBCDSP,R2      ;SET BUFFER FILL FOR 2848 DIAG
010016  105737  012544                  TSTB    TSTTYP          ;IS TEST BEING RUN FOR 2848 RESPONDER
010022  001402                          BEQ     10$             ;YES, FILL BUFFER WITH EBCDIC SPACE
010024  113702  012546                  MOVB    FILLCH,R2       ;NO, USE CURRENT FILL CHARACTER
010030  110224                  10$:    MOVB    R2,(R4)+        ;MOVE FILL CHARACTER TO BUFFER
010032  005305                          DEC     R5              ;DECR COUNTER
010034  001375                          BNE     10$             ;NOT DONE, DO NEXT CHAR
010036  005063  000004                  CLR     SCURS(DTAB)     ;RESET THE CURSOR
010042  105063  000002                  CLRB    SSENSE(DTAB)    ;CLEAR SENSE BYTE
010046  112763  000003  000000          MOVB    #CEDE,SCMD(DTAB);CHANGE COMMAND TO PRESENT END SEQ
                                ;
                                ;       FALL THROUGH TO PRESENT ENDING STATUS
                                ;


                                ;
                                ;       PRESENT ENDING STATUS TO CHANNEL
                                ;
010054  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.
                                ;
```

```
 4090  010062  013702  013122          STOUT:  MOV     DEVCON,R2       ;OUTPUT DEVICE ADDRESS
 4091  010066  060002                          ADD     DEV,R2
 4092  010070  110277  002404                  MOVB    R2,@DXCA
 4093  010074  132763  000002  000003          BITB    #UCHK,SSTAT(DTAB) ;IS THE UNIT CHECK BIT SET?
 4094  010102  001403                          BEQ     10$             ;NO,TRANSMIT THE STATUS
 4095  010104  112763  000002  000003          MOVB    #UCHK,SSTAT(DTAB) ;YES, THEN SEND ONLY UNIT CHECK
 4096                                  ;
 4097                                  ;       IF MULTIPLEXER CHANNEL
 4098                                  ;               CLEAR ANY PENDING STATUS IN SPW STATUS ENTRY
 4099                                  ;                       (PROBABLY 'BUSY')
 4100                                  ;
 4101  010112  105737  012540          10$:    TSTB    CHTYPE          ;SELECTOR CHANNEL?
 4102  010116  001004                          BNE     20$             ;YES, DON'T CLEAR STATUS IN SPW TABLE
 4103  010120  060202                          ADD     R2,R2           ;COMPUTE ADDRESS OF SPW STATUS ENTRY
 4104  010122  063702  013126                  ADD     STSPW,R2        ;OFFSET BY BASE OF SPW TABLE
 4105  010126  105012                          CLRB    (R2)            ;CLEAR SPW STATUS ENTRY
 4106                                  ;
 4107                                  ;       OUTPUT THE STATUS TO THE CHANNEL
 4108                                  ;
 4109  010130  116377  000003  002346  20$:    MOVB    SSTAT(DTAB),@DXOS ;OUTPUT STATUS TO CHANNEL
 4110  010136  152777  000007  002336          BISB    #DXST,@DXCS     ;PRESENT TO CHANNEL
 4111  010144  110037  013102                  MOVB    DEV,DXACT       ;SET DX ACTIVE FLAG
 4112  010150  000425                          BR      DXEXIT          ;RETURN FROM INTERRUPT
 4113
 4114
 4115                                  ;
 4116                                  ;       SENSE COMMAND DESIRED BY 360
 4117                                  ;
 4118  010152  012777  000002  002326  SENSCM: MOV     #SSENSE,@DXBA   ;SET UP ADDRESS OF SENSE BYTE
 4119  010160  060377  002322                  ADD     DTAB,@DXBA
 4120  010164  163777  013116  002314          SUB     PHYOFF,@DXBA    ;FOR MEMORY MANAGEMENT - OFFSET FOR PHY ADDRESS
 4121  010172  013702  013122                  MOV     DEVCON,R2       ;COMPUTE DEVICE ADDRESS
 4122  010176  060002                          ADD     DEV,R2
 4123  010200  110277  002274                  MOVB    R2,@DXCA
 4124  010204  012777  177777  002276          MOV     #-1,@DXBC       ;TRANSFER 1 BYTE
 4125  010212  110037  013102                  MOVB    DEV,DXACT       ;SET DX ACTIVE FLAG
 4126  010216  052777  000005  002256          BIS     #DXRD,@DXCS     ;START TRANSFER
 4127
 4128
 4129                                  ;
 4130                                  ;       EXIT FROM THE DX ISR
 4131                                  ;
 4132  010224  012605                  DXEXIT: MOV     (SP)+,R5        ;RESTORE REGISTERS
 4133  010226  012604                          MOV     (SP)+,R4
 4134  010230  012603                          MOV     (SP)+,R3
 4135  010232  012602                          MOV     (SP)+,R2
 4136  010234  012601                          MOV     (SP)+,R1
 4137  010236  012600                          MOV     (SP)+,R0
 4138  010240  000002                          RTI
```

```
4139                                            .SBTTL  DX11-B ISR (UTILITY SUBROUTINES)
4140                                    ;
4141                                    ;       SET UP ADDR OF DEVICE STATUS TABLE
4142                                    ;
4143                                    ;       CALLING SEQUENCE
4144                                    ;       ..... R0 = DEV #
4145                                    ;       JSR     PC,SUDEV
4146                                    ;       ..... RETURN
4147                                    ;             R3 = ADDRESS OF DEVICE TABLE
4148                                    ;
4149                                    ;       ONLY REGISTER R3 IS MODIFIED BY THIS SUBROUTINE
4150                                    ;
4151  010242  013703  013076            SUDEV:  MOV     SDEVTB,DTAB     ;START AT DEV 1
4152  010246  010146                            MOV     R1,-(SP)        ;SAVE R1
4153  010250  010001                            MOV     DEV,R1
4154  010252  005301                    SUD10:  DEC     R1              ;DEC DEVICE NUMBER
4155  010254  001403                            BEQ     SUDEX           ;DONE, EXIT
4156  010256  062703  002000                    ADD     #2000,DTAB      ;INCR TO NEXT DEV TABLE
4157  010262  000773                            BR      SUD10           ;TRY AGAIN
4158  010264  012601                    SUDEX:  MOV     (SP)+,R1        ;RETURN TO CALLER
4159  010266  000207                            RTS     PC
4160
4161
4162                                    ;
4163                                    ;       CLEAR DEVICE STATUS TABLE
4164                                    ;
4165                                    ;       CALLING SEQUENCE
4166                                    ;       ..... R0 = DEV #
4167                                    ;       JSR     PC,CDEVST
4168                                    ;       ..... RETURN
4169                                    ;             R3 = ADDRESS OF DEVICE TABLE
4170                                    ;                  THE FOLLOWING TABLE ENTRIES ARE CLEARED
4171                                    ;                  SCMD
4172                                    ;                  SSTAT
4173                                    ;                  SBUFA
4174                                    ;                  SRBYTC
4175                                    ;                  SLCMD
4176                                    ;
4177                                    ;       ONLY REGISTER R3 IS AFFECTED BY THIS SUBROUTINE
4178                                    ;
4179  010270  004737  010242            CDEVST: JSR     PC,SUDEV                ;SET UP ADDR OF DEVICE STAT TABLE
4180  010274  105063  000000                    CLRB    SCMD(DTAB)      ;RESET CURRENT COMMAND ENTRY
4181  010300  105063  000003                    CLRB    SSTAT(DTAB)     ;RESET DEVICE STATUS ENTRY
4182  010304  005063  000012                    CLR     SBUFA(DTAB)     ;RESET CURRENT BUFFER ADDRESS POINTER
4183  010310  005063  000014                    CLR     SRBYTC(DTAB)    ;RESET REMAINING BYTE COUNT
4184  010314  105063  000001                    CLRB    SLCMD(DTAB)     ;RESET LAST COMMAND ENTRY
4185  010320  000207                            RTS     PC                      ;RETURN TO THE CALLER
4186
4187
4188                                    ;
4189                                    ;       CSPWST -- CLEAR SPW STATUS BYTE
4190                                    ;
4191                                    ;       CALLING SEQUENCE
4192                                    ;       ........DTAB (R3) POINTS TO CURRENT DEVICE STATUS TABLE
4193                                    ;       ........DEV (R0) CONTAINS CURRENT DEVICE NUMBER
4194                                    ;       JSR     PC,CSPWST
```

```
4195                                    ;       ........RETURN TO CALLER WITH DEVICE STATUS BYTE RESET
4196                                    ;
4197                                    ;       ALL REGISTERS ARE PRESERVED ACCROSS THIS SUBROUTINE
4198                                    ;
4199  010322  010546                    CSPWST: MOV     R5,-(SP)        ;SAVE REGISTER FOR SUBROUTINE USAGE
4200  010324  105763  000016                    TSTB    SONLF(DTAB)     ;IS DEVICE ON-LINE?
4201  010330  001007                            BNE     10$             ;NO, JUST EXIT
4202  010332  010005                            MOV     DEV,R5          ;GET DEVICE NUMBER AND COMPUTE
4203  010334  063705  013122                    ADD     DEVCON,R5
4204  010340  060505                            ADD     R5,R5           ;ADDRESS OF SPW STATUS BYTE
4205  010342  063705  013126                    ADD     STSPW,R5
4206  010346  105015                            CLRB    (R5)            ;RESET SPW STATUS BYTE
4207  010350  012605                    10$:    MOV     (SP)+,R5        ;RESTORE REGISTER
4208  010352  000207                            RTS     PC
```

