

MPG

COMMON SECTIONS (MPG)
MD-11-DTECA-B

EP-DTECA-B-DL-B
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

| | | | | | |
|----------|----------|----------|----------|----------|----------|
| Frame 1 | Frame 2 | Frame 3 | Frame 4 | Frame 5 | Frame 6 |
| Frame 7 | Frame 8 | Frame 9 | Frame 10 | Frame 11 | Frame 12 |
| Frame 13 | Frame 14 | Frame 15 | Frame 16 | Frame 17 | Frame 18 |
| Frame 19 | Frame 20 | Frame 21 | Frame 22 | Frame 23 | Frame 24 |
| Frame 25 | Frame 26 | Frame 27 | Frame 28 | Frame 29 | Frame 30 |
| Frame 31 | Frame 32 | Frame 33 | Frame 34 | Frame 35 | Frame 36 |
| Frame 37 | Frame 38 | Frame 39 | Frame 40 | Frame 41 | Frame 42 |
| Frame 43 | Frame 44 | Frame 45 | Frame 46 | Frame 47 | Frame 48 |
| Frame 49 | Frame 50 | Frame 51 | Frame 52 | Frame 53 | Frame 54 |
| Frame 55 | Frame 56 | Frame 57 | Frame 58 | Frame 59 | Frame 60 |

Small vertical text or markings at the bottom right corner of the microfiche card.

100
101
102
103
104
105
106
107
108
109
110
111
112

.....

- OCT 75 REMOVED TYPING OF A CR/LF WHEN ENCOUNTERING A 00 BYTE ON THE CONSOLE TERMINAL.
- OCT 75 REMOVED 'DEV NOT RDY' MESSAGE FROM THE PAPER TAPE HANDLER. WILL NOW REPORT 'DEV ERR'. ALSO ADDED CODE THAT IGNORES LEADING 0'S ON THE FIRST DATA BLOCK.
- OCT 75 CORRECTED BUGS THAT CAUSED INCORRECT PRINTING WHEN THE LIST DEVICE WAS AN LV11.
- AUG 75 INITIAL RELEASE

114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167

:SBITL MPG EXEC BUILD INFORMATION
:TITLE MAINTDEC-11-DTECA-B COMMON SECTION FOR ALL MPG EXECUTIVES
;REVISION "B"

MACY11:

DTECA?,DTECA?/CRF:SYM/DOC=DTECA?.P11

OR FOR MIMIC, DTECM?,DTECM?/CRF:SYM/EQ:MIMIC/DOC=DTECA?.P11

NOTE: "/EQ:NLIST" WILL SUPPRESS DISPLAY OF ALL
UNUSED CONDITIONAL CODE IN THE LISTING.

LNKX11:

TC11 = TCMPG.BIN/B:0,TCMPG.MAP+DTECA?,DTEDA?/E
RK11 = RKMPG.BIN/B:0,RKMPG.MAP+DTECA?,DTE1A?/E
TM11 = TMMPG.BIN/B:0,THMPG.MAP+DTECA?,DTE2A?/E
TM02 = THMPG.BIN/B:0,THMPG.MAP+DTECA?,DTE3A?/E
RX11 = RXMPG.BIN/B:0,RXMPG.MAP+DTECA?,DTE4A?/E
RPO4 = RBMPG.BIN/B:0,RBMPG.MAP+DTECA?,DTE5A?/E

OR FOR UNDER MIMIC:

RK11 = RKMPG.BIN/B:0,RKMPG.MAP+DTECM?,DTE1A?,DTMGA?,XXXX/E

FOR RK11 THIS RESULTS IN A 578 BLOCK DISK.
WHERE XXXX IS AN OPTIONAL DEVICE ROUTINE.

THE LINK MAPS, WHICH ARE IN MAINDEC DOCUMENT #
11-DTEMA-?, PROVIDE THE ACTUAL MEMORY ADDRESSES
OF THE VARIOUS MODULES IN EACH VERSION OF THE
EXECUTIVE.

FOR PAPER TAPE OUTPUT:

PUNCH TCMPG.BIN/FILE:ELEV
PUNCH RKMPG.BIN/FILE:ELEV
PUNCH TMMPG.BIN/FILE:ELEV
PUNCH THMPG.BIN/FILE:ELEV
PUNCH RXMPG.BIN/FILE:ELEV
PUNCH RBMPG.BIN/FILE:ELEV

TCMPG UNIQUE MODULES = MAINDEC # DTEDA-?
RKMPG UNIQUE MODULES = MAINDEC # DTE1A-?
TMMPG UNIQUE MODULES = MAINDEC # DTE2A-?
THMPG UNIQUE MODULES = MAINDEC # DTE3A-?
RXMPG UNIQUE MODULES = MAINDEC # DTE4A-?
RBMPG UNIQUE MODULES = MAINDEC # DTE5A-?

169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223

.SBTTL MPG EXEC GLOBALS AND EQUATES

.DSABL GBL
.IIF DF NLIST, .NLIST CND

;NOTE: THE SIZE OF THE LINKER OUTPUT OF ANY VERSION OF THE
; EXECUTIVE CANNOT EXCEED OCTAL 20000 BYTES.

;EXTERNAL GLOBAL NAMES

.GLOBL CFTCHI,CSAVEI,DIRBLK,EXCID

;INTERNAL GLOBAL NAMES

.GLOBL BCDCV,BCLEAR,BKREAD,BLKWRT,BMOVE
.GLOBL BUF,CHROUT,CLRBUF,CMPNAM,CRLF,CRPMAP
.GLOBL DATUPK,DELAY,DELEA,DELFI
.GLOBL DEVERR,DEVFUL,EXECST
.GLOBL FLMOF,FSTMOD,ITOA,LDUNUM,LSTFLG,MAPBUF
.GLOBL MBUF,MES,MMAP,NXTBLK
.GLOBL PAKNAM,PIPFLG,READBK,RSTO4,SAVO4
.GLOBL TAB,TEXT,TXNAM,UPKNAM,WRTBLK
.GLOBL WRTLC,SBUF,SBUF10,SBUF2
.GLOBL SMAPBF,SMBUF,SMMAP,STPNM1,STXNAM,STYPNM

000000
000001
000002
000003
000004
000005
000006
000006
000007
001200

000020
000040

000001
000040
002000
004000
010000
020000
040000
100000

177572
172516

R0 =%0
R1 =%1
R2 =%2
R3 =%3
R4 =%4
R5 =%5
R6 =%6
SP =%6
PC =%7
STACK =1200

CTPRIO =20 ;BIT DEF. FOR USER PROG FLGWD
SETDED =40

MMVER =1 ;BIT DEF. FOR MPG SYSTEM FLGWD

UNIMAP =40
CUSPGR =2000
CHPINT =4000
CNTRLO =10000
CTUDED =20000
CRUDED =40000
CTRQBP =100000

MMR0 =177572
MMR3 =172516

;DEVICE DESCRIPTOR BLOCK EQUATES

| | | | | |
|-----|--------|--------|------|--|
| 225 | | | | |
| 226 | | | | |
| 227 | | | | |
| 228 | 177714 | XDHSU | ==64 | ;INDEX TO DEV HANDLER SET UP |
| 229 | 177716 | XWCTR | ==62 | ;INDEX TO WRITE COUNTER. |
| 230 | 177720 | XFLMOD | ==60 | ;INDEX TO FILE MODE INDICATOR |
| 231 | 177722 | XFLCNT | ==56 | ;INDEX TO FILE COUNT |
| 232 | 177724 | XSVMAP | ==54 | ;INDEX TO BLOCK # FOR MAP BLOCK |
| 233 | 177726 | XSVCNT | ==52 | ;INDEX TO ENTRY # IN UFD BLOCK |
| 234 | 177730 | XSVBLK | ==50 | ;INDEX TO BLK # OF UFD FOR CURR FILE |
| 235 | 177732 | XSVNAM | ==46 | ;INDEX TO FILE NAME IN RAD50 (2 WORDS) |
| 236 | 177736 | XSVEXT | ==42 | ;INDEX TO FILE EXTENSION NAME IN RAD50 |
| 237 | 177740 | XSVDAT | ==40 | ;INDEX TO DATE IN DOS FORMAT |
| 238 | 177742 | XSVXX | ==36 | ;INDEX TO (NOT USED?) |
| 239 | 177744 | XISTBK | ==34 | ;INDEX TO BLK # OF FILE'S FIRST DATA BLOCK |
| 240 | 177746 | XBKLG | ==32 | ;INDEX TO # OF BLOCKS IN THE FILE |
| 241 | 177750 | XLSTBK | ==30 | ;INDEX TO # OF LAST BLK WRITTEN |
| 242 | 177752 | XSVUPT | ==26 | ;INDEX TO (NOT USED?) |
| 243 | 177754 | XBT | ==24 | ;INDEX TO BOOT ROUTINE. |
| 244 | 177756 | DRT | ==22 | ;INDEX TO DIRECTORY ROUTINE |
| 245 | 177760 | ZER | ==20 | ;INDEX TO ZERO ROUTINE |
| 246 | 177762 | DLT | ==16 | ;INDEX TO DELETE ROUTINE |
| 247 | 177764 | CLS | ==14 | ;INDEX TO CLOSE ROUTINE |
| 248 | 177766 | ETR | ==12 | ;INDEX TO ENTER ROUTINE |
| 249 | 177770 | SRH | ==10 | ;INDEX TO LOOKUP ROUTINE |
| 250 | 177772 | ALC | ==6 | ;INDEX TO ALLOCATE ROUTINE |
| 251 | 177774 | XSV | ==4 | ;INDEX TO SERVICE ROUTINE (DRIVER). |
| 252 | 177776 | XDN | ==2 | ;INDEX TO DRIVE NUMBER |
| 253 | 000000 | XCM | =0 | ;INDEX TO COMMAND REGISTER |
| 254 | 000002 | XWC | =2 | ;INDEX TO WORD COUNT |
| 255 | 000004 | XBA | =4 | ;INDEX TO BUS ADDRESS |
| 256 | 000006 | XDT | =6 | ;INDEX TO BLOCK NUMBER |
| 257 | 000010 | XCO | =10 | ;INDEX TO COMMAND |
| 258 | 000012 | XRD | =12 | ;INDEX TO READ COMMAND |
| 259 | 000014 | XWT | =14 | ;INDEX TO WRITE COMMAND |
| 260 | 000016 | XBC | =16 | ;INDEX TO REQUESTED BLOCK COUNT |
| 261 | 000020 | XDR | =20 | ;INDEX TO 1ST DIR BLOCK POINTER. |
| 262 | 000022 | XNB | =22 | ;INDEX TO LAST BLOCK # ALLOCATED. |
| 263 | 000024 | XXNAM | =24 | ;INDEX TO ASCII NAME IN DOB |

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282

.SBTTL LOAD MEDIA BOOTSTRAP LOADER

000000'

.CSECT LOADER

: THE SELECTED DEVICE LOADER RESIDES IN THIS
: AREA BETWEEN OCTAL ADR'S 0 TO 577.

: THE LISTINGS THAT CONTAIN THE VARIOUS LOADERS WHICH
: MAY OCCUPY THIS AREA ARE:

: TCMPG = DTE0A-? - TC11 BOOT LOADER
: RKMPG = DTE1A-? - RK11 BOOT LOADER
: TMMPG = DTE2A-? - TM11 BOOT LOADER
: THMPG = DTE3A-? - TM02 BOOT LOADER
: RXMPG = DTE4A-? - RX11 BOOT LOADER
: RBMPG = DTE5A-? - RPO4 BOOT LOADER

```

284 .SBTTL HOUSEKEEPING EXTENSION & INITIAL ENTRY
285
286
287 000000' .CSECT EXEC
288
289
290 ;INITIAL ENTRY AND HOUSEKEEPING EXTENSION
291
292 000000' LOC600:
293 001 .IF DF MIMIC
294 JMP DISKWR ;GO TO DISK CREATE
295 .ENDC
296 000000' 012706 001200 EXECST: MOV #STACK,R6 ;SET UP STACK POINTER
297 000004' 004567 000016' JSR R5,EITHKP ;GO DO 1 TIME HOUSEKEEPING
298 001 .IF NDF MIMIC
299 000010' 016700 000456 MOV MPGST,R0 ;GET ADR OF WHERE MPG WILL GO
300 000014' 016701 000446 MOV ENDMEM,R1 ;GET END OF MEM ADR
301 000020' 160001 SUB R0,R1 ;CALC AMNT OF MEM AFTER THE EXEC
302 000022' 010167 000022 MOV R1,EFMSZ ;STORE IT IN LINK INFO
303 000026' 012720 046524 MOV #TM,(R0)+ ;STORE MPG'S FILE NAME WHICH
304 000032' 012720 MOV (PC)+,(R0)+ ;IS 'TMGA??' OR 'TMMA??'
305 000034' 040507 MPGID: .ASCII /GA/
306 000036' 012710 037477 MOV #??,(R0)
307 000042' 004567 003176 JSR R5,LOADVR ;GO LOAD MPG
308 000046' 000016' MPGADR: .WORD MPGBGN ;MEM ADR
309 000050' 000000 EFMSZ: .WORD XXXX ;MAX BYTE CNT
310 000052' 000060 .WORD LMPGDE- ;DEV ERR ADR
311 000054' 000050 .WORD LMPGIM- ;INSUF MEM ERR ADR
312 000056' 000002 .WORD LMPGIN- ;NORMAL RET ADR
313 000 .ENDC
314 000060' 012700 LMPGIN: MOV (PC)+,R0 ;GET STORED ADR OF INTERFACE ADR'S
315 000062' 000000 ESGADR: .WORD XXXX
316 000064' 016701 000402 MOV MPGST,R1 ;SET UP ADR IN MPG FOR THEM
317 000070' 062701 000006 ADD #6,R1
318 000074' 012702 MOV (PC)+,R2 ;GET # OF ADR'S STORED
319 000076' 000000 ESGCNT: .WORD XXXX
320 000100' 012021 10$: MOV (R0)+,(R1)+ ;MOVE INTERFACE ADR TO MPG
321 000102' 005302 DEC R2 ;DECR COUNT
322 000104' 001375 BNE 10$ ;DONE ALL? (Y,N-10$)
323 ;R1 IS NOW POINTING AT MPG'S
324 ;ENTRY POINT
325 000106' 012706 001200 MOV #STACK,R6 ;SET UP STACK ADR TO BE SURE
326 000112' 016700 000350 MOV ENDMEM,R0 ;SET UP END OF MEM ADR
327 000116' 012702 MOV (PC)+,R2 ;SET UP MPG'S POSITION ADJ FACTOR
328 000120' 000000 POSADJ: .WORD 0 ;MAY BE CHANGED IF MEM MGMNT
329 000122' 000111 JMP (R1) ;GO TO MPG
330
331
332 000124' 004567 005524 LMPGIM: JSR R5,MES ;ISSUE INSUFF MEM ERR MSG
333 000130' 000136' .WORD ISUFM
334 000132' 000000 LMPGDE: HALT ;ERROR DURING MPG LOAD
335 000134' 000721 BR EXECST ;GO TRY AGAIN
336
337 000136' 042452 025122 044440 ISUFM: .ASCIZ /*ER* INSUF MEM FOR MPG/<015><012>
000144' 051516 043125 046440
000152' 046505 043040 051117

```

```

000160' 046440 043520 005015
000166'      000
338      000170'          .EVEN
339
340
341
342          ;LIGHT THE LIGHTS S/R
343
344      000400'          .=      LOC600+400
345
346      000400' 012701      LIGHTS: MOV      (PC)+,R1          ;GET CURRENT BIT PATTERN
347      000402' 000000      LTSPAT: .WORD    0          ;BIT PATTERN STORAGE
348      000404' 012702 000240      MOV      #160.,R2          ;SET UP LOOP CNT
349      000410' 050101      10$:  BIS      R1,R1          ;BIT PATTERN TO THE LIGHTS
350      000412' 050101      BIS      R1,R1
351      000414' 050101      BIS      R1,R1
352      000416' 005302      DEC      R2          ;DECR LOOP CNT
353      000420' 001373      BNE     10$          ;CNT = 0? (Y,N-10$)
354      000422' 005327      DEC      (PC)+          ;DECR 100 X COUNT
355      000424' 000144      L2MCNT: .WORD    100          ;100X CNT
356      000426' 001016      BNE     LTSEX          ;CNT = 0? (Y,N-LTSEX)
357      000430' 012767 000144 177766      MOV      #100.,L2MCNT      ;RESTORE 100 X CNT
358      000436' 005701      TST     R1          ;BIT PATTERN FULL?
359      000440' 100003      BPL     20$          ;Y,N-20$
360      000442' 005067 177734      CLR     LTSPAT          ;INITIALIZE THE PATTERN
361      000446' 000406      BR      LTSEX          ;GO TO EXIT
362      000450' 000261      20$:  SEC          ;EXPAND BIT PATTERN
363      000452' 106067 177724      RORB   LTSPAT
364      000456' 000261      SEC
365      000460' 106167 177717      ROLB   LTSPAT+1
366      000464' 000207      LTSEX: RTS      PC          ;EXIT IN-LINE

