

DH11

DEVICE ROUTINE (MPG)
MD-11-DTDHA-B

EP-DTDHA-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN U.S.A.

This microfiche card contains a grid of frames on the left side, each containing data. The data is organized into columns and rows, with some frames containing headers and others containing numerical values. The frames are arranged in a grid that is approximately 10 frames wide and 15 frames high. The data appears to be organized into several columns, with some frames containing headers and others containing numerical values. The frames are arranged in a grid that is approximately 10 frames wide and 15 frames high. The data appears to be organized into several columns, with some frames containing headers and others containing numerical values.

.REN 2

IDENTIFICATION

PRODUCT CODE: MADEC-11-DTDMA-B
PRODUCT NAME: DH11 DEVICE ROUTINE FOR MPG
DATE: APRIL 1976
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: W. R. GREENE

COPYRIGHT (C) 1975, 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

2

MADEC-11-DTDMA-B
D77
D77

.SBTTL REVISION HISTORY

:	APR 76	DTDA-B RELEASE
:	JAN 76	ADDED MEMORY MANAGEMENT SUPPORT
:	OCT 75	CREATED FULL SUPPORT DEVICE ROUTINE
:	AUG 75	DTDA-A INITIAL RELEASE (MINIMUM SUPPORT DEVICE ROUTINE)

DTDA-B P11

```

:SBTT: STANDARD DEVICE ROUTINE TABLE
:TITLE MAINDEC-11-DTDM4-B DH11 DEVICE ROUTINE FOR MPG
:REVISION B
:FILENAME OF "TDH4B.MPG" ON MPG/100P MEDIA
:MACY11: DTDM4B.DTDM4B/CRF:SYN/DOC-DTDM4B.P11
:LNKX11: DTDM4B.MPG/0:0-DTDM4B/E
:PAPER TAPE: PUNCH DTDM4B.MPG/FILE:ELEV

```

```

000000*
177776

```

```

.CSECT DH11
.DSABL CR
.PS 177776

```

```

:THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED
:TO INTERFACE WITH MPG.

```

```

000000* 007522
000002* 000000

```

```

LOC2: .WORD DVREN0-
FLAG0: .WORD 0

```

```

:DEVICE ROUT SIZE IN BYTES
:DEVICE ROUT FLAGWORD

```

```

000000 100000
000001 0740000
000002 0200000
000003 0000010
000004 0000000
000005 0000000
000006 0000000
000007 0000001

```

```

DMWAIT: 100000
MBSY: 40000
RBSY: 20000
BIFLG: 10
CLINCT: 1
CLPVC: 1
LDFLG: 1

```

```

: DEVICE ROUTINE WAIT FLAG
: WRITE BUSY FLAG
: READ BUSY FLAG
: BREAK INST FLAG
: CLEAR WRITE VECTOR FLAG
: CLEAR READ VECTOR FLAG
: LEADER FLAG

```

```

000008 0000000
000009 0000000
000010 0000000
000011 0000000
000012 0000000
000013 0000000
000014 0000000
000015 0000000
000016 0000000
000017 0000000
000018 0000000
000019 0000000
000020 0000000
000021 0000000
000022 0000000
000023 0000000
000024 0000000
000025 0000000
000026 0000000
000027 0000000
000028 0000000
000029 0000000
000030 0000000
000031 0000000
000032 0000000
000033 0000000
000034 0000000
000035 0000000
000036 0000000
000037 0000000
000038 0000000
000039 0000000
000040 0000000
000041 0000000
000042 0000000
000043 0000000
000044 0000000
000045 0000000
000046 0000000
000047 0000000
000048 0000000
000049 0000000
000050 0000000
000051 0000000
000052 0000000
000053 0000000
000054 0000000
000055 0000000
000056 0000000
000057 0000000
000058 0000000
000059 0000000
000060 0000000
000061 0000000
000062 0000000
000063 0000000
000064 0000000
000065 0000000
000066 0000000
000067 0000000
000068 0000000
000069 0000000
000070 0000000
000071 0000000
000072 0000000
000073 0000000
000074 0000000
000075 0000000
000076 0000000
000077 0000000
000078 0000000
000079 0000000
000080 0000000
000081 0000000
000082 0000000
000083 0000000
000084 0000000
000085 0000000
000086 0000000
000087 0000000
000088 0000000
000089 0000000
000090 0000000
000091 0000000
000092 0000000
000093 0000000
000094 0000000
000095 0000000
000096 0000000
000097 0000000
000098 0000000
000099 0000000

```

```

SIZE: 0-000000
DEV: 0
DREG0: 0
VCT0: 0
RUS0: 0
RUS1: 0
RUS2: 0
RUS3: 0
RUS4: 0
RUS5: 0
RUS6: 0
RUS7: 0
RUS8: 0
RUS9: 0
RUS10: 0
RUS11: 0
RUS12: 0
RUS13: 0
RUS14: 0
RUS15: 0
RUS16: 0
RUS17: 0
RUS18: 0
RUS19: 0
RUS20: 0
RUS21: 0
RUS22: 0
RUS23: 0
RUS24: 0
RUS25: 0
RUS26: 0
RUS27: 0
RUS28: 0
RUS29: 0
RUS30: 0
RUS31: 0
RUS32: 0
RUS33: 0
RUS34: 0
RUS35: 0
RUS36: 0
RUS37: 0
RUS38: 0
RUS39: 0
RUS40: 0
RUS41: 0
RUS42: 0
RUS43: 0
RUS44: 0
RUS45: 0
RUS46: 0
RUS47: 0
RUS48: 0
RUS49: 0
RUS50: 0
RUS51: 0
RUS52: 0
RUS53: 0
RUS54: 0
RUS55: 0
RUS56: 0
RUS57: 0
RUS58: 0
RUS59: 0
RUS60: 0
RUS61: 0
RUS62: 0
RUS63: 0
RUS64: 0
RUS65: 0
RUS66: 0
RUS67: 0
RUS68: 0
RUS69: 0
RUS70: 0
RUS71: 0
RUS72: 0
RUS73: 0
RUS74: 0
RUS75: 0
RUS76: 0
RUS77: 0
RUS78: 0
RUS79: 0
RUS80: 0
RUS81: 0
RUS82: 0
RUS83: 0
RUS84: 0
RUS85: 0
RUS86: 0
RUS87: 0
RUS88: 0
RUS89: 0
RUS90: 0
RUS91: 0
RUS92: 0
RUS93: 0
RUS94: 0
RUS95: 0
RUS96: 0
RUS97: 0
RUS98: 0
RUS99: 0

```

```

: INTERFACE WORD 0 1 (NOT USED)
: INTERFACE WORD 0 2 (NOT USED)
: INTERFACE WORD 0 3 (NOT USED)
: INTERFACE WORD 0 4 (NOT USED)
: INTERFACE WORD 0 5 (NOT USED)
: INTERFACE WORD 0 6 (NOT USED)
: 0 OF BYTES TRANSFERRED / UNITOP FLG
: ERROR ON LAST I/O INDICATOR
: FIRST DEVICE REGISTER ADR
: INTERRUPT VECTOR ADR
: INT PROC STATUS WORD (BR 5)
: INT PROC STATUS WORD (BR 5)
: HOUSEKEEPING ROUT REL ADR
: REPORT ROUT REL ADR
: KILL ROUT REL ADR
: DATA ERROR COUNTER REL ADR
: TIME OUT ERROR ROUT REL ADR
: I/O BUSY BRANCH ADR
: DEVICE ERROR BRANCH ADR
: USER MODE PRINT ROUTINE BRANCH ADR
: CMD MODE PRINT ROUTINE BRANCH ADR
: CONVERT BINARY TO ASCII ROUT BR ADR
: CONVERT BINARY TO DECIMAL ASCII BR ADR
: CONVERT PACKED DECIMAL TO ASCII BR ADR

```


001162*	001162*	HSKIPST= .			
001163*	001163*	ISTAT= .			: STORAGE FOR DEV REG'S AT INT
001164*	001164*	SCR= .			
001165*	001165*	TRC= .			
001166*	001166*	CLTR= .			
001167*	001167*	CR= .			
001168*	001168*	BYCR= .			
001169*	001169*	BR= .			
001170*	001170*	BCR= .			
001171*	001171*	SSR= .			
001202*	000010	STAT= .	BLM	0	: DEV REG CURRENT VALUE STORAGE
001203*	000000	BYRD= .	WORD	0	: BYTES READ COUNT
001204*	000000	BYWR= .	WORD	0	: BYTES WRITTEN COUNT
001205*	000000	RDINT= .	WORD	0	: READ CHRD COUNT
001206*	000000	WRINT= .	WORD	0	: WRITE CHRD COUNT
001207*	000000	BRKINT= .	WORD	0	: BREAK CHRD COUNT
001208*	000000	MISINT= .	WORD	0	: MISC. CHRD COUNT (CRESET,
001209*	000000	RDINT= .	WORD	0	: READ INTERRUPT COUNT
001210*	000000	WRINT= .	WORD	0	: WRITE INTERRUPT COUNT
001211*	000000	NEVERR= .	WORD	0	: NON-EXIST MEM ERR CNT
001212*	000000				: FDUPLX, HDUPLX)
001213*	000000	OVERRUN= .	WORD	0	: OVERRUN ERRORS COUNT
001214*	000000	FRAMER= .	WORD	0	: FRAMING ERRORS COUNT
001215*	000000	PARERR= .	WORD	0	: PARITY ERRORS COUNT
001216*	000000	SOFERR= .	WORD	0	: SILO OVERFLOW COUNT
001217*	000000	DATER= .	WORD	0	: DATA ERROR COUNT
001218*	000000	TOECNT= .	WORD	0	: # OF ENTRIES INTO T/O ERROR ROUT
001264*	000000	FLAG= .	WORD	0	: FLAGWORD STORAGE
001266*	000000	HSKIPEN= .			
000001	000000	XOXX= .	0		: VALUE TO BE TAILORED BY DEV ROUT
120000	000001	HWVER= .	1		: SYSTEM FLOWD BIT DEF
000002	120000	PSCONS= .	20000		: INT SRVC VIRT PAGE BASE
	000002	LSMTPS= .	2		: MTPS INST LEGAL FLAG

.SBTTL DH11 FUNCTION ROUTINES

;TIMEOUT ERROR HANDLER

```

TOUTER: INC TOECNT          ; INCR TIME OUT ERROR COUNT
          TOPCNT,06        ; EXCEEDED 6 TIMEOUTS*
          BR000            ; YES - CONTINUE
          JSR SAVREG       ; NO - RETURN
          PC, SUPTR0       ; SAVE REGISTERS
          BIC #14107,(R3)  ; P TBL ADDR TO R3
          BRZ              ; CLEAR WAIT FOR I/O TERM
          JSR PRINT        ; PRINT TIMEOUT ERR MSG
          TOEMSG-         ;
          BR000            ;
          JSR KILL         ; KILL THE PROGRAM
          PC, ERDTRG       ; DISPLAY STATUS & STANT 0
          JSR RESREG       ; RESTORE REGISTERS
          (R3),R5          ; GO DISPLAY DEVICE REGS
          JMP SCUPGR