```
4209                                    ;
4210                                    ;       MUXEND -- HANDLE DATA TRANSFEP COMPLETIONS FOR MUX
4211                                    ;
4212                                    ;       CALLING SEQUENCE
4213                                    ;       ........R3 (DTAB) CONTAINS THE ADDRESS OF THE DEVICE STATUS TABLE
4214                                    ;       JSR     PC,MUXEND
4215                                    ;       ........RETURN  C-BIT SET   - MUX DATA TRANS DONE
4216                                    ;                       C-BIT RESET - SEL CHAN OR DATA TRANSFER NOT DONE
4217                                    ;
4218                                    ;       NO REGISTERS ARE AFFECTED BY THIS SUBROUTINE
4219                                    ;
4220  010354  105737  012540           MUXEND: TSTB    CHTYPE          ;SELECTOR OR MULTIPLEXER CHANNEL??
4221  010360  001006                           BNE     5$              ;SELECTOR CHANNEL -- EXIT
4222                                    ;
4223                                    ;       MULTIPLEXER CHANNEL
4224                                    ;
4225  010362  162763  000004  000014           SUB     #4,SRBYTC(DTAB) ;DECR REAMINING BYTE COUNT
4226  010370  003004                           BGT     10$             ;IF > 1, DATA TRANSFER NOT COMPLETE YET
4227                                    ;
4228                                    ;       DATA TRANSFER COMPLETE ON MUX CHANNEL
4229                                    ;
4230  010372  005063  000014                   CLR     SRBYTC(DTAB)    ;INSURE REMAINING BYTE COUNT ZERO
4231  010376  000261                   5$:     SEC                     ;SET MUX TRANSFER COMPLETE FLAG
4232  010400  000404                           BR      30$             ;GOTO COMMON EXIT
4233                                    ;
4234                                    ;       DATA TRANSFER INCOMPLETE
4235                                    ;
4236  010402  062763  000004  000012   10$:    ADD     #4,SBUFA(DTAB)  ;BUMP BUFFER ADDRESS
4237  010410  000241                   20$:    CLC                     ;RESET FLAG TO INDICATE MUX CHAN NOT DONE
4238  010412  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
                                ;
010414  010046                 DISCTL: MOV     R0,-(SP)        ;SAVE REGSITERS USED BY SUBROUTINE
010416  010146                         MOV     R1,-(SP)
010420  010246                         MOV     R2,-(SP)
010422  126327  000001  000002         CMPB    SLCMD(DTAB),#CMRMI ;WAS IT A READ MANUAL INPUT COMMAND?
010430  001535                         BEQ     DRMI            ;IF YES, PERFORM READ MANUAL INPUT PROCEEDURE
010432  126327  000001  000012         CMPB    SLCMD(DTAB),#CMSRMI ;WAS IT A SHORT READ MANAUL INPUT?
010440  001542                         BEQ     DSRMI           ;IF YES, EXIT
010442  126327  000001  000006         CMPB    SLCMD(DTAB),#CMREAD ;WAS IT A READ FULL BUFFER COMMAND?
010450  001522                         BEQ     DREAD           ;YES, RESET CURSOR ON READ FULL BUFFER
                                ;
                                ;       THE COMMAND MUST HAVE BEEN A 360 WRITE
                                ;               DETERMINE TYPE OF TEST BEING RUN
                                ;
010452  105737  012544                 TSTB    TSTTYP          ;TYPE OF TEST 0 = 2848  1 = FRIEND
010456  001102                         BNE     DISFRN          ;FRIEND
                                ;
                                ;       FORMAT DISPLAY ALA 2260
                                ;
010460  016301  000006                 MOV     SINBF(DTAB),R1  ;GET ADDR OF START OF INPUT
010464  126327  000001  000005         CMPB    SLCMD(DTAB),#CMWTLA ;WAS LAST CMD A WRITE LINE ADDRESS?
010472  001016                         BNE     20$             ;NO, NORMAL WRITE
                                ;
                                ;       WRITE LINE ADDRESS COMMAND
                                ;               FIRST BYTE OF DATA BLOCK IS CURSOR LINE ADDRESS
                                ;
010474  005263  000014                 INC     SRBYTC(DTAB)    ;INCR BYTE COUNT
010500  112102                         MOVB    (R1)+,R2        ;GET LINE NUMBER
010502  042702  177760                 BIC     #177760,R2      ;GET ONLY LINE NUMBER
```

```
4295  010506  005063  000004                    CLR     SCURS(DTAB)     ;
4296
4297                                      ;       COMPUTE CURSOR ADDRESS  CURS = LINE # * LINESIZE
4298                                      ;
4299  010512  005702                     10$:    TST     R2              ;DONE?
4300  010514  001405                             BEQ     20$             ;YES, MORE DATA INTO DISPLAY BUF
4301  010516  062763  000050  000004             ADD     #LINSZ,SCURS(DTAB) ;INCR TO NEXT LINE
4302  010524  005302                             DEC     R2              ;DECR LINE COUNT
4303  010526  000771                             BR      10$
4304                                      ;
4305                                      ;       MOVE DATA RECEIVED INTO DISPLAY BUFFER
4306                                      ;               THE DATA RECEIVED MANY BE ANY CONFIGURATION
4307                                      ;               OF EIGHT BITS, 0 - 255. TO EMULATE A 2260 DISPLAY
4308                                      ;               THE RECEIVED CHARACTER IS "FOLDED" INTO ONE OF
4309                                      ;               THE 64 CHARACTERS USED BY THE 2260. THIS FOLDING
4310                                      ;               IS PERFORMED BY TRANSLATING THE CHARACTER TO ASCII
4311                                      ;               AND THEN BACK TO EBCDIC.
4312                                      ;
4313  010530  016302  000010             20$:    MOV     SOUTB(DTAB),R2  ;COMPUTE DISPLAY ADDR
4314  010534  066302  000004                     ADD     SCURS(DTAB),R2
4315  010540  026327  000014  000741             CMP     SRBYTC(DTAB),#DISPSZ+1 ;ALL CHARS PROCESSED?
4316  010546  103077                             BHIS    DISCEX          ;YES, EXIT
4317  010550  005263  000014                     INC     SRBYTC(DTAB)    ;INCREMENT THE BYTE COUNT
4318  010554  112100                             MOVB    (R1)+,R0        ;GET THE NEXT BYTE RECEIVED AND BUMP POINTER
4319  010556  042700  177400                     BIC     #177400,R0      ;STRIP SIGN EXTENSION BITS (IF ANY)
4320  010562  116000  011766                     MOVB    EBCDTB(R0),R0   ;FOLD CHARACTER INTO ASCII CHARACTER SET
4321  010566  042700  177400                     BIC     #177400,R0      ;STRIP SIGN EXTENSION BITS, IF ANY
4322  010572  162700  000040                     SUB     #40,R0          ;SCALE INTO ASCII TABLE RANGE
4323  010576  116012  012366                     MOVB    ATOETB(R0),(R2) ;COMPLETE FOLDING BY RETRANSLATING TO EBCDIC
4324  010602  005263  000004                     INC     SCURS(DTAB)     ;INCR CURSOR PTR
4325  010606  121227  000025                     CMPB    (R2),#NEWLNE    ;WAS A NEW LINE SPECIFIED?
4326  010612  001015                             BNE     60$
4327                                      ;
4328                                      ;       NEW LINE COMMAND - ADVANCE CURSOR TO BEG OF NEW LINE
4329                                      ;               CURSOR = (CURSOR/LINESIZE + 1) * LINESIZE.
4330                                      ;
4331  010614  005002                             CLR     R2              ;CLEAR LINE CTR
4332  010616  005202                     40$:    INC     R2              ;INCR LINE CTR
4333  010620  162763  000050  000004             SUB     #LINSZ,SCURS(DTAB)
4334  010626  003373                             BGT     40$             ;KEEP DIVIDING
4335  010630  005063  000004                     CLR     SCURS(DTAB)     ;CLEAR CURSOR
4336  010634  062763  000050  000004  50$:      ADD     #LINSZ,SCURS(DTAB)
4337  010642  005302                             DEC     R2
4338  010644  001373                             BNE     50$
4339                                      ;
4340                                      ;       CHECK FOR WRAP AROUND
4341                                      ;
4342  010646  026327  000004  000740  60$:      CMP     SCURS(DTAB),#DISPSZ ;CURSOR OVERFLOW DISPLAY BUFFER?
4343  010654  002725                             BLT     20$             ;CURSOR OK, PROCESS NEXT CHAR
4344  010656  005063  000004                     CLR     SCURS(DTAB)     ;OVERFLOW, RESTART CURSOR AT POS 0
4345  010662  000722                             BR      20$
4346
4347
4348                                      ;
4349                                      ;       FRIEND TEST, IF SEPARATE I/O BUFFERS DON'T COPY
4350                                      ;               INPUT TO OUTPUT BUFFER
```

