

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZITA-D-D
PRODUCT NAME: INTERPROCESSOR TEST PROGRAM (ITEP)
PROGRAM DATE: JANUARY 1977
MAINTAINER: DIAGNOSTICS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1973, 1977 BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11 SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

A. PDP-11 SYSTEM WITH AT LEAST 4K OF CORE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE

THIS PROGRAM AND ALL OVERLAYS ARE ASSEMBLED IN ABSOLUTE FORMATS. THE ABS LOADER IS USED TO LOAD THE PROGRAM AND OVERLAYS.

LOAD THE ITEP PROGRAM AND THE APPROPRIATE OVERLAY FOR THE TYPE OF INTERFACE YOU WISH TO TEST.

4.0 OPERATING PROCEDURES.

IF RUNNING ITEP ON AN LSI-11 ENVIRONMENT :

1. IF THE LINE CLOCK IS TO BE USED IT SHOULD BE ENABLED PRIOR TO PROGRAM EXECUTION.

A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED

1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY) DN11 AND DM11BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THIS LISTING.

1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
 - A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
 - B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
 - C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DMB.

IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS, VECTOR, ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DMBB.

2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
 - A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
 - B. TYPEIN ACTUAL BUS ADDRESS
3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
 - A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS

- B. TYPEIN ACTUAL VECTOR ADDRESS
 - 4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.
 - A. TYPE A CAR, RETURN TO USE DEFAULT VALUE
 - B. TYPEIN ACTUAL VALUE
 - 5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE /VERLAY.(SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - A. TYPE A CAR, RETURN TO USE DEFAULT VALUE
 - B. TYPEIN ACTUAL VALUE
 - 6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE OVERLAY.
 - A. TYPE A CAR, RETURN TO USE DEFAULT VALUE
 - B. ENTER ACTUAL VALUE
 - 7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
 - A. TYPE A CAR, RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
 - B. ENTER ACTUAL VALUE.
 - 8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DM11BB.
 - 9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
 - A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR, RETURN.
- NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES,THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR(INTERFACE SERVICE ROUTINE) SPECIFICATION
SWR14=1 SETUP DM-11BB ISR
SWR13=1 SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR (OVERLAY) (NOT DN-11 OR DM11BB)
SET APPROPRIATE SWITCHES AND HIT CONTINUE.
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11,DM11-BB THEN VARIABLE ISR. (FOR EACH ENTRY SET SWICHES AND THEN HIT CONT.)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE (200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.)
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND DMBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THIS LISTING)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.
 3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
 - A. PRESS CONTINUE TO START TESTING

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR
SW14=1 SINGLE PASS
 SW14 HAS NO EFFECT IF SW04=0
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
 IF SW12=0 AND SW04=1 END PASS IS TYPED
 AND TRANSMITTED/RECEIVED DATA IS TYPED.
SW11=1 USE PREVIOUSLY SPECIFIED DATA
SW10=1 DATA SELECT (WITH SW09)
SW09=1 DATA SELECT (WITH SW10)
 00=1 GET DATA FROM OPERATOR
 01=1 TEST MESSAGE #1 (\$A QUICK BROWN FOX)
 10=1 TEST MESSAGE #2 (\$B NUMERICS)
 11=1 TEST MESSAGE #3 (\$C COMTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
 * IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
 TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
 RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
 OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.

SW04=1 RETURN TO MONITOR FOR END PASS
 WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
SW03=1 INTERNAL LOOPBACK MODE
SW02=1 EXTERNAL LOOPBACK MODE
SW01=1 ONE-WAY-IN MODE
SW00=1 ONE-WAY-OUT MODE

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A
REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE.
TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN
^(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377)
SEPERATED BY SPACES AND TERMINATED BY ^(UP ARROW).
I.E. ABCD^ 000 123 377^ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES
INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED
DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING
IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES(TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177),
AND ARE FOLLOWED BY A CR(015), LF(012), RECEIVE TERMINATING CHARACTER(001),
4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION,
WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED. DURING RECEPTION,
WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF.
IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10(SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1(SW04=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SW07=0)
3. RETURNS TO MONITOR FOR "END PASS"(SW04=1) OR GO TO STEP 1 (SW04=0)

ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR GO TO STEP 1 (SW04=0)

- E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED,
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED,
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO
TRANSMIT DATA.

F. IF SW04=0 THE OVERLAY WILL CONTINUE TO TRANSMIT/RECEIVE DATA.

IF SW04=1 THE OVERLAY WILL RETURN TO THE MONITOR AND TYPE "END PASS".

IF BOTH SW04=1 AND SW14=1, THE PROGRAM WILL REQUEST NEW INTERFACE PARAMS AFTER ONE PASS OF THE SELECTED TEST MODE.

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.
LINE FEED = RESTART PROGRAM AT LOCATION 200.
QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)

THEN TYPE EITHER:

*WXXXXXX TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.

*BXXXXXX TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.

*C TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.
CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

IF THE OPERATOR WISHES TO MANUALLY EXAMINE THE TRANSMIT OR RECEIVE BUFFERS, DO THE FOLLOWING; TO FIND THE STARTING ADDRESS OF THE RECEIVE BUFFER, LOAD ADDRESS 11020 AND EXAMINE. TO FIND THE STARTING ADDRESS OF THE TRANSMIT BUFFER, LOAD ADDRESS 11022 AND EXAMINE.

5.1 NORMAL HALTS SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

THE ONLY ERROR REPORT FROM THE CONTROL PROGRAM OCCURS IF THE INTERFACE SPECIFIED IS NOT LOADED.

THE ERROR REPORTS FROM THE VARIOUS INTERFACE SERVICE ROUTINES ARE AS DEFINED IN THEIR DOCUMENTS

7.0 RESTRICTIONS

THE OPERATION OF THIS PROGRAM REQUIRES COORDINATION BETWEEN THE OPERATOR AND THE OPERATOR OF ANOTHER PDP-11 SYSTEM UNLESS ONE OF THE SYSTEMS IS ALWAYS OPERATING IN A FIXED MODE. THE FOLLOWING TABLE LISTS THE VALID COMBINATIONS:

NOTE: ONLY ONE MODE MAY BE SELECTED AT A TIME.

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL-DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTIONS APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.
EXAMPLE:
SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU S

8.0 MISCELLANEOUS

ITEP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

THE INTERPROCESSOR TEST PROGRAM (ITP) PROVIDES THE LINKAGE BETWEEN THE OPERATOR AND THE VARIOUS INTERFACE SERVICE ROUTINES (OVERLAY) WHICH PERFORM THE ACTUAL DATA MOVEMENT AND VERIFICATION TO AND FROM THE COMMUNICATION LINK. IN ADDITION, ITP CONTAINS VARIOUS INTERRUPT AND SUB ROUTINES WHICH ARE USED BY THE OVERLAY'S.

9.1 TRAP CATCHER

THIS IS A SERIES OF JUMP AND HALT INSTRUCTIONS PLACED IN ALL UNUSED VECTORS TO CATCH UNEXPECTED INTERRUPTS.

9.2 SWITCH REGISTER INPUT ROUTINE (MANIN:)

THIS ROUTINE IS ENTERED ONLY WHEN SWITCH 15 IS SET WHEN PROGRAM IS STARTED AT LOCATION 200. IT ACCEPTS PARAMETERS FOR THE ISR'S FROM THE CONSOLE SWITCHES AT A SERIES OF HALTS, AS SPECIFIED IN OPERATING INSTRUCTIONS.

9.3 PARAMETER INPUT ROUTINE (GETIT:)

THIS ROUTINE SOLICITS PARAMETERS FROM THE OPERATOR ON THE CONSOLE DEVICE AND PLACES THEM IN THE SPECIFIED ISR'S PARAMETER TABLE.
NOT USED OPTIONAL PARAMETER WORDS ARE INDICATED BY THE PRESENCE OF A NEGATIVE VALUE IN THE ISR'S TABLE. THIS SECTION OF CODE UTILIZES SUB/ROUTINE 'GETANY' WHICH PRINTS OUT THE WORD POINTED TO BY THE ADDRESS IN REGISTER 0.
IT THEN INPUTS A WORD OR CARRIAGE RETURN FROM THE OPERATOR. IF ONLY A CARRIAGE RETURN IS TYPED, THE PARAMETER IS LEFT AS IT IS, OTHERWISE IT IS REPLACED BY THE OPERATORS TYPE IN AND THE POINTER IN REGISTER 0 IS INCREMENTED TO THE NEXT WORD.

9.4 TTY INTERRUPT (TTYINT:)

THE TTY INTERRUPT IS USED TO INTERRUPT THE EXECUTION OF A TEST IN ORDER TO RESTART (TYPE A LINE FEED) OR TO SPECIFY NEW OPERATIONAL SWITCHES (TYPE A CARRIAGE RETURN)

9.5 SET SWITCH OPTIONS (SWRSET:)

THE PROGRAM WILL HALT (MANUAL PARAMETER ENTRY) OR WAIT FOR A CARRIAGE RETURN (TTY CONTROL) AT THIS POINT TO PERMIT THE OPERATOR TO SETUP THE OPERATIONAL SWITCH SETTINGS. THE TEST MODE(SW00-SW03) AND TEST DATA(SW08-SW11) MAY BE CHANGED ONLY AT THIS POINT. ALL OTHER SWITCHES MAY BE CHANGED WHILE A TEST IS RUNNING. IF NEW VARIABLE DATA IS SPECIFIED, THIS ROUTINE WILL REQUEST THAT THE DATA BE ENTERED AND UTILIZES THE 'GETSTR' SUB/ROUTINE TO INPUT THE DATA FROM THE OPERATOR.

9.6 SETUP TIMER (SUTIME:)

THE PROGRAM LOOKS FOR AND UTILIZES EITHER THE LINE CLOCK OR REAL TIME CLOCK IF EITHER IS PRESENT ON THE SYSTEM. A BUS ERROR(NO RESPONSE) IS USED TO INDICATE THE ABSENCE OF A CLOCK. IF NEITHER EXISTS, THE PROGRAM WILL STILL RUN BUT IS SUBJECT TO WAITING IN UNENDING LOOPS.

9.7 THE INTERFACE SERVICE ROUTINES (ISR'S) ARE ENTERED AT THIS POINT.

9.8 END OF PASS (\$EOP:)

THIS SECTION OF CODE WILL PRINT "END OF PASS XXXXXX" AND THEN SENSE FOR SW14. IF SWITCH 14 IS RESET THE OVERLAY'S ARE REENTERED. IF SWITCH 14 IS SET THE PROGRAM CHECKS TO SEE IF IT WAS LOADED BY A MONITOR (LOCATION 42 NOT EQUAL 0) AND IF IT WAS, CONTROL IS RETURNED TO THE MONITOR. OTHERWISE THE PROGRAM REQUESTS NEW PARAMETERS.