```

```

TOEMSG: .ASCII 'DH11 TIMEOUT ON I/O'

```

.EVEN

;KILL USER PROGRAM ROUTINE

```

KILL: MOV DREG0,R2          ; GET DEV REG ADDR
        BR0100,(R2)        ; ANY INT EBLs SET ?
        BR000            ; NO-EXIT
        JSR TRVECT        ; TEST READ INT VECTOR
        CLR INT           ; BRANCH IF NOT ME
        BIC #010100,(R2)  ; RESET RD INT EBL
        BR005Y,FLAG0      ; RESET READ BSY IN FLAG 0
        PC, RINTV         ; RESET INT VECTOR INFO
        JSR TRVECT        ; TEST WRITE INT VECTOR
        BR000            ; BRANCH IF NOT ME
        BIC #020000,(R2)  ; RESET WR INT EBL
        BR005Y,FLAG0      ; RESET WRITE BSY IN FLAG 0
        PC, RINTV         ; RESET INT VECTOR INFO
        CLR ERR           ; CLEAR ERROR INDICATOR
        CLR RCR           ; CLEAR INTERN CTL REG
        CLR ERRFLG        ; CLEAR INTERN ERR FLAG
        RTS              ; RETURN

```

;READ COMMAND HANDLER

```

READ: MOV R5,STANT         ; SAVE R5
        SUB R4,STANT       ; FOR STANT 0 REFERENCE
        MOV DREG0,R4      ; DEV REG ADDR TO R4
        BIT BR005Y,FLAG0  ; TEST READ BUSY
        BNE G1PTBS        ; BRANCH IF SET
        BIT #100,(R4)     ; TEST RECV INT EBL
        BR000            ; CONTINUE IF NOT SET

```

```

000000 000000 000000 176306 JSR      RS,ACIOBSY      ;OTHERWISE RELEASE CONTROL
000000 000000 000000 176232 SROBSY: BIS      SROBSY,FLAGD      ;SET READ BUSY
000000 000000 000000 176232 MOV      TVCTR0,108      ;INT VECTOR ADDR TO CALL
000000 000000 000000 176232 MOV      REUSR0,208      ;ALSO BUS PRIORITY
000000 000000 000000 176276 JSR      RS,ASETVEC      ;GO SET THE VECTOR
000000 000000 000000 177460 X00X     X00X           ;CLEAR TIME OUT ERR CNT
000000 000000 000000 010100 X00X     X00X           ;SET RECV INT ENB
000000 000000 000000 004376 GTPTBS: PC, SUPTR0      ;GET P TBL BASE IN R3
000000 000000 000000 176204 CLR      ERR             ;CLEAR ERROR INDICATOR
000000 000000 000000 000030 PTCNT(R3)          ;CLEAR TIME OUT ERR CNT
000000 000002 000000 000200 BSPOPER,POPSW(R3)    ;TEST MAINT BIT IN OPSW
000000 000000 000000 001000 R0INCL: BIC      R0INCL,R0INCL      ;SET CSR BIT TO SAME STATE
000000 000000 000000 001000 R0INST: MOV      R0INST,R0INST      ;GET ADDRESS
000000 000000 000000 002364 (R5)+,LSADDR      ;BYTE COUNT AND
000000 000000 000000 002362 (R5)+,LSADDR      ;LINE NBR FROM CALL
000000 000000 000000 002360 (R5)+,BYTES      ;ERR IF LINE SPECIFIED
000000 000000 000000 001011 (R5)+,R0           ;GET CURRENT DEV NBR
000000 000000 000000 116301 MOV      PCUR0V(R3),R1
000000 000000 000000 012700 R1,R0
000000 000000 000000 105701 TESTB   R1,R0
000000 000000 000000 006300 CALDNL: R0,R0
000000 000000 000000 005301 DEC      R0
000000 000000 000000 001174 CALDN   R0,R0
000000 000000 000000 000674 R0,R0
000000 000000 000000 002334 TESTRC: R0,R0
000000 000000 000000 001405 R0,R0
000000 000000 000000 176126 LOADRC: R5,ACIOBSY      ;ANY DESIRED LINE IN USE ?
000000 000000 000000 000006 R5,R5                ;NO - KICK OFF THIS REAC
000000 000000 000000 002314 READ    R5,R5                ;YES - WAIT
000000 000000 000000 177274 LOADRC: R0,R0
000000 000000 000000 0050067 R0,R0      ;SET LINE NBR IN RCR
000000 000000 000000 005267 R0CNT   R0CNT      ;INCR READ CNT COUNT
000000 000000 000000 010702 PC,R2
000000 000000 000000 0062702 R0CVTBL--,R2
000000 000000 000000 012701 R0,R1
000000 000000 000000 006000 FRCVTL: R0,R0
000000 000000 000000 103007 R0,R0      ;FILL RECEIVE TABLE
000000 000000 000000 016722 DRCVTL L$ADDR (R2)+
000000 000000 000000 002254 R0,R2      ;WITH ADDRESS
000000 000000 000000 005301 R0,R2      ;AND BYTE COUNT
000000 000000 000000 001370 DEC      R1
000000 000000 000000 000424 FRCVTL SETRSE
000000 000000 000000 006702 R1,R2
000000 000000 000000 005301 R1,R1
000000 000000 000000 001363 FRCVTL
000000 000000 000000 002763 BNT4TOT,PFLAGD(R3)
000000 000000 000000 100000 B0RWAIT,FLAGD      ;SET WAIT FOR I/O TERM
000000 000000 000000 001002 R0NOWT: B0RWAIT,FLAGD      ;TEST DEV ROUT NOWAIT
000000 000000 000000 000167 B0RWAIT,FLAGD      ;BRANCH IF SET
000000 000000 000000 042763 TSTIEB ;OTHERWISE WAIT
000000 000000 000000 000010 BNT4TOT,PFLAGD(R3) ;CLEAR WAIT FOR I/O TERM

```

```

002032' 000205      RTS      RS      ;RETURN IN LINE
;WRITE AND BREAK COMMAND HANDLER
000010 175740 WRITE: BIC      @BFLG, 1, @B0 ;CLEAR BREAK FLAG
000010 175730 BREAK: BIS      @BFLG, 0, @B0 ;SET BREAK FLAG
003774 003766 XMIT:  MOV      R4, STANT ;SAVE RS
000004 175734      MOV      @REG30, R4 ;FOR STANT & REFERENCE
040000 175704      BIT      @BBSY, @FLAG0 ;TEST WRITE BUSY
020000      BREQ      @ERR, @ERR ;BRANCH IF SET
000000 175734      BIT      @XMIT, @INT_EBL ;TEST XMIT INT EBL
000000      BNEQ      @CONT, @CONT ;CONTINUE IF NOT SET
000000      BR      @PCIOBSY ;OTHERWISE RELEASE CONTROL
040000 175660 SETBSY: BIS      @BBSY, @FLAG0 ;SET WRITE BUSY
000020 175700      MOV      @VC, @R0, 108 ;INT VECTOR ADDR TO CALL
000004 000012      ROR      @R0, 108 ;ADJUST FOR WRITE INT
000000 175670      MOV      @R0, @R0, 208 ;ALSO PASS BUS PRIORITY
000000 175716      SETVEC ;GO SET THE VECTOR
108:      .NOB0
208:      .NOB0
000000 177100      CLRERR: CLR      @ERR ;CLEAR TIME OUT ERR CNT
020000      CLR      @PTCNT, @R3 ;SET XMIT INT EBL
175630      CLR      @ERR ;CLEAR ERROR INDICATOR
000030      CLR      @PC, @R3 ;GET P TBL BASE IN R3
004006      CLR      @PSW, @R3 ;TEST MAINT BIT IN OPSW
000200 000002      BRANCL ;SET CSR BIT TO SAME STATE
001000      BRANCL ;SET CSR BIT TO SAME STATE
001000      BRANCL: BIC      @CSR, @R3 ;GET MS ADDRESS
002004      BRANCL: MOV      @R3, @MSADDR ;GET MS ADDRESS
002002      BRANCL: MOV      @R3, @LSADDR ;GET MS ADDRESS
002000      BRANCL: MOV      @R3, @BYTES ;GET MS ADDRESS
000000      BRANCL: MOV      @R3, @R0 ;GET MS ADDRESS
000036      TESTC   @R0, @R0 ;TESTC
000001      MOV      @R0, @R0 ;MOV
000000      TSTB   @R0 ;TSTB
000000      TESTC   @R0 ;TESTC
000000      MOV      @R0, @R0 ;MOV
000000      DYNPLP: DYNPLP ;DYNPLP
000012      TESTC   @R0, 12, @R4 ;ANY DESIRED LINE IN USE ?
000006      LOADTC: LOADTC ;NO - KICK OFF THIS WRITE
000006      MOV      @R5, @PCIOBSY ;YES - WAIT
001730      LOADTC: MOV      @R0, @TEMPOR ;SET LINE NGRS IN TEMP TOR
000002 175544      BIT      @BUSMPS, @CSYSFW ;SAVE PS AND INH INT
000000      MOV      @R0, @PSSAVE ;SAVE PS AND INH INT
001702      BISE   @R340, @RPS ;SAVE PS AND INH INT
177776
000340 177776