```

```

368                                     .SBTTL  MPG EXEC CONSTANTS & BUFFER AREAS
369
370
371 000466' 000000      ENDMEM: .WORD  0      ;LAST MEM WORD ADR STORAGE
372 000470' 000000      TODAY:  .WORD  0      ;TODAY'S DATE CODE
373 000472' 000016'     MPGST:  .WORD  MPGBGN  ;MPG'S MEM START ADR
374
375          000000      XXXX=   0      ;DATA THAT WILL BE TAILORED
376
377 000474' 001144'     $BUF:   .WORD  BUF      ;ADDRESSES NEEDED BY HANDLERS
378 000476' 001146'     $BUF2:  .WORD  BUF+2
379 000500' 001154'     $BUF10: .WORD  BUF+10
380 000502' 000544'     $MAPBF: .WORD  MAPBUF
381 000504' 000744'     $MBUF:  .WORD  MBUF
382 000506' 000754'     $MMAP:  .WORD  MMAP
383 000510' 000524'     $TXNAM: .WORD  TXNAM
384
385 000512' 001200      STKADR: .WORD  STACK
386 000514' 000      LSTFLG: .BYTE  0      ;DIRECTORY "ALL" FLAG
387 000515' 000      FSTMOD: .BYTE  0      ;DIRECTORY "FAST MODE" FLAG
388 000516' 000      PRTFLG: .BYTE  0      ;USE TTY OR PRNTR FLG
389 000517' 000
390
391 000520' 000000      CHKSUM: .WORD  0      ;COMPUTED CKSUM WD STORAGE
392 000522' 000      PIPFLG: .BYTE  0      ;PIP MODE FLAG
393 000523' 000
394 000524' 000      000      000      TXNAM: .BYTE  0,0,0,0,0,0 ;ASCII FILENAME STORAGE
395 000527' 000      000      000
396 000532' 000      000      000      TXEXT: .BYTE  0,0,0
397          000536'      .EVEN
398 000536' 000006      REALNM: .BLKB  6      ;FILENAME OF FILE ON A MATCH
399
400 000544' 000100      MAPBUF: .BLKW  64.     ;MAP BUFFER
401
402 000744' 000004      MBUF:   .BLKW  4      ;ANOTHER MAP BUFFER
403 000754' 000074      MMAP:   .BLKW  60.
404
405
406
407 001144'          BUF:          ;MAIN READ - WRITE BUFFER
408
409
410
411 :
412 :
413 :
414 :

```

```

:
:
:
:
THIS BUFFER AREA (256 WORDS) IS INITIALLY OCCUPIED BY ONE
TIME HOUSEKEEPING S/R'S AND MESSAGES. THE FOLLOWING CODE AND
MESSAGES UP TO 'BUFEND' WILL BE DESTROYED WHEN THE EXEC
LOADS THE MPG PROGRAM FILE.

```

.SBTTL TEMPORARY ONE TIME HOUSEKEEPING S/R'S

416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471

```

*****
ASCII TO BINARY CONVERSION ROUTINE
      JSR      PC,ATOI          S/R CALL
      USES VALUE AT "EKBPTR" AS INPUT DATA ADR
      R2 CONTAINS RESULT UPON EXIT
*****
ATOI:  JSR      R4,SAV04        ;SAVE WORK REG'S
      CLR      R2              ;CLEAR RESULT REG
      MOV      EKBPTR,R0       ;GET ASCII DATA ADR
10$:   MOVB    (R0)+,R3        ;GET ASCII CHAR
      CMPB    R3,#40          ;IS IT A SPACE?
      BEQ     10$             ;N,Y-10$
      SUB     #'0,R3          ;CONVERT IT TO BINARY
      BMI    30$             ;LESS THAN ASCII 0? (N,Y-30$)
      CMPB    R3,#9.         ;GREATER THAN ASCII 9?
      BHI    30$             ;N,Y-30$
      ASL     R2              ;MULT RUNNING TOTAL BY 10
      ADD     R2,R3           ;AND INCLUDE NEW DIGIT'S VALUE
      ADD     R3,R2
      MOVB    (R0)+,R3        ;GET NXT ASCII CHAR
      BR     20$             ;GO CHECK IT
      DEC     R0              ;BACK UP DATA PNTR
30$:   MOV     R0,EKBPTR      ;STORE IT
      MOV     R2,4(SP)        ;STORE RESULT ON STACK
      JSR     PC,RST04        ;RESTORE REG'S
      RTS     PC              ;EXIT IN-LINE

```

```

*****
READ AND CONVERT OCTAL ASCII TO BINARY S/R
      JSR     PC,OATOB        S/R CALL
      BR     LABEL          EXECUTED IF A DATA ERROR
      ON EXIT:
      R0 = 0 = NO DATA RECEIVED
      R1 = DATA RECEIVED IN BINARY

```

```

*****
OATOB: MOV     #EKYBUF,R0    ;SET UP REL ADR OF
      SUB     #5$,R0        ;KEYBOARD BUFFER
      MOV     R0,5$
      JSR     R5,CTRD        ;ISSUE READ FOR DATA
5$:   .WORD   XXXX
      .WORD   21.

```

```

001144' 004467 003546
001150' 005002
001152' 016700 001312'
001156' 112003
001160' 120327 000040
001164' 001774
001166' 162703 000060
001172' 100412
001174' 120327 000011
001200' 101007
001202' 006302
001204' 060203
001206' 006302
001210' 006302
001212' 060302
001214' 112003
001216' 000763
001220' 005300
001222' 010067 001312'
001226' 010266 000004
001232' 004767 003472
001236' 000207

```

```

472 001264' 005000          CLR      R0          ;RESET DATA FND FLAG
473 001266' 005001          CLR      R1          ;CLEAR ACCUM REG
474 001270' 012702 001314'  MOV      #EKYBUF,R2 ;INITIALIZE READ DATA PNTR
475 001274' 112203          10$:  MOVB   (R2)+,R3 ;GET DATA CHAR
476 001276' 122703 000040  CMPB   #40,R3 ;IS IT A SPACE?
477 001302' 001774          BEQ     10$ ;N,Y-10$
478 001304' 122703 000015  20$:  CMPB   #015,R3 ;IS IT A C/R?
479 001310' 001420          BEQ     40$ ;N,Y-40$
480 001312' 122703 000012  CMPB   #012,R3 ;IS IT A L/F?
481 001316' 001415          BEQ     40$ ;N,Y-40$
482 001320' 162703 000060  SUB     #60,R3 ;MAKE CHAR A BINARY VALUE
483 001324' 100420          BMI     80$ ;LESS THAN ASCII 0? (N,Y-80$)
484 001326' 020327 000007  CMP     R3,#07 ;GREATER THAN 7?
485 001332' 101015          BHI     80$ ;N,Y-80$
486 001334' 005200          INC     R0          ;SET DATA FND FLAG
487 001336' 006301          ASL     R1          ;MULT ANY PREV CHAR'S BY 8
488 001340' 006301          ASL     R1
489 001342' 006301          ASL     R1
490 001344' 060301          ADD     R3,R1 ;ADD IN CURRENT CHAR
491 001346' 112203          MOVB   (R2)+,R3 ;GET NEXT CHAR
492 001350' 000755          BR      20$ ;GO PROCESS IT
493 001352' 020127 000377  40$:  CMP     R1,#377 ;VALUE TOO LARGE?
494 001356' 101003          BHI     80$ ;N,Y-80$
495 001360' 062716 000002  ADD     #2,(SP) ;SET UP NORMAL RET POINT
496 001364' 000207          50$:  RTS     PC ;EXIT
497 001366' 004567 004262  80$:  JSR     R5,MES ;ISSUE ERROR MSG
498 001372' 001725'          .WORD  INVDMG
499 001374' 000773          BR      50$ ;RETURN FOR ERROR EXIT
500          001          .IF DF MIMIC
    
```

```

*****
OCTAL BINARY TO ASCII S/R
      JSR     PC,OBTOA          S/R CALL
      R1 = BINARY WORD TO CONVERT
      R2 = ADR TO STORE 6 ASCII DIGITS
      DESTROYS R0,R1,R2,R3
    
```

```

*****
515 OBTOA: CLR      R0          ;CLEAR ACCUM REG
516      MOV     #6,R3 ;SET UP DIGIT CNT
517      BR      20$ ;GO DO BIT 15
518 10$:  CLR     R0          ;CLEAR ACCUM REG
519      ROL     R1          ;SHIFT 3 BITS INTO R0
520      ROL     R0
521      ROL     R1
522      ROL     R0
523 20$:  ROL     R1
524      ROL     R0
525      BIS     #60,R0 ;MAKE CHAR ASCII
526      MOVB   R0,(R2)+ ;STORE IT IN OUTPUT AREA
527      DEC     R3 ;DECR DIGIT CNT
    
```

| | | | | |
|-----|-----|-------|------|-----------------------|
| 528 | | BNE | 10\$ | ;DONE ALL? (Y,N-10\$) |
| 529 | | RTS | PC | ;EXIT IN-LINE |
| 530 | 000 | .ENDC | | |

CHECK KEYBOARD REPLY S/R

| | | |
|-----|-----------|--------------------------|
| JSR | PC,CKRPLY | S/R CALL |
| BR | LABEL | EXECUTED IF INV DATA |
| BR | LABEL | EXECUTED IF Y, CR, OR LF |
| | | IN-LINE IF N |

| | | | | | | |
|-----|---------|--------|---------|-------------|-----------|-------------------------------|
| 544 | 001376' | 012702 | 001314' | CKRPLY: MOV | #EYBUF,R2 | ;INITIALIZE READ DATA POINTER |
| 545 | 001402' | 112203 | | 10\$: MOVB | (R2)+,R3 | ;GET DATA BYTE |
| 546 | 001404' | 122703 | 000040 | CMPB | #40,R3 | ;IS IT A SPACE? |
| 547 | 001410' | 001774 | | BEQ | 10\$ | ;N,Y-10\$ |
| 548 | 001412' | 122703 | 000015 | CMPB | #015,R3 | ;IS IT A C/R? |
| 549 | 001416' | 001413 | | BEQ | 20\$ | ;N,Y-20\$ |
| 550 | 001420' | 122703 | 000012 | CMPB | #012,R3 | ;IS IT A L/F? |
| 551 | 001424' | 001410 | | BEQ | 20\$ | ;N,Y-20\$ |
| 552 | 001426' | 122703 | 000131 | CMPB | #'Y,R3 | ;IS IT A "Y"? |
| 553 | 001432' | 001405 | | BEQ | 20\$ | ;N,Y-20\$ |
| 554 | 001434' | 122703 | 000116 | CMPB | #'N,R3 | ;IS IT AN "N"? |
| 555 | 001440' | 001004 | | BNE | 30\$ | ;Y,N-30\$ |
| 556 | 001442' | 062716 | 000002 | ADD | #2,(SP) | ;BYPASS SECOND BR |
| 557 | 001446' | 062716 | 000002 | 20\$: ADD | #2,(SP) | ;BYPASS FIRST BR |
| 558 | 001452' | 000207 | | 30\$: RTS | PC | ;EXIT |

```

560 .SBTTL TEMPORARY ONE TIME HOUSEKEEPING MESSAGE STORAGE AREA
561
562 001454' 005015 042077 052101 EDMSG: .ASCIZ <015><012>'?DATE (DD-MMM-YY)/<015><012>'*/
    001462' 020105 024040 042104
    001470' 046455 046515 054455
    001476' 024531 005015 000052
563 001504' 042452 025122 044440 EDERMG: .ASCIZ /*ER* INV DATE/
    001512' 053116 042040 052101
    001520' 000105
564 001522' 005015 047503 051516 CON1MG: .ASCII <015><012>'CONS TERMINAL CONSTANTS:'<015><012><015><012>
    001530' 052040 051105 044515
    001536' 040516 020114 047503
    001544' 051516 040524 052116
    001552' 035123 005015 005015
565 001560' 043077 046111 020114 .ASCIZ '?FILL AFTER: 012 / '
    001566' 043101 042524 035122
    001574' 020040 030460 020062
    001602' 020057 000
566 001605' 077 044506 046114 CON2MG: .ASCIZ '?FILL WITH: 000 / '
    001612' 053440 052111 035110
    001620' 020040 030040 030060
    001626' 027440 000040
567 001632' 043077 046111 020114 CON3MG: .ASCIZ '?FILL COUNT: 002 / '
    001640' 047503 047125 035124
    001646' 020040 030060 020062
    001654' 020057 000
568 001657' 077 047503 053116 CON4MG: .ASCIZ '?CONVERT LOWER CASE TO UPPER (Y/N) / '
    001664' 051105 020124 047514
    001672' 042527 020122 040503
    001700' 042523 052040 020117
    001706' 050125 042520 020122
    001714' 054450 047057 020051
    001722' 020057 000
569 001725' 052 051105 020052 INVDMG: .ASCIZ '*ER* INV DATA'<015><012>
    001732' 047111 020126 040504
    001740' 040524 005015 000
570 001745' 015 037412 047514 LDUREQ: .ASCIZ <015><012>'?LOAD/SAVE/FETCH DEV UNIT # / '
    001752' 042101 051457 053101
    001760' 027505 042506 041524
    001766' 020110 042504 020126
    001774' 047125 052111 021440
    002002' 027440 000040
571 002006' 005015 052477 042523 USEMM: .ASCIZ <015><012>'?USE MEM MGMNT (Y/N) / '
    002014' 046440 046505 046440
    002022' 046507 052116 024040
    002030' 027531 024516 027440
    002036' 000040
572 002040' 042440 042530 052503 EXCIDT: .ASCIZ ' EXECUTIVE FOR MPG'<015><012>
    002046' 044524 042526 043040
    002054' 051117 046440 043520
    002062' 005015 000
573 .EVEN
574 = BUF+512.
575 002144' 000000 BUFEND: .WORD 0

```

```

577 .SBTTL INPUT DEVICE DESCRIPTOR BLOCK (DOB)
578
579
580 002146' 000000 IDHSU: .WORD 0 ;XDHSU DEVICE HANDLER SET UP ADDRESS
581 002150' 000000 DOBSTR:
582 002150' 000000 IWCTR: .WORD 0 ;XWCTR OUTPUT FILE OPEN FLAG
583 002152' 000000 IFLMOD: .WORD 0 ;XFLMOD FILE MODE FLAG
584 002154' 000000 IFLCNT: .WORD 0 ;XFLCNT FILE COUNT
585 002156' 000000 ISVMAP: .WORD 0 ;XSVMAP BLK # OF FILE'S MAP BLOCK
586 002160' 000000 ISVCNT: .WORD 0 ;XSVCNT ENTRY # OF CURR FILE IN UFD
587 002162' 000000 ISVBLK: .WORD 0 ;XSVBLK BLK # OF CURR FILE'S UFD
588
589 002164' 000000 000000 ISVNAM: .WORD 0,0 ;XSVNAM FILE'S FILENAME IN RAD50 (2 WORDS)
590 002170' 000000 ISVEXT: .WORD 0 ;XSVEXT FILE'S EXTENSION IN RAD50
591 002172' 000000 ISVDAT: .WORD 0 ;XSVDAT FILE'S CREATION DATE IN DOS FORMAT
592 002174' 000000 ISVXX: .WORD 0 ;XSVXX (NOT USED?)
593 002176' 000000 IISTBK: .WORD 0 ;XISTBK BLOCK # OF FILE'S FIRST DATA BLK
594 002200' 000000 IBKLG: .WORD 0 ;XIBKLG # OF BLOCKS IN THE FILE
595 002202' 000000 ILSTBK: .WORD 0 ;XLSTBK BLOCK # OF LAST DATA BLOCK WRITTEN
596 002204' 000000 ISVUPT: .WORD 0 ;XSVUPT (NOT USED?)
597
598 002206' 000000 INBOOT: .WORD 0 ;XBT ADDRESS OF "BOOT" ROUTINE
599 002210' 000000 INDRT: .WORD 0 ;DRT ADDRESS OF "DIRECTORY" ROUTINE
600 002212' 000000 INZER: .WORD 0 ;ZER ADDRESS OF "ZERO" ROUTINE
601 002214' 000000 INDLT: .WORD 0 ;DLT ADDRESS OF "DELETE" ROUTINE
602 002216' 000000 INCLS: .WORD 0 ;CLS ADDRESS OF "CLOSE" ROUTINE
603 002220' 000000 INETR: .WORD 0 ;ETR ADDRESS OF "ENTER" (CREATE) ROUTINE
604 002222' 000000 INSRH: .WORD 0 ;SRH ADDRESS OF "LOOKUP" (SEARCH) ROUTINE
605 002224' 000000 INALC: .WORD 0 ;ALC ADDRESS OF "ALLOCATE" ROUTINE
606 002226' 000000 INSRV: .WORD 0 ;XSV ADDRESS OF DEVICE DRIVER ROUTINE
607 002230' 000000 INDRV: .WORD 0 ;XDN CURRENT DRIVE (UNIT) NUMBER
608 002232' 000000 INDEV: ;RS POINTS HERE
609 002232' 000000 INCM: .WORD 0 ;XCM ADDRESS OF DEVICE'S COMMAND REGISTER
610 002234' 000000 INWC: .WORD 0 ;XWC CURRENT WORD COUNT
611 002236' 000000 INBA: .WORD 0 ;XBA CURRENT BUS (MEMORY) ADDRESS
612 002240' 000000 INBLK:
613 002240' 000000 IND: .WORD 0 ;XDT CURRENT BLOCK NUMBER
614 002242' 000000 INCOM: .WORD 0 ;XCO CURRENT COMMAND CODE
615 002244' 000000 INPRC: .WORD 0 ;XRD READ COMMAND CODE
616 002246' 000000 INPWC: .WORD 0 ;XWT WRITE COMMAND CODE
617 002250' 000000 INBKCT: .WORD 0 ;XBC REQUESTED BLOCK COUNT
618 002252' 000000 INDIR: .WORD 0 ;XDR ADDRESS OF FIRST DIRECTORY BLK #
619 002254' 000000 INNBK: .WORD 0 ;XNB LAST BLOCK # ALLOCATED (NEXT BLK #)
620 002256' 000 000 000 IFNAM: .BYTE 0,0,0 ;XXNAM FILE'S NAME IN ASCII (9 CHAR'S)
621 002261' 000 000 000 .BYTE 0,0,0
622 002264' 000 000 000 .BYTE 0,0,0
623 002270' .EVEN
624 002270' INDEND= .