```
4351                                    ;
4352  010664  105737  012545    DISFRN: TSTB    IOBUF           ;SEPARATE I/O BUFFERS?
4353  010670  001026                    BNE     DISCEX          ;YES, DON'T COPY INPUT TO OUTPUT
4354  010672  016301  000006            MOV     SINBF(DTAB),R1  ;SET UP INPUT BUFFER ADDRESS
4355  010676  016302  000010            MOV     SOUTB(DTAB),R2  ;SET UP OUTPUT BUFFER ADDRESS
4356  010702  012700  000360            MOV     #DISPSZ/2,R0    ;TRANSFER THE INPUT BUFFER TO THE OUTPUT BUFFER
4357                                    ;
4358                                    ;       PERFORM COPY
4359                                    ;
4360  010706  012122            10$:    MOV     (R1)+,(R2)+     ;INPUT TO OUTPUT
4361  010710  005300                    DEC     R0              ;ARE WE DONE?
4362  010712  001375                    BNE     10$             ;NO, CONTINUE COPY
4363  010714  000414                    BR      DISCEX          ;PREPARE TO RETURN TO CALLER
4364
4365                                    ;
4366                                    ;       A READ FULL BUFFER WAS PERFORMED
4367                                    ;               THE CURSOR MUST BE RESET TO THE BEGINNING OF THE SCREEN
4368                                    ;
4369  010716  005063  000004    DREAD:  CLR     SCURS(DTAB)     ;RESET THE CURSOR
4370  010722  000411                    BR      DISCEX          ;AND PREPARE TO EXIT
4371
4372
4373
4374                                    ;
4375                                    ;       A READ MANUAL INPUT WAS PERFORMED
4376                                    ;               TO EMULATE THE 2260 SCREEN THE START OF MANUAL INPUT CHARCATER
4377                                    ;               MUST BE DELETED FROM THE SCREEN
4378                                    ;
4379  010724  016301  000020    DRMI:   MOV     SMINS(DTAB),R1  ;GET THE STARTING ADDRESS
4380  010730  005301                    DEC     R1              ;DECREMENT TO THE SMI CHAR
4381  010732  112711  000100            MOVB    #EBCDSP,(R1)    ;BLANK OUT THE CHARACTER
4382  010736  166301  000010            SUB     SOUTB(DTAB),R1  ;AND COMPUTE THE CURSOR POSITION
4383  010742  010163  000004            MOV     R1,SCURS(DTAB)
4384
4385
4386                                    ;
4387                                    ;       A SHORT READ MANUAL INPUT WAS PERFORMED
4388                                    ;               NO ACTION REQUIRED BY DISPLAY CONTROL ROUTINE
4389                                    ;
4390  010746                    DSRMI:
4391
4392
4393                                    ;
4394                                    ;       RESTORE REGISTERS AND RETURN TO CALLER
4395                                    ;
4396  010746  012602            DISCEX: MOV     (SP)+,R2        ;RESTORE SAVED REGISTERS
4397  010750  012601                    MOV     (SP)+,R1
4398  010752  012600                    MOV     (SP)+,R0
4399  010754  000207                    RTS     PC              ;RETURN TO THE CALLER
4400                                    .SBTTL  TELETYPE (CONSOLE) INPUT ISR
4401                                    ;
4402                                    ;       TELETYPE INPUT HANDLER (ISR)
4403                                    ;
4404                                    ;               CONTROL PASSES HERE ON A TELETYPE INPUT INTERRUPT
4405                                    ;
4406                                    ;       DATA IS INPUT FROM THE CONTROL CONSOLE AND STORED INTO
```

```
4407                                    ;       THE TELETYPE INPUT BUFFER (TBUF). WHEN ALL THE DATA IS
4408                                    ;       ENTERED, THE OPERATOR HITS A C/R TO END THE LINE, THEN
4409                                    ;       AN ACTIVE FLAG IS SET AND THE COMMAND EXECUTED BY THE
4410                                    ;       SYSTEM.
4411                                    ;
4412                                    ;               THE FOLLOWING CONTROL FUNCTIONS ARE AVAILABLE FOR
4413                                    ;               OPERATOR CONVENIENCE.
4414                                    ;
4415                                    ;                       C/R  =  LINE DELIMETER
4416                                    ;                            =  DELETE LAST CHARACTER
4417                                    ;                       \    = (BACKSLASH SHIFT L) = DELETE LAST LINE
4418                                    ;                       (CONTROL-C) = ABORT CURRENT COMMAND -- FOR DUMPS
4419                                    ;                       (RUB OUT) = DELETE LAST CHARACTER
4420                                    ;                       (CTL-R) = REENTER ALL PARAMETERS        VRG-2-FEB-82
4421                                    ;                       (CTL-U) = DELETE CURRENT INPUT LINE
4422                                    ;                       (CTL-S) = TEMPORARILY STOP OUTPUT TO CONSOLE
4423                                    ;                       (CTL-Q) = RESUME OUTPUT TO CONSOLE
4424                                    ;
4425                                    ;       NOTE -- A CONTROL Q MUST BE ISSUED AFTER A CONTROL S TO
4426                                    ;               RESUME CONSOLE OUTPUT
4427                                    ;
4428  010756  010046                   TKIN:   MOV     R0,-(SP)        ;SAVE REGISTERS
4429  010760  010146                           MOV     R1,-(SP)
4430  010762  017700  001502                   MOV     @TKB,R0         ;GET TELE CHARACTER
4431  010766  042700  177600                   BIC     #177600,R0      ;INSURE 7-BIT ASCII
4432  010772  013701  012754                   MOV     TPTR,R1         ;BUFFER PTR
4433  010776  020027  000022                   CMP     R0,#CTL.R       ;;CONTROL -R ?                  VRG-2-FEB-82
4434  011002  001002                           BNE     3$              ;NO
4435  011004  000137  001002                   JMP     RSTART          ;YES, ALLOW OPERATOR TO REENTER ALL PARAMETERS
4436  011010  020027  000023           3$:     CMP     R0,#CTL.S       ;CONTROL-S , TEMPORALILY STOP CONSOLE OUTPUT?
4437  011014  001003                           BNE     6$              ;NO, CONTINUE
4438  011016  105237  013105                   INCB    TTYSTP          ;YES, SET FLAG TO STOP TTY OUTPUT
4439  011022  000503                           BR      100$            ;       AND EXIT FROM INTERRUPT
4440  011024  020027  000021           6$:     CMP     R0,#CTL.Q       ;CONTROL-Q, RESUME CONSOLE OUTPUT?
4441  011030  001010                           BNE     10$             ;NO, CONTINUE
4442  011032  105037  013105                   CLRB    TTYSTP          ;YES, RESET CONSOLE STOP FLAG
4443  011036  105737  013103                   TSTB    PCTR            ;       CHECK TO INSURE OUTPUT TO RESUME
4444  011042  001473                           BEQ     100$            ;       NO OUTPUT -- EXIT
4445  011044  004737  011264                   JSR     PC,PROUT        ;               RESTART CONSOLE OUTPUT
4446  011050  000470                           BR      100$            ;       AND EXIT FROM THE INTERRUPT
4447  011052  020027  000003           10$:    CMP     R0,#CTL.C       ;COMMAND ABORT -- CTL C?
4448  011056  001011                           BNE     20$             ;NO
4449  011060  105737  013065                   TSTB    TCMACT          ;IS A COMMAND ACTIVE?
4450  011064  001457                           BEQ     90$             ;NO, TREAT AS A DELETE LAST LINE
4451  011066  105237  013066                   INCB    TCMDAB          ;YES, SET ABORT FLAG
4452  011072  012701  012652                   MOV     #TBUF,R1        ;SET UP BUFFER POINTER
4453  011076  000137  011232                   JMP     100$            ;EXIT
4454  011102  105737  013065           20$:    TSTB    TCMACT          ;TELE CMD CURRENTLY ACTIVE?
4455  011106  001051                           BNE     100$            ;YES, IGNORE CHARACTER
4456  011110  110021                           MOVB    R0,(R1)+        ;STORE CHAR INTO BUFFER - INC PTR
4457  011112  020027  000015                   CMP     R0,#CR          ;LINE DELLIMETER -- C/R?
4458  011116  001005                           BNE     30$             ;NO
4459  011120  012701  012652                   MOV     #TBUF,R1        ;RESET BUFFER PTR
4460  011124  105237  013065                   INCB    TCMACT          ;YES, SET COMMAND ACTIVE FLAG
4461  011130  000440                           BR      100$            ;DONT PRINT THE LINE DELIMITER
4462  011132  020027  000177           30$:    CMP     R0,#RUBOUT      ;A RUBOUT?
```

```
 4463  011136  001002                        BNE     40$             ;NOPE
 4464  011140  012700  000137                MOV     #'_,R0          ;YES, TREAT AS A DELETE LAST CHARACTER
 4465  011144  120027  000025        40$:    CMPB    R0,#CTL.U       ;CONTROL-U? (DELETE CURRENT INPUT LINE)
 4466  011150  001002                        BNE     50$             ;NOPE, CONTINUE
 4467  011152  112700  000134                MOVB    #'\,R0          ;YES, TREAT AS DELETE LAST LINE (BACKSLASH)
 4468  011156  004737  011400        50$:    JSR     PC,PCHAR        ;ECHO THE CHARCTER BACK
 4469  011162  020027  000137                CMP     R0,#'_          ;DELETE LAST CHAR -- BACK ARROW?
 4470  011166  001004                        BNE     60$             ;NO
 4471  011170  124141                        CMPB    -(R1),-(R1)     ;YES, DECR POINTER BY 2
 4472  011172  020127  012652                CMP     R1,#TBUF        ;ARE WE  BEYOND BEG OF THE BUFFER?
 4473  011176  003403                        BLE     70$             ;YES, RESET TO BEG OF BUFFER
 4474  011200  020027  000134        60$:    CMP     R0,#'\          ;DELETE CUR LINE -- BACK SLASH?
 4475  011204  001004                        BNE     80$             ;NO
 4476  011206  012701  012652        70$:    MOV     #TBUF,R1        ;YES, RESET BUFFER PTR
 4477  011212  004737  011342                JSR     PC,CRLF         ;NEW LINE FOR NEW COMMAND
 4478  011216  020127  012752        80$:    CMP     R1,#TBUFE       ;WERE LIMITS EXCEEDED?
 4479  011222  001003                        BNE     100$            ;NOPE, EXIT
 4480  011224  012700  000134        90$:    MOV     #'\,R0          ;THEY WERE -- TREAT AS A LINE ABORT
 4481  011230  000740                        BR      30$
 4482  011232  010137  012754        100$:   MOV     R1,TPTR         ;SAVE BUFFER PTR
 4483  011236  012601                        MOV     (SP)+,R1        ;RESTORE REGISTERS + EXIT
 4484  011240  012600                        MOV     (SP)+,R0
 4485  011242  000002                        RTI
```

```
                        .SBTTL  TELETYPE (CONSOLE) OUTPUT ISR
                ;
                ;       TELETYPE OUTPUT DRIVER (ISR) -- PRINT
                ;
                ;               CONTROL PASSES HERE ON A TELE OUT INTERRUPT
                ;
011244  105037  013064  PISR:   CLRB    PIUFL           ;CLEAR PRINTER BUSY FLAG
011250  105737  013103          TSTB    PCTR            ;ANY MORE DATA TO PRINT?
011254  001402                  BEQ     10$             ;NO, EXIT
011256  004737  011264          JSR     PC,PROUT        ;OUTPUT ANOTHER CHAR
011262  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
                                ;
011264  105737  013064        PROUT:  TSTB    PIUFL           ;IS IT BUSY?
011270  001023                       BNE     20$             ;YES, EXIT
011272  105737  013105               TSTB    TTYSTP          ;HAS CONSOLE OUTPUT BEEN SUSPENDED?
011276  001020                       BNE     20$             ;YES, RETURN IMMEDIATELY TO CALLER
011300  105237  013064               INCB    PIUFL           ;NO, SET BUSY FLAG
011304  105337  013103               DECB    PCTR            ;DECR CHAR COUNTER
011310  117777  001544  001156       MOVB    @PFPTR,@TPB     ;OUTPUT NEXT CHAR
011316  005237  013060               INC     PFPTR           ;INCR PRINT FETCH POINTER
011322  023727  013060  013060       CMP     PFPTR,#PBFE     ;TIME TO WRAP AROUND?
011330  001003                       BNE     20$             ;NO, EXIT
011332  012737  012756  013060       MOV     #PBFS,PFPTR     ;YES, RESTORE TO START OF BUFFER
011340  000207                20$:    RTS     PC                      ;RETURN TO CALLER