9.10 HALT HANDLER (\$HLT:)

THIS ROUTINE IS USED TO SENSE THE OPERATIONAL SWITCHES AND PROVIDE ERROR CONTROL. IT WILL PRINTOUT THE ADDRESS OF THE ERROR HLT IF SWITCH 13 (DELETE ERROR TYPEOUTS) IS DOWN (NOT SET)

- 9.11 READ A CHARACTER ROUTINE (\$READC:)
THIS ROUTINE GETS A CHARACTER FROM THE TTY AND PLACES IT ON THE STACK
- 9.12 READ A STRING ROUTINE (\$READS)
THIS ROUTINE GETS A STRING OF CHARACTERS FROM THE TTY AND PLACES THEM IN A BUFFER SPECIFIED BY THE ADDRESS FOLLOWING THE SUB/ROUTINE CALL.
THE ROUTINE WILL ALSO ACCEPT OCTALLY REPRESENTED CHARACTERS WHEN THEY ARE PRECEDED AND FOLLOWED BY UP ARROWS, AND SPERATED BY SPACES OR COMMAS.
- 9.13 OCTAL INPUT ROUTINE(\$ACCEPT:)
THIS ROUTINE READS AN OCTALLY REPRESENTED WORD FROM THE TTY AND PLACES IT IN THE LOCATION INDICATED BY THE ADDRESS FOLLOWING THE SUB/ROUTINE CALL.
- 9.14 CLOCK INTERRUPT ROUTINE (TIMER:)
THIS ROUTINE IS ENTERED ON INTERUPTS FROM EITHER THE LINE CLOCK OR REAL TIME CLOCK EVERY 16 MILLISECONDS IF EITHER IS PRESENT.
IT WILL INCREMENT LOCATION 'TIME:' IN THE OVERLAY'S PARAMETER TABLE EVERY SECOND.
- 9.15 BINARY TO OCTAL ROUTINE (\$B2O16)
THIS ROUTINE WILL PRINTOUT THE OCTAL REPRESENTATION OF A WORD ON THE STACK.
- 9.16 POWER DOWN ROUTINE (\$PWRDN:)
THIS ROUTINE SAVES THE STATUS OF THE MACHINE WHEN POWER IS LOST.
- 9.17 POWER UP ROUTINE (\$PWRUP:)
THIS ROUTINE RESTORES THE STATE OF THE MACHINE WHEN POWER IS RESTORED AND RESTARTS AT ADDRESS 200.
- 9.18 VARIABLE INTERFACE SERVICE ROUTINE (VISR:)
THESE LOCATIONS ARE RESERVED FOR AND WILL BE OVERLAID BY THE VARIABLE ISR'S.
THE FIRST 2 WORDS CONTAIN A 3 CHARACTER ISR NEMONIC FOLLOWED BY A ZERO CHARACTER.
THE NEXT 3 WORDS CONTAIN THE BUS ADDRESS, VECTOR ADDRESS AND PRIORITY.
THE NEXT 2 WORDS MAY CONTAIN OPTIONAL PARAMETERS. THEY WILL CONTAIN ALL ONES IF THEY ARE NOT REQUIRED
THE NEXT WORD MAY CONTAIN THE ADDRESS OF AN INPUT BUFFER IF THE ISR REQUIRES AN ASCII PARAMETER. IT WILL CONTAIN

ALL ONES IF THE PARAMETER IS NOT REQUIRED.
LOCATION 'CLOCK:' WILL BE INCREMENTED EVERY SECOND WHILE
THE TEST IS BEING RUN IF THERE IS A LINE CLOCK OR REAL TIME CLOCK
ON THE SYSTEM. IT MAY BE USED AS A ELASPED TIMER BY THE ISR.

10.0 PARAMETERS FOR THE DM11BB AND THE DN11

10.1 DM11BB PARAMETERS

PARAM#1 IS LOADED INTO THE CONTROL AND STATUS REGISTER OF THE DM11BB
TO SELECT THE LINE NUMBER IN OCTAL (BITS 0-3). ALL OTHER BITS MUST BE 0'S.
THIS IS THE ONLY PARAMETER USED BY THE DM11BB.

10.2 DN11 PARAMETERS

ONLY PARAM#3 IS USED BY THE DN11, IT CONTAINS THE NUMBER THE DN WILL DIAL.

MAINDEC-11-DZITA-D INTERPROCESSOR TEST PROGRAM MACY11 27(1006) 01-DEC-76 11:01 PAGE 13
DZITAD.P11 01-DEC-76 11:00

SEQ 0013

```

535 ;BASIC DEFINITIONS
536 ;*****
537 ;INITIAL ADDRESS OF THE STACK POINTER
538 001070 STACK= 1070
539
540 ;*****
541 ,EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
542
543
544 ;REGISTER DEFINITION
545 000000 R0= %0 ;GENERAL REGISTER
546 000001 R1= %1 ;GENERAL REGISTER
547 000002 R2= %2 ;GENERAL REGISTER
548 000003 R3= %3 ;GENERAL REGISTER
549 000004 R4= %4 ;GENERAL REGISTER
550 000005 R5= %5 ;GENERAL REGISTER
551 000006 R6= %6 ;GENERAL REGISTER
552 000007 R7= %7 ;GENERAL REGISTER
553 000000 MODE= %0
554 000006 R6=SP
555 000007 R7=PC
556
557
558 ;SWITCH DEFINITION
559 100000 SW15= 100000
560 040000 SW14= 40000
561 020000 SW13= 20000
562 010000 SW12= 10000
563 004000 SW11= 4000
564 002000 SW10= 2000
565 001000 SW09= 1000
566 000400 SW08= 400
567 000200 SW07= 200
568 000100 SW06= 100
569 000040 SW05= 40
570 000020 SW04= 20
571 000010 SW03= 10
572 000004 SW02= 4
573 000002 SW01= 2
574 000001 SW00= 1
575 ,EQUIV SW09,SW9
576 ,EQUIV SW08,SW8
577 ,EQUIV SW07,SW7
578 ,EQUIV SW06,SW6
579 ,EQUIV SW05,SW5
580 ,EQUIV SW04,SW4
581 ,EQUIV SW03,SW3
582 ,EQUIV SW02,SW2
583 ,EQUIV SW01,SW1
584 ,EQUIV SW00,SW0
585 000000 PRTY0= 0
586 000040 PRTY1= 40
587 000100 PRTY2= 100
588 000140 PRTY3= 140
589 000200 PRTY4= 200
590 000240 PRTY5= 240
    
```

```

591 000300 PRTY6= 300
592 000340 PRTY7= 340
593
594 ;MISCELLANEOUS BIT ASSIGNMENT
595 100000 BIT15= 100000
596 040000 BIT14= 40000
597 020000 BIT13= 20000
598 010000 BIT12= 10000
599 004000 BIT11= 4000
600 002000 BIT10= 2000
601 001000 BIT09= 1000
602 000400 BIT08= 400
603 000200 BIT07= 200
604 000100 BIT06= 100
605 000040 BIT05= 40
606 000020 BIT04= 20
607 000010 BIT03= 10
608 000004 BIT02= 4
609 000002 BIT01= 2
610 000001 BIT00= 1
611 ,EQUIV BIT09,BIT9
612 ,EQUIV BIT08,BIT8
613 ,EQUIV BIT07,BIT7
614 ,EQUIV BIT06,BIT6
615 ,EQUIV BIT05,BIT5
616 ,EQUIV BIT04,BIT4
617 ,EQUIV BIT03,BIT3
618 ,EQUIV BIT02,BIT2
619 ,EQUIV BIT01,BIT1
620 ,EQUIV BIT00,BIT0
621
622 ;VECTOR ADDRESSES
623 000004 ERRVEC= 4
624 000010 RESVEC= 10
625 000014 TBITVEC=14
626 000014 TRTVEC= 14
627 000014 BPTVEC= 14
628 000020 IOTVEC= 20
629 000024 PMRVEC= 24
630 000030 EMTVEC= 30
631 000034 TRAPVEC=34
632 ,EQUIV R4,CSR
633 ,EQUIV R4,RCSR
634
635 000000 ,=0
636 ;TRAP CATCHER IN UNUSED LOCATIONS FROM 0 - 776
637 ;LOCATION 0 WILL CATCH IMPROPERLY LOADED VECTORS
638 ,=24
639 000024 006672 $PWRDN
640 000026 000340 340
641 000030 005032 $HLT
642 000032 000340 340
643 000034 006544 $TRAP
644 000036 000340 340
645
646
    
```

```

647          000100          ,=100
648 000100 006312          TIMER
649 000102 000340          340
650
651          000174          ,=174
652 000174 000000          DISPRG:0
653 000176 000000          SWREG: 0
654
655          000200          ,=200
656
657 000200 000137 003254    JMP      0#BEGIN      ;JUMP TO STARTING ADDRESS OF PROGRAM
658 000204 000137 004116    JMP      0#SWPRNT     ;RESTART AT 204, DO THE RESTART,
    
```

```

659          001100          ;*****
660          001100          ,=1100
661
662          ;ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
663          ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
664          ;NOTE1: NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
665          ;NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
666
667          ;CALL:
668          ;1) USING A TRAP INSTRUCTION
669          ;      TYPE      ,MESADR      ;MESADR IS FIRST ADDRESS OF AN ASCII STRING
670          ;OR
671          ;      TYPE
672          ;      MESADR
673
674 001100 010046          STYPE: MOV      R0,-(SP)      ;SAVE R0
675 001102 017600 000002    MOV      02(SP),R0      ;GET ADDRESS OF ASCII STRING
676 001106 062766 000002 000002 ADD      #2,2(SP)      ;ADJUST RETURN PC
677 001114 112046          1S:  MOVB   (R0)+,-(SP)    ;PUSH CHARACTER TO BE TYPED ONTO STACK
678 001116 001003          BNE     2S              ;BR IF IT ISN'T THE TERMINATOR
679 001120 005726          TST    (SP)+          ;IF TERMINATOR POP IT OFF THE STACK
680 001122 012600          MOV    (SP)+,R0      ;RESTORE R0
681 001124 000002          RTI                    ;RETURN
682 001126 004737 001160    2S:  JSR    PC,5$        ;GO TYPE THIS CHARACTER
683 001132 122726 000012    3S:  CMPB  #12,(SP)+    ;CHECK IF THE CHAR. TYPED WAS A LINE FEED
684 001136 001366          BNE     16              ;GO GET NEXT CHAR. IF NOT LINE FEED
685 001140 013746 001524    MOV    NULL,-(SP)    ;GET # OF FILLER CHARS. NEEDED
686
687 001144 105366 000001    4S:  DECB  1(SP)        ;DOES A NULL NEED TO BE TYPED?
688 001150 002770          BLT    3S              ;BR IF NO--GO POP THE NULL OFF OF STACK
689 001152 004737 001160    JSR    PC,5$        ;GO TYPE A NULL
690 001156 000772          BR     4S              ;LOOP
691 001160 105777 000334    5S:  TSTR  @TPS        ;!AIT UNTIL PRINTER IS READY
692 001164 100375          BPL    5S              ;
693 001166 116677 000002 000326 MOVH   2(SP),@TPB     ;LOAD CHAR TO BE TYPED INTO DATA REG.
694 001174 000207          RTS    PC              ;
695
    
```

```

696 ;*****
697 ;   DEBUG DUMP ROUTINE
698 ;*****
699     =1200
700 001200 013746 011042 DMPHLT: MOV FLAG,-(SP)
701 001204 042737 000070 011042 BIC #70,FLAG ;INIT FLAGS
702 001212 104400 001532 TYPE ,ASTRK ;TYPE *
703 001216 104402 GETCHR
704 001220 012637 001314 MOV (SP)+,68
705 001224 104400 001314 TYPE ,68
706 001230 122737 000127 001542 CMPB #127,6CHAR ;W? FOR WORD
707 001236 001004 BNE 16
708 001240 052737 000010 011042 BIS #BIT3,FLAG ;SET FLAG BIT
709 001246 000430 BR 38
710 001250 122737 000102 001542 18: CMPB #102,6CHAR ;B? FOR BYTE
711 001256 001004 BNE 26
712 001260 052737 000020 011042 BIS #BIT4,FLAG
713 001266 000420 BR 38
714 001270 122737 000103 001542 28: CMPB #103,6CHAR ;C? FOR CONTINUE
715 001276 001014 BNE 36
716 001300 012637 011042 MOV (SP)+,FLAG
717 001304 052737 000040 011042 BIS #BIT5,FLAG
718 001312 000413 BR DUMP
719 001314 000000 68: 000000
720 001316 104400 001526 46: TYPE ,QUES
721 001322 104400 001536 TYPE ,CRLF
722 001326 000724 BK DMPHLT
723 001330 005037 001544 38: CLR WORK
724 001334 005726 TST (SP)+
725 001336 104406 001544 ACCEPT ,WORK
726 001342 012700 001476 DUMP: MOV #DMPHLT,R0 ;INIT DUMP LIST
727 001346 062710 000020 ADD #20, (R0) ;BUMP ADDRESS
728 001352 032737 000040 011042 BIT #BIT5,FLAG
729 001360 001005 BNE L1
730 001362 013737 001544 ;01476 MOV WORK,DMPHLT
731 001370 001001 BNE L1
732 001372 022020 CMP (R0)+, (R0)+ ;SKIP 1ST TWO ENTRIES
733
734 001374 012001 L1: MOV (R0)+, R1 ;GET ADDR OF DATA FROM LIST
735 001376 001700 BEQ DMPHLT ;BR IF END OF LIST
736 001400 104400 001536 TYPE ,CRLF
737 001404 010146 MOV R1, -(SP) ;PUSH ADDR ON STACK
738 001406 004037 006350 JSR R0,#B2016 ;PRINT OUT ADDRESS
739
740 001412 032737 000010 011042 BIT #BIT3,FLAG
741 001420 001014 BNE L3
742 001422 012702 000020 MOV #20, R2 ;SET WORD COUNTER = 8
743 001426 005046 L2: CLR -(SP)
744 001430 112116 MOVB (R1)+,(SP)
745 001432 104400 007152 TYPE ,MSG00
746 001436 004037 006336 JSR R0,#B2016
747 001442 003 .BYTE 3
748 001443 001 .BYTE 1
749 001444 005302 DEC R2 ;DECREMENT WORD COUNTER
750 001446 001367 BNE L2 ;BR IF NOT = 0
751 001450 000751 BR L1 ;GET NEXT ENTRY
    
```

```

752
753 001452 012702 000010 L3: MOV #10,R2
754 001456 012146 18: MOV (R1)+,-(SP)
755 001460 104400 007152 TYPE ,MSG00
756 001464 004037 006350 JSR R0,#B2016
757 001470 005302 DEC R2 ;DECREMENT THE WORD COUNT
758 001472 001371 BNE 16
759 001474 000737 BR L1 ;GET NEXT ENTRY
760 001476 000000 DMPHLT: 0 ;RESERVED FOR SW, REG
761 001500 000000 0 ;END OF TABLE FOR SW, REG
762 001502 000001 .RX: .BLKW 1
763 001504 000001 .TX: .BLKW 1
764 001506 000000 0
765 001510 000000 0
766 001512 000000 0
767 001514 177560 TKS: 177560 ;TTY KEYBOARD STATUS REG. ADDRESS
768 001516 177562 TKB: 177562 ;TTY KFYBOARD DATA BUFFER REG. ADDRESS
769 001520 177564 TPS: 177564 ;TTY PRINTER STATUS REG. ADDRESS
770 001522 177566 TPB: 177566 ;TTY PRINTER BUFFER REG. ADDRESS
771 001524 000000 NULL: .WORD 0 ;CONTAINS NULL CHARACTER FOR FILLS
    
```



```

772 ;*****
773 ;   CONSTANTS AND WORKING STORAGE
774 ;*****
775
776
777 001526 037440 000040 QUES:  ,ASCIZ  " ? "
778 001532 005015 000052 ASTRK: ,ASCIZ  <15><12>*"
779 001536 015      000000 CRLF:  ,ASCII  <15>
780 001537 012      000      LF:    ,ASCIZ  <12>
781      001542      ,EVEN
782      001544      $WORK=WORK
783 001542 000000 $CHAR:  0
784 001544 000000 WORK:   0
785 001546 000000 WORK1:  0
786 001550 000000 WORK2:  0
787 001552 000000 WORK3:  0
788 001554 000000 WORK4:  0
789 001556 000000 WORK5:  0
790 001560 000000 WORK6:  0
791 001562 000000 FLAGS:  0
792
793 001564 002604 ;DATA PATTERN ADDRESS TABLE AND PATTERNS
794 001566 003004 DAT:   ,WORD  VDB      ;ADDRESS OF VARIABLE DATA BUFFER
795 001570 003076      ,WORD  DP1      ;ADDRESS OF MESSAGE 1
796 001572 003126      ,WORD  DP2      ;ADDRESS OF MESSAGE 2
797 001574 002604      ,WORD  DP3      ;ADDRESS OF MESSAGE 3
798 001576 003004      ,WORD  DP4      ;ADDRESS OF MESSAGE 4
799 001600 003076      ,WORD  DP5      ;ADDRESS OF MESSAGE 5
800 001602 003126      ,WORD  DP6      ;ADDRESS OF MESSAGE 6
801      ,WORD  DP7      ;ADDRESS OF MESSAGE 7
802
803 001604 000400 ;RECEIVER DATA BUFFER STARTS HERE..
804 002604      IBUF:  ,BLKW  400
805 002604 000100 VDB:   ,BLKW  100      ;VAHIABLE DATA BUFFER
806
807 003004 177      177      DP1:   ,BYTE  177,177
808 003006 040444 052040 042510 ,ASCIZ  "SA THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG."<15><12><001><177><177><177><1
809      ,EVEN
810 003076 177      177      DP2:   ,BYTE  177,177
811
812 003100 041044 030040 031061 ,ASCIZ  "SB 0123456789"<15><12><001><177><177><177><177>
813 003126 177      177      DP3:   ,BYTE  177,177
814
815 003130 041444 041440 046517 ,ASCII  "SC COM-TEST MAYNARD THE QUICK BROWN FOX JUMPED OVER THE LAZY DUG"
816 003230 030040 031061 032063 ,ASCIZ  " 0123456789"<15><12><001><177><177><177><177>
817      ,EVEN
818
819      002604      DP4=VDB      ;SPARE
820      003004      DP5=DP1      ;SPARE
821      003076      DP6=DP2      ;SPARE
822      003126      DP7=DP3      ;SPARE
823
824
825
826
827
828
829
830
831
832
833