```

```

000233 0001666 108: BR 208
000234 000340 HTPS PCSAVE
000235 001660 208: MOV B,40
MSADDR,R1 ;GET MEMORY EXTEN BITS
000236 001660 CLC
000237 001660 R1
000238 001660 R1
000239 001660 R1
000240 001660 R1
000241 001660 R1
000242 000060 R1
000243 001660 BIC B,0,(R4)
000244 001660 BIS R1,(R4) ;JAM INTO SCR
000245 001660 FSNOTB: MOV PC,R2
PC,R2
000246 001660 ROD B,NOTBL--,R2
000247 001660 CLR R1
000248 001660 FSNOTL: R0
R0 ;TEST LINE BIT
000249 001660 BCC DSNOTL ;BYPASS IF ZERO
000250 001660 BIC B,17,(R4) ;CLEAR LINE BITS
000251 001660 BIS R1,(R4) ;SELECT LINE
000252 001660 NOB ;TO MAKE LISTING MATCH EVAL VER
000253 001660 MOV LSADDR,6(R4) ;STORE HARD ADDR
000254 001660 MOV BYTES,10(R4) ;STORE HARD B.C.
000255 001660 NEG 10(R4) ;PUT IN TWOS COMP FORM
000256 001660 MOV MSADDR,(R2)+ ;STORE SOFT ADDR
000257 001660 MOV LSADDR,(R2)+
000258 001660 MOV BYTES,(R2)+ ;STORE SOFT B.C.
000259 001660 INC R1
000260 001660 CHB R1
000261 001660 CHB R1 ;REPEAT FOR 16 LINES
000262 001660 FSNOTL
000263 001660 SETTCR ;BYPASS THIS LINE
000264 001660 ROD R4,R2
000265 001660 INC R1
000266 001660 CHB R1
000267 001660 CHB R1 ;REPEAT IF NOT LAST
000268 001660 FSNOTL ;SET WAIT FOR I/O TERM
000269 001660 BIT B,410T,PFLAGD(R3) ;SET LINE NGRS IN HARD TCR
000270 001660 TESTOR,12(R4) ;AND SOFTWARE TCR
000271 001660 TESTOR,SYCR ;IF BREAK FLAG SET
000272 001660 TESTOR,SYCR
000273 001660 BIFLG,FLAGD
000274 001660 INCARC
000275 001660 BIS TESTOR,14(R4) ;SET LINE NGRS IN SCR
000276 001660 INC BKCNT ;INCR BREAK CMD COUNT
000277 001660 BR SETSCN
000278 001660 INC WRCNT ;INCR WRITE CMD COUNT
000279 001660 BIT BUSHTPS,2CSYSFW ;RESTORE PS AND EBL INT
000280 001660 BNE 108
000281 001660 MOVB PSSAVE,20PS
000282 001660 BR 208
000283 001660 208
000284 001660 PSSAVE
000285 001660 BIT B,WAIT,FLAGD ;TEST DEV ROUT NOWAIT
000286 001660 TESTIEB ;IF RESET TEST INT EBL
000287 001660 BIFLG,FLAGD ;CLEAR WAIT FOR I/O TERM
000288 001660 BIT B,410T,PFLAGD(R3) ;OTHERWISE RETURN
000289 001660 RETURN: RTS
;WAIT COMMAND HANDLER
002622 042767 10000 175152 WAIT: BIC B,0WAIT,FLAGD ;CLEAR DEV ROUT NOWAIT

```

```

000000 016704 175170      MOV      DREG00,R4
000001 000000 020100      BIT      @20100,(R4)
000002 175132 TRMTST: RELEAS
000003 000000 000000      BCLR    VCT,FLAGD
000004 000000 004002      JS      TRVECT
000005 000000 002330      JS      @2330
000006 175110 000000      PC, @RINTV
000007 175102 108:      BCLR    VCT,FLAGD
000008 000000 004002      ERRTST
000009 000000 004002      JS      TRVECT
000010 000000 002330      ERRTST
000011 175060 100004      PC, @RINTV
000012 001322 001322      BCLR    VCT,FLAGD
000013 175064 000001      ERRTST
000014 003276 001322      RETURN
000015 000001 175064      B1,ERR
000016 003276 020000      PC, SUIPTRD
000017 000001 000001      BRONER,POPSW(R3)
000018 002700      ERREXT
000019 005204      ABBREV
000020 000167      R4
000021 175060      ERRTST
000022 175060      RELEAS
000023 000717      BR
;NOWAIT COMMAND HANDLER
002776 002767 100000 174776 NOWAIT: BIS      @NOWAIT,FLAGD
003004 000705      BR
;CRESET COMMAND HANDLER
003006 016704 175012      CRESET: MOV      DREG00,R4
003007 000000 004000      B1,0000,(R4)
003008 000000 176216      MISCNT
003009 174772      (R4)
003010 001214      ERR
003011 001206      ERRFLG
003012 000000      R4
;SETUP COMMAND HANDLER
003013 000002      SETUP: MOV      (R5)+,LINE
003014 000000      LINE:  BR      R4
003015 000000      WORD   0
;BOLD COMMAND HANDLER
003016 000000      BOLD:  JS      PC, @RINTV
003017 000000      JS      @RINTV
003018 000000      JS      @RINTV
;POINT R4 AT REG ADDR
;TEST INT EBLS
;IF SET, RELEASE CONTROL
;TEST IF VECTOR CLR REGD
;BRANCH IF NOT
;TEST READ VECTOR
;BRANCH IF NOT ME
;GO RESET THE VECTOR
;CLEAR THE REG FLAG
;TEST IF VECTOR CLR REGD
;BRANCH IF NOT
;TEST WRITE VECTOR
;BRANCH IF NOT ME
;GO RESET THE VECTOR
;CLEAR THE REG FLAG
;TEST FOR ANY ERROR
;RETURN IF NONE
;SET ERROR INDICATOR
;TEST JONT PRINT ON ERR BIT
;EXIT IF SET
;SET ABBREVIATED RPT FLG
;OTHERWISE RELEASE CONTROL
;SET NOWAIT FLAG
;SET MASTER CLEAR
;INCR MISC CNT COUNT
;CLEAR ALL CONTROL BITS
;CLEAR ERROR INDICATOR
;CLEAR INTERN ERR FLAG
;CLEAR INTERN CTL REG
;RETURN
;SAVE LINES TO BE USED
;RETURN
;CALL RECV BOLD AND
;WAIT BOLD SUBR
;ADJUST R5

```

```

000014      RBAUD: RTS          : RETURN
000015      RBAUD: RTS          : CALL RECV BAUD SUBR
000016      RBAUD: RTS          : ADJUST RS
000017      RBAUD: RTS          : RETURN
000018      RBAUD: RTS          : CALL XMIT BAUD SUBR
000019      RBAUD: RTS          : ADJUST RS
000020      RBAUD: RTS          : RETURN

000136      TBAUD: RTS          : RECEIVE BAUD COMMAND SUBROUTINE

000205      RBAUD: RTS          : SAVE RS
000206      RBAUD: RTS          : FOR STANT & REFERENCE
000207      RBAUD: RTS          : GET BAUD PARAM
000208      RBAUD: RTS          : CHECK IF WITHIN RANGE
000209      RBAUD: RTS          : BRANCH IF OK
000210      RBAUD: RTS          : ELSE PRINT ERROR MESSAGE

002740      RBAUD: RTS          : AND TAKE ERROR EXIT
002741      RBAUD: RTS          : MOVE RECEIVE
002742      RBAUD: RTS          : BAUD CODE
002743      RBAUD: RTS          : INTO POSITION
002744      RBAUD: RTS          : SET UP
002745      RBAUD: RTS          : PARAMETER MASK
002746      RBAUD: RTS          : GO UPDATE PARAM REGS
002747      RBAUD: RTS          : RETURN

002748      RBAUD: RTS          : BAUD SELECTION CODE ) 15/
002749      RBAUD: RTS          : ASCII
002750      RBAUD: RTS          : .EVEN
002751      RBAUD: RTS          : TRANSMIT BAUD COMMAND SUBROUTINE

002752      TBAUD: RTS          : SAVE RS
002753      TBAUD: RTS          : FOR STANT & REFERENCE
002754      TBAUD: RTS          : GET BAUD PARAM
002755      TBAUD: RTS          : CHECK IF WITHIN RANGE
002756      TBAUD: RTS          : BRANCH IF OK
002757      TBAUD: RTS          : ELSE PRINT ERR MSG
002758      TBAUD: RTS          : AND TAKE ERROR EXIT
002759      TBAUD: RTS          : MOVE TRANSMIT
002760      TBAUD: RTS          :
002761      TBAUD: RTS          :
002762      TBAUD: RTS          :
002763      TBAUD: RTS          :
002764      TBAUD: RTS          :
002765      TBAUD: RTS          :
002766      TBAUD: RTS          :
002767      TBAUD: RTS          :
002768      TBAUD: RTS          :
002769      TBAUD: RTS          :
002770      TBAUD: RTS          :
002771      TBAUD: RTS          :
002772      TBAUD: RTS          :
002773      TBAUD: RTS          :
002774      TBAUD: RTS          :
002775      TBAUD: RTS          :
002776      TBAUD: RTS          :
002777      TBAUD: RTS          :
002778      TBAUD: RTS          :
002779      TBAUD: RTS          :
002780      TBAUD: RTS          :
002781      TBAUD: RTS          :
002782      TBAUD: RTS          :
002783      TBAUD: RTS          :
002784      TBAUD: RTS          :
002785      TBAUD: RTS          :
002786      TBAUD: RTS          :
002787      TBAUD: RTS          :
002788      TBAUD: RTS          :
002789      TBAUD: RTS          :
002790      TBAUD: RTS          :
002791      TBAUD: RTS          :
002792      TBAUD: RTS          :
002793      TBAUD: RTS          :
002794      TBAUD: RTS          :
002795      TBAUD: RTS          :
002796      TBAUD: RTS          :
002797      TBAUD: RTS          :
002798      TBAUD: RTS          :
002799      TBAUD: RTS          :
002800      TBAUD: RTS          :
002801      TBAUD: RTS          :
002802      TBAUD: RTS          :
002803      TBAUD: RTS          :
002804      TBAUD: RTS          :
002805      TBAUD: RTS          :
002806      TBAUD: RTS          :
002807      TBAUD: RTS          :
002808      TBAUD: RTS          :
002809      TBAUD: RTS          :
002810      TBAUD: RTS          :
002811      TBAUD: RTS          :
002812      TBAUD: RTS          :
002813      TBAUD: RTS          :
002814      TBAUD: RTS          :
002815      TBAUD: RTS          :
002816      TBAUD: RTS          :
002817      TBAUD: RTS          :
002818      TBAUD: RTS          :
002819      TBAUD: RTS          :
002820      TBAUD: RTS          :
002821      TBAUD: RTS          :
002822      TBAUD: RTS          :
002823      TBAUD: RTS          :
002824      TBAUD: RTS          :
002825      TBAUD: RTS          :
002826      TBAUD: RTS          :
002827      TBAUD: RTS          :
002828      TBAUD: RTS          :
002829      TBAUD: RTS          :
002830      TBAUD: RTS          :
002831      TBAUD: RTS          :
002832      TBAUD: RTS          :
002833      TBAUD: RTS          :
002834      TBAUD: RTS          :
002835      TBAUD: RTS          :
002836      TBAUD: RTS          :
002837      TBAUD: RTS          :
002838      TBAUD: RTS          :
002839      TBAUD: RTS          :
002840      TBAUD: RTS          :
002841      TBAUD: RTS          :
002842      TBAUD: RTS          :
002843      TBAUD: RTS          :
002844      TBAUD: RTS          :
002845      TBAUD: RTS          :
002846      TBAUD: RTS          :
002847      TBAUD: RTS          :
002848      TBAUD: RTS          :
002849      TBAUD: RTS          :
002850      TBAUD: RTS          :
002851      TBAUD: RTS          :
002852      TBAUD: RTS          :
002853      TBAUD: RTS          :
002854      TBAUD: RTS          :
002855      TBAUD: RTS          :
002856      TBAUD: RTS          :
002857      TBAUD: RTS          :
002858      TBAUD: RTS          :
002859      TBAUD: RTS          :
002860      TBAUD: RTS          :
002861      TBAUD: RTS          :
002862      TBAUD: RTS          :
002863      TBAUD: RTS          :
002864      TBAUD: RTS          :
002865      TBAUD: RTS          :
002866      TBAUD: RTS          :
002867      TBAUD: RTS          :
002868      TBAUD: RTS          :
002869      TBAUD: RTS          :
002870      TBAUD: RTS          :
002871      TBAUD: RTS          :
002872      TBAUD: RTS          :
002873      TBAUD: RTS          :
002874      TBAUD: RTS          :
002875      TBAUD: RTS          :
002876      TBAUD: RTS          :
002877      TBAUD: RTS          :
002878      TBAUD: RTS          :
002879      TBAUD: RTS          :
002880      TBAUD: RTS          :
002881      TBAUD: RTS          :
002882      TBAUD: RTS          :
002883      TBAUD: RTS          :
002884      TBAUD: RTS          :
002885      TBAUD: RTS          :
002886      TBAUD: RTS          :
002887      TBAUD: RTS          :
002888      TBAUD: RTS          :
002889      TBAUD: RTS          :
002890      TBAUD: RTS          :
002891      TBAUD: RTS          :
002892      TBAUD: RTS          :
002893      TBAUD: RTS          :
002894      TBAUD: RTS          :
002895      TBAUD: RTS          :
002896      TBAUD: RTS          :
002897      TBAUD: RTS          :
002898      TBAUD: RTS          :
002899      TBAUD: RTS          :
002900      TBAUD: RTS          :
002901      TBAUD: RTS          :
002902      TBAUD: RTS          :
002903      TBAUD: RTS          :
002904      TBAUD: RTS          :
002905      TBAUD: RTS          :
002906      TBAUD: RTS          :
002907      TBAUD: RTS          :
002908      TBAUD: RTS          :
002909      TBAUD: RTS          :
002910      TBAUD: RTS          :
002911      TBAUD: RTS          :
002912      TBAUD: RTS          :
002913      TBAUD: RTS          :
002914      TBAUD: RTS          :
002915      TBAUD: RTS          :
002916      TBAUD: RTS          :
002917      TBAUD: RTS          :
002918      TBAUD: RTS          :
002919      TBAUD: RTS          :
002920      TBAUD: RTS          :
002921      TBAUD: RTS          :
002922      TBAUD: RTS          :
002923      TBAUD: RTS          :
002924      TBAUD: RTS          :
002925      TBAUD: RTS          :
002926      TBAUD: RTS          :
002927      TBAUD: RTS          :
002928      TBAUD: RTS          :
002929      TBAUD: RTS          :
002930      TBAUD: RTS          :
002931      TBAUD: RTS          :
002932      TBAUD: RTS          :
002933      TBAUD: RTS          :
002934      TBAUD: RTS          :
002935      TBAUD: RTS          :
002936      TBAUD: RTS          :
002937      TBAUD: RTS          :
002938      TBAUD: RTS          :
002939      TBAUD: RTS          :
002940      TBAUD: RTS          :
002941      TBAUD: RTS          :
002942      TBAUD: RTS          :
002943      TBAUD: RTS          :
002944      TBAUD: RTS          :
002945      TBAUD: RTS          :
002946      TBAUD: RTS          :
002947      TBAUD: RTS          :
002948      TBAUD: RTS          :
002949      TBAUD: RTS          :
002950      TBAUD: RTS          :
002951      TBAUD: RTS          :
002952      TBAUD: RTS          :
002953      TBAUD: RTS          :
002954      TBAUD: RTS          :
002955      TBAUD: RTS          :
002956      TBAUD: RTS          :
002957      TBAUD: RTS          :
002958      TBAUD: RTS          :
002959      TBAUD: RTS          :
002960      TBAUD: RTS          :
002961      TBAUD: RTS          :
002962      TBAUD: RTS          :
002963      TBAUD: RTS          :
002964      TBAUD: RTS          :
002965      TBAUD: RTS          :
002966      TBAUD: RTS          :
002967      TBAUD: RTS          :
002968      TBAUD: RTS          :
002969      TBAUD: RTS          :
002970      TBAUD: RTS          :
002971      TBAUD: RTS          :
002972      TBAUD: RTS          :
002973      TBAUD: RTS          :
002974      TBAUD: RTS          :
002975      TBAUD: RTS          :
002976      TBAUD: RTS          :
002977      TBAUD: RTS          :
002978      TBAUD: RTS          :
002979      TBAUD: RTS          :
002980      TBAUD: RTS          :
002981      TBAUD: RTS          :
002982      TBAUD: RTS          :
002983      TBAUD: RTS          :
002984      TBAUD: RTS          :
002985      TBAUD: RTS          :
002986      TBAUD: RTS          :
002987      TBAUD: RTS          :
002988      TBAUD: RTS          :
002989      TBAUD: RTS          :
002990      TBAUD: RTS          :
002991      TBAUD: RTS          :
002992      TBAUD: RTS          :
002993      TBAUD: RTS          :
002994      TBAUD: RTS          :
002995      TBAUD: RTS          :
002996      TBAUD: RTS          :
002997      TBAUD: RTS          :
002998      TBAUD: RTS          :
002999      TBAUD: RTS          :
003000      TBAUD: RTS          :