```


674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729

.SBTTL FILE ROUTINES REQUIRED FOR MPG

;512 BYTE DATA BLOCK FORMAT

;NEXT BLOCK # (NOT USED FOR PAPER TAPE) - 1 WORD
;SYNC WORD (WORD OF 000001) - 1 WORD
;BYTE COUNT (# OF DATA BYTES + 6) - 1 WORD
;REL MEMORY ADR FOR DATA - 1 WORD
;DATA BYTES - MAX OF 492.
;CHECKSUM BYTE - 1 BYTE
;UNUSED BYTES - 2 WORDS (GAP IF PAPER TAPE)
;UNUSED BYTES - 7 BYTES (WILL CONTAIN THE "EOF" BLK
; IF LAST BLOCK OF THE FILE.)

;READ LINKED FILE BLOCK

;JSR R5,GET S/R CALL
;.WORD ERRADR-. REL ERROR RET ADR
;.WORD XXXX ABS DATA START ADR ON RETURN
;.WORD XXXX DATA BYTE COUNT ON RETURN

GET: JSR R5,CREGSV ;SAVE REG'S R0-R5
TSTB EPARBK ;PARTIAL BLOCK FLAG SET?
BNE 85\$;N,Y-85\$
10\$: JSR PC,ERDLNK ;READ A LINKED BLOCK
BR EGERT ;ERROR ON READ? (N,Y-EXERET)
MOV #BUF+2,R0 ;SET UP BUFFER ADR
20\$: CLR EUSRCT ;CLEAR USER DATA BYTE COUNT
CLR CHKSUM ;INITIALIZE CHECKSUM VALUE
JSR PC,EGETBY ;GET A BYTE (WILL BE IN R3)
DEC R3 ;IS IT THE 1 OF A POSSIBLE SYNC WORD?
BNE 20\$;Y,N-20\$
JSR PC,EGETBY ;GET ANOTHER BYTE (SYNC IS WORD OF 1)
TSTB R3 ;IS IT THE 0 OF A SYNC WD?
BNE 20\$;Y,N-20\$
JSR PC,EGETWD ;GET DATA BYTE COUNT
SUB #6,R3 ;ALLOW FOR THE 3 HEADER WORDS
MOV R3,EDTACT ;SAVE DATA COUNT
BNE 30\$;THIS THE EOF BLK? (Y,N-30\$)
INCB ERDEOF ;SET THE EOF BLK FLAG
30\$: JSR PC,EGETWD ;GET MEM ADR/START ADR WORD
MOV R3,EMSADR ;SAVE IT
MOV R0,EUSRSA ;SAVE BUF POINTER AS USER START ADR
50\$: MOVB (R0)+,R3 ;GET A DATA BYTE
ADD R3,CHKSUM ;INCLUDE IT IN THE CKSUM
DEC EDTACT ;DECR BLOCK DATA COUNT
BPL 65\$;WAS IT CKSUM BYTE? (Y,N-65\$)
TSTB CHKSUM ;IS THE CHECK SUM CORRECT?
BEQ 55\$;N,Y-55\$
725: JMP CKS'ER ;GO REPORT CKECK SUM ERROR
55\$: TSTB ERDEOF ;IS THIS THE EOF BLK?
RNE 80\$;N,Y-80\$
CMP R0,#BUF+512. ;AT THE END OF THE BUFFER?
BNE 60\$;Y,N-60\$

002412' 004567 001554
002416' 105767 000362
002422' 001116
002424' 004767 001750
002430' 000551
002432' 012700 001146'
002436' 005067 000340
002442' 005067 176052
002446' 004767 000270
002452' 005303
002454' 001370
002456' 004767 000260
002462' 105703
002464' 001364
002466' 004767 000222
002472' 162703 000006
002476' 010367 000270
002502' 001002
002504' 105267 000275
002510' 004767 000200
002514' 010367 000254
002520' 010067 000254
002524' 112003
002526' 060367 175766
002532' 005367 000234
002536' 100021
002540' 105767 175754
002544' 001402
002546' 000167 003672
002552' 105767 000227
002556' 001026
002560' 020027 002144'
002564' 001003

| | | | | | | | | |
|-----|---------|--------|---------|---------|-------|----------------|--|--------------------------------------|
| 730 | 002566' | 105067 | 000212 | | CLRB | EPARBK | | : RESET PARTIAL BLOCK FLG |
| 731 | 002572' | 000415 | | | BR | 75\$ | | : GO TO EXIT |
| 732 | 002574' | 010067 | 000176 | 60\$: | MOV | RO, ENXDTA | | : SAVE CURR POINTER AS NEXT DATA ADR |
| 733 | 002600' | 000410 | | | BR | 70\$ | | : GO CHECK FOR EXIT |
| 734 | 002602' | 005267 | 000174 | 65\$: | INC | EUSRCT | | : ADD 1 TO USER DATA COUNT |
| 735 | 002606' | 020027 | 002144' | | CMP | RO, #BUF+512. | | : AT THE END OF THE BUFFER? |
| 736 | 002612' | 001344 | | | BNE | 50\$ | | : Y, N-50\$ |
| 737 | 002614' | 012767 | 001146' | 000154 | MOV | #BUF+2, ENXDTA | | : SET NXT DATA ADR TO START OF BUF |
| 738 | 002622' | 105267 | 000156 | 70\$: | INCB | EPARBK | | : SET PARTIAL BLOCK FLAG |
| 739 | 002626' | 005767 | 000150 | 75\$: | TST | EUSRCT | | : USER DATA CNT = 0? |
| 740 | 002632' | 001412 | | | BEQ | 85\$ | | : N, Y-85\$ |
| 741 | 002634' | 016706 | 175652 | 80\$: | MOV | STKADR, R6 | | : RESTORE STACK POINTER |
| 742 | 002640' | 004567 | 006174 | | JSR | RS, CRTREG | | : RESTORE REGISTERS |
| 743 | 002644' | 005725 | | | TST | (R5)+ | | : POINT PAST ERR RET ADR |
| 744 | 002646' | 016725 | 000126 | | MOV | EUSRSA, (R5)+ | | : STORE DATA START ADR IN LINK INFO |
| 745 | 002652' | 016725 | 000124 | | MOV | EUSRCT, (R5)+ | | : DO SAME WITH DATA COUNT |
| 746 | 002656' | 000205 | | | RTS | R5 | | : RETURN TO THE USER |
| 747 | | | | | | | | |
| 748 | 002660' | 016700 | 000112 | 85\$: | MOV | ENXDTA, RO | | : GET NEXT DATA ADR |
| 749 | 002664' | 020027 | 001146' | | CMP | RO, #BUF+2 | | : IS IT AT THE START OF THE BUF? |
| 750 | 002670' | 001262 | | | BNE | 20\$ | | : Y, N-20\$ |
| 751 | 002672' | 004767 | 001502 | | JSR | PC, ERDLNK | | : READ NEXT LINKED BLOCK |
| 752 | 002676' | 000475 | | | BR | EXERET | | : ERROR ON READ? (N, Y-EXERET) |
| 753 | 002700' | 066767 | 000076 | 000066 | ADD | EUSRCT, EMSADR | | : ADD PREV DATA CNT TO MEM START ADR |
| 754 | 002706' | 005067 | 000070 | | CLR | EUSRCT | | : CLEAR USER'S DATA BYTE COUNT |
| 755 | 002712' | 000702 | | | BR | 40\$ | | : GO PROCESS USER DATA |
| 756 | | | | | | | | |
| 757 | | | | | | | | |
| 758 | 002714' | 004767 | 000022 | EGETWD: | JSR | PC, EGETBY | | : GET A BYTE |
| 759 | 002720' | 110327 | | | MOVB | R3, (PC)+ | | : STORE IT |
| 760 | 002722' | 000000 | | ETMP: | .WORD | 0 | | : WORK WORD |
| 761 | 002724' | 004767 | 000012 | | JSR | PC, EGETBY | | : GET NEXT BYTE |
| 762 | 002730' | 110367 | 177767 | | MOVB | R3, ETMP+1 | | : STORE IT ALSO |
| 763 | 002734' | 016703 | 177762 | | MOV | ETMP, R3 | | : GET BOTH STORED BYTES |
| 764 | 002740' | 000207 | | | RTS | PC | | : EXIT IN-LINE |
| 765 | | | | | | | | |
| 766 | 002742' | 020027 | 002144' | EGETBY: | CMP | RO, #BUF+512. | | : EXHAUSTED DATA BLOCK? |
| 767 | 002746' | 001005 | | | BNE | EGETKG | | : Y, N-EGETKG |
| 768 | 002750' | 004767 | 001424 | | JSR | PC, ERDLNK | | : READ NEXT BLOCK |
| 769 | 002754' | 000446 | | EGERET: | BR | EXERET | | : ERROR ON READ? (N, Y-EXERET) |
| 770 | 002756' | 012700 | 001146' | | MOV | #BUF+2, RO | | : RESTORE DATA ADR |
| 771 | 002762' | 112003 | | EGETKG: | MOVB | (RO)+, R3 | | : GET BYTE FROM BLOCK |
| 772 | 002764' | 060367 | 175530 | | ADD | R3, CHKSUM | | : ADD IT INTO CHECK SUM |
| 773 | 002770' | 000207 | | | RTS | PC | | : EXIT IN-LINE |
| 774 | | | | | | | | |
| 775 | | | | | | | | |
| 776 | 002772' | 000000 | | EDTACT: | .WORD | 0 | | : COUNT OF DATA BYTES IN BLOCK |
| 777 | 002774' | 000000 | | EMSADR: | .WORD | 0 | | : MEMORY ADR OR START ADR |
| 778 | 002776' | 000000 | | ENXDTA: | .WORD | 0 | | : NEXT DATA ADR |
| 779 | 003000' | 000000 | | EUSRSA: | .WORD | 0 | | : USER'S DATA START ADR |
| 780 | 003002' | 000000 | | EUSRCT: | .WORD | 0 | | : USER'S DATA BYTE COUNT |
| 781 | 003004' | 000 | | EPARBK: | .BYTE | 0 | | : PARTIAL BLOCK FLAG |
| 782 | 003005' | 000 | | ERDEOF: | .BYTE | 0 | | : PROCESSING EOF BLK FLAG |

```

784                                     ;OPEN LINKED INPUT FILE
785
786                                     ;JSR   R5,OPENL           S/R CALL
787                                     ;.WORD ERRADR-.        RELATIVE ERROR RETURN ADR
788                                     ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
789
790 003006' 004767 001040      OPENL: JSR   PC,ESUFNM           ;SET UP ".USR" EXT & FTCH I.D.
791 003012' 004567 001154      LDROPN: JSR  R5,CREGSV          ;SAVE REG'S R0-R5
792 003016' 005067 177762          CLR   EPARBK           ;RESET PARTIAL BLK & EOF BLK FLAGS
793 003022' 004767 001006          JSR   PC,ESUFT1        ;DO COMMON SET UP FUNCTIONS
794 003026' 012765 000001 177744  MOV   #1,XISTBK(R5)     ;SET UP DUMMY BLK #
795 003034' 012767 002256' 003322  MOV   #IFNAM,IFNAME    ;INIT FILENAME ADR
796 003042' 004775 177770          JSR   PC,JSR4(R5)      ;GO TO DEV HDLR & LOOK FOR FILE
797 003046' 000417          BR    EJNFND           ;FILE FOUND? (Y,N-FLNOTF)
798 003050' 016565 177744 000006  MOV   XISTBK(R5),XDT(R5) ;MOVE 1ST BLK # TO CURR BLK #
799 003056' 016706 175430      EXNRET: MOV  STKADR,R6    ;RESTORE STACK POINTER
800 003062' 004567 005752          JSR   R5,CRTREG        ;RESTORE REGISTERS
801 003066' 005725      EXDRET: TST  (R5)+      ;INCR PAST ERROR RET ADR
802 003070' 000205          RTS   R5              ;EXIT TO MPG
803
804 003072' 016706 175414      EXERET: MOV  STKADR,R6    ;RESTORE STACK POINTER
805 003076' 004567 005736          JSR   R5,CRTREG        ;RESTORE REGISTERS
806 003102' 061505          ADD   (R5),R5         ;SET UP ERROR RETURN ADR
807 003104' 000205          RTS   R5              ;EXIT TO MPG ERROR ADR
808
809 003106' 000167 003244      EJNFND: JMP  FLNOTF     ;GO TO NOT FOUND ERROR
810
811                                     ;CREATE OUTPUT LINKED FILE
812
813                                     ;JSR   R5,CREATE         S/R CALL
814                                     ;.WORD NNNN           # OF BLOCKS NEEDED
815                                     ;.WORD ERRADR-.        REL ERR PET ADR
816                                     ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
817
818
819 003112' 005725      CREATE: TST  (R5)+      ;BYPASS # OF BLOCKS
820 003114' 004567 001052          JSR   R5,CREGSV          ;SAVE REG'S R0-R5
821 003120' 004767 001700          JSR   PC,CLRBUF         ;CLEAR BUFFER AREA
822 003124' 004767 000632          JSR   PC,ESUSAV        ;DO COMMON SETUP FUNCTIONS
823 003130' 004767 001156          JSR   PC,EINTOL        ;INITIALIZE FOR OUTPUT LINKED FILE
824 003134' 000750          BR    EXNRET          ;EXIT TO MPG VIA NORMAL RETURN
825
826
827                                     ;WRITE LINKED ABS. FORMAT BLOCK
828                                     ;CLOSE LINKED ABS. FORMAT FILE
829
830                                     ;JSR   R5,PUT           PUT S/R CALL
831                                     ;.CLOSE          CLOSE S/R CALL
832                                     ;.WORD ERRADR-.        RELERR RET ADR
833
834 003136' 005067 176002      CLOSE: CLR   BUF          ;INDICATE THIS IS THE LAST BLOCK
835 003142' 004567 001024      PUT:   JSR   R5,CREGSV    ;SAVE REG'S R0-R5
836 003146' 005003          CLR   R3              ;INITIALIZE CHECKSUM VALUE
837 003150' 012700 001146'      MOV   #BUF+2,R0        ;GET BUFFER START ADR
838 003154' 016002 000002      MOV   2(R0),R2         ;GET BYTE COUNT
839 003160' 012710 000001      MOV   #1,(R0)         ;PLACE SYNC WORD IN BLOCK