```

```

814 ;*****
815 ;   START OF PROGRAM
816 ;*****
817 BEGIN:
818 003254 012706 001070      MOV   $STACK,SP      ;SETUP THE STACK POINTER
819 003260 104414 000340      STPS,PRTY7
820 003264 104420      SUSWR
821 003266 005037 004774      CLP   $PASS          ;CLEAR THE PASS COUNT
822 003272 012737 003254 003462      MOV   $BEGIN,$LPADR ;INITILIZE THE LOOP ADDRESS FOR SCOPE
823 003300 005037 001562      CLR   FLAGS          ;RESET FLAGS
824 003304 000005      RESET
825
826 ;*****
827 ;   GET PARAMETERS FROM OPERATOR
828 ;*****
829 003306 022777 100000 005530      CMP   $10000,$SWR    ;MANUAL INPUT??
830 003314 001063      BNE   GETIT          ;BR IF NO
831
832 ;*****
833 ;   SWITCH REG INPUT ROUTINE
834 ;*****
835 003316 012701 010164      MANIN: MOV   $DN+4,R1      ;PRESET POINTER FOR DN-11
836 003322 005000      CLR   R0             ;CLEAR DISPLAY
837 003324 000000      HALT                ;HALT FOR ISR REQUEST
838 003326 017737 005512 001562      MOV   $SWR, FLAGS    ;SAVE ISR REQUEST INDICATORS
839 003334 032777 020000 005502      BIT   $20000,$SWR    ;IS DN11 SETUP REQUESTER?
840 003342 001402      BEQ   .+6            ;BR IF NO
841 003344 004737 003404      JSR   PC,MANBA       ;GO SETUP DN11
842
843 003350 012701 010406      MOV   $DMB+4,R1      ;PRESET DMB ISR ADDRESS
844 003354 032737 040000 001562      BIT   $40000,FLAGS    ;IS DMB SETUP REQUESTED?
845 003362 001402      BEQ   .+6            ;BR IF NO
846 003364 004737 003404      JSR   PC,MANBA       ;GO SETUP DM11-B
847
848 003370 012701 011004      MOV   $VISR+4,R1     ;PRESET VARIABLE ISR ADDRESS
849 003374 004737 003404      JSR   PC,MANBA       ;GO SETUP VARIABLE ISR
850
851 003400 000137 004200      JMP   SWRSET          ;GO GET OPERATIONAL SWITCHES
852
853 003404 011100      MANBA: MOV   (R1), R0    ;DISPLAY BUS ADDR
854 003406 000000      HALT                ;HALT FOR BUS ADDR
855 003410 017721 005430      MOV   $SWR, (R1)+    ;DISPLAY VECTOR ADDR
856 003414 011100      MOV   (R1), R0      ;HALT FOR VECTOR ADDR
857 003416 000000      HALT
858 003420 017721 005420      MOV   $SWR, (R1)+    ;DISPLAY PRIORITY
859 003424 011100      MOV   (R1), R0      ;HALT FOR PRIORITY
860 003426 000000      HALT
861 003430 017721 005410      MOV   $SWR, (R1)+    ;DISPLAY PARAM #1
862 003434 011100      MOV   (R1), R0      ;BR IF PARAM NOT REQUIRED
863 003436 100410      BMI  MANINX         ;HALT FOR PARAM #1
864 003440 000000      HALT
865 003442 017721 005376      MOV   $SWR, (R1)+    ;
866 003446 011100      MOV   (R1), R0      ;DISPLAY PARAM #2
867 003450 100403      BMI  MANINX         ;BR IF PARAM NOT REQUIRED
868 003452 000000      HALT
869 003454 017721 005364      MOV   $SWR, (R1)+    ;
870 003460 000207      MANINX: RTS          PC