```

000000	006100	000000	000000	208:	ROL	R0		: BALD CODE
000001	006101	000000	000000		DEC	R1		: INTO POSITION
000002	006102	000000	000000		BNE	R1		: SET UP
000003	006103	000000	000000		MOV	R2, R0		: PARAMETER MASK
000004	006104	000000	000000		JSR	R2, LPRUPD		: GO UPDATE PARAM REGS
000005	006105	000000	000000		RTS	R0		: RETURN
					: EVEN COMMAND HANDLER			
000020	012700	000020	000020	EVEN:	MOV	R2, R0		
000021	012701	000020	000020		MOV	R1, R0		
000022	012702	000020	000020		JSR	R2, LPRUPD		
000023	012703	000020	000020		RTS	R0		
					: ODD COMMAND HANDLER			
000060	012700	000060	000060	ODD:	MOV	R2, R0		
000061	012701	000060	000060		MOV	R1, R0		
000062	012702	000060	000060		JSR	R2, LPRUPD		
000063	012703	000060	000060		RTS	R0		
					: NO PARITY COMMAND HANDLER			
000200	005000	000200	000200	NOPAR:	CLR	R0		
000201	012701	000200	000200		MOV	R2, R0		
000202	012702	000200	000200		JSR	R2, LPRUPD		
000203	012703	000200	000200		RTS	R0		
					: ONE STOP BIT COMMAND HANDLER			
000204	005000	000204	000204	ONESTP:	CLR	R0		
000205	012701	000204	000204		MOV	R2, R0		
000206	012702	000204	000204		JSR	R2, LPRUPD		
000207	012703	000204	000204		RTS	R0		
					: TWO STOP BITS COMMAND HANDLER			
000208	012700	000208	000208	TWOSTP:	MOV	R2, R0		
000209	012701	000208	000208		MOV	R1, R0		
000210	012702	000208	000208		JSR	R2, LPRUPD		
000211	012703	000208	000208		RTS	R0		
					: BITS PER CHARACTER COMMAND HANDLER			
002374	010567	002402	002374	BITS:	MOV	R5, STANT		: SAVE R5
002403	162767	000004	002403		MOV	R2, STANT		: FOR STANT 8 REFERENCE
002404	004767	002526	002404		MOV	R1, STANT		
002405	012500	000005	002405		MOV	R0, STANT		: GET NBR OF BITS
002406	162700	000005	002406		MOV	R0, R0		: SUBTRACT 5
002407	020027	000004	002407		MOV	R0, R0		: IF LESS THAN 4
002408	103411	000004	002408		MOV	R0, R0		: ENTRY OK
002409	005004	003070	002409		MOV	R0, R0		: ELSE PRINT ERR MSG
002410	004767	002214	002410		MOV	R0, R0		: THEN TAKE ERR EXIT