```

```

840 003164' 112004          10$:  MOVB  (R0)+,R4          ;GET BYTE OF BLOCK
841 003166' 060403          ADD   R4,R3          ;ADD IT INTO CKSUM
842 003170' 005302          DEC   R2             ;DECR BYTE COUNT
843 003172' 001374          BNE  10$           ;CNT = 0? (Y,N-10$)
844 003174' 105403          NEGB R3             ;NEGATE THE CKSUM
845 003176' 110320          MOVB  R3,(R0)+      ;STORE CKSUM IN THE BLOCK
846 003200' 005767 175740  TST   BUF           ;THIS THE LAST BLOCK?
847 003204' 001014          BNE  20$           ;Y,N-20$
848 003206' 062700 000004  ADD   #4,R0         ;ALLOW FOR 4 BLANKS
849 003212' 112720 000001  MOVB  #1,(R0)+      ;STORE EOF BLOCK DATA
850 003216' 105020          CLRB (R0)+         ; SYNC WORD
851 003220' 112720 000006  MOVB  #6,(R0)+      ; CNT OF 6
852 003224' 105020          CLRB (R0)+
853 003226' 105020          CLRB (R0)+         ; START ADR OF 0
854 003230' 105020          CLRB (R0)+
855 003232' 112720 000371  MOVB  #371,(R0)+    ; CKSUM
856 003236' 004767 001272  20$: JSR   PC,EWRITL    ;GO WRITE LINKED BLOCK
857 003242' 000705          BR   EXMRET        ;EXIT TO MPG VIA NORMAL RETURN
858
859
860          ;LOAD DEVICE ROUTINE
861
862          ;ALSO USED TO LOAD THE MPG PROGRAM FILE AND
863          ;THE VALID DEVICES FILE
864
865          ;JSR   R5,LOADVR      S/R CALL
866          ;.WORD MEMADR        ABS STARTING MEM ADR FOR DATA
867          ;.WORD BYTCNT        SIZE OF MEM AREA IN BYTES
868          ;.WORD DEVERR-        REL RET ADR IF DEVICE ERR
869          ;.WORD INSFADR-        REL RET ADR IF INSUFF MEM
870          ;.WORD NORMADR-        REL NORMAL RET ADR
871          ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
872
873 003244' 012527          LOADVR: MOV  (R5)+,(PC)+      ;GET STARTING MEM ADR
874 003246' 000000          LDRMST: .WORD 0           ;STARTING & CURR MEM ADR
875 003250' 012527          MOV  (R5)+,(PC)+      ;GET BYTE COUNT
876 003252' 000000          LDRBYC: .WORD 0           ;MEM AREA BYTE COUNT
877 003254' 004567 000712  JSR   R5,CREGSV      ;SAVE ALL REG'S
878 003260' 012700 000004G MOV  #CFTCHI+4,R0     ;GET FETCH INFO ADR
879 003264' 012046          MOV  (R0)+,-(SP)     ;SAVE FETCH'S MDL CODE &
880 003266' 011046          MOV  (R0)-,(SP)     ;UNIT #
881 003270' 016010 177774  MOV  -4(R0),(R0)     ;SET FETCH TO THE LOAD DEV
882 003274' 016040 177772  MOV  -6(R0),-(R0)
883 003300' 016746 175206  MOV  STKADR,-(SP)   ;SAVE STORED STACK POINTER
884 003304' 012767 050115 176752  MOV  #'MP,IFNAM+6    ;SET UP ".MPG" FILENAME EXT
885 003312' 112767 000107 176746  MOVB #'G,IFNAM+8.
886 003320' 012767 047514 003314  MOV  #'L0,EFDEVN    ;SET FUNC DEV NAME TO "LOAD"
887 003326' 012767 042101 003310  MOV  #'AD,EFDEVN+2
888 003334' 004567 177452  JSR   R5,LDRPN      ;GO OPEN DEVICE ROUT FILE
889 003340' 000134          .WORD LDEVER-        ;
890 003342' 005077 177700  CLR  #LDRMST        ;CLEAR FIRST WORD OF DEV ROUT AREA
891 003346' 004567 177040  LDRGET: JSR  R5,GET   ;GET A BLOCK OF THE FILE
892 003352' 000122          .WORD LDEVER-        ;
893 003354' 000000          LDRDTA: .WORD XXXX    ; ADR OF DATA IN BUF
894 003356' 000000          LDRDTC: .WORD XXXX    ; # OF DATA BYTES IN BUF
895 003360' 016700 177770  MOV  LDRDTA,R0      ;GET ADR OF DATA IN BUF

```

```

896 003364' 016702 177766      MOV      LDRDTC,R2      ;GET # OF DATA BYTES IN BUF
897 003370' 001415      BEQ      LDRNEX      ;EOF BLK? (N,Y-LDRNEX)
898 003372' 016701 177650      MOV      LDRMST,R1      ;GET CURR MEM ADR FOR DATA
899 003376' 005767 177650      20$:    TST      LDRBYC      ;MEMORY AREA FILLED?
900 003402' 001440      BEQ      LDRISM      ;N,Y-LDRISM
901 003404' 112021      MOVB     (R0)+,(R1)+    ;MOVE DATA BYTE TO DESIGNATED AREA
902 003406' 005367 177640      DEC      LDRBYC      ;DECR MEMORY AREA BYTE CNT
903 003412' 005302      DEC      R2          ;DECR BUF DATA BYTE CNT
904 003414' 001370      BNE      20$        ;CNT = 0? (Y,N-20$)
905 003416' 010167 177624      MOV      R1,LDRMST    ;STORE NEW CURR MEM ADR
906 003422' 000751      BR      LDRGET      ;GO GET NEXT FILE BLOCK
907 003424' 012767 000003 177270 LDRNEX: MOV      #3,ETMP    ;SET EXIT ADR TO NORMAL RET
908 003432' 012667 175054      LDREXC: MOV     (SP)+,STKADR ;GET ORG STACK POINTER
909 003436' 012667 000006G      MOV      (SP)+,CFTCHI+6 ;RESTORE FETCH UNIT # &
910 003442' 012667 000004G      MOV      (SP)+,CFTCHI+4 ;MDL CODE
911 003446' 016706 175040      MOV      STKADR,R6    ;RESTORE STACK POINTER
912 003452' 004567 005362      JSR      R5,CATREG    ;RESTORE REGISTERS
913 003456' 005367 177240      30$:    DEC      ETMP      ;DECR EXIT RETURN ADR CNT
914 003462' 001402      BEQ      40$        ;POINTING AT CORR ADR? (N,Y-40$)
915 003464' 005725      TST      (R5)+      ;INCR PAST CURR ADR
916 003466' 000773      BR      30$        ;GO CK IF THIS IS THE ADR
917 003470' 061505      40$:    ADD      (R5),R5    ;SET UP RETURN ADR
918 003472' 000205      RTS      R5         ;RETURN TO CALLER
919
920 003474' 012767 000001 177220 LDEVER: MOV      #1,ETMP    ;SET RET ADR TO DEV ERROR ADR
921 003502' 000753      BR      LDREXC      ;GO TO COMMON EXIT
922 003504' 012767 000002 177210 LDRISM: MOV      #2,ETMP    ;SET RET ADR TO INSUFF MEM ADR
923 003512' 000747      BR      LDREXC      ;GO TO COMMON EXIT
924
925
926      ;DELETE LINKED ".USR" FILE
927
928      ;JSR      R5,DELETE      S/R CALL
929      ;.WORD  ERRADR-.      REL ERR RET ADR
930      ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
931
932 003514' 004567 000452      DELETE: JSR      R5,CREGSV    ;SAVE REG'S R0-R5
933 003520' 004767 000236      JSR      PC,ESUSAV    ;DO COMMON SETUP FUNCTIONS
934 003524' 012767 002400' 002632 MOV      #OFNAM,NFNAME ;INIT FILENAME ADR
935 003532' 004775 177762      JSR      PC,ADLT(R5)  ;GO TO DEVICE'S DELETE ROUT
936 003536' 000167 177314      EJNRET: JMP      EXNRET    ;EXIT TO MPG VIA NORMAL RETURN
937
938
939      ;ZERO SAVE DEVICE'S DIRECTORY
940
941      ;JSR      R5,ZERO      S/R CALL
942      ;.WORD  ERRADR-.      REL ERR RET ADR
943
944 003542' 004567 000424      ZERO:   JSR      R5,CREGSV    ;SAVE REG'S R0-R5
945 003546' 004767 001252      JSR      PC,CLRBUF    ;INITIALIZE BUFFER AREA
946 003552' 004567 001222      JSR      R5,BCLEAR    ;CLEAR OUTPUT DDB
947 003556' 002270'      .WORD   ODHSU
948 003560' 000000      .WORD   0
949 003562' 000122      .WORD   DOBEND-ODHSU
950 003564' 004767 000172      JSR      PC,ESUSAV    ;DO COMMON SETUP FUNCTIONS
951 003570' 004775 177760      JSR      PC,ZZER(R5)  ;GO TO ZERO DIR ROUTINE

```

```

952 003574' 000760      BR      EJRNET      ;EXIT VIA NORMAL RETURN
953
954
955                      ;LIST FETCH DEV DIRECTORY
956
957                      ;JSR      RS LIST      S/R CALL
958                      ;.WORD  ERRADR-.      REL ERR RET ADR
959
960 003576' 012567 174712  LIST:  MOV      (RS)+,LSTFLG      ;GET "ALL" & "FAST" FLAGS FROM MPG
961 003602' 004567 000364      JSR      RS,CREGSV      ;SAVE REG'S R0-R5
962 003606' 105267 174704      INCB     PRIFLG      ;SET PRINTER/TTY FLAG
963 003612' 004767 001730      JSR      PC,CRLF      ;ISSUE CR/LF
964 003616' 004767 000206      JSR      PC,ESUFTH     ;DO COMMON SET UP
965 003622' 016703 174642      MOV      TODAY,R3     ;GET TODAY'S DATE CODE
966 003626' 004567 001146      JSR      RS,BCLEAR     ;SET FILENAME TO ALL '?'S
967 003632' 002256      .WORD   IFNAM
968 003634' 037477      .ASCII  '??/'
969 003636' 000011      .WORD   9
970 003640' 004767 002172      JSR      PC,DATUPK     ;CONVERT & DISPLAY IT
971 003644' 004775 177756      JSR      PC,@DRT(R5)   ;GO TO DIRECTORY LIST ROUT
972 003650' 004767 001672      JSR      PC,CRLF      ;ISSUE A CR/LF
973 003654' 000730      BR      EJRNET      ;EXIT VIA NORMAL RETURN
974
975
976                      ;BOOT FROM FETCH DEVICE
977
978                      ;JSR      RS BOOT     S/R CALL
979                      ;.WORD  ERRADR-.      REL ERR RET ADR
980
981 003656' 004567 000310  BOOT:  JSR      RS,CREGSV     ;SAVE REG'S R0-R5
982 003662' 004767 000142      JSR      PC,ESUFTH     ;DO COMMON SET UP
983 003666' 004775 177754      JSR      PC,@XBT(R5)   ;DO PARTIAL BOOT
984 003672' 005065 000004      CLR      XBA(R5)      ;SET READ'S MEM ADR TO 0
985 003675' 010065 000006      MOV      RO,XDT(R5)   ;STORE BOOT BLOCK #
986 003702' 004767 000610      JSR      PC,BKRDO     ;DO READ OF THE BOOT BLK
987 003706' 016500 000012      MOV      XRD(R5),RO   ;GET DEVICE'S READ CMND
988 003712' 042700 000001      SIB      #1,RO        ;RESET THE "GO" BIT
989 003716' 010075 000000      MOV      RO,@(RS)     ;LOAD DEV CSR WITH UNIT #
990 003722' 011500      MOV      (RS),RO      ;STORE CSR ADR IN RO
991 003724' 005077 005210      CLR      @CTRORA      ;RESET TTY INT ENABLE
992 003730' 032777 000001 000012'  BIT      #MMVER,@CSYSFW ;RUNNING UNDER MEM MGMNT?
993 003736' 001410      BEQ     10$          ;Y,N-10$
994 003740' 005037 177572      CLR      @MMMRO       ;DISABLE MEM MGMNT
995 003744' 032777 000040 000012'  BIT      #UNIMAP,@CSYSFW ;USING THE UNIBUS MAP?
996 003752' 001402      BEQ     10$          ;Y,N-10$
997 003754' 005037 172516      CLR      @MMMR3       ;DISABLE IT
998 003760' 005007      CLR      PC          ;BRANCH TO CODE JUST READ - (LOC J)

```

```

1000 .SBTTL SUBROUTINES FOR REQUIRED MPG FILE ROUTINES
1001
1002
1003 ;"SAVE" & OUTPUT DOB COMMON SETUP
1004
1005 ;JSR PC,ESETUP S/R CALL
1006
1007 003762' 012767 051525 176416 ESUSAV: MOV #US,OFNAM+6 ;SET ".USR" EXT IN OUTPUT DOB
1008 003770' 112767 000122 176412 MOVB #R,OFNAM+8.
1009 003776' 012767 040523 002636 MOV #SA,EFDEVN ;SET FUNCT DEV NAME TO "SAVE"
1010 004004' 012767 042526 002632 MOV #VE,EFDEVN+2
1011 004012' 004767 000100 JSR PC,ESUOS ;SET UP FOR SAVE DEV HNDLR
1012 004016' 016702 000006G MOV CSAVEI+6,R2 ;GET SAVE'S UNIT #
1013 004022' 004767 000176 JSR PC,ESUOPT ;INITIALIZE OUTPUT DOB
1014 004026' 000207 RTS PC ;EXIT IN-LINE
1015
1016
1017 ;"FETCH" & INPUT DOB COMMON SET UP
1018
1019 ;JSR PC,ESUFTH S/R CALL
1020 ;JSR PC,ESUFT1 S/R CALL
1021 ;JSR PC,ESUFNM S/R CALL
1022
1023 004030' 004767 000016 ESUFTH: JSR PC,ESUFNM ;SET UP "USR" EXT & FTCH I.D.
1024 004034' 004767 000044 ESUFT1: JSR PC,ESUIF ;SET UP FOR FETCH DEV HNDLR
1025 004040' 016702 000006G MOV CFTCHI+6,R2 ;GET FETCH'S UNIT #
1026 004044' 004767 000146 JSR PC,ESUIPT ;INITIALIZE INPUT DOB
1027 004050' 000207 RTS PC ;EXIT IN-LINE
1028
1029 004052' 012767 051525 176204 ESUFNM: MOV #US,IFNAM+6 ;SET ".USR" EXT IN INPUT DOB
1030 004060' 112767 000122 176200 MOVB #R,IFNAM+8.
1031 004066' 012767 052106 002546 MOV #FT,EFDEVN ;SET FUNCT DEV NAME TO "FTCH"
1032 004074' 012767 044103 002542 MOV #CH,EFDEVN+2
1033 004102' 000207 RTS PC ;EXIT IN-LINE
1034
1035
1036 ;SET UP DEVICE HANDLER ADR IN INPUT OR OUTPUT DOB
1037
1038 ;JSR PC,ESUIF S/R CALL
1039 ;JSR PC,ESUOS S/R CALL
1040
1041 004104' 012700 000004G ESUIF: MOV #CFTCHI+4,R0 ;SET UP "FETCH" MDL CODE ADR
1042 004110' 012701 002146' MOV #IDHSU,R1 ;SET UP INPUT DOB ADR
1043 004114' 000404 BR ESUCOM ;GO TO COMMON PROCESSING
1044
1045 004116' 012700 000004G ESUOS: MOV #CSAVEI+4,R0 ;SET UP "SAVE" MDL CODE ADR
1046 004122' 012701 002270' MOV #ODHSU,R1 ;SET UP OUTPUT DOB ADR
1047 004126' 012702 012006' ESUCOM: MOV #EMDLTB,R2 ;GET ADR OF MDL TBL
1048 004132' 021022 20$: CMP (R0),(R2)+ ;THIS THE DEV'S MDL?
1049 004134' 001404 BEQ 30$ ;N,Y-30$
1050 004136' 005722 TST (R2)+ ;POINT TO NXT ENTRY
1051 004140' 005712 TST (R2) ;END OF THE TBL?
1052 004142' 001412 BEQ 40$ ;N,Y-40$ -- GO TO EXIT
1053 004144' 000772 BR 20$ ;GO CK NXT ENTRY
1054 004146' 011211 30$: MOV (R2),(R1) ;STORE DEVICE'S SET UP ADR
1055 004150' 020227 012010' CMP R2,#PTSUAD ;IS THIS PAPER TAPE?