```

```

870 003462 000000          $LPADR: 0
871
872
873
874
875 003464 012700 011000    ;*****
876 003470 104400 007154    ;ISR PARAM INPUT ROUTINE
877 003474 104400 011000    ;*****
878 003500 104400 001526    ;*****
879
880 003504 104404 001544    GETSTR  ,WORK          ;READIN OPERATOR'S RESPONSE
881 003510 123727 001544 000015  CHPB   ,WORK, #15     ;IS IT CAR. RET?
882 003516 001431          BEQ    ,GETBA         ;BR IF YES
883
884 003520 012700 010160    MOV    ,DN, R0        ;PRESET DN ISR ADDR
885 003524 023737 010160 001544  CMP    ,DN, WORK     ;IS IT DN?
886 003532 001004          BNE   ,DMBST         ;BR IF NO
887 003534 052737 020000 001562  BIS   ,DMBST,FLAGS   ;SET DN11 FLAG
888 003542 000417          BR    ,GETBA         ;GO GET DN11 PARAMS
889
890 003544 012700 010402    DMBTST: MOV   ,DMB, R0     ;PRESET DM11B ISH
891 003550 023737 010402 001544  CMP   ,DMB, WORK    ;IS IT DM?
892 003556 001004          BNE  ,NOISR         ;BR IF NO
893 003560 052737 040000 001562  BIS  ,NOISR,FLAGS   ;SET DM11-B FLAG
894 003566 000405          BR   ,GETBA         ;GO GET DM11-B PARAMS
895
896 003570          NOISR:
897 003570 104400 007177    TYPE  ,MSG02         ;<15><12> ISR NOT LOADED!
898
899 003574 000000          HALT
900 003576 000137 003254    JMP   ,GETBA         ;TRY AGAIN
901 003602 010004          MOV   ,R0, R4        ;SAVE POINTER
902 003604 022020          CMP   ,R0+, (R0)+   ;INCREMENT ISR POINTER
903 003606 104400 007222    TYPE  ,MSG03         ;<15><12> BUS ADDRESS=
904 003612 004737 004776    JSR  ,PC,GETANY     ;GET THE BUS ADDR
905
906 003616 104400 007241    GETVA: TYPE ,MSG04     ;<15><12> VECTOR ADDRESS=
907 003622 004737 004776    JSR  ,PC,GETANY     ;GET THE VECTOR ADDR.
908
909 003626 104400 007263    GETPRI: TYPE ,MSG05    ;<15><12> PRIORITY=
910 003632 004737 004776    JSR  ,PC,GETANY     ;GET THE PRIORITY
911
912 003636 005710          TST  ,R0             ;PARAM #1 REQUIRED?
913 003640 100412          BHI  ,GETPRM        ;BR IF NO
914 003642 104400 007277    GETPRM: TYPE ,MSG06    ;<15><12> PARAMS #1=
915 003646 004737 004776    JSR  ,PC,GETANY     ;GET PARAM
916
917 003652 005710          TST  ,R0             ;PARAM #2 REQUIRED?
918 003654 100404          BHI  ,GETPRM        ;BR IF NO
919 003656 104400 007314    TYPE  ,MSG07         ;<15><12> PARAMS #2=
920 003662 004737 004776    JSR  ,PC,GETANY     ;GET PARAM
921
922 003666 016437 000016 003704  GETP3: MOV   ,R4, ARIA   ;IS ASCII PARAM REQUIRED?
923 003674 100424          BHI  ,GETEX         ;BR IF NO
924 003676 104400 007331    TYPE  ,MSG08         ;<15><12> ASCII PARAM=
925 003702 104400          TYPE  ,PRINTOUT ASCII PARAM
    
```

```

926 003704 000000          ARIA: 0
927 003706 104400 001526    TYPE  ,QUES
928
929 003712 104404          GETSTR
930 003714 001604          IBUF
931 003716 012702 001604    MOV   ,IBUF, R2     ;GET ASCII INPUT AND
932 003722 122712 000015 001604  CHPB  ,R15, (R2)    ;PUT IT HERE
933 003726 001407          BEQ  ,GETEX         ;SETUP POINTER
934
935 003730 013703 003704    MOV   ,ARIA, R3     ;WAS NEW DATA EN(ERED)?
936 003734 112223          MOV  ,R2+, (R3)+   ;BR IF NO
937 003736 122712 000015 003704  CHPB  ,R15, (R2)    ;SETUP DEST. POINTER
938 003742 001374          BNE  ,-6           ;MOV INPUT TO DEST.
939 003744 105013          CLR  ,R3           ;LAST DIGIT?
940
941
942 003746 020427 011000    GETEX: CMP   ,R4, #VISR ;WAS THIS THE VARIABLE ISR
943 003752 001461          BEQ  ,SWRNT         ;BR IF YES
944 003754 000137 003464    JMP  ,GETIT         ;GET ANOPHER
945
946
947
948 003760 017701 175532    TTYINT: MOV  ,R1, #TKB ;CLEAR TTY BUFFER
949 003764 042777 000100 175522  BIC  ,R1, #TKS     ;RESET INT, ENABLE
950 003772 042701 177700    BIC  ,R1, #R17700 ;STRIP JUNK
951 003776 022701 000007    CMP  ,R1, #R1
952 004002 001013          BNE  ,R1, #R1
953 004004 022737 000176 011044  CMP  ,R1, #SWREG,SWR
954 004012 001003          BNE  ,R1, #R1
955 004014 052737 000001 011042  BIS  ,R1, #BIT0,FLAG
956 004022 052777 000100 175464 181  BIS  ,R1, #TKS
957 004030 000002          RTI
958 004032 022701 000077 281  CMP  ,R1, #R1
959 004036 001014          BNE  ,R1, #R1
960 004040 012700 001476    MOV  ,R0, #DMP1ST ;IS IT ?
961 004044 012710 001604    MOV  ,R0, #IBUF    ;BR IF NO
962 004050 017737 004744 001502  MOV  ,R0, #IBUF    ;SETUP DUMP LIST
963 004056 017737 004740 001504  MOV  ,R0, #IBUF    ;TO PRINTOUT INPUT BUFFER
964 004064 000137 001374    JMP  ,R0, #IBUF    ;IF SWITCH REG. =0 PRINT RX BUFFER,
965
966 004070 022701 000012    NOQ:  CMP  ,R1, #R1  ;" = " " PRINT TX BUFFER
967 004074 001002          BNE  ,R1, #R1
968 004076 000137 003254    JMP  ,R1, #R1
969
970 004102 104400 010155    RSTART: TYPE ,MFILL
971 004106 000005          RESET
972 004110 105227 000000    INCB  ,R0
973 004114 001375          BNE  ,R0, #R0
974 004116 022737 000176 011044  SWRNT: CMP  ,R0, #SWREG,SWR
975 004124 001007          BNE  ,R0, #SWPNT
976 004126 052737 000002 011042  BIS  ,R0, #BIT1,FLAG
977 004134 104400 007350    TYPE  ,MSG09
978 004140 104422          SETSWI
979 004142 000417          BR   ,R0, #R0
980 004144 104400 007350    XSWPNT: TYPE ,MSG09
981 004150 105777 175340    TSTB ,R0, #TKS
    
```

```

982 004154 100375          BPL      ,-4          ;LOOP
983 004156 017702 175334    MOV      @TKB,R2      ;RESET DONE FLAG
984 004162 017746 004656    MOV      @SWR, -(SP)
985 004166 004037 006350    JSR     R0,@B2016    ;PRINTOUT SWITCHES
986 004172 104400 001536    TYPE    ,CRLF
987 004176 000401          BR       ,+4          ;SKIP OVER HALT
988                                     ;*****
989                                     ;   SET SWITCH OPTIONS
990                                     ;*****
991
992 004200 000000          SWRSET: HALT          ;HALT FOR SWITCH SETUP
993                                     ;SW00=ONE WAY OUT
994                                     ;SW01=ONE WAY IN
995                                     ;SW02=EXTERNAL LOOPBACK
996                                     ;SW03=INTERNAL LOOPBACK
997                                     ;SW04=LOOP ON DATA
998                                     ;SW05=MONITOR INPUT
999                                     ;SW06=MONITOR OUTPUT
1000                                    ;SW07=NO DATA COMPARE
1001                                    ;SW08=EXTERNAL DATA
1002                                    ;SW09=DATA SELECT
1003                                    ;SW10=DATA SELECT
1004                                    ;SW11=DATA SELECT
1005                                    ;SW12=
1006                                    ;SW13=INHIBIT ERROR TYPEOUTS
1007                                    ;SW14=LOOP ON TEST
1008                                    ;SW15=HALT ON ERROR
1009
1010 004202 012737 004210 003462 REST:  MOV    #RESTR1,@LPADR ;SETUP LOOP
1011 004210 017701 004630    RESTR1: MOV    @SWR,R1
1012 004214 000301          SWAB    R1
1013 004216 032777 000017 004620    BIT     @17,@SWR          ;WAS SOME MODE SELECTED?
1014 004224 001003          BNE    ,+10          ;BR IF YES
1015 004226 104400 007637    TYPE    ,MSG21          ;<15><12>NO MODE SELECTED.
1016 004232 000723          BR     RSTART          ;GO GET SWITCH REGISTER.
1017 004234 042701 177761    BIC    @177761, R1      ;STRIP JUNK
1018 004240 016137 001564 011022    MOV    DAT(1), IXDA     ;SETUP INIT DATA ADDR FROM TABLE
1019 004246 005701          TST    R1              ;VARIABLE DATA SPECIFIED?
1020 004250 001010          HNE    SUXCC           ;BR IF NO
1021 004252 032777 000400 004564    BIT     @400, @SWR      ;USE EXTERNAL DATA?
1022 004260 001004          BNE    SUXCC           ;BR IF YES
1023                                     ;*****
1024                                     ;   GET VARIABLE DATA
1025                                     ;*****
1026 004262 104400 007372          TYPE    ,MSG10          ;<15><12> ENTER DATA <15><12>
1027 004266 104404          GETSTR
1028 004270 002604          VDB
1029 004272 012737 001604 011020    SUXCC: MOV    @IBUF, IRDA ;SETUP READ BUFFER ADDR
1030
1031 004300 032777 000400 004536    BIT     @400, @SWR      ;EXTERNAL DATA?
1032 004306 001403          BEQ    SWRNXT          ;BR IF NO
1033 004310 012737 002604 011022    MOV    @VDB, IXDA      ;SETUP BUFFER ADDRESS
1034 004316
1035 004316 012737 003760 000060    SWRNXT: MOV    @TTYINT,@#60 ;SETUP TTY VECTOR
1036 004324 012737 000340 000062    MOV    @340, @#62      ;
1037 004332 012777 000100 175154    MOV    @100, @TKS      ;AND ENABLE INTERRUPTS
    
```

```

1038 004340 012702 001604          MOV     @IBUF, R2
1039 004344 005022          CLR    (R2)+           ;CLEAR INPUT BUFFER
1040 004346 022702 002004          CMP    @IBUF+200,R2
1041 004352 001374          BNE    CLRIB
    
```

```

1042 ;*****
1043 ; SETUP TIMER *
1044 ;*****
1045
1046 004354 012737 000060 006334 SUTIME: MOV #60, MSEC8 ;PRESET COUNTER
1047 004362 012737 006312 000100 MOV #TIMER, 100 ;SETUP LINE CLOCK VECTOR
1048 004370 012737 000340 000102 MOV #340, 102 ;AND PRIORITY
1049 004376 012737 004422 000004 MOV #NOLC, 4 ;SETUP BUS ERROR VECTOR
1050 004404 012737 000340 000006 MOV #340, 6 ;SET UP PRIORITY TO 7
1051 004412 052737 000100 177546 BIS #100, 177546 ;ENABLE LINE CLOCK
1052 004420 000441 BR NORTC
1053
1054 ;BUS ERROR RETURNS HERE IF NO LINE CLOCK
1055
1056 004422 012737 006312 000104 NOLC: MOV #TIMER, 104 ;SETUP RTC VECTOR
1057 004430 012737 000340 000106 MOV #340, 106 ;AND PRIORITY
1058 004436 012737 004472 000004 MOV #16, 4 ;SETUP BUS ERROR VECTOR
1059 004444 012737 000340 000006 MOV #340, 6 ;SET PRIORITY
1060 004452 012737 003100 172542 MOV #1600,, 172542 ;SET COUNTER BUFFER.
1061 004460 012737 000111 172540 MOV #111, 172540 ;ENABLE REAL TIME TIME CLOCK
1062 NOP ;WAIT
1063 BR NORTC ;CONTINUE.
1064 004472 012737 004516 000100 1$: MOV #2$,100 ;TRY LSI CLOCK
1065 004500 104414 000000 STPS,PRTY0 ;LOWER PSW TO 0
1066 004504 005227 000000 3$: INC #0
1067 004510 001375 BNE 3$
1068 004512 104400 007570 TYPE,MSG19 ;NO CLOCK AVAILABLE
1069 004516 012737 006312 000100 2$: MOV #TIMER,100
1070
1071 004524 000137 004530 NORTC: JMP .+4 ;SPARE JUMP
1072 004530 012737 000006 000004 MOV #6,0#4 ;SET TRAP VECTOR
1073 004536 005037 000006 CLR #6 ;SET BUS ERROR VECTOR
1074 004542 012706 001070 MOV #STACK, SP ;SETUP STACK