:READ INTERRUPT HANDLER

001320		ROINT:	JSR	RD, SAVREG	:SAVE REGISTERS
001366			JSR	RS, STSTAT	:STORE DEVICE REG CONTENTS
174372			ISUB	ISTAT-	
001394			MOVB	R1, R2	:INCR READ INT COUNT
000000			MOVB	PC, SPTAD	:SET UP INTERN PTRS
174276			MOVB	SI, (R4)	:SILO OVERFLOW?
000020	177352	RENFL:	MOVB	RENFL	:YES-REMEMBER IT
174354			MOVB	NR, R1	:GET STORED CHAR
000002		STBYT:	MOVB	FRSBYT	:PROCESS IT
177760		FRSBYT:	MOVB	SI, ERRFLG	:REMEMBER SILO OVERFL
177322			MOVB	SO, ERR	:INCR SILO OVERFL COUNT
000002			MOVB	STO, RD	:ABORT READ
177760			MOVB	(R4), R1	:GET DATA FROM RBUF
177322			MOVB	CH, VL	:OR IF CH NOT VALID
000002			MOVB	R1, R2	:SAVE IN R2
177760			MOVB	R1, R1	:FILTER OUT OTHER BITS
177322			MOVB	R1, R1	:USE AS TABLE INDEX
000002			MOVB	PC, R3	
174256			MOVB	BC, TBL--R3	:POINT RD AT TBL ENTRY
000001			MOVB	R1, R3	:ADD INDEX
010000			MOVB	2(R3)	:TEST BYTE COUNT
174256			MOVB	ATERM	:BRANCH IF ZERO
000001			MOVB	PC, R1	
010000			MOVB	BYRD+2--R1	
174256			MOVB	SI, (R1)	:INCR BYTES READ BY 1
000001			MOVB	(R1)	
020000			MOVB	SI, ERRFLG	:TEST FOR PARITY ERROR
174256	177250	NOPERR:	MOVB	NO, PERR	:BRANCH IF NOT
000001			MOVB	SI, ERRFLG	:OTHERWISE INCR COUNT
020000			MOVB	SI, ERRFLG	:REMEMBER PARITY ERROR
174256			MOVB	NO, FERR	:TEST FOR FRAMING ERROR
000002	177230	NOFERR:	MOVB	NO, FERR	:BRANCH IF NOT
000000			MOVB	SI, ERRFLG	:OTHERWISE INCR COUNT
174216	177210	NOOERR:	MOVB	SI, ERRFLG	:REMEMBER FRAMING ERROR
000004			MOVB	NO, OERR	:TEST FOR OVERRUN ERROR
173030			MOVB	SI, ERRFLG	:BRANCH IF NOT
177776			MOVB	(R3)+, RD	:OTHERWISE INCR COUNT
174140			MOVB	R2, R1	:REMEMBER OVERRUN ERROR
172724			MOVB	PC, SPTBYT	
000001			MOVB	(R3)	:MOVE BYTE TO MEMORY
177760			MOVB	(R3)	:INCREMENT ADDRESS
			MOVB	ATERM	:DECREMENT BYTE COUNT
			MOVB	CH, VL	:TERMINATE IF BC ZERO
		ATERM:	MOVB	PC, R1	
			MOVB	BYRD+2--R1	
			MOVB	(R1), SIZE	:UPDATE ACTUAL BYTES XFERRED
			MOVB	SI, R1	
			MOVB	R2	
		ATERMLP:	MOVB	SI, ERRFLG	:FILTER ALL BUT LINE NBR
			MOVB	R2	


```

000014 .WORD 12.
000015 JSR RS,DISPST ;GO DISPLAY STATUS AT LAST INT
000016 .WORD 17357
000017 JSR RS,DISPST ;GO DISPLAY STATUS AT LAST INT
000018 .WORD 17357
000019 JSR RS,PRINT ;ISSUE 'CURRENTLY' MSG
000020 .WORD 17357
000021 JSR CLMSG-
000022 .WORD 10.
000023 JSR RS,DISPST ;GO DISPLAY CURRENT STATUS
000024 .WORD 17357
000025 JSR CSTAT-
000026 .WORD 01,R4 ;DISPLAY COUNTS?
000027 DISCNT: BIT RPTEND ;Y,N-RPTEND
000028 BEQ RPTEND ;SET UP # OF WORDS
000029 MOV R0,R0 ;SET UP ADR OF CNTS
000030 MOV PC,R1
000031 MOV R0,R0 ;SET UP TBL ADR
000032 MOV PC,R2
000033 RPTBL: .WORD B0YRD-RPTBRS
000034 .WORD (R2)+,RPTBRS ;MOV MSG ADR TO S/R LINAGE
000035 JSR R0,SAVEG ;SAVE ALL REG'S
000036 .WORD (R1),R0 ;GET CURRENT COUNT
000037 RPTBRS: JSR RS,ABINASC ;CONVERT IT TO ASCII
000038 .WORD XXXX ;RESTORE REG'S
000039 JSR R0,RESREG ;POINT AT NXT CNT
000040 TST (R1)+ ;DONE ALL WORDS?
000041 DEC R0 ;Y,N-RPTLP
000042 BNE RPTLP ;GO ISSUE COUNTS MSG
000043 JSR RS,PRINT
000044 .WORD CNTSMG-
000045 RPTEND: JSR CNTSEN-CNTSMG ;ISSUE "END OF REPORT" MSG
000046 .WORD RS,PRINT
000047 .WORD RENMG-
000048 DVREX: .WORD -13.
000049 JSR R0,RESREG ;RESTORE REGISTERS
000050 TST (RS)+ ;SET UP RETURN POINT
000051 RTS RS ;EXIT IN-LINE
000052 .WORD 0
000053 .WORD B0YRD-RPTBRS
000054 .WORD B0YRD+6-RPTBRS
000055 .WORD B0YRD-RPTBRS
000056 .WORD B0YRD+6-RPTBRS
000057 .WORD C0C0D-RPTBRS
000058 .WORD C0C0A-RPTBRS
000059 .WORD C0C0K-RPTBRS
000060 .WORD C0C0S-RPTBRS
000061 .WORD R0INPS-RPTBRS
000062 .WORD R0INPS-RPTBRS
000063 .WORD ERCHET-RPTBRS
000064 .WORD ERCOVER-RPTBRS
000065 .WORD ERCPAN-RPTBRS
000066 .WORD ERCPAN-RPTBRS
000067 .WORD ERCSOF-RPTBRS
000068 .WORD ERCDTA-RPTBRS
000069 .WORD
000070 .WORD
000071 .WORD
000072 .WORD
000073 .WORD
000074 .WORD
000075 .WORD
000076 .WORD
000077 .WORD
000078 .WORD
000079 .WORD
000080 .WORD
000081 .WORD
000082 .WORD
000083 .WORD
000084 .WORD
000085 .WORD
000086 .WORD
000087 .WORD
000088 .WORD
000089 .WORD
000090 .WORD
000091 .WORD
000092 .WORD
000093 .WORD
000094 .WORD
000095 .WORD
000096 .WORD
000097 .WORD
000098 .WORD
000099 .WORD
000100 .WORD
000101 .WORD
000102 .WORD
000103 .WORD
000104 .WORD
000105 .WORD
000106 .WORD
000107 .WORD
000108 .WORD
000109 .WORD
000110 .WORD
000111 .WORD
000112 .WORD
000113 .WORD
000114 .WORD
000115 .WORD
000116 .WORD
000117 .WORD
000118 .WORD
000119 .WORD
000120 .WORD
000121 .WORD
000122 .WORD
000123 .WORD
000124 .WORD
000125 .WORD
000126 .WORD
000127 .WORD
000128 .WORD
000129 .WORD
000130 .WORD
000131 .WORD
000132 .WORD
000133 .WORD
000134 .WORD
000135 .WORD
000136 .WORD
000137 .WORD
000138 .WORD
000139 .WORD
000140 .WORD
000141 .WORD
000142 .WORD
000143 .WORD
000144 .WORD
000145 .WORD
000146 .WORD
000147 .WORD
000148 .WORD
000149 .WORD
000150 .WORD
000151 .WORD
000152 .WORD
000153 .WORD
000154 .WORD
000155 .WORD
000156 .WORD
000157 .WORD
000158 .WORD
000159 .WORD
000160 .WORD
000161 .WORD
000162 .WORD
000163 .WORD
000164 .WORD
000165 .WORD
000166 .WORD
000167 .WORD
000168 .WORD
000169 .WORD
000170 .WORD
000171 .WORD
000172 .WORD
000173 .WORD
000174 .WORD
000175 .WORD
000176 .WORD
000177 .WORD
000178 .WORD
000179 .WORD
000180 .WORD
000181 .WORD
000182 .WORD
000183 .WORD
000184 .WORD
000185 .WORD
000186 .WORD
000187 .WORD
000188 .WORD
000189 .WORD
000190 .WORD
000191 .WORD
000192 .WORD
000193 .WORD
000194 .WORD
000195 .WORD
000196 .WORD
000197 .WORD
000198 .WORD
000199 .WORD
000200 .WORD

```


:DH11 ERROR REPORT ROUTINE

```

005620 004767 000004 ERRAPT: JSR PC,ERRDIS
005621 000177 172220 JYP 2CUPGER
005622 010701 ERRDIS: MOV PC,R1 ;POINT R1 AT ERR MSG
005623 062701 000236 R00 EMSGBF--,R1
005624 012767 000014 000054 MOV R12,ERRBCT
005625 010700 MOV PC,R0 ;POINT R0 AT ERR MSG TBL
005626 062700 000144 R00 ERCDTB--,R0
005627 105710 18: TSTB (R0)
005628 001416 BEQ ERTBEN ;BRANCH IF R0 AT TBL END
005629 132067 176366 BITB (R0)+,ERRFLG ;TEST FOR PARTICULAR ERR
005630 001003 BNE 38 ;BRANCH IF FOUND
005631 062700 000005 28: R00 R0
005632 000770 BR 18,R0
005633 012702 000005 38: MOV R5,R2
005634 112021 48: MOVB (R0)+,(R1)+ ;MOVE MSG CODE TO ERR MSG
005635 005237 000014 INC ERMBCT ;BUMP BYTE COUNT
005636 005302 DEC R2
005637 001373 BNE 48
005638 000760 BR 18
005639 004567 000656 ERTBEN: JSR R5,PRINT ;CHECK IF MORE
005640 000136 .WORD EMSGBD- ;PRINT ERROR MSG
005641 000014 ERMBCT: .WORD 12
005642 004567 000556 ERDING: JSR R5,DISPST ;DISPLAY DEVICE REGS
005643 173234 .WORD 15STAT-
005644 016300 000022 ERRSNM: MOV PSRCST(R3),R0 ;GET ADDR OF SRC STANTS
005645 111001 108: MOVB (R0),R1 ;SAVE STANT LENGTH
005646 026067 003004 000106 CMP #4(R0),STANT ;ERROR OCCUR ON THIS STANT?
005647 001402 BEQ 208 ;YES - BRANCH
005648 060100 R00 R1,R0 ;POINT AT NEXT STATEMENT
005649 000771 BR 108 ;GO ON NEXT STANT
005650 005720 208: TST (R0)+ ;SET UP ADDR OF STANT & DATA
005651 010701 MOV PC,R1 ;SET UP DATA OUTPUT ADDR
005652 062701 000172 R00 R5,STANUM-,R1
005653 004577 172074 JSR R5,DECRSC ;CONVERT IT TO ASCII
005654 012767 020040 000160 MOV R2,0040,STANUM+4 ;SET 2 LOW DIGITS TO SPACES
005655 004567 000574 JSR R5,PRINT ;ISSUE STANT & MSG
006000 .WORD STANTG-
006001 .WORD -14
006002 177767 CLR ERRFLG ;CLEAR ERROR FLAG
006003 005067 176240 RTS PC
006012 020001 050040 051101 ERCDTB: .ASCII (<001>/ PRR/ ;ERROR MSG CODE TABLE
006013 020002 043040 046522 .ASCII (<002>/ FRM/
006014 020004 047440 051128 .ASCII (<004>/ OVR/
006015 020010 047040 046508 .ASCII (<010>/ NEW/
006016 020020 051440 043117 .ASCII (<020>/ SDF/
006017 .BYTE 0
006018 .EVEN
006052 000000 STANT: .WORD 0 ;SAVED R5 FOR STANT #
006054 044104 030461 042440 EMSGBD: .ASCII /DH11 ERROR: /
006055 051122 051117 020072

```

1201	006070	000050			EMSGBF: .BLKB	NO
1202	006140	052123	047115	020124	STAMPG: .ASCII	/STANT # /
1203	006150	054130	054130	054130	STAMP: .ASCII	/XXXXXXXX/