```

```

1056 004154' 001005          BNE      40$          ;Y,N-40$
1057 004156' 020027 000004G  CMP      RO,#CSAVEI+4 ;IS THIS FOR THE SAVE DEV (PUNCH)?
1058 004162' 001002          BNE      40$          ;Y,N-40$
1059 004164' 062711 000010  ADD      #8.,(R1)      ;POINT ADR AT PT PUNCH SET UP ADR
1060 004170' 000207          RTS      PC           ;EXIT IN-LINE
1061
1062
1063                          ;STORE REGISTERS & SAVE STACK POINTER
1064
1065                          ;JSR      R5,CREGSV      S/R CALL
1066
1067 004172' 010446          CREGSV: MOV     R4,-(SP)   ;SAVE REG'S R4 THRU R0
1068 004174' 010346          MOV     R3,-(SP)   ;R5 SAVED BY JSR TO THIS S/R
1069 004176' 010246          MOV     R2,-(SP)
1070 004200' 010146          MOV     R1,-(SP)
1071 004202' 010046          MOV     R0,-(SP)
1072 004204' 010667 174302  MOV     SP,STKADR   ;SAVE STACK POINTER
1073 004210' 105067 174302  CLR     PRIFLG     ;RESET PRINTER/TTY FLAG
1074 004214' 010507          MOV     R5,PC      ;EXIT IN-LINE
    
```

M02

1076 .SBTTL FILE INITIALIZATION S/R'S REQUIRED FOR LOAD MEDIA HANDLERS

1077
 1078
 1079 ; INITIALIZE INPUT OR OUTPUT DOB S/R
 1080

```

1081 ;JSR PC,ESUIPT S/R CALL
1082 ;R2 CONTAINS UNIT # UPON ENTRY
1083
1084 004216' 012705 002232' ESUIPT: MOV #INDEV,R5 ;SET UP INPUT DOB ADR
1085 004222' 000402 BR ESUIOC ;GO TO COMMON POINT
1086 004224' 012705 002354' ESUOPT: MOV #OUTDEV,R5 ;SET UP OUTPUT DOB ADR
1087 004230' 004775 177714 ESUIOC: JSR PC,2XDHSU(R5) ;GO TO HANDLER'S SET UP ADR
1088 004234' 010067 000020 MOV R0,12$ ;SET DEV PARAM INFO ADR
1089 004240' 010567 000012 MOV R5,10$ ;SET DOB ADR
1090 004244' 062767 177754 000004 ADD #XBT,10$
1091 004252' 004567 000466 JSR R5,BMOVE ;MOVE DEV INFO TO DOB
1092 004256' 000000 10$: .WORD XXXX ;DEST
1093 004260' 000000 12$: .WORD XXXX ;SRC
1094 004262' 000050 .WORD 40 ;CNT
1095 004264' 010504 MOV R5,R4 ;MOVE ASCII FILENAME TO DOB
1096 004266' 062704 000024 ADD #XXNAM,R4
1097 004272' 016703 174174 MOV MPGST,R3
1098 004276' 012324 MOV (R3)+,(R4)+
1099 004300' 012324 MOV (R3)+,(R4)+
1100 004302' 011314 MOV (R3),(R4)
1101 004304' 005065 177722 CLR XFLCNT(R5) ;RESET FILES PROCESSED CNT
1102 004310' 000207 RTS PC ;EXIT IN-LINE
    
```

1103
 1104
 1105 ; INITIALIZE OUTPUT FOR LINKED FILE

```

1106 ;JSR PC,EINTOL S/R CALL
1107
1108
1109 004312' 012705 002354' EINTOL: MOV #OUTDEV,R5 ;SET UP OUTPUT DOB ADR
1110 004316' 005065 177716 CLR XWCTR(R5) ;RESET BLOCKS WRITTEN CNT
1111 004322' 016767 174142 175764 MOV TODAY,OSVDAT ;SET TODAY'S DATE IN DOB
1112 004330' 005067 175766 CLR OBKLG1 ;CLEAR FILE LENGTH
1113 004334' 004775 177766 JSR PC,2ETR(R5) ;GO TO DEV'S "ENTER" ROUT
1114 004340' 012767 177777 176024 MOV #-1,0UBKCT ;SET UP LINKED FILE CODE
1115 004346' 004775 177772 JSR PC,2ALC(R5) ;ALLOCATE THE FIRST BLOCK
1116 004352' 005265 177716 INC XWCTR(R5) ;INCR BLOCKS WRITTEN CNT
1117 004356' 016767 175742 175734 MOV OLSTBK,01STBK ;STORE FIRST BLOCK #
1118 004364' 005267 175732 INC OBKLG1 ;INCR # OF BLKS USED CNT
1119 004370' 012767 000001 174546 MOV #1,BUF ;INITIALIZE BLOCK # IN BUF
1120 004376' 000207 RTS PC ;EXIT IN-LINE
    
```

```

1122 .SBTTL FILE READ S/R'S FOR LOAD MEDIA HANDLERS
1123
1124
1125 ;READ LINKED FILE BLOCK INTO BUF
1126
1127 ;JSR PC,ERDLNK S/R CALL
1128 ;BR ERRADR EXECUTED IF AN ERROR
1129
1130 004400' 012705 002232' ERDLNK: MOV #INDEV,R5 ;GET INPUT DDB ADR
1131 004404' 112767 000001 174110 MOVB #1,PIPFLG ;SET LINK MODE FLAG
1132 004412' 005765 000006 TST XDT(R5) ;IS THERE ANOTHER BLOCK?
1133 004416' 001410 BEQ ERDERR ;Y N-ERDERR
1134 004420' 004767 000060 JSR PC,BKREAD ;GO READ THE BLOCK
1135 004424' 016765 174514 000006 MOV BUF,XDT(R5) ;SAVE NEXT BLOCK ADDRESS
1136 004432' 062716 000002 ENMRET: ADD #2,(SP) ;SET RET ADR FOR NORMAL +2 POINT
1137 004436' 000207 ERET: RTS PC ;EXIT IN-LINE
1138
1139 004440' 012705 006626' ERDERR: MOV #EOFMSG,R5 ;SET UP END OF FILE MSG ADR
1140 004444' 004767 002062 JSR PC,EXERPT ;ISSUE EOF ERROR MSG
1141 004450' 000565 BR CLRBUF ;CLEAR THE BUFFER & EXIT
1142
1143
1144 ;INPUT OR OUTPUT NEXT BLOCK IN FILE
1145
1146 ;JSR PC,NXTBLK S/R CALL
1147
1148 004452' 012765 000400 000002 NXTBLK: MOV #256,XWC(R5) ;SET UP WORD COUNT
1149 004460' 012765 001144' 000004 MOV #BUF,XBA(R5) ;SET UP DATA ADR
1150 004466' 016765 174452 000006 MOV BUF,XDT(R5) ;GET NEXT BLOCK #
1151 004474' 001760 BEQ ERET ;THIS LAST BLOCK? (N,Y-ERET)
1152 004476' 062716 000002 ADD #2,(SP) ;SET UP FOR NORMAL RETURN
1153 004502' 000471 BR EXYBK ;GO DO READ OR WRITE
1154
1155
1156 ;READ A BLOCK INTO THE BUFFER
1157
1158 ;JSR PC,BKREAD S/R CALL
1159 ;JSR PC,READBK S/R CALL
1160
1161 004504' 004767 000314 BKREAD: JSR PC,CLRBUF ;CLEAR THE BUFFER
1162 004510' 012765 001144' 000004 MOV #BUF,XBA(R5) ;SET UP READ DATA ADR
1163 004516' 012765 000400 000002 BKREAD: MOV #256,XWC(R5) ;SET UP WORD COUNT
1164 004524' 016565 000012 000010 READBK: MOV XRD(R5),XCO(R5) ;SET UP THIS DEV'S READ CMND
1165 004532' 000455 BR EXYBK ;GO DO READ OR WRITE

```

```

1167 .SBTTL FILE WRITE S/R'S FOR LOAD MEDIA HANDLERS
1168
1169
1170 ;WRITE A LINKED FILE BLOCK
1171
1172 ;JSR PC,EWRITL S/R CALL
1173
1174 004534' 004467 000156 EWRITL: JSR R4,SAV04 ;SAVE WORK REGISTERS
1175 004540' 012705 002354' MOV #OUTDEV,RS ;GET OUTPUT DEV ADR
1176 004544' 112767 000001 173750 MOVB #1,PIPF LG ;SET LINK MODE FLAG
1177 004552' 016767 175546 175602 MOV OLSTBK,OUTDT ;SET UP BLOCK NUMBER
1178 004560' 005767 174360 TST BUF ;THIS LAST BLK TO BE WRITTEN?
1179 004564' 001415 BEQ IOS ;N,Y-10$
1180 004566' 012767 177777 175576 MOV #-1,OUBKCT ;SET ALLOCATE LINKED FILE CODE
1181 004574' 004775 177772 JSR PC,ALC(RS) ;GO HAVE DEV ALLOCATE NXT BLK
1182 004600' 005267 175516 INC OBKLG T ;INCR REQUESTED BLOCK COUNT
1183 004604' 016767 175514 174332 MOV OLSTBK,BUF ;LINK CURR BLOCK TO NEXT
1184 004612' 004767 000014 JSR PC,WRTLC ;GO WRITE LINKED BLOCK & CLEAR BUF
1185 004616' 000427 BR BKWRT1 ;GO TO EXIT
1186 004620' 004767 000006 IOS: JSR PC,WRTLC ;GO WRITE LINKED BLOCK & CLEAR BUF
1187 004624' 004767 000054 JSR PC,FCLOSE ;GO CLOSE THE FILE
1188 004630' 000422 BR BKWRT1 ;GO TO EXIT
1189
1190
1191 ;WRITE 256 WORD BLK, INCR BLK CNT, & CLEAR BUFFER
1192
1193 ;JSR PC,WRTLC S/R CALL
1194
1195 004632' 004767 000006 WRTLC: JSR PC,BLKWRT ;GO WRITE THE BLOCK
1196 004636' 005265 177716 INC XWCTR(RS) ;INCR BLKS WRITTEN CNT
1197 004642' 000470 BR CLRBUF ;GO CLEAR BUF & EXIT
1198
1199
1200 ;WRITE 256 WORD BLOCK FROM BUF (BLK # ALREADY SET)
1201
1202 ;JSR PC,BLKWRT S/R CALL
1203 ;JSR PC,WRTBLK S/R CALL
1204
1205 004644' 012765 001144' 000004 BLKWRT: MOV #BUF,XBA(RS) ;SET UP DATA ADR
1206 004652' 012765 000400 000002 BKWRTD: MOV #256.,XWC(RS) ;SET UP WORD COUNT
1207 004660' WRTDIR:
1208 004660' 016565 000014 000010 WRTBLK: MOV XWT(RS),XCO(RS) ;SET UP THIS DEV'S WRITE CMND
1209 004666' 004467 000024 EXYBK: JSR R4,SAV04 ;SAVE REGISTERS
1210 004672' 004775 177774 JSR PC,XSV(RS) ;GO READ OR WRITE THIS BLOCK
1211 004676' 004767 000026 BKWRT1: JSR PC,RST04 ;RESTORE REG'S 0-4
1212 004702' 000207 RTS PC ;EXIT IN-LINE
1213
1214
1215 ;CLOSE OUTPUT FILE
1216
1217 ;JSR PC,FCLOSE S/R CALL
1218
1219 004704' 004775 177764 FCLOSE: JSR PC,CCLS(RS) ;GO TO DEVICE'S CLOSE ROUT
1220 004710' 005065 177716 CLR XWCTR(RS) ;CLEAR "OUTPUT FILE OPEN" FLAG
1221 004714' 000207 RTS PC ;EXIT IN-LINE

```

.SBTTL DATA S/R'S REQUIRED FOR LOAD MEDIA HANDLERS

1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278

004716' 010346
004720' 010246
004722' 010146
004724' 010046
004726' 010407

004730' 012604
004732' 012600
004734' 012601
004736' 012602
004740' 012603
004742' 000204

004744' 012767 112120 000012
004752' 004467 177740
004756' 012500
004760' 012501
004762' 012502
004764' 000000

004766' 005302
004770' 001375
004772' 004767 177732
004776' 000205

```

:*****
: SAV04: SAVE REGISTERS R0 THRU R4 ON THE STACK
: LINK: JSR R4, SAV04 S/R CALL
:*****
SAV04: MOV R3, -(SP) ;SAVE R3 THRU R0
      MOV R2, -(SP)
      MOV R1, -(SP)
      MOV R0, -(SP)
      MOV R4, PC ;EXIT -- R4 IS ALREADY SAVED

:*****
: RST04: RESTORE REGISTERS R0 THRU R4 SAVED BY SAV04 S/R
: LINK: JSR PC, RST04 S/R CALL
:*****
RST04: MOV (SP)+, R4 ;SET UP RETURN ADR
      MOV (SP)+, R0 ;RELOAD R0 THRU R3
      MOV (SP)+, R1
      MOV (SP)+, R2
      MOV (SP)+, R3
      RTS R4 ;EXIT IN-LINE & RELOAD R4

:*****
: BMOVE: MOVE BYTE FIELDS S/R
: LINK: JSR R5, BMOVE S/R CALL
: .WORD DESTADR DESTINATION ADR (ABSOLUTE)
: .WORD SRCADR SOURCE ADR (ABSOLUTE)
: .WORD COUNT NUMBER OF BYTES TO MOVE
:*****
BMOVE: MOV #112120, EMINST ;SET UP "MOVB (R1)+, (R0)+" INST
EMCOM: JSR R4, SAV04 ;SAVE REGISTERS
      MOV (R5)+, R0 ;GET DESTINATION ADR
      MOV (R5)+, R1 ;GET SOURCE ADR OR CLEAR DATA
      MOV (R5)+, R2 ;GET BYTE COUNT
EMINST: .WORD XXXX ;"MOVB (R1)+, (R0)+" OR
      ;"MOVB R1, (R0)+" INST
      DEC R2 ;DECR BYTE COUNT
      BNE EMINST ;CNT = 0? (Y, N-EMINST)
ECOMEX: JSR PC, RST04 ;RESTORE SAVED REGISTERS
      RTS R5 ;EXIT IN-LINE
    
```

1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324

```

*****
BCLEAR: CLEAR BYTE FIELDS TO A SPECIFIC VALUE S/R
LINK: JSR RS,BCLEAR S/R CALL
      .WORD DEST,ADR DESTINATION ADR (ABSOLUTE)
      .WORD DATA DATA BYTE TO CLEAR WITH
      .WORD COUNT NUMBER OF BYTES TO CLEAR
      USES "BMOVE" S/R
*****

```

```

005000' 012767 110120 177756 BCLEAR: MOV #110120,EMINST ;SET UP "MOVB R1,(R0)+" INST
005006' 000761 BR EMCOM ;GO TO "BMOVE" S/R

```

```

*****
CRPMAP: CLEAR 128 BYTES TO 0'S STARTING AT "MAPBUF"
LINK: JSR PC,CRPMAP S/R CALL
*****

```

```

005010' 004567 177764 CRPMAP: JSR RS,BCLEAR ;GO CLEAR THE MAPBUF
005014' 000544 .WORD MAPBUF
005016' 000000 .WORD 0
005020' 000200 .WORD 128.
005022' 000207 RTS PC ;EXIT IN-LINE

```

```

*****
CLRBUF: CLEAR 510 BYTES TO 0'S STARTING AT "BUF+2"
LINK: JSR PC,CLRBUF S/R CALL
*****

```

```

005024' 004567 177750 CLRBUF: JSR RS,BCLEAR ;GO CLEAR THE BUFFER
005030' 001146 .WORD BUF+2
005032' 000000 .WORD 0
005034' 000776 .WORD 510.
005036' 000207 RTS PC ;EXIT IN-LINE

```

1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371

CMPNAM: COMPARE TWO 9 BYTE FIELDS

LINK: JSR RS,CMPNAM S/R CALL
BR XXXX BR IF UNEQUAL

IF DATA IS UNEQUAL, RETURN WILL BE SO THAT THE BR
INST IS EXECUTED. IF EQUAL, RETURN WILL BE IN-LINE AFTER
THE BR INST.