1075 004546 104414 000000 STPS,PRTY0
1076 004552 012737 006350 011030 MOV #B2016,B2016 ;SETUP BIN TO OCT ADDR
1077
1078 ;*****
1079 ; DO TESTING NOW
1080 ;*****
1081
1082 004560 032737 020000 001562 BIT #20000, FLAGS ;WAS A DN11 SETUP
1083 004566 001402 BEQ DMCHK ;BR IF NO
1084 004570 004737 010200 JSR PC,DMGO ;GO TO DN11 ISR
1085
1086 004574 032737 040000 001562 DMCHK: BIT #40000, FLAGS ;WAS A DM11-B SETUP?
1087 004602 001402 BEQ VIGO ;BR IF NO
1088 004604 004737 010422 JSR PC,DMBGO ;GO TO DM11-B ISR
1089
1090 ;*****
1091 ; GOTO THE MODULE AND RUN *
1092 ;*****
1093
1094 004610 004777 004222 VIGO: JSR PC,#ISR+36 ;GO TO ISR
    
```

```

1095 ;*****
1096 ; END OF PASS ROUTINE
1097 ;*****
1098 004614 005237 004774 EOP: INC SPASS ;INCREMENT PASS COUNTER
1099 004620 005746 TST -(SP) ;PUSH DOWN AND PROTECT STACK.
1100 004622 104416 KBDIN
1101 004624 032777 010000 004212 BIT #SW12,0SWR ;INHIBIT TYPEOUTS?
1102 004632 001034 BNE 2$ ;BF IF YES
1103 004634 104400 007411 TYPE ,MSG11 ;<15><12> END OF PASS
1104 004640 013746 004774 MOV SPASS, -(SP)
1105 004644 004037 006350 JSR R0,$B2016 ;PRINTOUT PASS COUNT
1106 004650 104400 001536 TYPE ,CRLF
1107 004654 032700 000002 BIT #0W1,MODE ;SKIP TRANSMIT TYPEOUT IF OWI
1108 004660 001012 HNE 3$ ;BR IF YES
1109 004662 104400 010060 TYPE ,MSG26 ;TRANSMITTED DATA=
1110 004666 013737 011022 004676 MOV IXDA,4$ ;SET POINTER TO TXBUF
1111 004674 104400 TYPE TXBUFFER
1112 004676 000000 4$: 0
1113 004700 032700 000001 BIT #0W0,MODE ;SKIP RECEIVE TYPEOUT IF OWO
1114 004704 001007 BNE 2$ ;BR IF YES
1115 004706 104400 010106 3$: TYPE ,MSG27 ;RECEIVED DATA=
1116 004712 013737 011020 004722 MOV IPDA,5$ ;SET POINTER TO RXBUF
1117 004720 104400 TYPE RXBUFFER
1118 004722 000000 5$: 0
1119 004724 032777 040000 004112 2$: BIT #BIT14,0SWR ;LOOP ON TEST?
1120 004732 001005 BNE 1$ ;BR IF NO...
1121 004734 016600 000002 MOV 2(SP),R0 ;GET RETURN ADDRESS
1122 004740 104414 000000 STPS,PRTY0
1123 004744 000110 JMP (R0) ;GO BACK TO MODULE.
1124 004746 012706 001070 1$: MOV #STACK, SP ;RESET THE STACK POINTER.
1125 004752 013700 000042 MOV #0#42, R0 ;GET MONITOR ADDRESS
1126 004756 001404 BEQ $DOAGN ;BR IF NONE
1127 004760 004710 JSR PC,(R0) ;GO TO MONITOR
1128 004762 000240 NOP ;SAVE ROOM FOR
1129 004764 000240 NOP ;ACT-11
1130 004766 000240 NOP
1131 004770 000137 000200 $DOAGN: JMP #0#200 ;RESTART TEST
1132 004774 000000 $PASS: 0
1133
1134 ;*****
1135 ; SUBROUTINE TO INPUT OCTAL WORD FROM OPERATOR
1136 ;*****
1137
1138 004776 011046 GETANY: MOV (R0), -(SP) ;PUT WORD ON STACK
1139 005000 004037 006350 JSR R0,$B2016 ;AND TYPE IT
1140 005004 104400 001526 TYPE ,QUES
1141 005010 011037 001544 MOV (R0),WORK ;PRESET FOR DEFAULT (CR)
1142 005014 104406 001544 ANYMOR: ACCEPT ,WORK ;OCTAL READIN
1143 005020 013710 001544 MOV WORK, (R0) ;MOVE IT TO ISR
1144 005024 005720 ANYEX: TST (R0)+ ;BUMP POINTER
1145 005026 000240 NOP
1146 005030 000207 RTS PC ;SUB/ROUTINE EXIT
1147
1148 ;*****
1149 ; ERROR HLT HANDLER
1150 ;*****
    
```

```

1151 005032          $HLT:
1152 005032 104414 000140 STPS,PRTY3 ;LOWER PSW PRIORITY TO 3
1153 005036 005237 005730 $HLOT: INC $ERTTL ;INCREMENT ERROR COUNTER
1154 005042 001775 BEQ $HLOT ;MAKE SURE ITS NOT ZERO
1155 005044 011637 005726 MOV (SP) , $HLTAD ;SAVE ADDRESS OF HLT
1156 005050 162737 000002 005726 SUB #2, $HLTAD ;AND BACK IT UP
1157 005056 010146 MOV R1, -(SP) ;SAVE R1
1158
1159 005060 032777 020000 003756 BIT #BIT13, $SWR ;INHIBIT ERR TYPEOUTS?
1160 005066 001070 TRX ;BR IF YES
1161
1162 005070 104400 001536 TYPE ,CRLF
1163 005074 117701 000626 MOV $HLTAD,R1 ;EXTRACT HLT CODE
1164 005100 006301 ASL R1 ;AND ALIGN IT
1165 005102 016137 005270 005112 MOV EMTAB(R1),,+10 ;GET HEADER ADDRESS
1166 005110 104400 005320 TYPE ,EMO ;AND PRINT HEADER
1167 005114 104400 007430 TYPE ,MSG12 ; < AT LOC >
1168 005120 013746 005726 MOV $HLTAD,-(SP) ;GET HLT ADDRESS
1169 005124 004037 006350 JSR R0,$B2016 ;AND PRINT IT
1170 005130 005701 TST R1 ;HLT CODE = 0?
1171 005132 001446 BEQ TRX ;BR IF YES
1172
1173
1174 005134 022701 000016 CMP #16,R1 ;IS IT HLT+??
1175 005140 001023 BNE 16 ;BR IF NO
1176 005142 005702 TST R2 ;PRINTOUT BAD DATA?
1177 005144 001406 BEQ 26 ;BR IF NO
1178 005146 104400 007767 TYPE ,MSG23 ;< BAD DATA= >
1179 005152 110246 MOV R2,-(SP) ;GET DATA
1180 005154 004037 006336 JSR R0,$B20CT ;AND PRINT IT
1181 005160 003 ;.BYTE 3
1182 005161 001 ;.BYTE 1
1183 005162 005703 26: TST R3 ;PRINT OUT GOOD DATA?
1184 005164 001410 BEQ 36 ;BR IF NO
1185 005166 104400 010003 TYPE ,MSG24 ;< GOOD DATA= >
1186 005172 110346 MOV R3,-(SP) ;GET DATA
1187 005174 004037 006336 JSR R0,$B20CT ;AND PRINT IT
1188 005200 003 ;.BYTE 3
1189 005201 001 ;.BYTE 1
1190 005202 104400 001536 TYPE ,CRLF
1191 005206 000420 36: BR TRX
1192
1193
1194 005210 005702 16: TST R2 ;PRINTOUT RCV CSR?
1195 005212 001405 BEQ TR3 ;BR IF NO
1196 005214 104400 007441 TYPE ,MSG13 ; < RCV CSR=>
1197 005220 010246 MOV R2, -(SP) ;GET DATA
1198 005222 004037 006350 JSR R0,$B2016 ;AND PRINT IT
1199
1200 005226 005703 TR3: TST R3 ;PRINTOUT XMIT CSR?
1201 005230 001407 BEQ TRX ;BR IF NO
1202 005232 104400 007453 TYPE ,MSG14 ; < XMIT CSR=>
1203 005236 010346 MOV R3, -(SP) ;GET DATA
1204 005240 004037 006350 JSR R0,$B2016 ;AND PRINT IT
1205 005244 104400 001536 TYPE ,CRLF
1206
    
```

```

1207 005250 032777 100000 003566 TRX: BIT #BIT15, $SWR ;HALT ON ERROR?
1208 005256 001401 BEQ HLTX ;BR IF NO
1209 005260 000000 HALT
1210
1211 005262 104416 HLTX: KBDIN
1212 005264 012601 MOV (SP)+, R1 ;RESTORE R1
1213 005266 000002 RTI ;AND RETURN TO PROGRAM
1214
1215 005270 005320 EMTAB: EMO
1216 005272 005333 EM1
1217 005274 005343 EM2
1218 005276 005363 EM3
1219 005300 005407 EM4
1220 005302 005422 EM5
1221 005304 005446 EM6
1222 005306 005542 EM7
1223 005310 005565 EM10
1224 005312 005642 EM11
1225 005314 005667 EM12
1226 005316 005711 EM13
1227
005320 051105 047522 020122 EM0: .ASCIZ "ERROR HALT"
005333 127 044501 044524 EM1: .ASCIZ "WAITING"
005343 127 044501 044524 EM2: .ASCIZ "WAITING TO XMIT"
005363 104 026516 030461 EM3: .ASCIZ "DN=11 NOT AVAILAHLE"
005407 104 030516 026461 EM4: .ASCIZ "DN11-ERROR"
005422 047104 030461 041440 EM5: .ASCIZ "DN11 CALL ABANDONED"
005446 041522 020126 052502 EM6: .ASCIZ "RCV BUFFER FULL,END OF MESSAGE CHARACTER(001) WAS NOT FOUND"
005542 040504 040524 041440 EM7: .ASCIZ "DATA COMPARE ERROR"
005565 105 051122 051117 EM10: .ASCIZ "ERROR RCV CSR<CONTENTS OF SELECT 0 REGISTER"
005642 047125 054105 042520 EM11: .ASCIZ "UNEXPECTED INTERRUPT"
005667 116 046530 050040 EM12: .ASCIZ "NXM PRINCIPAL CAR"
005711 116 046530 040440 EM13: .ASCIZ "NXM ALT CAR"
005726 .EVEN
1228 005726 000000 $HLTAD: 0
005730 000000 $ERTTL: 0
    
```

```

1229 ;*****
1230 ; READ A CHAR. ROUTINE
1231 ;*****
1232
1233 ; CALL= GETCHR ;INPUT A CHAR FROM TTY
1234 ; RETURNS HERE WITH CHAR ON STACK
1235 005732 011646 ;READC: MOV (SP),-(SP) ;PUSH THE PC
1236 005734 016666 000004 000002 MOV 4(SP),2(SP) ;SAVE THE PS
1237 005742 105777 173546 TSTB PTKS ;IS RECEIVE DONE
1238 005746 100375 BPL ,-4 ;LOOP IF NO
1239 005750 017737 173542 001542 MOV @TKB ,SCHAR ;SAVE THE CHAR.
1240 005756 042737 177600 001542 BIC #177600,SCHAR ;&STRIP JUNK
1241 005764 013766 001542 000004 MOV SCHAR,4(SP) ;PUT CHAR ON STACK
1242 005772 000002 RTI ;EXIT
1243 ;*****
1244 ; HEAD A STRING ROUTINE
1245 ;*****
1246
1247 ; CALL= GETSTR ;INPUT A STRING OF CHARS FROM TTY
1248 ; ADDR ;TO THIS ADDRESS
1249 ;
1250 005774 011602 ;READS: MOV (SP), R2 ;TERMINATE INPUT WITH LINE FEED
1251 005776 012201 MOV (R2)+, R1 ;SETUP ADDRESS OF INPUT BUFFER
1252 006000 010216 MOV R2, (SP) ;INCREMENT RETURN ADDRESS
1253 ; AND PUT BACK ON STACK
1254 006002 104402 ;GETIC: GETCHR ;GET A CHAR
1255 006004 104400 001542 TYPE ,SCHAR
1256 006010 122726 000136 CMPE #136, (SP)+ ;IS IT BINARY DELIMITER
1257 006014 001011 HNE GOTIC ;BR IF NO
1258 ;
1259 006016 104406 001544 ;OCT: ACCEPT ,SWORK ;GET OCT. CHAR
1260 006022 113721 001544 MOV SWORK,(R1)+ ;STORE OCT. CHAR.
1261 006026 122737 000136 001542 CMPE #136,SCHAR ;TERMINATOR=BIN. DELIMITER?
1262 006034 001370 BNE OCT ;BR IF NO
1263 006036 000761 BR GETIC ;
1264 006040 ;GOTIC:
1265 006040 113721 001542 MOV SCHAR, (R1)+ ;STORE CHAR. IN BUFFER
1266 006044 022737 000015 001542 CMP #15, SCHAR ;IS IT END OF INPUT (CAP. RETURN)
1267 006052 001353 BNE GETIC ;BR IF NO
1268 006054 112721 000012 MOV #12,(R1)+
1269 006060 104400 001537 TYPE ,LF ;TYPE A LINE FEED
1270 006064 112721 000001 MOV #001,(R1)+ ;INSERT RX TERM.
1271 006070 112721 000177 MOV #177, (R1)+ ;AND A FILL
1272 006074 112721 000177 MOV #177, (R1)+ ;INSERT ANOTHER FILL
1273 006100 112721 000177 MOV #177, (R1)+ ;INSERT 3RD FILL
1274 006104 112721 000177 MOV #177, (R1)+ ;INSERT 4TH FILL
1275 006110 105011 CLRB (R1) ;PUT ZEROS AT END
1276 006112 000002 RTI
1277
    
```

```

1278 ;*****
1279 ;ROUTINE TO ACCEPT AN OCTAL NUMBER FROM THE TTY
1280 ;CALL:
1281 ; ACCEPT ,ADDR ;PUT OCTAL NUMBER IN ADDR
1282 ;
1283 ;SACCEPT:
1284 006114 MOV R0,-(SP) ;SAVE P0
1285 006116 010146 MOV R1,-(SP) ;SAVE R1
1286 006120 010246 MOV R2,-(SP) ;SAVE R2
1287 006122 010346 MOV R3,-(SP) ;SAVE R3
1288 006124 016600 000010 MOV 10(SP),R0 ;GET ADDRESS OF WHERE TO PUT NUMBER
1289 006130 005001 18: CLR R1 ;CLEAR PARTIAL NUMBER
1290 006132 012702 000006 MOV #6,R2 ;MAX. # OF DIGITS ALLOWED
1291 006136 104402 28: GETCHP ;GET ONE CHARACTER
1292 006140 112603 MOV R3,(SP)+,R3 ;AND PUT IT IN R3
1293 006142 110337 006310 MOV R3,68
1294 006146 104400 006310 TYPE ,68 ;ECHO THE CHARACTER
1295 006152 022703 000025 CMP #25,R3
1296 006156 001451 BEQ 88
1297 006160 022703 000015 CMP #15,R3 ;WAS THIS CHARACTER A "CR"?
1298 006164 001427 BEQ 56 ;BR IF YES
1299 006166 022703 000040 CMP #40,R3 ;WAS "SPACE" HIT?
1300 006172 001433 BEQ 76 ;BR IF YES
1301 006174 022703 000136 CMP #136,R3 ;WAS " " HIT?
1302 006200 001430 BEQ 76 ;BR IF YES
1303 006202 032703 000110 BIT #110,R3 ;INSURE THE CHARACTER IS
1304 006206 001011 BNE 48 ;A DIGIT BETWEEN 0 AND 7.
1305 ;
1306 006210 005302 DEC R2 ;CHECK NUMBER OF CHARACTERS
1307 006212 002407 BLT 48 ;BR IF TO MANY
1308 006214 006301 ASL R1 ;POSITION PARTIAL NUMBER
1309 006216 006301 ASL R1 ;FOR THIS DIGIT
1310 006220 006301 ASL R1
1311 006222 042703 177770 BIC #'C<7>,R3 ;GET RID OF THE ASCII JUNK
1312 006226 050301 BIS R3,R1 ;COMBINE THIS DIGIT WITH PARTIAL
1313 006230 000742 BR 28 ;GO GET ANOTHER DIGIT