000272 005726
000274 000205

TST (SP)+ :CLEAN UP THE STACK
RTS RS :EXIT IN-LINE

:LINE PARAMETER REGISTER UPDATE SUBROUTINE

:JSR RS,LPRUPD S/R CALL

:R0 CONTAINS PARAMETER
:R1 CONTAINS BIT MASK
:DESTROYS R2,R3,R4

000000 000000
000001 000001
000002 000002
000003 000003
000004 000004
000005 000005
000006 000006
000007 000007
000008 000008
000009 000009
000010 000010
000011 000011
000012 000012
000013 000013
000014 000014
000015 000015
000016 000016
000017 000017
000018 000018
000019 000019
000020 000020
000021 000021
000022 000022
000023 000023
000024 000024
000025 000025
000026 000026
000027 000027
000028 000028
000029 000029
000030 000030
000031 000031
000032 000032
000033 000033
000034 000034
000035 000035
000036 000036
000037 000037
000038 000038
000039 000039
000040 000040
000041 000041
000042 000042
000043 000043
000044 000044
000045 000045
000046 000046
000047 000047
000048 000048
000049 000049
000050 000050
000051 000051
000052 000052
000053 000053
000054 000054
000055 000055
000056 000056
000057 000057
000058 000058
000059 000059
000060 000060
000061 000061
000062 000062
000063 000063
000064 000064
000065 000065
000066 000066
000067 000067
000068 000068
000069 000069
000070 000070
000071 000071
000072 000072
000073 000073
000074 000074
000075 000075
000076 000076
000077 000077
000078 000078
000079 000079
000080 000080
000081 000081
000082 000082
000083 000083
000084 000084
000085 000085
000086 000086
000087 000087
000088 000088
000089 000089
000090 000090
000091 000091
000092 000092
000093 000093
000094 000094
000095 000095
000096 000096
000097 000097
000098 000098
000099 000099

LPRUPD:
108:
208:
308:
PARAM:
PMASK:

MOV R0,PARAM :SAVE PARAMETER
AND R1,PMASK :AND PARAM MASK
MOV R2,R0 :POINT R2 AT DEV REGS
MOV R3,R0 :POINT R3 AT P TABLE
MOV R4,R0 :CHECK SIMULTANEOUS LINES
MOV R5,R0 :BRANCH IF ANY SET
MOV R6,R0 :ELSE GET ASSIGNED LINE
MOV R7,R0 :CONVERT BIT POSITION
MOV R8,R0 :REPRESENTING LINE #
MOV R9,R0 :TO
MOV R10,R0 :BINARY NUM IN R1
MOV R11,R0 :CLEAR LINE BIT
MOV R12,R0 :CLEAR LINE SELECT
MOV R13,R0 :JMP INTO LINE SELECT
MOV R14,R0 :GET PARAMETER
MOV R15,R0 :AND MASK
MOV R16,R0 :CLEAR PARAM BIT IN P REG
MOV R17,R0 :JMP PARAM IN P REG
MOV R18,R0 :ANY MORE LINES?
MOV R19,R0 :YES - DO IT AGAIN
MOV R20,R0 :NO - EXIT
MOV R21,R0 :INIT TO NORM MSG LNTH
MOV R22,R0 :GET CURRENT UNIT #
MOV R23,R0 :VALID UNIT #
MOV R24,R0 :Y,N-DISUIV
MOV R25,R0 :CONVERT # TO DECIMAL ASCII
MOV R26,R0 :MOVE ASCII # TO 1ST TWO DIGITS
MOV R27,R0 :SET UP ERR COND MSG LNTH
MOV R28,R0 :RESET HIGH BYTE

:DISPLAY CURRENT UNIT #

:JSR PC,DISUN# S/R CALL
:R3 MUST CONTAIN PROG TAB ADDR
:DESTROYS R0,R1,R2

000000 012767 000022 000056
000001 01116308 000035
000002 01010079 000020
000003 0004577 171420
000004 016767 000424 000416
000005 012767 000026 000020
000006 042700 177400

DISUN#:
DISUIV:
DISUN#:
DISUIV:

MOV R0,DISUN# :INIT TO NORM MSG LNTH
MOV R1,DISUN# :GET CURRENT UNIT #
MOV R2,DISUN# :VALID UNIT #
MOV R3,DISUN# :Y,N-DISUIV
MOV R4,DISUN# :CONVERT # TO DECIMAL ASCII
MOV R5,DISUN# :MOVE ASCII # TO 1ST TWO DIGITS
MOV R6,DISUN# :SET UP ERR COND MSG LNTH
MOV R7,DISUN# :RESET HIGH BYTE

MOV R8,DISUN# :INIT TO NORM MSG LNTH
MOV R9,DISUN# :GET CURRENT UNIT #
MOV R10,DISUN# :VALID UNIT #
MOV R11,DISUN# :Y,N-DISUIV
MOV R12,DISUN# :CONVERT # TO DECIMAL ASCII
MOV R13,DISUN# :MOVE ASCII # TO 1ST TWO DIGITS
MOV R14,DISUN# :SET UP ERR COND MSG LNTH
MOV R15,DISUN# :RESET HIGH BYTE

```

11317 006464 004577 171366      JSR    RS,2BINASC      ;CONVERT BINARY TO ASCII
006470 000378      WORD    UNASC-        ;
006472 000076      DISUPR: JSR    RS,PRINT ;GO ISSUE UNIT 0 MSG
006476 000000      .WORD    UNITNG-      ;
006500 000020      DISUPL: .WORD    16.   ;
075502 000207      RTS     PC             ;EXIT INLINE

;TAILOR STATUS MSG & PRINT IT

;JSR    RS,DISPST      S/R CALL
;WORD    STATADR-      REL ADR OF STATUS DATA
;DESTROYS R0,R1,R2

DISPST: MOV    RS,R2      ;GET REL DATA ADR
        (RS)+,R2      ;MOVE IT R0
        PC,R1         ;SET UP ADR OF REG NAMES IN ASCII
        MOVREGS-R1    ;
        MOVREGS-DVREGS/6,R0 ;GET # OF REGISTERS TO DISPLAY
108:    (R1)+,DVREGS   ;MOVE REG NAME TO MSG
        (R1)+,DVREGS+2 ;
        (R1)+         ;
        R0,SAVEREG    ;BYPASS DISP VALUE
        (R2),R0       ;SAVE REG'S R0 - R5
        JSR    RS,2BINASC ;GET REG'S STORED VALUE
        DVASC-        ;CONVERT IT TO ASCII
        JSR    RS,PRINT ;PRINT THE STATUS MSG
        DVASC-        ;
        R0,RESREG     ;RESTORE R0 - R4
        (R2)+         ;POINT AT NXT REG VALLE
        R0,R0         ;DECR REG CNT
        R0,R0         ;DONE ALL? (Y,N-108)
        RTS          ;EXIT IN-LINE

```

:ISSUE MSG TO LIST DEVICE

```

:JSR   RS,PRINT           S/R CALL
:WORD  FLAGADR-          REL ADR OF MSG
:WORD  BYTCNT           MSG BYTE CNT (IF NEGATIVE,
                        RESET PRY DEV DEDICATED.)
:R3 = PROG TRM ADR
:R4 = FLAGADR -- IF NEGATIVE, USE CTRD NODE PRINT
:DESTROYS R0,R1,R2

```

```

000000 010050 PRINT: MOV R5,R0 ;GET MSG ADR & MAKE IT ABS
000001 010051 MOV R5,R0 ;GET MSG ADR & MAKE IT ABS
000002 010052 TST R5 ;GET BYTE COUNT
000003 010053 PC,R2 ;USE CTRD NODE PRINT?
000004 010054 PC,R2 ;Y,N=108
000005 010055 PC,R2 ;SET UP LINK INFO ADR
000006 010056 PC,R2 ;MAKE MSG ADR REL
000007 010057 PC,R2 ;STORE MSG ADR
000008 010058 PC,R2 ;STORE MSG'S BYTE COUNT
000009 010059 PC,R2 ;CNT NEG? (Y,N=108)
000010 010060 PC,R2 ;CNT POSITIVE
000011 010061 PC,R2 ;STORE PROG'S # IN MSG
000012 010062 PC,R2 ;ISSUE PROG #
000013 010063 PC,R2 ;ISSUE MSG SPECIFIED
000014 010064 PC,R2 ;ISSUE A (CR) & (LF)
000015 010065 PC,R2 ;GO TO EXIT
000016 010066 PC,R2 ;STORE MSG'S ABS ADR
000017 010067 PC,R2 ;STORE ITS BYTE CNT
000018 010068 PC,R2 ;GO TO RPG TO ISSUE THE MSG
000019 010069 PC,R2 ;EXIT IN-LINE
000020 010070 PC,R2
000021 010071 PC,R2
000022 010072 PC,R2
000023 010073 PC,R2
000024 010074 PC,R2
000025 010075 PC,R2
000026 010076 PC,R2
000027 010077 PC,R2
000028 010078 PC,R2
000029 010079 PC,R2
000030 010080 PC,R2
000031 010081 PC,R2
000032 010082 PC,R2
000033 010083 PC,R2
000034 010084 PC,R2
000035 010085 PC,R2
000036 010086 PC,R2
000037 010087 PC,R2
000038 010088 PC,R2
000039 010089 PC,R2
000040 010090 PC,R2
000041 010091 PC,R2
000042 010092 PC,R2
000043 010093 PC,R2
000044 010094 PC,R2
000045 010095 PC,R2
000046 010096 PC,R2
000047 010097 PC,R2
000048 010098 PC,R2
000049 010099 PC,R2
000050 010100 PC,R2
000051 010101 PC,R2
000052 010102 PC,R2
000053 010103 PC,R2
000054 010104 PC,R2
000055 010105 PC,R2
000056 010106 PC,R2
000057 010107 PC,R2
000058 010108 PC,R2
000059 010109 PC,R2
000060 010110 PC,R2
000061 010111 PC,R2
000062 010112 PC,R2
000063 010113 PC,R2
000064 010114 PC,R2
000065 010115 PC,R2
000066 010116 PC,R2
000067 010117 PC,R2
000068 010118 PC,R2
000069 010119 PC,R2
000070 010120 PC,R2
000071 010121 PC,R2
000072 010122 PC,R2
000073 010123 PC,R2
000074 010124 PC,R2
000075 010125 PC,R2
000076 010126 PC,R2
000077 010127 PC,R2
000078 010128 PC,R2
000079 010129 PC,R2
000080 010130 PC,R2
000081 010131 PC,R2
000082 010132 PC,R2
000083 010133 PC,R2
000084 010134 PC,R2
000085 010135 PC,R2
000086 010136 PC,R2
000087 010137 PC,R2
000088 010138 PC,R2
000089 010139 PC,R2
000090 010140 PC,R2
000091 010141 PC,R2
000092 010142 PC,R2
000093 010143 PC,R2
000094 010144 PC,R2
000095 010145 PC,R2
000096 010146 PC,R2
000097 010147 PC,R2
000098 010148 PC,R2
000099 010149 PC,R2
000100 010150 PC,R2

```

:TEST READ INTERRUPT VECTOR S/R

```

006710 016767 171112 000010 TRVECT: MOV IVCTRD,208 ;GET CURR INT VECT ADR
006711 016768 000004 MOV FLAGADR(R3),-(SP) ;STORE FLAG ADR TO IDENTIFY ME
006712 016769 171114 JSR RS,STSTVEC ;DO I HAVE VECTOR CONTROL?
006713 016770 000008 208: .WORD X00X ;RPG WILL TELL ME SINCE I CAN'T
006714 016771 171116 .WORD RDINT- ;GET AT LOWER MEN IF MEN RIGHT
006715 016772 000001 BR TRVECT ;BR IF I DONT HAVE CONTROL
006716 016773 000001 TST (R5)+ ;BYPASS BR INST IN S/R CALL
006717 016774 000205 TRVECT: RTS R5 ;EXIT IN-LINE