005040' 004467 177652
005044' 012700 000024
005050' 066600 000012
005054' 012701 000524
005060' 012702 000011
005064' 122021
005066' 001011
005070' 005302
005072' 001374
005074' 004567 177644
005100' 000536'
005102' 000524'
005104' 000006
005106' 005725
005110' 000730
005112' 126027 177777 000077
005120' 001324
005122' 000762

CMPNAM: JSR R4,SAVD4 ;SAVE WORK REGISTERS
MOV #XXNAM,R0 ;SET UP DOB DISPLACEMENT
ADD 12(SP),R0 ;ADD IN CURR DOB ADR
MOV #TXNAM,R1 ;SET UP SECOND FILENAME ADR
MOV #9,R2 ;SET CNT TO 9
10\$: CMPB (R0)+,(R1)+ ;DATA CHAR MATCH?
BNE 30\$;Y,N-30\$
20\$: DEC R2 ;DECR CHAR CNT
BNE 10\$;CNT = 0? (Y,N-10\$)
JSR RS,BMOVE ;SAVE FILE'S NAME IN CASE THERE
 .WORD REALNM ;WERE '?'S IN DOB'S FILE NAME
 .WORD TXNAM
 .WORD 6
TST (RS)+ ;INCR PAST BR INST
BR ECOMEX ;GO TO EXIT AFTER BR
30\$: CMPB -1(R0),#'?' ;THIS A "DON'T CARE" CHAR?
BNE ECOMEX ;Y,N-ECOMEX
BR 20\$;GO CK NEXT CHAR

DELAY: THIS S/R PERFORMS A COMPUTE BOUND DELAY

LINK: JSR PC,DELAY S/R CALL

005124' 005046
005126' 005316
005130' 100776
005132' 005726
005134' 000207

DELAY: CLR -(SP) ;INITIALIZE LOOP COUNT
40\$: DEC (SP) ;DECR LOOP CNT
BMI 40\$;DONE LOOP 32K TIMES? (Y,N-40\$)
TST (SP)+ ;CLEAR CNTR OFF STACK
RTS PC ;EXIT IN-LINE

.SBTTL RADIX 50 CONVERSION S/R'S FOR LOAD MEDIA HANDLERS

1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428

005136' 004467 177554
005142' 012501
005144' 012500
005146' 004567 000012
005152' 062700 000004
005156' 004567 000012
005162' 000456

005164' 012767 177776 000006
005172' 000403

005174' 012727 177777
005200' 000000
005202' 004467 177510
005206' 012704 177775
005212' 005002
005214' 112103
005216' 122703 000040
005222' 001410
005224' 122703 000044
005230' 001411
005232' 120327 000101
005236' 103404
005240' 162703 000040
005244' 162703 000016
005250' 162703 000011
005254' 162703 000011

```
*****
:
: PAKNAM: CONVERT 9 CHARACTER ASCII FILE NAME TO
: 3 RADIX 50 WORDS S/R
:
: LINK: JSR R5,PAKNAM S/R CALL
: .WORD ASCIADR ADR OF 9 ASCII CHAR'S
: .WORD R5OADR ADR OF 3 OUTPUT RAD50 WORDS
:
*****
:
: PAKNAM: JSR R4,SAVO4 ;SAVE WORK REG'S
: MOV (R5)+,R1 ;GET ADR OF ASCII DATA
: MOV (R5)+,R0 ;GET ADR OF RAD50 WORDS
: JSR R5,PACK ;PACK FIRST 6 ASCII INTO 2 RAD50 WORDS
: ADD #4,R0 ;INCR TC EXT STORAGE
: JSR R5,PACK1 ;PACK LAST 3 ASCII INTO 1 WORD
: BR EPKEX ;GO TO EXIT
:
*****
:
: PACK: DEPENDING UPON ENTRY POINT, EITHER CONVERTS
: PACK1: 6 OR 3 ASCII WORDS TO EITHER 2 OR 1
: RAD50 WORDS. REG R1 CONTAINS ADR OF ASCII
: DATA UPON ENTRY AND POINTS TO THE NEXT LOCATION
: AFTER THE ASCII DATA UPON EXIT. REG R0 CONTAINS
: ADR OF OUTPUT RAD50 WORDS UPON ENTRY.
:
: LINK: JSR R5,PACK 6/2 S/R CALL
: ,PACK1 3/1 S/R CALL
:
*****
:
: PACK: MOV #-2,EPKCNT ;SET OUTPUT WORD CNT TO 2
: BR EPKCOM ;GO TO COMMON POINT
:
: PACK1: MOV #-1,(PC)+ ;SET OUTPUT WORD CNT TO 1
: EPKCNT: .WORD 0 ;S/R LOOP COUNT
: EPKCOM: JSR R4,SAVO4 ;STORE WORK REG'S
: 10$: MOV #-3,R4 ;INITIALIZE INTERNAL LOOP CNT
: CLR R2 ;CLEAR SUM
: 20$: MOVB (R1)+,R3 ;GET NEXT ASCII CHAR
: CMPB #40,R3 ;CHAR A BLANK?
: BEQ 30$ ;N,Y-30$
: CMPB #'$,R3 ;A DOLLAR SIGN?
: BEQ 50$ ;N,Y-50$
: CMPB R3,#'A ;LESS THAN AN "A"?
: BLO 40$ ;N,Y-40$
: SUB #40,R3 ;ADJUST FOR A-Z CHAR
: 30$: SUB #16,R3 ;ADJUST FURTHER
: 40$: SUB #11,R3
: 50$: SUB #11,R3
```

| | | | | | | | |
|------|---------|--------|--------|-------------|-----------|--|---|
| 1429 | 005260' | 006302 | | ASL | R2 | | |
| 1430 | 005262' | 006302 | | ASL | R2 | | :MULTIPLY CURRENT CHAR TOTAL |
| 1431 | 005264' | 006302 | | ASL | R2 | | :BY DECIMAL 40 |
| 1432 | 005266' | 010246 | | MOV | R2, -(SP) | | |
| 1433 | 005270' | 006302 | | ASL | R2 | | |
| 1434 | 005272' | 006302 | | ASL | R2 | | |
| 1435 | 005274' | 062602 | | ADD | (SP)+, R2 | | |
| 1436 | 005276' | 060302 | | ADD | R3, R2 | | :INCLUDE CURR CHAR |
| 1437 | 005300' | 005204 | | INC | R4 | | :DONE 3 CHAR'S? |
| 1438 | 005302' | 002744 | | BLT | 20\$ | | :Y N-20\$ |
| 1439 | 005304' | 010220 | | MOV | R2, (R0)+ | | :STORE RAD50 WORD |
| 1440 | 005306' | 005267 | 177666 | INC | EPKCN | | :DONE # OF WORDS SPECIFIED? |
| 1441 | 005312' | 002735 | | BLT | 10\$ | | :Y N-10\$ |
| 1442 | 005314' | 010166 | 000002 | EPKPEX: MOV | R1, 2(SP) | | :REPLACE R1 ON STK WITH FINAL ASCII ADR |
| 1443 | 005320' | 004767 | 177404 | EPKEX: JSR | PC, RST04 | | :RESTORE REG'S |
| 1444 | 005324' | 000205 | | RTS | R5 | | :EXIT IN-LINE |

1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501

:UPKNAME: UNPACK 3 RAD50 WORDS TO 9 ASCII CHARACTERS
: WHICH ARE A FILENAME.
:LINK: JSR R5,UPKNAME S/R CALL
: .WORD ASCIADR ADR OF 9 ASCII CHAR'S OF OUTPUT
: .WORD R5OADR ADR OF 3 INPUT RAD50 WORDS
:*****

005326' 004467 177364
005332' 012501
005334' 012500
005336' 004567 000012
005342' 062700 000004
005346' 004567 000012
005352' 000762

UPKNAME: JSR R4,SAVO4 ;SAVE WORK REG'S
MOV (R5)+,R1 ;GET ASCII OUTPUT DATA ADR
MOV (R5)+,R0 ;GET INPUT RAD50 WORDS ADR
JSR R5,UNPACK ;GO UNPACK FIRST TWO WORDS
ADD #4,R0 ;ADJUST INPUT DATA ADR
JSR R5,UNPACK1 ;GO UNPACK LAST WORD
BR EPKEX ;GO TO EXIT

:UNPACK: DEPENDING UPON ENTRY POINT, EITHER CONVERTS
:UPACK1: 2 OR 1 RAD50 WORDS TO EITHER 6 OR 3
: ASCII CHARACTERS. REG R1 CONTAINS ADR OF ASCII
: OUTPUT DATA UPON ENTRY AND POINTS TO THE NEXT
: LOCATION AFTER THE ASCII DATA UPON EXIT. REG R0
: CONTAINS THE ADR OF THE INPUT RAD50 WORDS UPON ENTRY.
:LINK: JSR R5,UNPACK 2/6 S/R CALL
: ,UNPACK1 1/3 S/R CALL
:*****

005354' 012767 177776 177616
005362' 000403
005364' 012767 177777 177606
005372' 004467 177320
005376' 012704 177775
005402' 011000
005404' 012702 005504'
005410' 005003
005412' 020012
005414' 103403
005416' 161200
005420' 005203
005422' 000773
005424' 105703
005426' 001406
005430' 120327 000033
005434' 001407
005436' 003004
005440' 062703 000040
005444' 062703 000016
005450' 062703 000011

UNPACK: MOV #-2,EPKCNT ;SET INPUT WORD CNT TO 2
BR EUPKCM ;GO TO COMMON POINT
UNPACK1: MOV #-1,EPKCNT ;SET INPUT WORD CNT TO 1
EUPKCM: JSR R4,SAVO4 ;SAVE WORK REG'S
SS: MOV #-3,R4 ;INITIALIZE INTERNAL LOOP CNT
MOV (R0),R0 ;GET CURR RAD50 WORD
MOV #COEFF,R2 ;GET COEFFICIENT TBL ADR
10\$: CLR R3 ;CLEAR QUOTIENT REG
20\$: CMP R0,(R2) ;DONE WITH PSEUDO DIVIDE?
BLO 30\$;N,Y-30\$
SUB (R2),R0 ;SUBT CURR COEFF VALUE
INC R3 ;INCR QUOTIENT
BR 20\$;LOOP AGAIN
30\$: TSTB R3 ;QUOTIENT ZERO? (A BLANK)
BEQ 40\$;N,Y-40\$
CMPB R3,#33 ;QUOT A "S"?
BEQ 60\$;N,Y-60\$
BGT 50\$;"." OR "0-9"? (N,Y-50\$)
ADD #40,R3 ;MUST BE "A-Z": ADJUST ACCORDINGLY
40\$: ADD #16,R3 ;ADJUST VALUE FURTHER
50\$: ADD #11,R3

| | | | | | | | |
|------|---------|--------|--------|---------|-------|----------|-----------------------------|
| 1502 | 005454' | 062703 | 000011 | 60\$: | ADD | #11,R3 | |
| 1503 | 005460' | 110321 | | | MOVW | R3,(R1)+ | ;STORE OUTPUT ASCII CHAR |
| 1504 | 005462' | 005722 | | | TST | (R2)+ | ;POINT TO NXT COEFF |
| 1505 | 005464' | 005204 | | | INC | R4 | ;DONE 3 OUTPUT CHAR'S? |
| 1506 | 005466' | 002750 | | | BLT | 10\$ | ;Y,N-10\$ |
| 1507 | 005470' | 011600 | | | MOV | (SP),R0 | ;RESTORE ORG RAD50 WORD ADR |
| 1508 | 005472' | 005720 | | | TST | (R0)+ | ;POINT IT AT NEXT WORD |
| 1509 | 005474' | 005267 | 177500 | | INC | EPKCNT | ;DONE # OF WORDS SPECIFIED? |
| 1510 | 005500' | 002736 | | | BLT | 5\$ | ;Y,N-5\$ |
| 1511 | 005502' | 000704 | | | BR | EPKPEX | ;GO TO EXIT |
| 1512 | | | | | | | |
| 1513 | 005504' | 003100 | | ECOEFF: | .WORD | 1600. | |
| 1514 | 005506' | 000050 | | | .WORD | 40. | |
| 1515 | 005510' | 000001 | | | .WORD | : | |

.SBTTL TYPE S/R'S REQUIRED FOR LOAD MEDIA HANDLERS

1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572

```

*****
CHROUT: TYPE THE ASCII CHAR IN REG R2 S/R
LINK: JSR PC,CHROUT S/R CALL
*****
CHROUT: MOVB R2,ETYPWK ;STORE CHAR AT COMMON LOC
ECOMTY: JSR R4,SAVO4 ;SAVE WORK REGISTERS
JSR R5,CLIST ;GO DO THE WRITE OF ONE CHAR
.WORD ETYPWK-.
.WORD 1
BR ETYEX ;GO TO EXIT
ETYPWK: .BLKW 5 ;TYPE WORK/CHAR AREA
*****
CRLF: THESE TWO S/R'S ARE USED TO TYPE A CARRIAGE
TAB: RETURN/LINE FEED AND TAB, RESPECTIVELY
LINK: JSR PC,CRLF S/R CALL
LINK: JSR PC,TAB S/R CALL
*****
CRLF: JSR R4,SAVO4 ;SAVE REGISTERS
JSR R5,CLIST ;ISSUE CR/LF CHARACTERS
.WORD EXERM-.
.WORD 2
BR ETYEX ;GO TO EXIT
TAB: MOVB #011,ETYPWK ;SET UP TAB CHAR
BR ECOMTY ;GO TYPE IT
*****
$STPNM: $STPNM WILL TYPE A CR/LF FOLLOWED BY THE NINE
$STPNM1: CHARACTER FILENAME STARTING AT THE ADDRESS STORED AT
LOCATION "$STXNAM". IN THE DISPLAYED DATA A PERIOD WILL BE
INSERTED BETWEEN THE FILENAME AND ITS EXTENSION.
$STPNM1 DOES THE SAME FUNCTION BUT WITHOUT THE
LEADING CR/LF.
LINK: JSR PC,$STPNAM S/R CALL
,$STPNM1
*****
$STPNM: JSR PC,CRLF ;ISSUE CR/LF
$STPNM1: JSR R4,SAVO4 ;SAVE WORK REGISTERS
MOV $STXNAM,R0 ;GET FILENAME ADR

```

```

005512' 110267 000016
005516' 004467 177174
005522' 004567 001512
005526' 000006
005530' 000001
005532' 000536
005534' 000005
005546' 004467 177144
005552' 004567 001462
005556' 001026
005560' 000002
005562' 000522
005564' 112767 000011 177742
005572' 000751
005574' 004767 177746
005600' 004467 177112
005604' 016700 172700

```

```

1573 005610' 012701 005534'      MOV      #ETYPWK,R1      ;GET CHAR WORK AREA ADR
1574 005614' 012702 001406      MOV      #1406,R2      ;SET UP TWO LOOP COUNTS
1575 005620' 112021      10$:    MOVB   (R0)+,(R1)+   ;MOVE CHAR OF FILENAME
1576 005622' 105302      DECB   R2              ;DONE 6 CHAR FILENAME?
1577 005624' 001375      BNE    10$            ;Y,N-10$
1578 005626' 112721 000056      MOVB   #'',(R1)+     ;PUT PERIOD IN TYPE DATA
1579 005632' 000302      SWAB   R2              ;GET SECOND LOOP COUNT
1580 005634' 112021      20$:    MOVB   (R0)+,(R1)+   ;MOVE CHAR OF EXT
1581 005636' 105302      DECB   R2              ;DONE 3 CHAR EXT?
1582 005640' 001375      BNE    20$            ;Y,N-20$
1583 005642' 004567 001372      JSR    R5,CLIST       ;TYPE THE FILENAME & EXT
1584 005646' 177666      .WORD  ETYPWK-        ;
1585 005650' 000012      .WORD  10             ;
1586 005652' 000466      BR     ETYEX          ;GO TO S/P EXIT
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599

```

```

MES:  THIS S/R WILL TYPE THE ASCII DATA STARTING AT THE
      SUPPLIED ADDRESS AND UP TO BUT NOT INCLUDING
      THE FIRST 00 CHARACTER.

```

```

LINK:  JSR    R5,MES      S/R CALL
       .WORD  DATADR     DATA ADR (ABSOLUTE)

```

```

1600 005654' 004467 177036      MES:    JSR    R4,SAV04  ;SAVE WORK REG'S
1601 005660' 012500      MOV     (R5)+,R0      ;GET DATA ADR
1602 005662' 010001      MOV     R0,R1        ;COMPUTE CHAR CNT
1603 005664' 105721      40$:    TSTB   (R1)+        ;
1604 005666' 001376      BNE    40$           ;
1605 005670' 160001      SUB    R0,R1         ;
1606 005672' 005301      DEC    R1            ;
1607 005674' 005701      TST    R1            ;CHAR CNT = 0?
1608 005676' 001411      BEQ    EMESEX        ;N,Y-EMESX
1609 005700' 012702 005716'      MOV     #44$,R2      ;GET S/R INFO STORAGE ADR
1610 005704' 160200      SUB    R2,R0         ;MAKE DATA ADR RELATIVE
1611 005706' 010022      MOV     R0,(R2)+     ;STORE DATA ADR
1612 005710' 010112      MOV     R1,(R2)     ;STORE CHAR CNT
1613 005712' 004567 001322      JSR    R5,CLIST       ;TYPE THE MSG
1614 005716' 000000      44$:    .WORD  XXXX
1615 005720' 000000      .WORD  XXXX
1616 005722' 004767 177002      EMESEX: JSR    PC,RST04  ;RESTORE WORK REG'S
1617 005726' 000205      RTS     R5           ;EXIT IN-LINE

```

1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632 005730' 004467 176762
1633 005734' 010300
1634 005736' 004577 000010'
1635 005742' 000034
1636 005744' 012500
1637 005746' 012701 000006
1638 005752' 160001
1639 005754' 012703 005776'
1640 005760' 060103
1641 005762' 112302
1642 005764' 004767 177522
1643 005770' 005300
1644 005772' 001373
1645 005774' 000752
1646
1647 005776' 000003
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659 006004' 004467 176706
1660 006010' 010300
1661 006012' 004577 000006'
1662 006016' 177760
1663 006020' 004567 001214
1664 006024' 177752
1665 006026' 000006
1666 006030' 004767 176674
1667 006034' 000207

```

*****
BCDCV: CONVERTS THE BINARY NUMBER IN R3 TO THE NUMBER OF
        DECIMAL ASCII DIGITS SPECIFIED BY THE COUNT WHICH FOLLOWS
        THE JSR. ALSO TYPE THESE DIGITS.

LINK:   JSR    R5,BCDCV      S/R CALL
        .WORD  DIGITCNT     # OF DIGITS TO PRINT

        R3 CONTAINS BINARY # TO CONVERT
    
```

```

*****
BCDCV: JSR    R4,SAV04      ;SAVE WORK REGISTERS
        MOV    R3,R0        ;SET WORD UP FOR S/R
        JSR    R5,@BTASLB   ;CONVERT # TO DECIMAL ASCII
        .WORD  EBCDWA-
        MOV    (R5)+,R0     ;GET # OF DIGITS WANTED
        MOV    #6,R1        ;SET UP MAX # OF DIGITS
        SUB    R0,R1        ;GET THE DIFFERENCE
        MOV    #EBCDWA,R3   ;GET ASCII DATA ADR
        ADD    R1,R3        ;INDEX UP TO DESIRED DIGITS
10$:   MOVB   (R3)+,R2      ;GET DIGIT TO TYPE
        JSR    PC,CHROUT    ;TYPE IT
        DEC    R0           ;DONE ALL DIGITS?
        BNE   10$          ;Y N-10$
        BR    EMESEX       ;GO TO EXIT

EBCDWA: .BLKW  3
    
```

```

*****
ITOA:  CONVERT BINARY WORD IN REG R3 TO SIX ASCII DIGITS
        AND TYPE THEM.

LINK:  JSR    PC,ITOA      S/R CALL
    
```

```

*****
ITOA:  JSR    R4,SAV04      ;SAVE WORK REG'S
        MOV    R3,R0        ;SET WORD UP FOR S/R
        JSR    R5,@BINASC   ;CONVERT IT TO ASCII
        .WORD  EBCDWA-
        JSR    R5,CLIST     ;TYPE THE 6 DIGITS
        .WORD  EBCDWA-
        .WORD  6
ETYEX: JSR    PC,RST04     ;RESTORE WORK REG'S
        RTS    PC          ;EXIT IN-LINE
    
```