1314 006232 104400 001526 48: TYPE ,QUES ;TYPE "?"
1315 006236 104400 001536 TYPE ,CRLF ;TYPE CARRAGE RETURN AND LINE FEED.
1316 006242 000732 BR 18 ;GO START OVER
1317 006244 104400 001537 56: TYPE ,LF ;FOLLOW "CR" WITH A "LF"
1318 006250 022702 000006 CMP #6,R2 ;WERE ANY DIGITS INPUT
1319 006254 001002 BNE ,+6 ;BR IF YES
1320 006256 013001 MOV @R0+,R1 ;USE OLD DATA
1321 006260 005740 TST -(R0) ;BACKUP R0--
1322 006262 010130 78: MOV R1,@R0+ ;PASS THE NUMBER TO THE USER
1323 006264 010066 000010 MOV R0,10(SP) ;SET FOR RETURN
1324 006270 012603 MOV (SP)+,R3 ;RESTORE R3
1325 006272 012602 MOV (SP)+,R2 ;RESTORE R2
1326 006274 012601 MOV (SP)+,R1 ;RESTORE R1
1327 006276 012600 MOV (SP)+,R0 ;RESTORE R0
1328 006300 000002 RTI
1329 006302 104400 010150 88: TYPE, CTLU
1330 006306 000710 BR 18
1331 006310 000 000 68: .BYTE 0,0 ;STORAGE FOR ASCII CHAR, AND TERMINATOR
    
```

```

1332 ;*****
1333 ; CLOCK INTERRUPT ROUTINE
1334 ;*****
1335 006312 005337 006334 TIMER: DEC MSECS ;COUNT 60 CYCLES
1336 006316 001005 BNE TINEX ;BR IF NOT 60
1337 006320 012737 000060 006334 MOV #60, MSECS ;RESTORE COUNT
1338 006326 005237 011032 INC TIME ;INCREMENT SECONDS
1339 006332 TIMEX:
1340 006332 000002 RTI ;RETURN FROM INTERRUPT
1341 006334 000000 MSECS: 0
1342 ;*****
1343 ;BINARY TO OCTAL (ASCII) AND TYPE
1344 ;%B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1345 ;CALL:
1346 ; MOV NUM,-(SP) ;NUMBER TO BE TYPED
1347 ; JSR RO,%B2OCT ;CALL FOR TYPEOUT
1348 ; .BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1349 ; .BYTE M ;M=1 OR 0
1350 ; ;1=TYPE LEADING ZEROS
1351 ; ;0=SUPPRESS LEADING ZEROS
1352 ;
1353 ;%B201---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST %B5OCT OR %B2016
1354 ;CALL:
1355 ; MOV NUM,-(SP)
1356 ; JSR RO,%B201
1357 ;
1358 ;%B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1359 ;CALL:
1360 ; MOV NUM,-(SP)
1361 ; JSR RO,%B2016
1362 ;
1363 006336 112037 006543 %B2OCT: MOVB (RO)+,%SOMODE+1 ;PICKUP THE NUMBER OF DIGITS TO TYPE
1364 006342 112037 006541 MOVB (RO)+,%SOFILL ;GET THE ZERO FILL SWITCH
1365 006346 000406 BK $R201
1366 006350 112737 000001 006541 %B2016: MOVB #1,%SOFILL ;SET THE ZERO FILL SWITCH
1367 006356 112737 000006 006543 MOVB #6,%SOMODE+1 ;SET FOR SIX(6) DIGITS
1368 006364 112737 000005 006540 %B201: MOVB #5,%SOCNT ;SET THE ITERATION COUNT
1369 006372 010346 MOV R3,-(SP) ;SAVE R3
1370 006374 010446 MOV R4,-(SP) ;SAVE R4
1371 006376 010546 MOV R5,-(SP) ;SAVE R5
1372 006400 113704 006543 MOVB %SOMODE+1,K4 ;GET THE NUMBER OF DIGITS TO TYPE
1373 006404 005404 NEC R4
1374 006406 062704 000006 ADD #6,R4 ;SUBTRACT IT FOR MAX. ALLOWED
1375 006412 110437 006542 MOV R4,%SOMODE ;SAVE IT FOR USE
1376 006416 113704 006541 MOVB %SOFILL,R4 ;GET THE ZERO FILL SWITCH
1377 006422 016605 000010 MOV 10(SP),R5 ;PICKUP THE INPUT NUMBER
1378 006426 005003 CLR R3 ;CLEAR THE OUTPUT WORD
1379 006430 006105 1$: ROL R5 ;ROTATE MSB INTO "C"
1380 006432 000404 BK 36 ;GO DO MSB
1381 006434 006105 2$: ROL R5 ;FORM THIS DIGIT
1382 006436 006105 ROL R5
1383 006440 006105 ROL R5
1384 006442 010503 MOV R5,R3
1385 006444 006103 3$: ROL R3 ;GET LSB OF THIS DIGIT
1386 006446 105337 006542 DECB %SOMODE ;TYPE THIS DIGIT?
1387 006452 100016 BPL 7$ ;BR IF NO
    
```

```

1388 006454 042703 177770 BIC #177770,R3 ;GET RID OF JUNK
1389 006460 001002 BNE 4$ ;TEST FOR 0
1390 006462 005704 TST R4 ;SUPPRESS THIS 0?
1391 006464 001403 BEQ 5$ ;BR IF YES
1392 006466 005204 4$: INC R4 ;DON'T SUPPRESS ANYMORE 0'S
1393 006470 052703 000060 BIS #'0,R3 ;MAKE THIS DIGIT ASCII
1394 006474 052703 000040 5$: BIS #' ,R3 ;MAKE ASCII IF NOT ALREADY
1395 006500 110337 006536 MOVB R3,%$ ;SAVE FOR TYPING
1396 006504 104400 006536 TYPE #,$ ;GO TYPE THIS DIGIT
1397 006510 105337 006540 7$: DECB %SOCNT ;COUNT BY 1
1398 006514 003347 HGT 2$ ;BR IF MORE TO DO
1399 006516 002402 BLT 6$ ;BR IF DONE
1400 006520 005204 INC R4 ;INSURE LAST DIGIT ISN'T A BLANK
1401 006522 000744 BK 2$ ;GO DO THE LAST DIGIT
1402 006524 012605 6$: MOV (SP)+,R5 ;RESTORE R5
1403 006526 012604 MOV (SP)+,R4 ;RESTORE R4
1404 006530 012603 MOV (SP)+,R3 ;RESTORE R3
1405 006532 012616 MOV (SP)+,(SP) ;SET THE STACK FOR RETURNING
1406 006534 000200 RTS R0 ;RETURN
1407 006536 000 .BYTE 0 ;STORAGE FOR ASCII DIGIT
1408 006537 000 .BYTE 0 ;TERMINATOR FOR TYPE ROUTINE
1409 006540 000 %SOCNT: .BYTE 0 ;OCTAL DIGIT COUNTER
1410 006541 000 %SOFILL: .BYTE 0 ;ZERO FILL SWITCH
1411 006542 000000 %SOMODE: 0 ;NUMBER OF DIGITS TO TYPE
1412
1413
1414 ;*****
1415 ;TRAP HANDLER
1416 006544 010046 STRAP: MOV RO,-(SP) ;SAVE RO
1417 006546 016600 000002 MOV 2(SP),RO ;GET TRAP ADDRESS
1418 006552 005740 TST -(RO) ;BACKUP BY 2
1419 006554 111000 MOVB (RO),RO ;GET RIGHT BYTE OF TRAP
1420 006556 016000 006564 MOV $TRPAD(RO),RO ;INDEX TO TABLE
1421 006562 000200 RTS R0 ;GO TO ROUTINE
1422
1423 ;TRAP TABLE
1424 ; ROUTINE
1425 ; -----
1426
1427 006564 001100 STRPAD: $TYPE
1428 104400 TYPE=TRAP+0
1429 006566 005732 $READC GETCHR=TRAP+2
1430 104402
1431 006570 005774 $READS GETSTR=TRAP+4
1432 104404
1433 006572 006114 $ACCEPT ACCEPT=TRAP+6
1434 104406
1435 006574 006610 $RWAIT RWAIT=TRAP+10
1436 104410
1437 006576 006620 $XWAIT XWAIT=TRAP+12
1438 104412
1439 006600 007042 .STPS STPS=TRAP+14
1440 104414
1441 006602 006732 .KBDIN KBDIN=TRAP+16
1442 104416
1443 006604 007064 .SUSWR
    
```

```

1444          104420          .SUSWR=TRAP+20
1445 006606 006750          .SETSWI
1446          104422          SETSWI=TRAP+22
1447
1448 ;*****
1449 ; SPECIAL PRINTOUT ROUTINES
1450 ;*****
1451
1452 006610 104400 007462 $RWAIT: TYPE ,MSG15 ;<15><12> WAITING AT LOC <SP>
1453 006614 000137 006624 JMP WAI1PO
1454
1455 006620 104400 007505 $XWAIT: TYPE ,MSG16 ;<15><12> WAITING FOR CLEAR TO SEND AT LOC <SP><SP>
1456
1457 006624 011646 WAITPU: MOV (SP), -(SP) ;SETUP ADDRESS OF CALL FOR PRINTOUT
1458 006626 004037 006350 JSR RO,$B2016 ;PRINTOUT ADDRESS
1459
1460 006632 104400 010021 TYPE ,MSG25 ; <DMBB LINE STATUS REG= >
1461
1462 006636 016646 000004 MOV 4(SP), -(SP) ;MOV CSR TO BOTTOM OF STACK
1463 006642 016666 000004 000006 MOV 4(SP), 6(SP) ;MOVE PSW UP A WORD
1464 006650 016666 000002 000004 MOV 2(SP), 4(SP) ;MOVE PC UP A WORD
1465 006656 012616 MOV (SP)+, (SP) ;MOVE CSR UP A WORD
1466 006660 004037 006350 JSR RO,$B2016 ;GO PRINT CSR
1467
1468 006664 104400 007564 TYPE ,MSG18 ;<,><15><12>
1469 ;GIVE IT LINE FEED
1470
1471 006670 000002 RTI ;AND RETURN TO CALLER
1472
1473 ;*****
1474 ;
1475 ;POWER DOWN ROUTINE.
1476 ;SINCE INIT IS ISSUED IN A PWR DN/UP SEQUENCE,
1477 ;PROGRAM MUST BE RESTARTED AGAIN.
1478 ;
1479
1480 006672 012737 006704 000024 $PWRDN: MOV $SPWRUP,@#24
1481 006700 000000 HALT
1482 006702 000776 BR .-2
1483
1484 ;*****
1485 ;
1486 ;POWER UP ROUTINE.
1487 ;MESSAGE "POWER HAS FAILED..." WILL BE PRINTED OUT.
1488 ;PROGRAM WILL BE RESTARTED.
1489 ;
1490
1491 006704 012737 006672 000024 $SPWRUP: MOV $PWRDN,@#24
1492 006712 012706 001070 MOV $STACK,SP
1493 006716 104400 007616 TYPE ,MSG20 ;<15><12> POWER FAILED..
1494 006722 104414 000000 STPS,PRTY0
1495 006726 000137 000200 JMP @#200
1496
1497 006732 032737 000001 011042 .KBDIN: BIT $BIT0,FLAG ;TEST "G FLAG
1498 006740 001437 BEQ OUT ;NO, EXIT
1499 006742 042777 000100 172544 BIC $100,@TK5 ;CLEAR TTY IE
    
```

```

1500 006750 104400 010131 .SETSWI:TYPE ,SWEQ ;TYPE SWR=
1501 006754 017746 002064 MOV @SWR,-(SP) ;SET UP OCTAL TYPEOUT
1502 006760 004037 006350 JSR RO,$B2016 ;DO IT
1503 006764 104400 010141 TYPE ,NEW ;TYPE NEW=
1504 006770 017737 002050 001544 MOV @SWR,WORK ;SET UP FOR CR DEFAULT
1505 006776 104406 001544 ACCEPT ,WORK ;GET VALUE
1506 007002 013777 001544 002034 MOV WORK,@SWR ;REPLACE IT
1507 007010 032737 000002 011042 BIT $BIT1,FLAG ;SEE HOW WE GOT HERE
1508 007016 001005 BNE 1$ ;WRUNG WAY?
1509 007020 005777 172472 TST @TKB ;CLEAR BUFFER
1510 007024 052777 000100 172462 BIS $100,@TK5 ;RESET TTY IE
1511 007032 042737 000003 011042 1$: BIC $BIT0+BIT1,FLAG ;CLEAR FLAG BITS
1512 007040 000002 OUT: RTI ;EXIT
1513
1514 007042 042766 000340 000002 .STPS: BIC $PRTY7,2(SP) ;CLEAR OUT PRIORITY BITS
1515 007050 057666 000000 000002 BIS @(SP),2(SP) ;SET NEW PRIORITY
1516 007056 062716 000002 ADD $2,(SP) ;SETUP EXIT
1517 007062 000002 RTI ;EXIT
1518
1519 007064 013746 000006 .SUSWR: MOV 6,-(SP) ;SAVE 6 ON STACK
1520 007070 013746 000004 MOV 4,-(SP) ;SAVE 4 ON STACK
1521 007074 012737 007122 000004 MOV #18,4 ;SETUP TIMEOUT
1522 007102 012737 000340 000006 MOV #340,6 ;SET PRIORITY
1523 007110 022777 177777 001726 CMP #-1,@SWR ;TEST FOR 177570
1524 007116 001402 BEQ 2$ ;NOT ALL 1'S
1525 007120 000407 BR 3$ ;IT'S THERE - EXIT
1526 007122 022626 1$: CMP (SP)+,(SP)+ ;ADJUST STACK AFTER TRAP
1527 007124 012737 000176 011044 2$: MOV $SWREG,SWR ;REPLACE HARDWARE REGISTERS
1528 007132 012737 000174 011046 MOV $DISPREG,DISPLAY ;WITH SOFTWARE REGISTERS
1529 007140 012637 000004 3$: MOV (SP)+,4 ;RESTORE 4
1530 007144 012637 000006 MOV (SP)+,6 ;RESTORE 6
1531 007150 000002 RTI ;EXIT
    
```



```

1532 ;*****
1533 ;
1534 ; AREA RESERVED FOR MOST ASCIZ MESSAGES.
1535 ;
1536 ;*****
1537
1538
1539
1540 007152 000040 MSG00: .ASCIZ / /
007154 005015 047111 042524 MSG01: .ASCIZ <15><12>/INTERFACE TYPE /
007177 015 044412 051123 MSG02: .ASCIZ <15><12>/ISR NOT LOADED!!/
007222 005015 052502 020123 MSG03: .ASCIZ <15><12>/BUS ADDRESS=/
007241 015 053012 041505 MSG04: .ASCIZ <15><12>/VECTOR ADDRESS=/
007263 015 050012 044522 MSG05: .ASCIZ <15><12>/PRIORITY=/
007277 015 050012 051101 MSG06: .ASCIZ <15><12>/PARAMS #1=/
007314 005015 040520 040522 MSG07: .ASCIZ <15><12>/PARAMS #2=/
007331 015 040412 041523 MSG08: .ASCIZ <15><12>/ASCII PARAM=/
007350 005015 042523 020124 MSG09: .ASCIZ <15><12>/SET SWITCHES.../
007372 005015 047105 042524 MSG10: .ASCIZ <15><12>/ENTER DATA/<15><12>
007411 015 042412 042116 MSG11: .ASCIZ <15><12>/END OF PASS /
007430 040440 020124 047514 MSG12: .ASCIZ / AT LOC /
007441 040 041522 020126 MSG13: .ASCIZ / RCV CSR=/
007453 040 041530 051123 MSG14: .ASCIZ / XCSR=/
007462 005015 053440 044501 MSG15: .ASCIZ <15><12>/ WAITING AT LOC /
007505 015 053412 044501 MSG16: .ASCIZ <15><12>/WAITING FOR CLEAR TO SEND AT LOC /
007551 040 020041 041440 MSG17: .ASCIZ / ! CSR= /
007564 006456 000012 MSG18: .ASCIZ /,<15><12>
007570 005015 047516 041440 MSG19: .ASCIZ <15><12>/NO CLOCKS AVAILABLE/
007616 005015 047520 042527 MSG20: .ASCIZ <15><12>/POWER FAILED../