                                ;
                                ;       PRINT A CR/LF
                                ;
                                ;       CALLING SEQUENCE
                                ;       JSR     PC,CRLF
                                ;       ........RETURN
                                ;
                                ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
                                ;
011342  010246                CRLF:   MOV     R2,-(SP)        ;SAVE THE R2 REGISTER
011344  012702  105215               MOV     #105215,R2      ;DO A CRLF
011350  004737  011360               JSR     PC,PRINT2       ;PRINT IT
011354  012602                       MOV     (SP)+,R2        ;RESTORE THE R2 REGISTER
011356  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
                                ;
```

```
 4553  011360  010237  011374          PRINT2: MOV     R2,P2BF
 4554  011364  004137  011504                  JSR     R1,MESG
 4555  011370  011374                          .WORD   P2BF
 4556  011372  000207                          RTS     PC
 4557  011374  000000                  P2BF:   .WORD   0
 4558  011376     377     377                  .BYTE   377,377
 4559
 4560
 4561                                  ;
 4562                                  ;       PRINT 1 CHARACTER
 4563                                  ;
 4564                                  ;       CALLING SEQUENCE
 4565                                  ;       ........R0 CONTAINS THE CHARACTER TO BE PRINTED
 4566                                  ;       JSR     PC,PCHAR
 4567                                  ;       ........RETURN WITH THE DATA IN THE PRINT BUFFER
 4568                                  ;
 4569                                  ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
 4570                                  ;
 4571  011400  110037  011414          PCHAR:  MOVB    R0,P1BF
 4572  011404  004137  011504                  JSR     R1,MESG
 4573  011410  011414                          .WORD   P1BF
 4574  011412  000207                          RTS     PC              ;RETURN TO THE CALLER
 4575  011414     000     377          P1BF:   .BYTE   0,377
```

```
4576                                    ;
4577                                    ;       PRMESG PRINT A CHARACTER STRING
4578                                    ;
4579                                    ;       CALLING SEQ
4580                                    ;       ........R2 CONTAINS THE STARTING ADDRESS OF THE MESSAGE
4581                                    ;       JSR     PC,PRMESG
4582                                    ;       ......RETURN
4583                                    ;
4584                                    ;       NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4585                                    ;
4586                                    ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4587  011416  010246                    PRMESG: MOV     R2,-(SP)        ;SAVE REGS
4588  011420  010346                            MOV     R3,-(SP)
4589  011422  013703  013062                    MOV     PPPTR,R3        ;GET PRINT OUTPUT POINTER
4590  011426  12122?  000377            10$:    CMPB    (R2),#377       ;END OF MESSAGE?
4591  011432  001417                            BEQ     40$             ;YES, EXIT
4592  011434  112223                            MOVB    (R2)+,(R3)+     ;NO MOVE NEXT CHAR TO PRINT BUFFER
4593  011436  105237  013103                    INCB    PCTR            ;INCR CHAR COUNTER
4594  011442  020327  013060                    CMP     R3,#PBFE        ;AT END OF BUFFER?
4595  011446  001002                            BNE     20$             ;NO
4596  011450  012703  012756                    MOV     #PBFS,R3        ;YES, WRAP AROUND TO BEG OF BUFFER
4597  011454  004737  011264            20$:    JSR     PC,PROUT        ;CAN WE START PRINT?
4598  011460  123737  013103  013142    30$:    CMPB    PCTR,PMAX       ;IS PRINT BUFFER FULL?
4599  011466  001774                            BEQ     30$             ;YES, WAIT TILL ROOM AVAILABLE
4600  011470  000756                            BR      10$             ;GET NEXT CHAR
4601  011472  010337  013062            40$:    MOV     R3,PPPTR        ;EXIT, RESTORE PUT PTR
4602  011476  012603                            MOV     (SP)+,R3        ;RESTORE REGS
4603  011500  012602                            MOV     (SP)+,R2
4604  011502  000207                            RTS     PC              ;RETURN TO THE CALLER
4605
4606
4607                                    ;
4608                                    ;       MESG -- PRINT A CHARACTER STRING ON THE SYSTEM CONSOLE
4609                                    ;
4610                                    ;       CALLING SEQUENCE
4611                                    ;       JSR     R1,MESG
4612                                    ;       .WORD   ADDRESS OF START OF MESSAGE
4613                                    ;       ........RETURN
4614                                    ;
4615                                    ;       NOTE -- MESSAGE MUST BE TERMINATED BY A 377
4616                                    ;
4617                                    ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4618                                    ;
4619  011504  010246                    MESG:   MOV     R2,-(SP)        ;SAVE REGISTER
4620  011506  012102                            MOV     (R1)+,R2        ;GET ADDRESS OF MESSAGE AND BUMP FOR RETURN
4621  011510  004737  011416                    JSR     PC,PRMESG       ;MORE MESSAGE PROCESSING
4622  011514  012602                            MOV     (SP)+,R2        ;RESTORE SOILED REGISTER
4623  011516  000201                            RTS     R1              ;RETURN TO THE CALLER
4624
4625
4626                                    ;
4627                                    ;       INMES PRINT A CHARACTER STRING
4628                                    ;
4629                                    ;       CALLING SEQUENCE
4630                                    ;       JSR     R1,INMES
4631                                    ;       .WORD ADDRESS OF MESSAGE
```

```
4632                                    ;       ......RETURN
4633                                    ;
4634                                    ;       INMES IS USED FOR ROUTINES AT THE ISR LEVEL AND DOES
4635                                    ;               NOT CHECK TO SEE IF DATA WILL BE OVERLAYED IN
4636                                    ;               TELEBUFFER
4637                                    ;
4638                                    ;       NOTE -- THE MESSAGE MUST BE TERMINATED BY A 377
4639                                    ;
4640                                    ;       NO REGISTERS ARE MODIFIED BY THIS SUBROUTINE
4641                                    ;
4642  011520  010246                    INMES:  MOV     R2,-(SP)
4643  011522  113746  013142                    MOVB    PMAX,-(SP)      ;CHEAT, SAVE PMAX
4644  011526  112737  000377  013142            MOVB    #377,PMAX       ;AND MAKE VERY LARGE
4645  011534  012102                            MOV     (R1)+,R2
4646  011536  004737  011416                    JSR     PC,PRMESG       ;USE STANDARD MESSAGE PROCESSOR
4647  011542  112637  013142                    MOVB    (SP)+,PMAX      ;RESTORE PRINT MAX
4648  011546  012602                            MOV     (SP)+,R2
4649  011550  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
                                ;
011552  005003                 COTB:   CLR     R3
011554  005005                         CLR     R5
011556  112204                 10$:    MOVB    (R2)+,R4        ;GET NEXT CHAR
011560  120427  000067                 CMPB    R4,#'7          ;CHAR GT 7?
011564  003013                         BGT     20$             ;YES EXIT
011566  120427  000060                 CMPB    R4,#'0          ;CHAR LT 0?
011572  002410                         BLT     20$             ;YES, EXIT
011574  042704  177770                 BIC     #177770,R4      ;SAVE ONLY L.S. 3 BITS
011600  006303                         ASL     R3              ;SHIFT OLD RESULT BY 8
011602  006303                         ASL     R3
011604  006303                         ASL     R3
011606  060403                         ADD     R4,R3           ;ADD IN NEW NUMBER
011610  005205                         INC     R5              ;INCR CHAR COUNT
011612  000761                         BR      10$             ;GET NEXT CHAR
011614  000207                 20$:    RTS     PC              ;RETURN TO CALLER
```

```
4680                                    ;
4681                                    ;       CHTB -- CONVERT ASCII HEX TO BINARY
4682                                    ;
4683                                    ;       CALLING SEQUENCE
4684                                    ;       ......R2 = ADDRESS OF FIRST CHARCTER TO BE CONVERTED
4685                                    ;       JSR     PC,CHTB
4686                                    ;       ........RETURN
4687                                    ;       UPON RETURN
4688                                    ;               R2 = NEXT CHAR POSITION NOT CONVERTED
4689                                    ;               R3 = BINARY RESULT
4690                                    ;               R4 = (BITS 0-7) FIRST NON HEX CHARACTER
4691                                    ;               R5 = NUMBER OF CHARACTERS CONVERTED
4692                                    ;
4693  011616  005003                    CHTB:   CLR     R3
4694  011620  005005                            CLR     R5
4695  011622  112204                    10$:    MOVB    (R2)+,R4        ;GET THE FIRST CHARACTER
4696  011624  162704  000060                    SUB     #'0,R4          ;SCALE RELATIVE TO ASCII ZERO
4697  011630  100422                            BMI     30$             ;NOT A VALID HEX CHAR
4698  011632  020427  000012                    CMP     R4,#10.         ;IS RESULT STILL GT 10?
4699  011636  002410                            BLT     20$             ;YES, WE HAVE A VALID HEX DIGIT
4700  011640  162704  000007                    SUB     #7,R4
4701  011644  020427  000020                    CMP     R4,#16.         ;IS IT A LETTER?
4702  011650  002012                            BGE     30$             ;NO, INVALID CHAR
4703  011652  020427  000012                    CMP     R4,#10.         ;AND GT 10
4704  011656  002407                            BLT     30$             ;NO,ILLEGAL CHAR
4705  011660  006303                    20$:    ASL     R3              ;MAKE ROOM FOR NEW ENTRY
4706  011662  006303                            ASL     R3
4707  011664  006303                            ASL     R3
4708  011666  006303                            ASL     R3
4709  011670  060403                            ADD     R4,R3           ;INSERT NEW ENTRY
4710  011672  005205                            INC     R5              ;INCR CHAR COUNT
4711  011674  000752                            BR      10$             ;AND CONVERT NEXT CHAR
4712  011676  005302                    30$:    DEC     R2              ;GET THE ILLEGAL CHARCTER
4713  011700  112204                            MOVB    (R2)+,R4        ;AND PUT IT R4
4714  011702  000207                            RTS     PC              ;ITS TIME TO RETURN TO THE CALLER
```