1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724

006036' 004467 176654
006042' 012704 000105
006046' 042703 100000
006052' 005204
006054' 162703 001750
006060' 003374
006062' 062703 001750
006066' 005046
006070' 032704 000003
006074' 001005
006076' 020327 000073
006102' 101402
006104' 005216
006106' 005303
006110' 012700 006217'
006114' 111001
006116' 020301
006120' 101404
006122' 160103
006124' 062700 000004
006130' 000771
006132' 005726
006134' 001404
006136' 020027 006223'
006142' 001001
006144' 005203
006146' 004567 177556
006152' 000002
006154' 012703 006212'
006160' 114043
006162' 114043
006164' 114043
006166' 004567 177462
006172' 006206'
006174' 010403
006176' 004567 177526
006202' 000002
006204' 000711
006206' 054055 054130 000055
006214' 040512 017516
006220' 042506 016102
006224' 040515 017522

: DATUPK:
: DATE UNPACK AND TYPE S/R
: LINK: JSR PC,DATUPK S/R CALL
: REG R3 CONTAINS PACKED DATE
:*****

DATUPK: JSR R4, SAV04
MOV #69, R4
BIC #100000, R3
10\$: INC R4
SUB #1000., R3
BGT 10\$
ADD #1000., R3
CLR -(SP)
BIT #3, R4
BNE 20\$
CMP R3, #59.
BLOS 20\$
INC (SP)
DEC R3
20\$: MOV #EMONTB+3, R0
30\$: MOVB (R0), R1
CMP R3, R1
BLOS 40\$
SUB R1, R3
ADD #4, R0
BR 30\$
40\$: TST (SP)+
BEQ 50\$
CMP R0, #EMONTB+7
BNE 50\$
INC R3
50\$: JSR R5, BCDCV
.WORD 2
MOV #EMONMG+4, R3
MOVB -(R0), -(R3)
MOVB -(R0), -(R3)
MOVB -(R0), -(R3)
JSR R5, MES
.WORD EMONMG
MOV R4, R3
JSR R5, BCDCV
.WORD 2
BR ETYEX

: SAVE WORK REGISTERS
: INITIALIZE YEAR TO BASE -1
: RESET CONTIG BIT
: ADD 1 TO YEAR VALUE
: DECR DATE VALUE BY YEAR INCR
: YEAR FOUND? (Y,N-10\$)
: RESTORE DAY & MONTH
: CLEAR LEAP YEAR FLAG
: THIS A LEAP YEAR?
: Y,N-30\$
: IS DAY LATER THAN FEB 28?
: Y,N-20\$
: SET LEAP YEAR FLAG
: ADJUST FOR FEB 29
: SET UP MONTH TABLE ADR
: GET # OF DAYS IN CURR MONTH
: DAY VALUE =< CURR MONTH?
: N,Y-40\$
: DECR DAY VALUE BY CURR MONTH
: POINT TO NEXT MONTH
: GO PROCESS NXT MONTH
: LEAP YEAR FLAG SET?
: Y,N-50\$
: IS MONTH = FEB?
: Y,N-50\$
: INCR DAY FOR FEB 29
: CONVERT & PRINT 2 DAY DIGITS
: SET UP MSG ADR
: MOV ASCII MONTH NAME TO MSG
: PRINT MONTH IN -XXX- FORMAT
: SET YEAR VALUE IN REG FOR S/R
: CONVERT & PRINT YEAR VALUE
: GO TO EXIT

EMONMG: .ASCIZ /-XXX-/
EMONTB: .ASCII /JAN/<31.>
.ASCII /FEB/<28.>
.ASCII /MAR/<31.>

| | | | | | |
|------|---------|--------|--------|--------|------------|
| 1725 | 006230' | 050101 | 017122 | .ASCII | /APR/<30.> |
| 1726 | 006234' | 040515 | 017531 | .ASCII | /MAY/<31.> |
| 1727 | 006240' | 052512 | 017116 | .ASCII | /JUN/<30.> |
| 1728 | 006244' | 052512 | 017514 | .ASCII | /JUL/<31.> |
| 1729 | 006250' | 052501 | 017507 | .ASCII | /AUG/<31.> |
| 1730 | 006254' | 042523 | 017120 | .ASCII | /SEP/<30.> |
| 1731 | 006260' | 041517 | 017524 | .ASCII | /OCT/<31.> |
| 1732 | 006264' | 047516 | 017126 | .ASCII | /NOV/<30.> |
| 1733 | 006270' | 042504 | 017503 | .ASCII | /DEC/<31.> |
| 1734 | | | | | |

.SBTTL ERROR RETURN POINTS REQUIRED FOR LOAD MEDIA HANDLERS

1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776

006274' 004567 000162
006300' 042504 020114 051105
006306' 000122
006310' 004567 000146
006314' 042504 020114 046117
006322' 000104
006324' 004567 000132
006330' 042504 020126 051105
006336' 000122
006340' 004567 000116
006344' 042504 020126 052506
006352' 046114 000
006356' 004567 176362
006362' 000524'
006364' 002256'
006366' 000011'
006370' 004767 177200
006374' 004567 000062
006400' 044506 042514 047040
006406' 052117 043040 042116
006414' 000
006416' 004567 000040
006422' 047111 020126 046503
006430' 042116 043040 051117
006436' 042040 053105 000
006444' 004567 000012
006450' 045503 052523 020115
006456' 051105 000122

THE FOLLOWING ARE RETURN POINTS FOR SYSTEM DEVICES
WHEN THEY ENCOUNTER ERRORS. THE CURRENT DEVICE
FUNCTION WILL BE ABORTED SINCE A RETURN ADR IS
NOT SUPPLIED.

LINK: JMP LABEL WHERE LABEL IS ONE OF
THE FOLLOWING:

DELERR: JSR R5,ERRCOM ;FILE DELETE ERROR
.ASCIZ /DEL ERR/
DELFIL: .EVEN ;FILE NEEDS TO BE DELETED ERROR
JSR R5,ERRCOM
.ASCIZ /DEL OLD/
DEVERR: .EVEN ;DEVICE ERROR
JSR R5,ERRCOM
.ASCIZ /DEV ERR/
DEVFUL: .EVEN ;DEVICE IS FULL ERROR
JSR R5,ERRCOM
.ASCIZ /DEV FULL/
FLNOTF: .EVEN ;MOVE FILENAME.EXT TO S/R AREA
JSR R5,BMOVE
.WORD TXNAM
NFNAME: .WORD IFNAM
.WORD 9
JSR PC,\$TYPNM ;TYPE NAME OF FILE NOT FOUND
JSR R5,ERRCOM ;FILE NOT FOUND ERROR
.ASCIZ /FILE NOT FND/
INVCMD: .EVEN ;INVALID CMND FOR DEV
JSR R5,ERRCOM
.ASCIZ /INV CMND FOR DEV/
CKSMER: .EVEN ;CHECK SUM ERROR
JSR R5,ERRCOM
.ASCIZ /CKSUM ERR/
.EVEN

```

1778                                     ;ERROR COMMON PROCESSING
1779
1780 006462' 004767 176436      ERRCOM: JSR    PC,DELAY      ;DELAY A LITTLE BIT
1781 006466' 004767 000040      JSR    PC,EXERPT    ;GO REPORT THE ERROR
1782 006472' 005727              TST    (PC)+        ;THIS AN ERROR ON A FORCED CLOSE?
1783 006474' 000000      ECFORC: .WORD    0
1784 006476' 001011              BNE    10$          ;N.Y-10$
1785 006500' 005767 173566      TST    0WCTR        ;HAS A FILE BEEN LEFT OPEN?
1786 006504' 001410              BEQ    20$          ;Y.N-20$
1787 006506' 005267 177762      INC    ECFORC       ;SET "FORCING A CLOSE" FLAG
1788 006512' 012705 002354      MOV    #OUTDEV,RS   ;SET UP OUTPUT DOB ADR
1789 006516' 004767 176162      JSR    PC,FCLOSE    ;GO CLOSE THE FILE
1790 006522' 005067 177746      10$: CLR    ECFORC   ;RESET FORCED CLOSE FLAG
1791 006526' 000167 174340      20$: JMP    EXERET   ;GO TO ERROR EXIT
1792
1793
1794                                     ;ISSUE STANDARD ERROR MSG S/R
1795
1796                                     ;JSR    PC,EXERPT    S/R CALL
1797                                     ;RS MUST CONTAIN ABS ADR OF ERROR MSG
1798
1799 006532' 010567 000036      EXERPT: MOV    RS,10$      ;STORE MSG ADR
1800 006536' 105067 171754      CLR    PRIFLG       ;RESET PRINTER/TTY FLAG
1801 006542' 042777 010000 000012' BIC    #CNTRLO,DCSYSFW ;RESET TYPING SUPPRESS FLAG
1802 006550' 004567 176170      JSR    RS,BMOVE     ;SET FUNCTIONAL DEV NAME IN MSG
1803 006554' 006613'          .WORD    ERMFDN
1804 006556' 006642'          .WORD    EFDEVN
1805 006560' 000004          .WORD    4
1806 006562' 004567 177066      JSR    RS,MES       ;ISSUE STANDARD PORTION OF
1807 006566' 006604'          .WORD    EXERM       ;ERR MSG
1808 006570' 004567 177060      JSR    RS,MES       ;ISSUE ERROR I.D.
1809 006574' 000000      10$: .WORD    XXXX
1810 006576' 004767 176744      JSR    PC,CRLF      ;ISSUE CR/LF
1811 006602' 000207          RTS    PC           ;EXIT IN-LINE
1812
1813 006604' 005015 042452 025122 EXERM: .ASCII <015><012> /*ER* /
1814 006612' 040
1815 006613' 130 054130 020130 ERMFDN: .ASCIZ /XXXX DEV: /
1816 006620' 042504 035126 000040
1817 006626' 047105 020104 043117 EOFMSG: .ASCIZ /END OF FILE/
1818 006634' 043040 046111 000105
1819                                     .EVEN
1820 006642' 047514 042101      EFDEVN: .ASCII /LOAD/      ;CURRENT DEVICE'S FUNCTIONAL NAME

```

.SBTTL CONSOLE TERMINAL CONTROL ROUTINES

1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875

THESE ROUTINES CONTROL THE ACCESS TO THE CONSOLE TERMINAL (READ AND WRITE) FOR BOTH MPG AND USER PROGRAM I/O FUNCTIONS. THESE ROUTINES SET UP THE NECESSARY INFORMATION REQUIRED BY THE INTERRUPT ROUTINES AND IN THE CASE OF THE WRITE, THE I/O TRANSFER IS ALSO INITIATED.

IF THE CONSOLE TERMINAL IS ALREADY BUSY WHEN ENTERED FOR AN MPG FUNCTION, THESE ROUTINES WILL AUTOMATICALLY WAIT UNTIL THE TERMINAL BECOMES FREE. THE I/O WILL THEN BE PERFORMED WITH RETURN BEING MADE IN-LINE ONLY AFTER THE ENTIRE I/O FUNCTION HAS BEEN COMPLETED.

FOR USER PROGRAM WRITE FUNCTIONS, RETURN WILL BE MADE TO THE SUPPLIED BUSY ADDRESS IF THE TERMINAL IS BUSY UPON ENTRY. NO OTHER ACTIONS WILL TAKE PLACE. IF THE TERMINAL IS FREE, THE USER I/O WILL BE INITIATED AND THEN THE 'WAIT FOR I/O TERM' FLAG WILL BE SET IN THE USER PROGRAM'S FLAGWORD AT THE ADDRESS SUPPLIED. RETURN WILL THEN BE MADE IN-LINE.

| | | | |
|--------------|-------|---------|-----------------------------|
| MPG LINKAGE: | JSR | RS,CTWR | MPG WRITE |
| | | CTRD | MPG READ WITH ECHO |
| | | CTRONE | MPG READ WITHOUT ECHO |
| | .WORD | TAG- | RELATIVE ADR OF I/O AREA |
| | .WORD | COUNT | # OF CHARACTERS TO TRANSFER |

| | | | |
|---------------|-------|-----------|------------------------------|
| USER LINKAGE: | JSR | RS,CTUSWR | USER WRITE |
| | .WORD | TAG- | RELATIVE ADR OF I/O AREA |
| | .WORD | COUNT | # OF CHARS TO XFER |
| | .WORD | BADR- | RELATIVE RETURN ADR IF BUSY |
| | .WORD | FWAOR | ABSOLUTE ADR OF PROG'S FLGWD |

DESTROYS REG'S: RC,R1,R2

RETURN WILL BE IN-LINE FOLLOWING THE ARGUMENTS UNLESS THE CONSOLE TERMINAL IS BUSY AND A USER I/O IS BEING ATTEMPTED.