007637 015 047012 020117 MSG21: .ASCIZ <15><12>/NO MODE SELECTED./
007663 015 044412 020106 MSG22: .ASCII <15><12>/IF CALLING,DIAL NUMBER/
007713 015 044412 020106 MSG22: .ASCIZ <15><12>/IF ANSWERING,PLACE MODEM IN AUTO-ANSWER/<15><12>
007767 073 041040 042101 MSG23: .ASCIZ /; BAD DATA=/
010003 040 020040 047507 MSG24: .ASCIZ / GOOD DATA=/
010021 015 020012 046504 MSG25: .ASCIZ <15><12>/ DMBB LINE STATUS REGISTER= /
010060 005015 051124 047101 MSG26: .ASCIZ <15><12>/TRANSMITTED DATA=/<15><12>
010106 005015 042522 042503 MSG27: .ASCIZ <15><12>/RECEIVED DATA=/<15><12>
010131 015 051412 051127 SWEQ: .ASCIZ <15><12>/SWR= /
010141 040 042516 036527 NEQ: .ASCIZ / NLW= /
010150 052536 005015 000 CTLU: .ASCIZ /~U/<15><12>
010155 177 000177 MPILL: .ASCIZ <177><177>
.EVEN
    
```

```

(1) ;*****
1541 ; DN-11 INTERFACE SERVICE PARAMS
1542 ;*****
1543 010160 047104 000040 DN: .ASCIZ "DN "
1544 010164 175200 DNBA: 175200 ;BUS ADDRESS
1545 010166 000350 DNIV: 350 ;INTERRUPT VECTOR
1546 010170 000200 DNPRI: 200 ;PRIORITY
1547 010172 177777 DNPARI: 177777 ;NOT USED
1548 010174 177777 DNPARI: 177777 ;NOT USED
1549 010176 010356 DNPARI: DIALNU ;ADDRESS OF DIAL #
1550
1551
1552
1553 100000 PWI=100000 ;POWER INDICATOR
1554 040000 ACR=40000 ;ABANDON CALL AND RETRY
1555 010000 DLO=10000 ;DATA LINE OCCUPIED
1556 000040 DSS=40 ;DATA SET STATUS
1557 000020 PND=20 ;PRESENT NEXT DIGIT
1558 000002 DP=2 ;DIGIT PRESENT
1559 000001 CRQ=1 ;CALL REQUEST
1560
    
```

```

1561 ;*****
1562 ; STAKT OF DN-11 CODE
1563 ;*****
1564 010200 013704 010164 DNGO: MOV DNBA, R4 ;SETUP BUS ADDR
1565 010204 005014 CLR (R4) ;RESET DN-11
1566 010206 005037 011032 CLR TIME ;RESET TIMER
1567 010212 032737 000002 011032 BIT #2, TIME ;AND WAIT 2 SECS
1568 010220 001774 BEQ #-6
1569 010222 012703 010356 MOV #DIALNO,R3 ;SETUP DIAL # ADDRESS
1570 010226 012714 000001 MOV #CRQ, #CSR ;SET CALL REQUEST
1571 010232 032714 100000 BIT #PW1, #CSR ;IS DN AVAILABLL
1572 010236 001425 BEQ DNL1X ;BR IF YES
1573
1574 010240 011402 MOV #CSR, R2 ;SETUP CSR FOR PRINTOUT
1575 010242 005003 CLR R3
1576 010244 104003 HLT+3 ;PRINTOUT "DN NOT AVAILABLE"
1577 010246 000137 010200 JMP DNGO ;RESTART
1578
1579 010252 032714 000200 DNL1: BIT #DONE, #CSR ;IS DONE FLAG SET?
1580 010256 001775 BEQ DNL1 ;WAIT IF NO
1581 010260 042714 000200 BIC #DONE, #CSR ;RESET DONG
1582 010264 032714 140000 BIT #ACP+PW1,#CSR ;ANY ERRORS?
1583 010270 001003 BNE DNL1E ;BR IF YES
1584
1585
1586 010272 032714 000020 BIT #PND, #CSR ;IS PRESENT NEXT DIGIT SET
1587 010276 001005 BNE DNL1X ;BR IF YES
1588
1589 010300 011402 DNL1E: MOV #CSR, R2 ;SETUP CSR FOR PRINTOUT
1590 010302 005003 CLR R3
1591 010304 104004 HLT+4 ;PRINTOUT "DN ERROR"
1592 010306 000137 010200 JMP DNGO ;RESTART
1593
1594 010312 112364 000001 DNL1X: MOVB (R3)+, 1(R4) ;LOAD NEXT DIGIT
1595 010316 052714 000002 BIS #DP, #CSR ;SET DIGIT PRESENT
1596 010322 105713 TSTH (R3) ;WAS THAT LAST CHAR?
1597 010324 001352 BNE DNL1 ;BR IF NO
1598
1599 010326 032714 000200 DNL2: BIT #DONE, #CSR ;WAIT FOR DONE FLAG
1600 010332 001775 BEQ DNL2
1601
1602 010334 032714 040000 BIT #ACR, #CSR ;WAS CALL ABANDONED?
1603 010340 001405 BEQ DNEX ;BR IF NO
1604
1605 010342 011402 MOV #CSP, R2 ;SETUP CSR FOR PRINTOUT
1606 010344 005003 CLR P3
1607 010346 104005 HLT+5 ;PRINTOUT "CALL ABANDONED"
1608 010350 000137 010200 JMP DNGO ;RESTART
1609
1610 010354 000207 DNEX: RTS PC ;RETURN TO ITEMP
1611
1612 010356 03447 032467 031460 DIALNO: .ASCIZ "8975030" ;NUMBER TO DIAL
1613 010364 000060
1614 010366 000000 000000 000000 0,0,0,0,0,0
1615 010374 000000 000000 000000
    
```

```

1616 ;*****
1617 ; DM11-B INTERFACE SERVICE PARAMS
1618 ;*****
1619
1620
1621 010402 046504 000102 DMB: .ASCIZ "DMB" ;ISR NAME
1622 010406 170500 MBA: 170500 ;BUS ADDRESS
1623 010410 000310 IV: 310 ;VECTOR ADDRESS
1624 010412 000200 PRIO: 200 ;PRIORITY
1625 010414 000000 PARA1: 0 ;PARAM #1
1626 010416 177777 PARA2: 177777 ;PARAM #2
1627 010420 177777 PARA3: 177777 ;PARAM #3
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
    
```

```

1638 ;*****
1639 ; DM-11B INTERFACE SERVICE ROUTINE
1640 ;*****
1641 DMBGO: NOP
1642 MOV MBA, R4 ;SETUP RUS ADUR INDEX
1643 CLR TIME ;RESET TIMER
1644
1645
1646 BIS #CS+CM, @CSR ;CLEAR DM-11 B
1647 BIT #BUSY, @CSR ;WAIT TIL FREE
1648 BNE ,-4
1649 MOV PARA1, @CSR ;SELECT LINE #
1650 BIS #DTR+LE,2(R4) ;SET DATA TERM RDY & LINE ENABLE
1651 BIT #20000,FLAGS ;HAS DM11 MADE CONNECTION YET?
1652 BNE IS ;BR IF YES
1653 TYPE ,MSG22 ;TYPE "MAKE CONNECTION"
1654
1655 BIT #OWO+XLB,@SWR ;IS MODE = OWO OR XLB
1656 BEQ REX1 ;BR IF NO
1657
1658
1659 STARTX: BIS #RQTS, 2(R4) ;SET REQUEST TO SEND
1660
1661 CTSW: BIT #CTS, LSTAT(R4) ;IS CLEAR TO SEND SET?
1662 BNE CTSOK ;BR IF YES
1663 CMP TIME, #36 ;30 SECS ELAPSED?
1664 BLO CTSW ;BR IF NO
1665 MOV LSTAT(R4),-(SP) ;TYPE CONTENTS OF RCSR
1666 BIT #SW12,@SWR ;INHIBIT TYPEOUTS?
1667 BNE IS ;BR IF YES
1668 XWAIT ;PRINTOUT "WAITING FOR CTS"
1669 CLR TIME ;RESET TIMER
1670 BR CTSW ;WAIT SOME MORE
1671
1672 CTSOK:
1673
1674 REX: MOV #-1,SETTLE ;SET UP DELAY FLAG
1675 CLR TEMP1
1676 MOV #14,TEMP2
1677 ADD #1,TEMP1
1678 BNE ,-6
1679 DEC TEMP2
1680 BNE ,-14
1681 REX1: RTS PC ;RETURN TO CONTROL PROGRAM
1682
1683 TEMP1: 0
1684 TEMP2: 0
1685 OWO=1
1686 OI=2
1687 TLB=4
1688 XLB=4
1689 ILB=10
1690
1691 ; RCSR EQUATES
1692 .EQUIV R4,RCSR
1693 .EQUIV R4,CSR
    
```

```

1694 RI=100000 ;RING INDICATOR
1695 CF=100 ;CARRIER FLAG
1696 CTS=40 ;CLEAR TO SEND
1697 SRD=10000 ;SEC. RECEIVE DATA
1698 CS=4000 ;CLEAR SCAN
1700 CM=2000 ;CLEAR MUX
1701 MM=1000 ;MAINT MODE
1702 STEP=400 ;STEP
1703 DONE=200 ;DONE
1704 IE=100 ;INTERRUPT ENABLE
1705 SE=40 ;SCAN ENABLE
1706 BUSY=20 ;BUST
1707 LINE=17 ;LINE NUMBER
1708
1709 ;LINE STATUS REGGRSTER EQUATES
1710 LSTAT=2
1711
1712 RQTS=4 ;REQUEST TO SEND
1713 DTR=2 ;DATA TERMINAL READY
1714 LE=1 ;LINE ENABLE
1715
    
```

```

1716          011000          .=11000
1717          ;*****
1718          ; THE INTERFACE SERVICE ROUTINE IS LOADED HERE
1719          ;*****
1720          ;THE FOLLOWING 18 WORDS ARE USED AS
1721          ;THE LINKAGE BETWEEN THE ISR AND THE
1722          ;CONTROL PROGRAM.
1723          ;
1724
1725 011000      VISR:
1726 011000 000002  ISR: .BLKW 2 ;.ASCIZ "DXX"
1727 011004 000001  BA: .BLKW 1 ;175610          ;BUS ADDRESS
1728 011006 000001  VA: .BLKW 1 ;300          ;VECTOR ADDRESS
1729 011010 000001  PRIOR: .BLKW 1 ;340          ;PRIORITY
1730 011012 000001  PARAM1: .BLKW 1 ;-1          ;PARAM #1
1731 011014 000001  PARAM2: .BLKW 1 ;-1          ;PARAM #2
1732 011016 000001  PARAM3: .BLKW 1 ;-1          ;PARAM #3
1733
1734 011020 000001  IRDA: .BLKW 1 ;.WORD 0          ;INITIAL READ DATA ADDRESS
1735 011022 000001  IXDA: .BLKW 1 ;.WORD 0          ;INITIAL XMIT DATA ADDRESS
1736 011024 000001  SETTLE: .BLKW 1 ;.WORD 0          ;LINE SETTLE DELAY FLAG
1737 011026 000001  IKCC: .BLKW 1 ;.WORD 0          ;INITIAL RCV CHAR COUNT
1738 011030 000001  B2O16: .BLKW 1 ;.WORD 0          ;ADDR OF BIN TO OCT TYPE ROUTINE
1739 011032 000001  TIME: .BLKW 1 ;.WORD 0          ;TIMER
1740 011034 000001  MODEA: .BLKW 1 ;.WORD 0          ;ADDR OF ITEP PARAMS
1741 011036 000001          .BLKW 1 ;.WORD START          ;ISR ENTRY ADDRESS
1742 011040
1743 011040 000001  TX.TERM: .BLKB 1 ;.BYTE 000          ;TRANSMITER TERMINATING CHAR.
1744 011041
1745 011041 000001  RX.TERM: .BLKB 1 ;.BYTE 012          ;RECEIVER TERMINATING CHAR.
1746
1747 011042 000001  FLAG: .BLKW 1
1748 011044 177570  SwR: 177570
1749 011046 177570  DISPLAY:177570
1750 011050 000001  START: .BLKW 1 ;NOP
1751 011052 000001          .BLKW 1 ;NOP
1752 011054 000001          .BLKW 1 ;NOP
1753 011056 000001          .BLKW 1 ;NOP
1754 011060 000001  CONT.: .BLKW 1 ;NOP
1755 011062 000001          .BLKW 1 ;NOP
1756 011064 000001          .BLKW 1 ;NOP
1757 011066 000001  FINI: .BLKW 1 ;MOV #340,PS          ;LOCK OUT INTERRUPTS.
1758 011070 000001          .BLKW 1 ;MOV #ENTER,2(SP)          ;SET FOR RETURN IF SW14=0
1759 011072 000001          .BLKW 1 ;JSR PC,SAVE05          ;GO SAVE YOUR REGISTERS.
1760 011074 000001          .BLKW 1 ;RTS PC          ;EXIT
1761
1762 011076 000001  ENTER: .BLKW 1 ;JSR PC,REST05          ;GO AND RESTORE REGISTERS
1763 011100 000001          .BLKW 1 ;MOV #-1,DELAY          ;INDICATE DELAY FOR TX.
1764 011102 000001          .BLKW 1 ;JMP CONT.          ;CONTINUE IN PROGRAM
1765 011104 000001          .END
    
```

```

ACCEPT= 104406          725 1142 1259 1434# 1505
ACR = 040000          1554# 1582 1602
ANYEX 005024          1144#
ANYMOR 005014          1142#
ARIA 003704          922# 935
ASTRK 001532          702 778#
BA 011004          1727#
BEGIN 003254          657 817# 822 900 968
BIT0 = 000001          620# 955 1497 1511
BIT00 = 000001          610# 620
BIT01 = 000002          609# 619
BIT02 = 000004          608# 618
BIT03 = 000010          607# 617
BIT04 = 000020          606# 616
BIT05 = 000040          605# 615
BIT06 = 000100          604# 614
BIT07 = 000200          603# 613
BIT08 = 000400          602# 612
BIT09 = 001000          601# 611
BIT1 = 000002          619# 976 1507 1511
BIT10 = 002000          600#
BIT11 = 004000          599#
BIT12 = 010000          598#
BIT13 = 020000          597# 1159
BIT14 = 040000          596# 1119
BIT15 = 100000          595# 1207
BIT2 = 000004          618#
BIT3 = 000010          617# 708 740
BIT4 = 000020          616# 712
BIT5 = 000040          615# 717 728
BIT6 = 000100          614#
BIT7 = 000200          613#
BIT8 = 000400          612#
BIT9 = 001000          611#
BPTVEC= 000014          627#
BUSY = 000020          1647 1706#
B2O16 011030          1076# 1738#
CF = 000100          1696#
CLRIB 004344          1039# 1041
CM = 002000          1646 1700#
CONT. 011060          1754#
CRLF 001536          721 736 779# 986 1106 1162 1190 1205 1315
CRQ = 000001          1559# 1570
CS = 004000          1646 1699#
CTLU 010150          1329 1540#
CTS = 000040          1661 1697#
CTSOK 010556          1662 1672#
CTSW 010512          1661# 1664 1670
DAT 001564          793# 1018
DIALNO 010356          1549 1569 1612#
DISPLA 011046          1528# 1749#
DISPRE 000174          652# 1528
DLO = 010000          1555#
DMB 010402          842 890 891 1621#
DMBGO 010422          1088 1641#
DMBTST 003544          886 889#
    
```