```
                                        .SBTTL  PROCESSOR ERROR TRAP HANDLERS
                                ;
                                ;       TRAP OUT ROUTINES
                                ;
                                ;
                                ;
                                ;       MEMORY TIME OUT ROUTINE
                                ;
011704  012702  014235          MTO:    MOV     #PMTO,R2        ;SET UP ADDRESS OF THE PRINT ROUTINE
011710  000404                  BR      TOUTRT          ;TO GENERALIZED TRAP OUT ROUTINE
                                ;
                                ;       MEMORY MANAGEMENT TRAP OUT ROUTINE
                                ;
011712  005037  177572          MMERR:  CLR     @#SR0           ;CLEAR THE MEMORY MANAGEMENT BIT
011716  012701  014261          MOV     #PMMERR,R1      ;SET UP ADDRES OF ERROR MESSAGE
011722  000005                  TOUTRT: RESET                   ;CLEAR ALL DEVICES
011724  105037  013064          CLRB    PIUFL           ;CLEAR PRINT IN USE FLAG
011730  005037  177776          CLR     PSW             ;LOWER PROCESSOR STATUS TO ALLOW INTERRUPTS TO CUM
011734  004737  011416          JSR     PC,PRMESG       ;PRINT THE ERROR MESSAGE
011740  105737  013103          10$:    TSTB    PCTR            ;IS PRINTING DONE?
011744  001375                  BNE     10$
011746  000000                  HALT                    ;YES, HALT
011750  000137  001000          JMP     START
                                ;
                                ;       INVALID UNIBUS ADDRESS TRAP
                                ;
011754  022626                  UNTRP:  CMP     (SP)+,(SP)+     ;POP THE PUSH STACK
011756  005037  177776          CLR     PSW             ;CLEAR THE PROCESSOR STATUS WORD
011762  000137  001206          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
                                ;
011766                          EBCDTB:
011766  040440  041502  042504          .ASCII  / ABCDEFGHI^.<(+!/      ;00 - 0F
011774  043506  044510  027136
012002  024074  020453
012006  045046  046113  057515          .ASCII  '%JKLM_OPQR_$*);]'      ;10 - 1F
012014  050117  051121  022137
012022  024452  056473
012026  027455  052123  053125          .ASCII  '-/STUVWXYZ',%[>?'      ;20 - 2F
012034  054127  055131  026042
012042  055445  037476
012046  030460  031462  032464          .ASCII  '0123456789:#@'=\'      ;30 - 3F
012054  033466  034470  021472
012062  023500  056075
012066  040440  041502  042504          .ASCII  / ABCDEFGHI^.<(+!/      ;40 - 4F
012074  043506  044510  027136
012102  024074  020453
012106  045046  046113  047115          .ASCII  '%JKLMNOPQR_$*);]'      ;50 - 5F
012114  050117  051121  022137
012122  024452  056473
012126  027455  052123  053125          .ASCII  '-/STUVWXYZ',%[>?'      ;60 - 6F
012134  054127  055131  026042
012142  055445  037476
012146  030460  031462  032464          .ASCII  '0123456789:#@'=\'      ;70 - 7F
012154  033466  034470  021472
012162  023500  056075
012166  040440  041502  042504          .ASCII  / ABCDEFGHI^.<(+!/      ;80 - 8F
012174  043506  044510  027136
012202  024074  020453
012206  045046  046113  047115          .ASCII  '%JKLMNOPQR_$*);]'      ;90 - 9F
012214  050117  051121  022137
012222  024452  056473
012226  027455  052123  053125          .ASCII  '-/STUVWXYZ',%[>?'      ;A0 - AF
012234  054127  055131  026042
012242  055445  037476
012246  030460  031462  032464          .ASCII  '0123456789:#@'=\'      ;B0 - BF
012254  033466  034470  021472
012262  023500  056075
012266  040440  041502  042504          .ASCII  / ABCDEFGHI^.<(+!/      ;C0 - CF
012274  043506  044510  027136
012302  024074  020453
012306  045046  046113  047115          .ASCII  '%JKLMNOPQR_$*);]'      ;D0 - DF
012314  050117  051121  022137
012322  024452  056473
012326  027455  052123  053125          .ASCII  '-/STUVWXYZ',%[>?'      ;E0 - EF
012334  054127  055131  026042
012342  055445  037476
012346  030460  031462  032464          .ASCII  '0123456789:#@'=\'      ;F0 - FF
012354  033466  034470  021472
012362  023500  056075
```

```
                                        ;
                                        ;       ASCII TO EBCDIC CONVERSION TABLE
                                        ;
012366  100     117     152     ATOETB: .BYTE   100,117,152,173,133,154,120,175 ;240-247
012371  173     133     154
012374  120     175
012376  115     135     134             .BYTE   115,135,134,116,153,140,113,141 ;250-257
012401  116     153     140
012404  113     141
012406  360     361     362             .BYTE   360,361,362,363,364,365,366,367 ;260-267
012411  363     364     365
012414  366     367
012416  370     371     172             .BYTE   370,371,172,136,114,176,156,157 ;270-277
012421  136     114     176
012424  156     157
012426  174     301     302             .BYTE   174,301,302,303,304,305,306,307 ;300-307
012431  303     304     305
012434  306     307
012436  310     311     321             .BYTE   310,311,321,322,323,324,325,326 ;310-317
012441  322     323     324
012444  325     326
012446  327     330     331             .BYTE   327,330,331,342,343,344,345,346 ;320,327
012451  342     343     344
012454  345     346
012456  347     350     351             .BYTE   347,350,351,155,177,137,112,025 ;330-337
012461  155     177     137
012464  112     025
                                        .SBTTL  PROGRAM CONSTANTS AND VARIABLES
                                        ;
                                        ;       CONSOLE UNIBUS ADDRESS CONSTANTS
                                        ;
012466  177560                  TKS:    .WORD   177560          ;KEYBOARD CONTROL STATUS REGISTER
012470  177562                  TKB:    .WORD   177562          ;KEYBOARD DATA BUFFER
012472  177564                  TPS:    .WORD   177564          ;PRINTER STATUS/CONTROL REGISTER
012474  177566                  TPB:    .WORD   177566          ;PRINTER DATA BUFFER
                                        ;
                                        ;       DX REGISTERS - ADDRESS GENERATED BY INITIALIZATION
                                        ;
012476  000000                  DXDS:   .WORD   0               ;DEVICE STATUS -- TT1
012500  000000                  DXCA:   .WORD   0               ;COMMAND AND ADDRESS -- TT2
012502  000000                  DXCS:   .WORD   0               ;CONTROL UNIT STATUS
012504  000000                  DXOS:   .WORD   0               ;OFFSET AND STATUS
012506  000000                  DXBA:   .WORD   0               ;BUS ADDRESS
012510  000000                  DXBC:   .WORD   0               ;BYTE COUNT
012512  000000                  DXMO:   .WORD   0               ;MAINTANCE OUT
012514  000000                  DXMI:   .WORD   0               ;MAINTANCE IN
012516  000000                  DXCB:   .WORD   0               ;CONTROL BITS
012520  000000                  DXND:   .WORD   0               ;NPR DATA
012522  000000                  DXES1:  .WORD   0               ;EXTRA SIGNALS
012524  000000                  DXMOB:  .WORD   0               ;BUFFERED BUS OUT
012526  000000                  DXES2:  .WORD   0               ;EXTRA SIGNALS
```

```
                                ;
                                ;       CONFIGURATION CONSTANTS
                                ;
012530  000000                  UNADDR: .WORD   0               ;UNIBUS ADDRESS
012532  000000                  VECTAD: .WORD   0               ;DX VECTOR ADDRESS
012534  000000                  SDEV:   .WORD   0               ;STARTING DEV NUMBER
012536  000000                  EDEV:   .WORD   0               ;ENDING DEV NUMBER
012540     000                  CHTYPE: .BYTE   0               ;CHANNEL TYPE 0 = MPX - 1 = SEL
012541     000                  MMRESP: .BYTE   0               ;MEMORY MANAGEMENT 0 = NO - 1 = YES
012542  000000                  BUFREL: .WORD   0               ;BUFFER RELOCATION ADDRESS
012544     000                  TSTTYP: .BYTE   0               ;TEST TYPE 0 = 2848 - 1 = FRIEND
012545     000                  IOBUF:  .BYTE   0               ;SEPERATE I/O BUFFER 0 = NO - 1 = YES
012546     000                  FILLCH: .BYTE   0               ;FILL CHARACTER
012547     000                  CONEND: .BYTE   0               ;EXTRA


                                ;
                                ;       SYSTEM PUSH STACK
                                ;
        012650                          .=.+100
        012650                  SSTACK  =       .