```

:TEST WRITE INTERRUPT VECTOR S/R

```

006740 016767 171062 000016 TRVECT: MOV IVCTRD,208 ;GET CURR INT VECT ADR
006741 0162767 000004 000010 R00 R4,208 ;ADJUST FOR WRITE INT

```


.SBTTL FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

: PROGRAM TABLE FORMAT

000242

PTLGTH= 162. ;PROGRAM TABLE LENGTH - NON NEW NIGHT VERSION OF MPG

;(PTLGTH= 212. ;PROGRAM TABLE LENGTH - NEW NIGHT VERSION OF MPG)

000000

PFLAGD= +0. ;PROGRAM FLAG WORD - 1 WORD

000002

URSTOP= 2	:	1	= USER HAS STOPPED THIS PROGRAM
ERSTOP= 4	:	1	= AN ERROR HAS STOPPED THIS PROGRAM
WT410T= 10	:	1	= WAITING FOR I/O TERMINATION
CTPRIO= 20	:	1	= CONSOLE OR PRINTER I/O IN PROGRESS
SETDED= 40	:	1	= THIS PROG SET THE PRY DEV DEDICATED FLAG
OCPRES= 100	:	1	= OBJ CODE IS PRESENT
USELIM= 200	:	1	= THIS PROG USES THE UNIBUS MAP (NEW NIGHT ONLY)
ACTIVE= 100000	:	1	= PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

000002

POPSiz= +2. ;PROGRAM'S OPERATION SWITCHES - 1 WORD

100000

STONER= 100000	:	1	= STOP PROG EXECUTION UPON ERROR
CYCPRG= 40000	:	1	= CYCLE PROGRAM (ON CURRENT DEVICE)
PRONER= 20000	:	1	= DO NOT PRINT ON ERROR
BIT12= 10000	:	1	= NOT USED
BIT11= 4000	:	1	= NOT USED
CYCDVL= 2000	:	1	= CYCLE THE DEVICE LIST
STRTKD= 1000	:	1	= CYCLE ON SAME DEVICE UPON ERROR
DOERCK= 400	:	1	= DON'T DO ERROR CHECKING
SOPER= 200	:	1	= DEVICE SPECIAL OPERATION
BIT6= 100	:	1	= NOT USED
DO10T= 40	:	1	= DO NOT PERFORM I/O TIMEOUT
AUTORP= 20	:	1	= DO NOT AUTOMATICALLY DISPLAY COUNTS
AURPEP= 10	:	1	= AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
HSKPEP= 4	:	1	= HOUSEKEEP COUNTS ONLY AT RUN COMMAND
PFBBOV= 2	:	1	= PRINT FIRST BAD BYTE ONLY ON VERIFY
NOCOMP= 1	:	1	= DO NOT PRINT PROG COMPLETED MSG

000004

PFLAGDR= +4. ;PROGRAM FLAGWORD ADDRESS - 1 WORD

000006

PASCIN= +6. ;PROGRAM'S NUMBER IN ASCII - 1 WORD

000010

PNAME= +8. ;PROGRAM'S NAME IN ASCII - 6 BYTES

000016

PROIOR= +14. ;ADDRESS OF READ I/O AREA - 1 WORD

000020

PWRIOR= +16. ;ADDRESS OF WRITE I/O AREA - 1 WORD

000022

PSRCST= +18. ;SOURCE STATEMENTS START ADDRESS - 1 WORD

000024

POBJST= +20. ;OBJECT CODE START ADDRESS - 1 WORD

000026

PLNGTH= +22. ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD

000030

PTOENT= +24. ;I/O TIMEOUT COUNT - 1 WORD

.....

000032	PMDLCO= +26.	:DEV ROUT MODEL 8 CODE - 1 WORD
000034	PPNTR= +28.	:CURRENT DEVICE NUMBER POINTER - 1 BYTE
000035	PCURDV= +29.	:CURRENT DEVICE 8 - 1 BYTE
000036	PONUMS= +30.	:DEVICE NUMBERS - 16 BYTES
000056	PTEM0= +46.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000060	PTEM1= +48.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000062	PTEM2= +50.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000064	PTEM3= +52.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000066	PTEM4= +54.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000070	PTEM5= +56.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000072	PTEM6= +58.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000074	PTEM7= +60.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000076	PTEM8= +62.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000100	PTEM9= +64.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000102	PTEM10= +66.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000104	PTEM11= +68.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000106	PTEM12= +70.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000110	PTEM13= +72.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000112	PTEM14= +74.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000114	PTEM15= +76.	:USER PROGRAM TEMPORARY STORAGE - 1 WORD
000116	PNBR= +78.	:NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
000120	PSRC= +80.	:DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
000122	PDST= +82.	:DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
000124	PSTKCT= +84.	:# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
000126	PSTKSV= +86.	:STACK WORDS STORAGE AREA - 30 WORDS
000222	PSVREG= +146.	:USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
000236	PUSRPC= +158.	:USER'S CURRENT PROGRAM COUNTER - 1 WORD

PC 00000
 PC 00001
 PC 00002
 PC 00003
 PC 00004
 PC 00005
 PC 00006
 PC 00007
 PC 00008
 PC 00009
 PC 00010
 PC 00011
 PC 00012
 PC 00013
 PC 00014
 PC 00015
 PC 00016
 PC 00017
 PC 00018
 PC 00019
 PC 00020
 PC 00021
 PC 00022
 PC 00023
 PC 00024
 PC 00025
 PC 00026
 PC 00027
 PC 00028
 PC 00029
 PC 00030
 PC 00031
 PC 00032
 PC 00033
 PC 00034
 PC 00035
 PC 00036
 PC 00037
 PC 00038
 PC 00039
 PC 00040
 PC 00041
 PC 00042
 PC 00043
 PC 00044
 PC 00045
 PC 00046
 PC 00047
 PC 00048
 PC 00049
 PC 00050
 PC 00051
 PC 00052
 PC 00053
 PC 00054
 PC 00055
 PC 00056
 PC 00057
 PC 00058
 PC 00059
 PC 00060
 PC 00061
 PC 00062
 PC 00063
 PC 00064
 PC 00065
 PC 00066
 PC 00067
 PC 00068
 PC 00069
 PC 00070
 PC 00071
 PC 00072
 PC 00073
 PC 00074
 PC 00075
 PC 00076
 PC 00077
 PC 00078
 PC 00079
 PC 00080
 PC 00081
 PC 00082
 PC 00083
 PC 00084
 PC 00085
 PC 00086
 PC 00087
 PC 00088
 PC 00089
 PC 00090
 PC 00091
 PC 00092
 PC 00093
 PC 00094
 PC 00095
 PC 00096
 PC 00097
 PC 00098
 PC 00099
 PC 00100

; FOLLOWING ENTRIES (PROIOX THRU PUBMAP) ARE ONLY IN NEW MGMT VERSION

- ; (PROIOX= +160. ; 18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
- ; (PROIOV= +164. ; 18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
- ; (PWIOX= +168. ; 18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
- ; (PWIOV= +172. ; 18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
- ; (PUPARS= +176. ; STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
- ; (PUPORS= +192. ; STORAGE AREA FOR USER'S POR'S 0 THRU 7 - 8 WORDS)
- ; (PUBMAP= +208. ; 1ST UNIBUS MAP REG 0 AND 8 OF REGS USED - 1 WORD)

; END OF NEW MGMT ONLY ENTRIES

000240

- PTSIZE= +160. ; PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON NEW MGMT
- ; (PTSIZE= +210. ; PROGRAM TABLE SIZE IN BYTES - 1 WORD - NEW MGMT VERSION)

000242

- PTEND= +162. ; END OF PROGRAM TABLE - NON NEW MGMT VERSION
- ; (PTEND= +212. ; END OF PROGRAM TABLE - NEW MGMT VERSION)

: DEVICE ROUTINE TABLE

000116	DRTLTH= 78.	:DEVICE ROUTINE TABLE LENGTH
000000	DEVSZ= +0.	:DEVICE ROUTINE SIZE IN BYTES - 1 WORD
000002	DEVFMD= +2.	:DEVICE ROUTINE FLAGWORD - 1 WORD
000004	DEVIM1= +4.	:DEVICE INTERFACE WORD # 1 - 1 WORD
000006	DEVIM2= +6.	:DEVICE INTERFACE WORD # 2 - 1 WORD
000010	DEVIM3= +8.	:DEVICE INTERFACE WORD # 3 - 1 WORD
000012	DEVIM4= +10.	:DEVICE INTERFACE WORD # 4 - 1 WORD
000014	DEVIM5= +12.	:DEVICE INTERFACE WORD # 5 - 1 WORD
000016	DEVIM6= +14.	:DEVICE INTERFACE WORD # 6 - 1 WORD
000020	DEVIM7= +16.	:DEVICE INTERFACE WORD # 7 - 1 WORD (SIZE)
000022	DEVIM8= +18.	:DEVICE INTERFACE WORD # 8 - 1 WORD (ERR)
000024	DEVORA= +20.	:DEVICE REGISTERS ADDRESS - 1 WORD
000026	DEVIVA= +22.	:DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
000030	DEVAPS= +24.	:DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
000032	DEVWPS= +26.	:DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
000034	DHPRAD= +28.	:DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
000036	DERPAD= +30.	:DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
000040	DKILAD= +32.	:DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
000042	DECTAD= +34.	:DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
000044	DTOERD= +36.	:DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
000046	DEVIOB= +38.	:DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
000050	DEVDER= +40.	:DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
000052	DVUPRT= +42.	:USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
000054	DVCPRT= +44.	:CMD MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
000056	DEVBTB= +46.	:CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
000060	DVBTDH= +48.	:CONVERT BINARY TO DECIMAL ASCII BR ADR (BTABLZ) - 1 WORD

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060

```

000062      DVPOTA= +50.      ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
000064      DVSFMD= +52.      ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
000066      DVSVEC= +54.      ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
000070      DVCVEC= +56.      ;CLEAR INTERRUPT VECTOR BR ADR (CLAVEC) - 1 WORD
000072      DVTVEC= +58.      ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
000074      DVRINT= +60.      ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
000076      DVGETB= +62.      ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
000100      DVPUTB= +64.      ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
000102      DEVSTP= +66.      ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
000104      DEVETP= +68.      ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
000106      DVPTPE= +70.      ;PACK TABLE EXTEN. REL POINTER - 1 WORD
000110      DVVTEP= +72.      ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
000112      DVCTEP= +74.      ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
000114      DVIMSP= +76.      ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
000116      DATEND= +78.      ;END OF DEVICE ROUTINE TABLE

000001      .END