VDB	002604	793	804*	808	1028	1033													
VIGO	004610	1087	1094*																
VISR	011000	847	875	877	942	1725*													
WAITPO	006624	1453	1457*																
WORK	001544	723*	725	730	782	784*	880	881	885	891	1141*	1142	1143	1504*					
		1505	1506																
WORK1	001546	785*																	
WORK2	001550	786*																	
WORK3	001552	787*																	
WORK4	001554	788*																	
WORK5	001556	789*																	
WORK6	001560	790*																	
XLB	= 000004	1655	1688*																
XSWPNT	004144	975	980*																
XWAIT	= 104412	1438*	1668																
XACCEP	006114	1283*	1433																
SBDOCT	006336	746	1180	1187	1363*														
SB201	006364	1365	1368*																
SB2016	006350	738	756	985	1076	1105	1139	1169	1198	1204	1366*	1458	1466	1502					
SCHAR	001542	706	710	714	783*	1239*	1240*	1241	1255	1261	1265	1266							
SDDAGN	004770	1126	1131*																
SERTTL	005730	1153*	1228*																
SHLT	005032	641	1151*																
SHLTAD	005726	1155*	1156*	1163	1168	1227*													
SHLOT	005036	1153*	1154																
SLPADR	003462	822*	870*	1010*															
SOCNT	006540	1368*	1397*	1409*															
SUMODE	006542	1363*	1367*	1372	1375*	1386*	1411*												
SPASS	004774	821*	1098*	1104	1132*														
SPWRDN	006672	639	1480*	1491															
SPWRUP	006704	1480	1491*																
SREADC	005732	1235*	1429																
SREADS	005774	1250*	1431																
SRWAIT	006610	1435	1452*																
STRAP	006544	643	1416*																
STRPAD	006564	1420	1427*																
STYPE	= 001100	674*	1427																
SWORK	= 001544	782*	1259	1260															
SXWAIT	006620	1437	1455*																
SOFILL	006541	1364*	1366*	1376	1410*														
.	= 011104	635*	638*	647*	651*	655*	660*	699*	762*	763*	781*	803*	805*	807*					
		839	844	938	973	982	987	1014	1071	1165*	1227*	1238	1319	1482					
		1568	1648	1678	1680	1716*	1726*	1727*	1728*	1729*	1730*	1731*	1732*	1734*					
		1735*	1736*	1737*	1738*	1739*	1740*	1741*	1743*	1745*	1747*	1750*	1751*	1752*					
		1753*	1754*	1755*	1756*	1757*	1758*	1759*	1760*	1762*	1763*	1764*							
		1441	1497*																
		762*	962*																
		1445	1500*																
		1439	1514*																
		1443	1519*																
		763*	963*																
XBDIN	006732																		
XR	001502																		
SETS*	006750																		
STPS	007042																		
SUSW*	007064																		
TX	001504																		

BOX	634*	696	772	814	825	831	872	988	1023	1079	1095	1135	1148	1229	1243				
	1332	1448	1540	1561	1616	1638	1717												
HELLU	1*																		
HLT	541*	1576	1591	1607															
.	ABS.	011104	000																

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DSKZ:DZITAD,DZITAD/SOL/CRF/EN:AMA=DZITAD.P11
 RUN-TIME: 9 14 1 SECONDS
 RUN-TIME RATIO: 107/25=4.1
 CORE USED: 11K (21 PAGES)