                                ;
                                ;       SYSTEM VARIABLES
                                ;
                                ;       THE FOLLOWING VARIABLES ARE RESET UPON START-UP
                                ;
012650  000000                  VSTRT:  .WORD   0               ;DUMMY
        012652                  TBUF    =       .               ;START OF TELETYPE INPUT BUFFER
        012752                          .=.+100
012752  000000                  TBUFE:  .WORD   0               ;END OF TELETYPE INPUT BUFFER
012754  000000                  TPTR:   .WORD   0               ;TELE IN PTR
012756  000000                  PBFS:   .WORD   0               ;START OF PRINT BUFFER
        013060                          .=.+100
        013060                  PBFE    =       .               ;END OF PRINT BUFFER
013060  000000                  PFPTR:  .WORD   0               ;PRINT FETCH PTR
013062  000000                  PPPTR:  .WORD   0               ;PRINT PUT PTR
013064     000                  PIUFL:  .BYTE   0               ;PRINTER IN USE FLAG
013065     000                  TCMACT: .BYTE   0               ;TELE COMMAND ACTIVE FLAG 0 = NON-ACT
013066     000                  TCMDAB: .BYTE   0               ;TEL COMMAND ABORT 1 = ABORT
013067     000                  LINECT: .BYTE   0               ;LINE CTR - CHARS / LINE
013070     000                  WK:     .BYTE   0               ;WORK LOC
013071     000                  WK1:    .BYTE   0               ;WORK LOC
013072  000000                  TTPTR:  .WORD   0               ;TUMBLE TABLE PTR
013074  000000                  TTADDR: .WORD   0               ;BEG OF TUMBLE TABLE
013076  000000                  SDEVTB: .WORD   0               ;START OF DEVICE TABLES
013100     000                  DXSTPF: .BYTE   0               ;DX STOP FLAG
013101     000                  MAXDEV: .BYTE   0               ;HIGHEST DEV # 1 - 8
013102     000                  DXACT:  .BYTE   0               ;DXACTIVE FLAG
013103     000                  PCTR:   .BYTE   0               ;PRINT BUFFER COUNTER
013104     000                  DXABFL: .BYTE   0               ;DX ABORT FLAG  0 = NO ABORT, 1 = ABORT
013105     000                  TTYSTP: .BYTE   0               ;CONSOLE OUTPUT STOP FLAG 0 = OUTPUT; 1 = NO OUTPUT
                                        .EVEN
013106  000000                  CMDCHF: .WORD   0               ;COMMAND CHAIN FLAG
013110  000000                  MDEV:   .WORD   0               ;DEV # IN MPXR EXEC
```

```
 4917            013110                  SELDEV  =       MDEV            ;DEV # IN SEL EXEC
 4918  013112    000000                  PBUFA:  .WORD   0               ;PHYSICAL BUFF ADDR - IN ,000'S
 4919  013114    000000                  VBUFA:  .WORD   0               ;VIRTUAL BUFF ADDR - IN ,000'S
 4920  013116    000000                  PHYOFF: .WORD   0               ;PHY OFFSET FOR MEMORY MANAGEMENT
 4921  013120    000000                  CDEV:   .WORD   0               ;CURRENT DX DEVICE -- INTER SERVICE ROUTINE
 4922  013122    000000                  DEVCON: .WORD   0               ;DEVICE ADDED TO THE DEVICE NUMBER = STARTING DEV NUMB -
 4923  013124    000000                  XADDR:  .WORD   0               ;EXTENDED ADDRESS BITS FOR THE DX CONTROL REGISTER -- IN
 4924  013126    000000                  STSPW:  .WORD   0               ;START OF THE PSW TABLE
 4925  013130    000000                  DSTOFF: .WORD   0               ;OFFSET TO THE DST TABLE
 4926  013132    000000                  SADDR:  .WORD   0               ;TELETYPE COMMAND STARTING BUFFER ADDRESS
 4927  013134    000000                  EADDR:  .WORD   0               ;TELETYPE COMMAND ENDING BUFFER ADDRESS
 4928  013136    000000                  DMPADR: .WORD   0               ;POINTER TO DUMP ROUTINE CURRENTLY BEING UTILIZED BY TEL
 4929  013140    000000                  VEND:   .WORD   0
 4930
 4931
 4932                                    ;
 4933                                    ;       THE FOLLOWING VARIABLES ARE NOT RESET ON START-UP
 4934                                    ;
 4935  013142    000102                  PMAX:   .WORD   PBFE-PBFS       ;SIZE OF PRINT BUFFER
 4936  013144    000000                  FTIMFL: .WORD   0               ;FIRST TIME FLAG
```

```
4937                                            .SBTTL  MESSAGES
4938                                            .NLIST  BEX
                                        ;
                                        ;       SYSTEM MESSAGES
                                        ;
013146     215     212          STMSG:  .BYTE   215,212
013150  055103  054104  026511          .ASCII /CZDXI-C  NEW DX11-B RESPONDER/
013205     377     377                  .BYTE   377,377
013207     215     212          CTRMSG: .BYTE   215,212
013211     116  052117  035105          .ASCII /NOTE: CONTROL P HAS BEEN CHANGED TO CONTROL R /
013267     377                          .BYTE   377
                                        .EVEN
013270     215     212          UNMSG:  .BYTE   215,212
013272  047125  041111  051525          .ASCII /UNIBUS ADDRESS -OCTAL- : /
013323     377                          .BYTE   377
                                        .EVEN
013324     215     212          VECTMS: .BYTE   215,212
013326  047111  042524  051122          .ASCII /INTERRUPT VECTOR ADDRESS -OCTAL- : /
013371     377                          .BYTE   377
                                        .EVEN
013372     215     212          DEVMES: .BYTE   215,212
013374  042504  044526  042503          .ASCII /DEVICE ADDRESSES -HEX- (XX,XX): /
013434     377                          .BYTE   377
        013436                          .EVEN
013436     215     212          CHTYMS: .BYTE   215,212
013440  044103  047101  042516          .ASCII /CHANNEL TYPE (M OR S): /
013467     377                          .BYTE   377
                                        .EVEN
013470     215     212          MMMES:  .BYTE   215,212
013472  042515  047515  054522          .ASCII /MEMORY MANAGEMENT (Y OR N): /
013526     377                          .BYTE   377
        013530                          .EVEN
013530     215     212          BFREMS: .BYTE   215,212
013532  052502  043106  051105          .ASCII /BUFFER RELOCATION, IF SPECIFIED - IN EVEN ,000'S -OCTAL- : /
013625     377                          .BYTE   377
                                        .EVEN
013626     215     212          TESTMS: .BYTE   215,212
013630  051106  042511  042116          .ASCII /FRIEND (F) OR 2848 DIAG(D): /
013664     377                          .BYTE   377
        013666                          .EVEN
013666     215     212          FIOMS:  .BYTE   215,212
013670  042523  040520  040522          .ASCII /SEPARATE I-O BUFFERS (Y OR N): /
013727     377                          .BYTE   377
                                        .EVEN
013730     215     212          FILLMS: .BYTE   215,212
013732  052517  050124  052125          .ASCII /OUTPUT BUFFER FILL CHARACTER -HEX- : /
013777     377                          .BYTE   377
                                        .EVEN
014000     207     207     215  NXMMSG: .BYTE   207,207,215,212
014004  047516  020116  054105          .ASCII /NON EX-MEM ERROR/
014024     215     212     377          .BYTE   215,212,377
        014030                          .EVEN
014030     207     207     215  PARMES: .BYTE   207,207,215,212
014034  040520  044522  054524          .ASCII /PARITY ERROR/
014050     212     215     377          .BYTE   212,215,377
        014054                          .EVEN
```

```
014054     207     207     215  ILLMES: .BYTE   207,207,215,212
014060  046111  042514  040507          .ASCII  /ILLEGAL DEVICE NUMBER/
014105     212     215     377          .BYTE   212,215,377
                                        .EVEN
014110  052503  051122  047105  STPMES: .ASCII  /CURRENT DEVICE NUMBER -- /
014141     377                          .BYTE   377
                                        .EVEN
014142     207     207     215  INVLDC: .BYTE   207,207,215,212
014146  047111  040526  044514          .ASCII /INVALID DX COMMAND/
014170     212     215     377          .BYTE   212,215,377
        014174                          .EVEN
014174  020040  047516  046440  PNOMM:  .ASCII  /  NO MEMORY MANAGEMENT AVAILABLE/
014234     377                          .BYTE   377
014235     215     212     207  PMTO:   .BYTE   215,212,207,207
014241     115  046505  051117          .ASCII  /MEMORY TIME OUT/
014260     377                          .BYTE   377
014261     215     212     207  PMMERR: .BYTE   215,212,207,207
014265     115  046505  051117          .ASCII  /MEMORY MANAGEMENT ERROR/
014314     377                          .BYTE   377
        014316                          .EVEN
014316     215     212          RNMESG: .BYTE   215,212
014320  054523  052123  046505          .ASCII  /SYSTEM INITIALIZED, TYPE 'R' TO ENABLE DX/
014371     377                          .BYTE   377
014372     015     012          HELPMS: .BYTE   CR,LF
014374  054104  030461  041055          .ASCII  /DX11-B 2848 EMULATOR TEST PACKAGE - OPERATIONAL INFORMATION/
014467     015     012                  .BYTE   CR,LF
014471     015     012                  .BYTE   CR,LF
014473     104  026440  020055          .ASCII  /D -- DUMP COMMAND/<CR><LF>
014516  020040  020040  020040          .ASCII  /       DTT,C     DUMP TUMBLE TABLE IN CODE "C"/<CR><LF>
014576  020040  020040  020040          .ASCII  /       DIN,C,XX  DUMP INPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014674  020040  020040  020040          .ASCII  /       DOT,C,XX  DUMP OUTPUT BUFFER FOR DEVICE XX IN CODE "C"/<CR><LF>
014773     105  026440  020055          .ASCII  /E -- ENABLE DEVICE ON DX/<CR><LF>
015025     040  020040  020040          .ASCII  /       EXX       ENABLE DEVICE XX/<CR><LF>
015070  020106  026455  043040          .ASCII  /F -- FILL BUFFER COMMAND/<CR><LF>
015122  020040  020040  020040          .ASCII  /       FIN,HH,XX FILL INPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015210  020040  020040  020040          .ASCII  /       FOT,HH,XX FILL OUTPUT BUFFER ON DEV XX WITH HH/<CR><LF>
015277     110  026440  020055          .ASCII  /H -- HELP COMMAND/<CR><LF>
015322  020040  020040  020040          .ASCII  /       THIS TEXT/<CR><LF>
015344  020113  026455  045440          .ASCII  /K -- KILL A DEVICE ON THE DX/<CR><LF>
015402  020040  020040  020040          .ASCII  /       KXX       KILL DEVICE XX/<CR><LF>
015443     122  026440  020055          .ASCII  /R -- ENABLE DX (RUN)/<CR><LF>
015471     040  020040  020040          .ASCII  /       R         RUN TEST/<CR><LF>
015524  020123  026455  042040          .ASCII  /S -- DISABLE DX (STOP)/<CR><LF>
015554  020040  020040  020040          .ASCII  /       S         STOP IMMEDIATELY/<CR><LF>
015617     040  020040  020040          .ASCII  /       SD        STOP AFTER NEXT DATA TRANSFER/<CR><LF>
015677     040  020040  020040          .ASCII  /       SE        STOP AFTER NEXT ENDING SEQUENCE/<CR><LF>
015761     040  020040  020040          .ASCII  /       SI        STOP ON NEXT SEL SEQ (ISS)/<CR><LF>
016036  020040  020040  020040          .ASCII  /       SP        STOP ON NEXT PARITY ERROR/<CR><LF>
016112  005015  044127  051105          .ASCII  <CR><LF>/WHERE:/<CR><LF>
016124  020040  020040  041442          .ASCII  /    "C" IS CODE FORMAT     O = OCTAL/<CR><LF>
016172  020040  020040  020040          .ASCII  /                           A = ASCII/<CR><LF>
016240  020040  020040  020040          .ASCII  /                           E = EBCDIC/<CR><LF>
016307     040  020040  020040          .ASCII  /                           H = HEX/<CR><LF>
016353     040  020040  021040          .ASCII  /    "XX" IS DX-11 DEVICE NUMBER IN HEX/<CR><LF>
016423     040  020040  021040          .ASCII  /    "HH" IS A HEX CHARACTER/<CR><LF><LF>
016461     103  047117  047523          .ASCII  /CONSOLE CONTROL CHARACTERS/<CR><LF>
```