```

ABBREV	005556R	002	DATAER	001260R	002	DVPTER	000106	HSKPEP	000004	PRRAN	006410R	002
ACTIVE	100000	0000	DECRSC	000062R	002	DVPLIT	000100	HSKPST	000116R	PRRER	001244R	002
ALARM	003677R	0000	DECTAD	000072	0000	DVREGG	000176	INCLNC	000125R	PASCIN	000006	0000
ALPMSG	004014R	0000	DEPRAD	000036	0000	DVREGS	000116R	INTEXT	000146R	PC	000000	0000
ATMSG	007003R	0000	DEVTRA	000056	0000	DVREJD	007522R	ISTAT	000116R	PJRDV	000003	0000
RUPPEP	000010	0000	DEVUER	000050	0000	DVREX	005546R	IVCTAD	000026R	PJULPS	000003	0000
RUTORP	000020	0000	DEVVAR	000024	0000	DVRGDT	007102R	KILL	000040R	PJWTR	000002	0000
BR	001176R	0000	DEVVETP	000104	0000	DVRGNG	007076R	KILLEX	000114R	POST	000122R	0000
BRALD	003052R	0000	DEVVMO	000002	0000	DVRIPI	000074	LALAM	000111R	PFB0V	000002	0000
BRALPS	003210R	0000	DEVVID	000076	0000	DVSFMD	000064	LBLUD	001111R	PLGAD	000000	0000
BRCHRO	007127R	0000	DEVVIR	000026	0000	DVSVEC	000006	LBITS	001111R	PLHGR	000002	0000
BRCHS	007114R	0000	DEVVIM1	000024	0000	DVTVEC	000072	LCOUNT	001111R	PLMGR	000002	0000
BRCH	001176R	0000	DEVVIM2	000026	0000	DVUPT	000052	LORST	001111R	PLSK	006412R	002
BRCHINSC	000006R	0000	DEVVIM3	000010	0000	DVWTEP	000110	LORFLG	000001	PLSCL	000002	0000
BRCHINSC	000006R	0000	DEVVIM4	000012	0000	ECHO	005570R	LECHG	001111R	PLSMA	000010	0000
BRCHINSC	000006R	0000	DEVVIM5	000014	0000	ENGLN	006070R	LEVEN	001111R	PLSMB	000116	0000
BRCHINSC	000006R	0000	DEVVIM6	000016	0000	ENSCG	006054R	LEXPX	001111R	PLSMD	006776R	002
BRCHINSC	000006R	0000	DEVVIM7	000020	0000	ERCOTR	007514R	LEXPX	001111R	POSTST	000024	0000
BRCHINSC	000006R	0000	DEVVIM8	000022	0000	ERCOTB	006012R	LINE	001111R	POPUL	000002	0000
BRCHINSC	000006R	0000	DEVVIM9	000023	0000	ERCOTF	007401R	LNECHG	001111R	PRIOG	000016	0000
BRCHINSC	000006R	0000	DEVVIM10	000023	0000	ERCHEN	007341R	LNORR	001111R	PRINT	006576R	0000
BRCHINSC	000006R	0000	DEVVIM11	000023	0000	ERCOVR	007361R	LNABIT	001111R	PRONM	007000R	0000
BRCHINSC	000006R	0000	DEVVIM12	000023	0000	ERCPAR	007420R	LNRGIC	001111R	PROMR	002000R	0000
BRCHINSC	000006R	0000	DEVVIM13	000023	0000	ERCSOF	007460R	LOADC	001111R	PRTEX	006700R	0000
BRCHINSC	000006R	0000	DEVVIM14	000023	0000	ERDTRG	005722R	LOC2	001111R	PJ	006700R	0000
BRCHINSC	000006R	0000	DEVVIM15	000023	0000	ERR	005720R	LOC	001111R	PJAC	000123	0000
BRCHINSC	000006R	0000	DEVVIM16	000023	0000	ERRDIS	005630R	LPA	001111R	PJACST	000023	0000
BRCHINSC	000006R	0000	DEVVIM17	000023	0000	ERRRXT	002764R	LPREST	001111R	PJSAVE	004230R	0000
BRCHINSC	000006R	0000	DEVVIM18	000023	0000	ERRFLG	004250R	LPALUD	001111R	PJKCT	000122	0000
BRCHINSC	000006R	0000	DEVVIM19	000023	0000	ERRMSG	007322R	LSADUR	001111R	PJKSV	000122	0000
BRCHINSC	000006R	0000	DEVVIM20	000023	0000	ERRPT	005620R	LSETU	001111R	PJREG	000223	0000
BRCHINSC	000006R	0000	DEVVIM21	000023	0000	ERRSW	005730R	LSTATS	001111R	PJEND	000005	0000
BRCHINSC	000006R	0000	DEVVIM22	000023	0000	ERRTST	002722R	LSTP1	001111R	PJEN1	000005	0000
BRCHINSC	000006R	0000	DEVVIM23	000023	0000	ERSTOP	000004	LSTP2	001111R	PJEN10	000101	0000
BRCHINSC	000006R	0000	DEVVIM24	000023	0000	ERTBEN	005712R	LTBRUD	001111R	PJEN11	000101	0000
BRCHINSC	000006R	0000	DEVVIM25	000023	0000	EVEN	003342R	LWAIT	001111R	PJEN12	000101	0000
BRCHINSC	000006R	0000	DEVVIM26	000023	0000	FDUPLX	003634R	LNISONT	001111R	PJEN13	000101	0000
BRCHINSC	000006R	0000	DEVVIM27	000023	0000	FLAG	001264R	NWER	000001	PJEN14	000101	0000
BRCHINSC	000006R	0000	DEVVIM28	000023	0000	FLAGD	000002R	NEROR	001274R	PJEN15	000101	0000
BRCHINSC	000006R	0000	DEVVIM29	000023	0000	FRAMR	001252R	NEVER	001274R	PJEN16	000101	0000
BRCHINSC	000006R	0000	DEVVIM30	000023	0000	FRCVTL	001750R	NOCOMP	000001	PJEN17	000101	0000
BRCHINSC	000006R	0000	DEVVIM31	000023	0000	FRBYT	004710R	NOECHG	003604R	PJEN18	000101	0000
BRCHINSC	000006R	0000	DEVVIM32	000023	0000	FSN0TB	002710R	NOERRR	005020R	PJEN19	000101	0000
BRCHINSC	000006R	0000	DEVVIM33	000023	0000	FSN0TL	002704R	NOERR	005040R	PJEN20	000101	0000
BRCHINSC	000006R	0000	DEVVIM34	000023	0000	GETBYT	000076R	NOERR	005040R	PJEN21	000101	0000
BRCHINSC	000006R	0000	DEVVIM35	000023	0000	GTBYTE	004704R	NOERR	005000R	PJEN22	000101	0000
BRCHINSC	000006R	0000	DEVVIM36	000023	0000	GTXTD	001000	NOERR	005000R	PJEN23	000101	0000
BRCHINSC	000006R	0000	DEVVIM37	000023	0000	GTPTBS	001606R	NWAIT	002776R	PJEN24	000101	0000
BRCHINSC	000006R	0000	DEVVIM38	000023	0000	KECRO	004612R	NRC	001116R	PJEN25	000101	0000
BRCHINSC	000006R	0000	DEVVIM39	000023	0000	PLX	003620R	CCPRES	000100	PLGTH	000240R	0000
BRCHINSC	000006R	0000	DEVVIM40	000023	0000	KEEP	005260R	COO	003360R	PTCNT	000030	0000
BRCHINSC	000006R	0000	DEVVIM41	000023	0000	HSKPEN	001266R	UNESTP	003412R	PTSIZE	000240	0000
BRCHINSC	000006R	0000	DEVVIM42	000023	0000			OVRUN	001250R	PUSRPC	000236	0000

PUBBYT	000100R	002	RERVEC	005204R	000	STRSCN	005120R	000	UNSC01	007066R	000	UNITAC	007046R	002
PUBIOR	000020		RESREG	006172R	000	STOPR0	005140R	000	TWVCT	006740R	000	USSTO	000000	
PSCONS	120000		RETJAN	007262R	000	STOPR1	004560R	000	TWVXT	006770R	000	USLST	000000	
RBALD	003306R	000	RICNT	001124R	000	STOPR2	004560R	000	ULIST	000050R	000	USLST	000000	
RBALSB	003106R	000	RPTBAS	000512R	000	STOPR3	004560R	000	UNSC01	007066R	000	USLST	000000	
RBSAB	000030R	000	RPTEND	000512R	000	STOPR4	004560R	000	UNSC01	007066R	000	USLST	000000	
RCA	000000R	000	RPTLP	000512R	000	STOPR5	004560R	000	UNSC01	007066R	000	USLST	000000	
RCVTBL	000000R	000	RPTINTV	000512R	000	STOPR6	004560R	000	UNSC01	007066R	000	USLST	000000	
ROBSY	002000		RTERM	000512R	000	STOPR7	004560R	000	UNSC01	007066R	000	USLST	000000	
ROCNT	001232R	000	RTRNINT	000512R	000	STOPR8	004560R	000	UNSC01	007066R	000	USLST	000000	
ROINS	007270R	000	RTRNLP	000512R	000	STOPR9	004560R	000	UNSC01	007066R	000	USLST	000000	
ROINT	000000R	000	RINTV	000512R	000	STOPR10	004560R	000	UNSC01	007066R	000	USLST	000000	
ROINCI	001640R	000	RIB	000000R	000	STOPR11	004560R	000	UNSC01	007066R	000	USLST	000000	
ROINCI	001640R	000	RIB	000000R	000	STOPR12	004560R	000	UNSC01	007066R	000	USLST	000000	
ROINOLT	002020R	000	RIB	000000R	000	STOPR13	004560R	000	UNSC01	007066R	000	USLST	000000	
RELEB	001500R	000	RIB	000000R	000	STOPR14	004560R	000	UNSC01	007066R	000	USLST	000000	
REMOF	007670R	000	RIB	000000R	000	STOPR15	004560R	000	UNSC01	007066R	000	USLST	000000	
REMOF	007670R	000	RIB	000000R	000	STOPR16	004560R	000	UNSC01	007066R	000	USLST	000000	
REPORT	005314R	000	SVREG	006160R	000	STOPR17	004560R	000	UNSC01	007066R	000	USLST	000000	
REPTBL	005560R	000	RTBSY	001160R	000	STOPR18	004560R	000	UNSC01	007066R	000	USLST	000000	
RES.	000000	000				STOPR19	004560R	000	UNSC01	007066R	000	USLST	000000	
DH11	007522	000				STOPR20	004560R	000	UNSC01	007066R	000	USLST	000000	

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DTDH4B/NL:TOC/DOC=DTDH4B.P11
RUN-TIME: 4 10 1 SECONDS
RUN-TIME RATIO: 40/16=2.4
CORE USED: 5K (9 PAGES)

DOCUMENT PAGES: 38

NO3