```
        016515     103  046124  041455          .ASCII  /CTL-C (^C) ABORT CURRENT COMMAND/<CR><LF>
        016557     103  046124  050455          .ASCII  /CTL-Q (^Q) RESUME OUTPUT AFTER TEMPORARILY STOPPING BY (^S)/<CR><LF>
        016654  052103  026514  020122          .ASCII  /CTL-R (^R) REQUESTS THE REENTRY OF CONTROL PARAMETERS/<CR><LF>  ;;VRG-2
        016743     103  046124  051455          .ASCII  /CTL-S (^S) TEMPORARILY STOP OUTPUT TO CONSOLE/<CR><LF>
        017022  052103  026514  020125          .ASCII  /CTL-U (^U) DELETE CURRENT INPUT LINE/<CR><LF>
        017070  052522  047502  052125          .ASCII  /RUBOUT -- DELETE LAST CHARACTER INPUT/<CR><LF>
        017137     015     012     012          .BYTE   CR,LF,LF,LF
                002551                  HELPLN  =       .-HELPMS
        017143     215     212          TOOC:  .BYTE   215,212
        017145     124  047517  041440          .ASCII  /TOO CLOSE TO 200000 BOUNDARY/
        017201     377                          .BYTE   377
                                                .LIST   BEX
  4939                                          .EVEN
  4940          000001                          .END
```

```
ACCESS  004062          2713    2987#
ASCDMP  004650          2855    3169#
ATOETB  012366          3103    4323    4808#
ATTN  = 000200          2095#   4052
BFREMS  013530          2330    4938#
BSY   = 000020          2098#   4031
BSYEN = 004000          2073#   2765
BUFREL  012542          2357*   2430    2432*   2433    2434    4870#
BUSOUT= 000040          2109#   3698
BYSS  = 000400          2053#
CDEV    013120          2821    3457*   4921#
CDEVST  010270          2745    3031    3055    3522    3555    3576    3611    4179#
CE    = 000010          2099#   2603    4079
CEDE  = 000003          2020#   3577    3740    4070
CERR    003040          2702#   2741    2800    2867    2992    3080    3281    3361
CHEND = 000040          2056#   3508
CHENDS= 001000          2052#
CHIS  = 000200          2054#   3505
CHTB    011616          2267    2626    2930    3353    4693#
CHTYMS  013436          2285    4938#
CHTYPE  012540          2283*   2291*   3767    4101    4220    4868#
CMDCHF  013106          2754*   3543*   3559*   3581    3583*   3614*   3709    3711*   3785    3899    4915#
CMDCHN= 000004          2059#   3612
CMDREJ= 000001          2061#   3664
CMREAD= 000006          2039#   4273
CMRMI = 000002          2037#   4269
CMSRMI= 000012          2040#   4271
CMWRT = 000001          2036#
CMWTLA= 000005          2038#   4286
CONEND  012547          4874#
CONUNE  007762          3814    3938    4045#
COTB    011552          2641    2987    3006    3268    3290    4665#
CR    = 000015          1976#   2212    2235    2274    2332    2654    2790    3011    3096    4457    4938
CRLF    011342          2195    2198    2674    2818    3171    3190    3225    4477    4536#
CSPWST  010322          2746    3523    3556    3573    3602    3660    4199#
CTL.C = 000003          1979#   4447
CTL.Q = 000021          1980#   4440
CTL.R = 000022          1981#   4433
CTL.S = 000023          1982#   4436
CTL.U = 000025          1983#   4465
CTRMSG  013207          2200    4938#
CUBUSY= 000400          2075#   3544
CUE   = 000040          2097#   4045
CUEND = 000020          2057#   3511
DE    = 000004          2100#   2603    4079
DEVCON  013122          2548*   2549*   3033    3058    3831    3875    3957    4012    4028    4090    4121    4203    4922#
DEVMES  013372          2257    4938#
DISCEX  010746          4316    4353    4363    4370    4396#
DISCTL  010414          3605    4266#
DISFRN  010664          4281    4352#
DISPSZ= 000740          2029#   2571    3082    3092    3318    3331    3530    3834    3873    3955    4010    4060    4315
                        4342    4356
DMPADR  013136          2868*   2886    2903    4928#
DONE  = 000200          2076#   3429    3458
DREAD   010716          4274    4369#
DRMI    010724          4270    4379#
```

```
DSRMI   010746          4272    4390#
DSTOFF  013130          2490*   3036    4925#
DTUMTB  003644          2870    2901#
DUMP    003444          2715    2853#
DXAB    006550          3476    3593    3679    3759#
DXABFL  013104          2681    2755*   2829*   3760*   4912#
DXACT   013102          2753*   3118    3542*   3558*   3578    3580*   3610*   3721*   3736*   3837*   3879*   3968*   4022*
                        4111*   4125*   4910#
DXBA    012506          3829*   3830*   3854*   3858    3863*   3871*   3872*   3956*   4011*   4118*   4119*   4120*   4850#
DXBC    012510          3741    3834*   3862*   3873*   3960*   3965*   4015*   4020*   4124*   4851#
DXCA    012500          3833*   3877*   3959*   4014*   4092*   4123*   4847#
DXCB    012516          4854#
DXCS    012502          2740    2742*   2743*   2764*   2765*   2766*   2817*   2830*   2831*   3429*   3458*   3544*   3759*
                        3839*   3881*   3969*   4032*   4110*   4126*   4848#
DXDS    012476          2414    2824    4846#
DXENB = 000100          2077#   2766    2817    3759
DXES1   012522          4856#
DXES2   012526          4858#
DXEXEC  006566          3134    3422    3767#
DXEXIT  010224          3394    3762    3787    3798    3840    3882    3901    3922    3970    4033    4112    4132#
DXISR   005346          2424    3380#
DXMI    012514          4853#
DXMO    012512          4852#
DXMOB   012524          4857#
DXND    012520          4855#
DXONLN= 001000          2074#   2740    2766    3759
DXOS    012504          2481*   4109*   4849#
DXRD  = 000005          2084#   3881    4032    4126
DXRST = 000001          2082#
DXST  = 000007          2085#   4110
DXSTPF  013100          2801*   3615    3647    3662    3738    3752*   4908#
DXWR  = 000003          2083#   3839    3969
EADDR   013134          2884*   2890    2942*   2948    3293*   3317*   3318*   3330*   3331*   4927#
EBCDMP  004604          2858    3147#
EBCDSP= 000100          2024#   2370    3531    4061    4381
EBCDTB  011766          3150    3153    4320    4756#
EDEV    012536          2255*   2260*   2268*   2269    2494    2544    4867#
ENABLE  004234          2717    3030#
EOM   = 000152          2026#   3112
EQPCHK= 000020          2110#   3744
ERASCM  010002          3813    3937    4059#
ESEND = 000100          2055#   3499
ESEQ    010054          3809    3852    3860    3933    3986    4079#
EXEC    002732          2596    2674#   2704    2767    2812    2832    2894    2914    2950    2973    3017    3039    3062
                        3120    3126
EXECMD  003054          2695    2708#
FILL    003730          2719    2929#
FILLCH  012546          2370*   2375*   2403*   2572    2931*   2947    3534    4064    4873#
FILLMS  013730          2401    4938#
FIOMS   013666          2387    4938#
FTIMFL  013144          2136*   2187    2207*   2585*   4936#
GDEV    005314          2874    2932    3030    3054    3077    3353#
GETPRM  001162          2188    2195#
GLERR   005116          3280#   3289    3292    3302    3314    3327    3337
GLEX    005304          3294    3336#
GLEX1   005302          3306    3319    3332#
```

```
GLIMIT  005062          2853    2929    3267#
GLMIN   005204          3274    3312#
GLMOT   005244          3276    3325#
GLMTT   005154          3272    3300#
GLOCT   005124          3270    3287#
HDMP    004772          2822    3226#
HELP    004016          2721    2959#
HELPLN= 002551          2960    4938#
HELPMS  014372          2959    4938#
HEXDMP  004766          2861    3225#
ILLMES  014054          3475    4938#
INFDSC= 004000          2050#   3493
INHEX   002636          2256    2400    2626#
INIT    002040          2189    2372    2388    2390    2402    2412#
INIT10  002404          2534#   2536
INMES   011520          3474    3591    3645    3677    4642#
INOCT   002644          2209    2232    2284    2312    2329    2367    2386    2641#
INPUT   004342          2723    3075#
INR5    002650          2627    2642#
INTREQ= 000100          2108#   3690
INVLDC  014142          3678    4938#
IOBUF   012545          2385*   2393*   4352    4872#
ISSREJ= 000010          2058#   3618
KILL    004300          2725    3054#
KILLEX  004336          3062#
KISAR0= 172340          1967#   2445
KISAR7= 172356          1968#   2456
KISDR0= 172300          1969#   2446
LF    = 000012          1977#   4938
LINECT  013067          2680*   3169    3172*   3173*   4902#
LINSZ = 000050          2027#   2029    4301    4333    4336
LOOP    005376          3388    3419#   3512    3545    3560    3584    3639    3712    3745
MAXDEV  013101          2547*   2576    2751    3357    3463    3540    3793    3917    4909#
MCMDTB  007340          3930    3931#
MDEV    013110          2762*   3902    3920    3929*   4916#   4917
MESG    011504          2196    2199    2301    2348    2594    2819    4554    4572    4619#
MEX     007222          3769    3899#
MMERR   011712          2165    4731#
MMMES   013470          2313    4938#
MMRESP  012541          2311*   2319*   2342    2435    4869#
MMSR0 = 177572          1970#   2321*   2460*   4731*
MREAD   007612          3936    3980    3982    4005#
MRMI  = 007514          3932    3978#
MSRMI   007514          3940    3979#
MTO     011704          2163    2412    4724#
MUXEND  010354          3722    3737    4220#
MWRITE  007364          3931    3935    3951#
NEWLNE= 000025          2023#   4325
NEWPRM  001206          2207#   2213    2215    2217    2219    2220    4748
NEWP10  001472          2258    2283#   2290
NEWP20  001716          2331    2347    2357#
NOLIN = 000014          2028#   2029
NOP   = 000003          2022#   3673
NXM   = 040000          2047#   3496
NXMMSG  014000          3592    4938#
OCTDMP  004702          2826    2864    2996    3190#
```

```
OCTEX   004756          3207#
ODMP    004706          3000    3191#
PARER = 100000          2046#   3502    3692    3742
PARMES  014030          3646    4938#
PBFE  = 013060          4521    4594    4896#   4935
PBFS    012756          2179    2180    4523    4596    4894#   4935
PBUFA   013112          2433*   2449    2477    2578    4918#
PCHAR   011400          2968    3200    3241    4468    4571#
PCHEND  006470          3510    3736#
PCHEN1  006500          3728    3738#
PCHEX   006524          3723    3742#
PCHIS   006306          3507    3660#
PCTR    013103          2887    2904    2966    4443    4493    4518*   4593*   4598    4737    4911#
PCUEND  006454          3513    3721#
PESEND  006126          3501    3602#
PFPTR   013060          2179*   4519    4520*   4521    4523*   4897#
PHYOFF  013116          2487*   2488*   3830    3863    3872    3954    3995    4009    4120    4920#
PNDSC   006040          3495    3573#
PISR    011244          2158    4492#
PIUFL   013064          4492*   4513    4517*   4734*   4899#
PMAX    013142          4598    4643    4644*   4647*   4935#
PMMERR  014261          4732    4938#
PMTO    014235          4724    4938#
PNOMM   014174          2302    4938#
PNXM    006114          3498    3591#
PPARER  006256          3504    3645#
PPPTR   013062          2180*   4589    4601*   4898#
PRINT2  011360          2676    2703    2998    3002    3174    4538    4553#
PRMESG  011416          2643    4587#   4621    4646    4736
PROUT   011264          4445    4495    4513#   4597
PSELRT  006010          3492    3555#
PSW   = 177776          1966#   2186*   2300*   3125    3127*   4735*   4747*
PSYSRT  005662          3460    3521#
P1BF    011414          4571*   4573    4575#
P2BF    011374          4553*   4555    4557#
QCUE  = 000010          2021#   3638
RNMESG  014316          2595    4938#
RSTART  001002          2136#   4435
RUBOUT= 000177          1984#   4462
RUN     003124          2727    2740#
SADDR   013132          2883*   2885    2889*   2890    2902    2906    2908*   2909*   2941*   2943    2993*   2994    2999
                        3014*   3015*   3287*   3303*   3304*   3316*   3317    3329*   3330    4926#
SATTN   007772          3815    3939    4052#
SBUFA = 000012          2003#   3953*   3954*   3956    3988*   3992    3995*   4008*   4009*   4011    4182*   4236*
SCMD  = 000000          1996#   3574    3577*   3606    3638*   3680*   3740*   3783    3790*   3802    3835    3874    3904
                        3908*   3926    3967    4005    4070*   4180*
SCMDRJ= 000200          2107#   3704
SCMDTB  006716          3806    3807#
SCURS = 000004          2000#   3082    3084    3086*   3088    3091*   3092    3113*   3527*   3856    3990    4068*   4295*
                        4301*   4314    4324*   4333*   4335*   4336*   4342    4344*   4369*   4383*
SDEV    012534          2254*   2259*   2269    2271    2492    2545    2548    3354    3461    4866#
SDEVTB  013076          2560*   3315    3328    4151    4907#
SELDEV= 013110          2550*   3781    3796    3805*   4917#
SELRST= 020000          2048#   3490
SENSCM  010152          3810    3934    4118#
SEX     006600          3768    3781#
```

```
SINBF = 000006   2001# 2567* 2568* 3316 3829 3953 4285 4354
SLCMD = 000001   1997# 3603 3609* 3835* 3874* 3967* 4005* 4184* 4269 4271 4273 4286
SMI   = 000112   2025# 3089
SMINS = 000020   2007# 2749* 3090* 3526* 3854 3988 4379
SONLF = 000016   2005# 3038* 3056* 3685 4200
SOUTB = 000010   2002# 2569* 2570* 3087 3329 3529 3855 3871 3989 4008 4059 4313 4355
                 4382
SPACE = 000040   1978# 3098 3102
SRBYTC= 000014   2004# 3741* 3838* 3880* 3951 3955* 3961 3963 3981 3994* 4006 4010* 4016
                 4018 4183* 4225* 4230* 4292* 4315 4317*
SRDRQ = 000017   2006# 2748* 3114* 3525* 3608* 3788 3850 3853* 3906 3984 3987*
SRD10   007154   3864 3874#
SREAD   007132   3812 3848 3871#
SRMI    007032   3808 3846#
SSENSE= 000002   1998# 2747* 3037* 3524* 3557* 3690* 3698* 3704* 3744* 3836* 3849* 3878* 3966*
                 3983* 4021* 4069* 4118
SSRMI   007032   3816 3847#
SSTACK= 012650   2147 2677 4881#
SSTAT = 000003   1999# 3528* 3649* 4045* 4052* 4079* 4093 4095* 4109 4181*
STAMOD= 000100   2096#
START   001000   2125 2132# 4740
STKSTB= 000002   2060#
STMSG   013146   2197 4938#
STOP    003276   2729 2789#
STOPDX  003352   2683 2791 2817#
STOUT   010062   4046 4053 4090#
STPDX   006544   3616 3648 3663 3739 3752#
STPMES  014110   2820 4938#
STSPW   013126   2486* 3035 3060 4030 4104 4205 4924#
SUDEV   010242   3081 3481 3637 3782 3903 4151# 4179
SUDEX   010264   4155 4158#
SUD10   010252   4154# 4157
SWRITE  006742   3807 3811 3829#
SYSINT  001006   2132 2147#
SYSRST= 010000   2049# 3459
TBUF  = 012652   2178 2650 2690 3005 4452 4459 4472 4476 4890#
TBUFE   012752   4478 4892#
TCHEND  005636   3506 3508#
TCHIS   005624   3503 3505# 3650
TCMACT  013065   2644* 2648 2678* 2684 2995* 3003 4449 4454 4460* 4900#
TCMDAB  013066   2645* 2646 2679* 2892 2912 2969 4451* 4901#
TCMDTB  003056   2693 2712#
TCUEND  005650   3509 3511#
TESEND  005600   3497 3499#
TESTMS  013626   2368 4938#
TINDSC  005554   3491 3493#
TKB     012470   4430 4837#
TKIN    010756   2156 4428#
TKS     012466   2184* 4836#
TNXM    005566   3494 3496#
TOOC    017143   2352 4938#
TOUTRT  011722   4725 4733#
TPARER  005612   3500 3502#
TPB     012474   4519* 4839#
TPS     012472   2185* 4838#
TPTR    012754   2178* 4432 4482* 4893#
```

```
TSELRS  005542          3490#
TSTTYP  012544          2366*  2376*  3075   3532   3634   3847   3979   4062   4280   4871#
TTADDR  013074          2505*  2756   3444   4906#
TTPTR   013072          2506*  2757*  3303   3386   3419   3445*  4905#
TTSIZE= 001000          2030#  2507   2758   2901   2906   2908   3304   3431   3441
TTYSTP  013105          4438*  4442*  4515   4913#
UCHK  = 000002          2101#  2498   2538   2608   2609   3061   3649   4093   4095
UCHKS = 002000          2051#
UEXP  = 000001          2102#
UNADDR  012530          2208*  2221*  2222*  2413   4864#
UNMSG   013270          2210   4938#
UNTRP   011754          2230   4746#
VBUFA   013114          2434*  2461*  2482   4919#
VCMDTB  002622          2533   2600#
VECTAD  012532          2231*  2243*  2423   4865#
VECTMS  013324          2233   4938#
VEND    013140          2171   4929#
VSTRT   012650          2170   2171   4889#
WK      013070          3147*  3148   3150*  3154   4903#
WK1     013071          3151   3153*  4904#
XADDR   013124          2584*  2764   4923#
.     = 017202          2116#  2124#  2131#  2611#  3978   4880#  4881   4890   4891#  4895#  4896   4938#

. ABS.  017202     000

 ERRORS DETECTED:  0

CZDXIC,CZDXIC/SOL/CRF/NL:TOC=CZDXIC.P11
RUN-TIME: 4 8 .6 SECONDS
RUN-TIME RATIO: 29/13=2.1
CORE USED:  7K  (13 PAGES)
```