;USER WRITE ENTRY POINT

| | | | | | | | |
|------|---------|--------|---------|---------|-------------|-------------------------|-------------------------------------|
| 1866 | 006646' | 032777 | 010000 | 000012' | CTUSWR: BIT | #CNTRLO, @CSYSFW | ; IS TYPING SUPPRESSED? |
| 1867 | 006654' | 001421 | | | BEQ | 30\$ | ; Y, N-30\$ |
| 1868 | 006656' | 032777 | 040000 | 000012' | BIT | #CRUDED, @CSYSFW | ; SUPPOSED TO RESET USER DEDICATED? |
| 1869 | 006664' | 001412 | | | BEQ | 20\$ | ; Y, N-20\$ |
| 1870 | 006666' | 042777 | 060000 | 000012' | BIC | #CRUDED+CTUDED, @CSYSFW | ; RESET SYSTEM DEDICATED FLGS |
| 1871 | 006674' | 016567 | 000006 | 002554 | MOV | 6(R5), UFWHLD | ; GET FLAGWORD ADR |
| 1872 | 006702' | 004577 | 000002' | | JSR | RS, @UFWCLR | ; RESET USER'S DEDICATED FLAG |
| 1873 | 006706' | 000040 | | | .WORD | SETDED | |
| 1874 | 006710' | 002546 | | | .WORD | UFWHLD- | |
| 1875 | 006712' | 062705 | 000010 | POS: | ADD | #8., R5 | ; SET UP RETURN ADR |

```

1876 006716' 000205          RTS      RS      ;EXIT IN-LINE
1877 006720' 032767 002042 002130 30$: BIT      #CTWBSY+CTEBSY+CTWAIT,CTFLGW ;WR BUSY OR ECHO BUSY SET?
1878 006726' 001404          BEQ      CTDUWR ;(Y,N-CTDUWR)
1879 006730' 062705 000004          CTBRET: ADD     #4,RS ;INCR RET ADR PAST ARGUMENTS
1880 006734' 061505          ADD     (RS),RS ;SET UP USER BUSY RET ADR
1881 006736' 000205          RTS      RS      ;EXIT TO USER'S BUSY ADR
1882 006740' 052767 100002 002110 CTDUWR: BIS     #CTWUSR+CTWBSY,CTFLGW ;SET USER WR & WR BUSY FLAGS
1883 006746' 000421          BR       CTWCOM ;GO TO COMMON PROCESSING
1884
1885 ;EXEC WRITE ENTRY POINT
1886
1887 006750' 032777 010000 000012' CTWR: BIT      #CNTRLO,@CSYSFW ;IS TYPING SUPPRESSED?
1888 006756' 001403          BEQ      40$ ;Y,N-40$
1889 006760' 062705 000004          ADD     #4,RS ;SET UP RETURN ADR
1890 006764' 000205          RTS      RS      ;EXIT IN-LINE
1891 006766' 032767 002042 002062 40$: BIT      #CTWBSY+CTEBSY+CTWAIT,CTFLGW ;WR BUSY OR ECHO BUSY FLAGS SET?
1892 006774' 001365          BNE     CTWR ;(N,Y-CTWR)
1893 006776' 042767 100000 002052          BIC     #CTWUSR,CTFLGW ;RESET USER WRITE FLAG
1894 007004' 052767 000002 002044          BIS     #CTWBSY,CTFLGW ;SET THE WRITE BUSY FLAG
1895 007012' 010500          CTWCOM: MOV     RS,RO ;GET DATA ADR INDEX VALUE AND
1896 007014' 062500          ADD     (RS)+,RO ;ADJUST TO ABSOLUTE ADR
1897 007016' 010067 002040          MOV     RO,CTWADR ;STORE DATA ADR FOR INT ROUTINE
1898 007022' 012567 002032          MOV     (RS)+,CTWCNT ;GET & STORE CHAR CNT
1899 007026' 005767 002024          TST     CTFLGW ;DOING A USER WRITE?
1900 007032' 100007          BPL     2$ ;(Y,N-2$)
1901 007034' 005725          TST     (RS)+ ;INCR PAST BUSY ADR
1902 007036' 012567 002022          MOV     (RS)+,CTWUFW ;MOVE USER FLGWD ADR TO INT RCT
1903 007042' 004577 000000'          JSR     RS,@UFWSET ;SET USER'S WAIT FOR
1904 007046' 000020          .WORD  CTRDNE ;CONSOLE/PRINTER I/O FLAG
1905 007050' 002014          .WORD  CTWUFW-
1906 007052' 052777 000100 002066 2$: BIS     #100,@CTWDRA ;SET WRITE INT ENABLE BIT
1907 007060' 005767 001772          TST     CTFLGW ;IS THIS A USER WRITE?
1908 007064' 100406          BMI     4$ ;(N,Y-4$)
1909 007066' 004767 171306 3$: JSR     PC,LIGHTS ;DISPLAY IDLE BIT PATTERN
1910 007072' 032767 000002 001756          BIT     #CTWBSY,CTFLGW ;IS 'WR BUSY' FLAG STILL SET?
1911 007100' 001372          BNE     3$ ;(N,Y-3$)
1912 007102' 000205 4$: RTS      RS      ;EXIT IN-LINE
1913
1914
1915
1916 ;EXEC READ WITHOUT ECHO ENTRY POINT
1917
1918 007104' 032767 000003 001744 CTRDNE: BIT     #CTRBSY+CTWBSY,CTFLGW ;READ BUSY OR WRITE BUSY?
1919 007112' 001374          BNE     CTRDNE ;(N,Y-CTRONE)
1920 007114' 042767 000200 001734          BIC     #CTRDEC,CTFLGW ;RESET THE ECHO FLAG
1921 007122' 000407          BR       CTRCOM ;GO TO COMMON PROCESSING
1922
1923 ;EXEC READ WITH ECHO ENTRY POINT
1924
1925 007124' 032767 000003 001724 CTRD: BIT     #CTRBSY+CTWBSY,CTFLGW ;READ BUSY OR WRITE BUSY?
1926 007132' 001374          BNE     CTRD ;(N,Y-CTRD)
1927          001          .IF DF MIMIC
1928          BR       CTRDNE ;DO NOT ECHO IF MIMIC
1929          000          .ENDC
1930 007134' 052767 000200 001714          BIS     #CTRDEC,CTFLGW ;SET THE ECHO FLAG
1931 007142' 042767 000040 001706 CTRCOM: BIC     #CTEBSY,CTFLGW ;RESET THE ECHO BUSY FLAG

```

```

1932 007150' 010500      MOV      R5,R0      ;GET DATA ADR INDEX VALUE AND
1933 007152' 062500      ADD      (R5)+,R0   ;ADJUST TO AN ABSOLUTE ADR
1934 007154' 012501      MOV      (R5)+,R1   ;GET CHARACTER COUNT
1935 007156' 012702 011066' MOV      #CTRCNT,R2 ;SET UP ADR OF RD STORAGE AREA
1936 007162' 010122      MOV      R1,(R2)+   ;STORE COUNT AND DATA ADR
1937 007164' 010022      MOV      R0,(R2)+   ;IN THE INT ROUT
1938 007166' 010122      MOV      R1,(R2)+
1939 007170' 010022      MOV      R0,(R2)+
1940 007172' 105767 001660 TSTB     CTFLGW     ;ECHO FLAG SET?
1941 007176' 100003      BPL      12$        ;(Y,N-12$)
1942 007200' 052767 000040 001650 BIS      #CTEBSY,CTFLGW ;SET THE ECHO BUSY FLAG
1943 007206' 052767 000001 001642 12$: BIS      #CTRBSY,CTFLGW ;SET THE READ BUSY FLAG
1944 007214' 012777 000100 001716 MOV      #100,CTRDR ;SET TTY RD INT ENABLE
1945          001      .IF DF MIMIC
1946          .IFT
1947          CTRMLP: MOV      #12,R0      ;*** MIMIC ***
1948          .IFTF
1949 007222' 004767 171152 CTRTRB: JSR      PC,LIGHTS      ;DISPLAY IDLE BIT PATTERN
1950 007226' 032767 000001 001622 BIT      #CTRBSY,CTFLGW      ;IS READ BUSY STILL SET?
1951          .IFF
1952 007234' 001372      BNE      CTRTRB      ;(N,Y-CTRTRB)
1953          .IFT
1954          BNE      CTRMCD      ;*** MIMIC ***
1955          .IFTF
1956 007236' 000205      RTS      R5          ;EXIT IN-LINE
1957          .IFT
1958          CTRMCD: DEC      R0          ;*** MIMIC ***
1959          BNE      CTRTRB      ;*** MIMIC ***
1960          SLPBKP: NOP          ;*** MIMIC ***
1961          BR      CTRMLP      ;*** MIMIC ***
1962          .ENDC
1963          000

```

.SBTTL LIST DEVICE CONTROL ROUTINE

1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986

```

*****
THIS ROUTINE DETERMINES IF A LIST DEVICE HAS BEEN ASSIGNED
AND THEN BRANCHES TO EITHER THE CONSOLE TERMINAL ROUTINE
OR TO THE PRINTER ROUTINE. IF THE PRINTER/TTY FLAG IS RESET
IT WILL ALWAYS GO TO THE CONSOLE TERMINAL.

LINKAGE:      JSR      RS CLIST      ROUTINE CALL
               .WORD   MSGADR-     REL ADR OF MSG DATA
               .WORD   COUNT       MSG BYTE COUNT

EXITS TO EITHER CTWR OR CPRT

*****
CLIST:  TSTB      PRIFLG      ;PRINTER/TTY FLAG SET?
        BEQ       CTWR       ;Y,N-CTWR
        TST       CLISTI+4   ;A PRINTER BEEN ASSIGNED?
        BEQ       CTWR       ;Y,N-CTWR
        BR        CPRT       ;GO TO PRINTER ROUTINE

```

```

007240' 105767 171252
007244' 001641
007246' 005767 002170
007252' 001636
007254' 000400

```

.SBTTL PRINTER CONTROL ROUTINE

1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043

THIS ROUTINE CONTROLS WRITE OPERATIONS TO THE MPG LIST DEVICE FOR BOTH MPG AND USER FUNCTIONS. IF AN MPG FUNCTION, THIS ROUTINE WILL WAIT FOR THE PRINTER IF IT IS ALREADY BUSY UPON ENTRY AND WILL WAIT UNTIL THE PRINTER COMPLETES THE SPECIFIED OPERATION BEFORE RETURNING.

FOR USER FUNCTIONS, RETURN WILL BE MADE TO THE SUPPLIED BUSY ADDRESS IF THE PRINTER IS BUSY UPON ENTRY. NO OTHER ACTIONS WILL TAKE PLACE. IF THE PRINTER IS FREE, THE USER'S WRITE WILL BE INITIATED AND THEN THE 'WAIT FOR I/O TERM' FLAG WILL BE SET IN THE USER PROGRAM'S FLAGWORD AT THE ADDRESS SUPPLIED. RETURN WILL THEN BE IMMEDIATELY MADE IN-LINE.

MPG LINKAGE: JSR R5, CPRT MPG PRINT
.WORD TAG- REL ADR OF I/O AREA
.WORD COUNT # OF CHARS TO TRANSFER

USER LINKAGE: JSR R5, CUSPRT USER PRINT
.WORD TAG- REL ADR OF I/O AREA
.WORD COUNT # OF CHARS TO XFER
.WORD BADR- REL RETURN ADR IF BUSY
.WORD FWADR ABSOLUTE ADR OF PROG'S FLGWD

DESTROYS REGISTERS: R0, R1, R2

;MPG ENTRY POINT

007256' 032777 000100 002162 CPRT: BIT #100, @PRTDRA ; IS THE PRINTER BUSY?
007264' 001404 BEQ 15 ; Y, N-15
007266' 032777 004000 000012' BIT #CHPINT, @CSYSFW ; INT ENABLE BEING HELD SET?
007274' 001770 BEQ CPRT ; Y, N-CPRT
007276' 042767 100000 002122 15: BIC #CPRUSR, CPRFWD ; RESET USER PRINT FLAG
007304' 000432 BR CPRCOM ; GO TO COMMON PROCESSING

;USER ENTRY POINT

007306' 032777 000100 002132 CUSPRT: BIT #100, @PRTDRA ; PRINTER BUSY?
007314' 001423 BEQ 55 ; Y, N-55
007316' 032777 004000 000012' BIT #CHPINT, @CSYSFW ; INT ENABLE BEING HELD SET?
007324' 001415 BEQ 35 ; Y, N-35
007326' 016567 000006 002122 MOV 6(R5), UFWHLD ; GET FLAGWORD ADR
007334' 004577 000004' JSR R5, @UFWTST ; BEING HELD SET FOR THIS USER
007340' 000040 .WORD SETDED
007342' 002114 .WORD UFWHLD-
007344' 000405 BR 35 ; Y, N-35
007346' 004577 000004' JSR R5, @UFWTST ; USER'S PREV PRINT COMPLETED
007352' 000020 .WORD CTPRIO
007354' 002102 .WORD UFWHLD-
007356' 000402 BR 55 ; N, Y-55

```

2044 007360' 000167 177344 3$: JMP CTBRET ;GO TO BUSY RETURN
2045 007364' 052767 100000 002034 5$: BIS #CPRUSR,CPRFWD ;SET USER PRINT FLAG
2046 007372' 012702 011426' CPRCOM: MOV #CPRFWD,R2 ;SET UP FLAGWORD ADR
2047 007376' 105777 002044 10$: TSTB @PRTDRA ;PRINTER READY SET?
2048 007402' 100003 BPL 15$ ;Y,N-15$
2049 007404' 005777 002036 TST @PRTDRA ;PRINTER ERROR?
2050 007410' 100021 BPL 30$ ;Y,N-30$
2051 007412' 105712 15$: TSTB (R2) ;PRINTER ERROR FLAG SET?
2052 007414' 100413 BMI 20$ ;N,Y-20$
2053 007415' 052712 000200 BIS #CPRERR,(R2) ;SET ERROR FLAG
2054 007422' 010246 MOV R2,-(SP) ;SAVE FLGWD ADR
2055 007424' 042777 010000 000012' BIC #CNTRLO,@CSYSFW ;RESET SUPPRESS TYPING FLAG
2056 007432' 004567 177312 JSR R5,CTWR ;ISSUE PRINTER ERROR MSG
2057 007436' 000142 .WORD CPRTER-.
2058 007440' 000026 .WORD 22.
2059 007442' 012602 MOV (SP)+,R2 ;RESTORE FLGWD ADR
2060 007444' 005712 20$: TST (R2) ;THIS A USER PRINT?
2061 007446' 100353 BPL 10$ ;Y,N-10$
2062 007450' 000167 177254 JMP CTBRET ;GO TAKE BUSY RETURN
2063 007454' 112712 000001 30$: MOVB #1,(R2) ;RESET PRT ERR & SET PRT BSY FLGS
2064 007460' 012700 011430' MOV #CPRADR,R0 ;SET UP ADR OF STORAGE AREA
2065 007464' 010510 MOV R5,(R0) ;GET DATA ADR & MAKE IT
2066 007466' 062520 ADD (R5)+,(R0)+ ;ABSOLUTE
2067 007470' 012520 MOV (R5)+,(R0)+ ;STORE CHAR CNT
2068 007472' 005712 TST (R2) ;USER PRINT?
2069 007474' 100015 BPL 40$ ;Y,N-40$
2070 007476' 005725 TST (R5)+ ;INCR PAST BUSY RET ADR
2071 007500' 012510 MOV (R5)+,(R0) ;STORE USER'S FLGWD ADR
2072 007502' 004577 000000' JSR R5,@UFWSET ;SET USER'S WAIT FOR
2073 007506' 000020 .WORD CTPRIO ;CONSOLE/PRINTER I/O FLAG
2074 007510' 001724 .WORD CPRUFW-.
2075 007512' 032777 020000 000012' BIT #CTUDED,@CSYSFW ;PRINTER DEDICATED TO USER?
2076 007520' 001403 BEQ 40$ ;Y,N-40$
2077 007522' 052777 004000 000012' BIS #CHPINT,@CSYSFW ;SET 'HOLD INT ENB SET' FLAG
2078 007530' 016701 001714 40$: MOV PRTIVA,R1 ;GET PRINTER'S INT VEC ADR
2079 007534' 012721 011162' MOV #CPRINT,(R1)+ ;STORE INT ROUT ADR AT VECTOR
2080 007540' 016711 001706 MOV PRTPSW,(R1) ;STORE PS WORD AT VECTOR
2081 007544' 005077 001676 CLR @PRTDRA ;MAY HAVE TO TOGGLE INT ENABLE
2082 007550' 012777 000100 001670 MOV #100,@PRTDRA ;SET PRINTER'S INT ENABLE
2083 007556' 005712 TST (R2) ;THIS A USER PRINT?
2084 007560' 100406 BMI 60$ ;N,Y-60$
2085 007562' 004767 170612 50$: JSR PC,LIGHTS ;DISPLAY IDLE BIT PATTERN
2086 007566' 032767 000001 001632 BIT #CPRBSY,CPRFWD ;PRINTER STILL BUSY?
2087 007574' 001372 BNE 50$ ;N,Y-50$
2088 007576' 000205 60$: RTS R5 ;EXIT TO IN-LINE ADR
2089
2090 007600' 042452 025122 046040 CPRTER: .ASCII /*ER* LST DEV: HW ERR/<015><012>
007606' 052123 042040 053105
007614' 020072 053510 042440
007622' 051122 005015
2091 .EVEN

```

.SBTTL CONSOLE TERMINAL INTERRUPT ROUTINES

;CONSOLE TERMINAL WRITE INTERRUPT

| | | | | | | | | | |
|------|---------|--------|----------------|---------|-------|------------------------|--|--|------------------------------------|
| 2093 | | | | | | | | | |
| 2094 | | | | | | | | | |
| 2095 | | | | | | | | | |
| 2096 | | | | | | | | | |
| 2097 | | | | | | | | | |
| 2098 | 007626' | 004567 | 001172 | CTWINT: | JSR | R5,CSVREG | | | ;GO SAVE ALL REGISTERS |
| 2099 | 007632' | 016702 | 001310 | | MOV | CTWDRA,R2 | | | ;SET UP DEV REG'S ADR |
| 2100 | 007636' | 005722 | | | TST | (R2)+ | | | ;ADJUST IT TO DEV BUF ADR |
| 2101 | 007640' | 012705 | 011076' | | MOV | #CTFTCT,R5 | | | ;SET UP TAB/FILL COUNT ADR |
| 2102 | 007644' | 005715 | | | TST | (R5) | | | ;TAB/FILL COUNT ZERO? |
| 2103 | 007646' | 001403 | | | BEQ | CTWRCE | | | ; (N,Y-CTWRCE) |
| 2104 | 007650' | 005325 | | | DEC | (R5)+ | | | ;DECR TAB/FILL CNT BY 1 |
| 2105 | 007652' | 011512 | | CTWFOT: | MOV | (R5),(R2) | | | ;MOVE FILL/SPACE CHAR TO DEV BUF |
| 2106 | 007654' | 000454 | | | BR | CTWREX | | | ;GO TO INT ROUT EXIT |
| 2107 | 007656' | 012704 | 011106' | CTWRCE: | MOV | #CTECNT,R4 | | | ;SET UP ADR OF ECHO BUF CNTRL WDS |
| 2108 | 007662' | 005724 | | | TST | (R4)+ | | | ;ECHO BUF CNT ZERO? |
| 2109 | 007664' | 001411 | | | BEQ | CTWRCD | | | ; (N,Y-CTWRCD) |
| 2110 | 007666' | 021464 | 000002 | | CMP | (R4),2(R4) | | | ;AT THE END OF THE BUF? |
| 2111 | 007672' | 001002 | | | BNE | 1\$ | | | ; (Y,N-1\$) |
| 2112 | 007674' | 162714 | 000024 | | SUB | #24,(R4) | | | ;SET UP BUF START ADR |
| 2113 | 007700' | 113400 | | 1\$: | MOVB | 2(R4)+,R0 | | | ;GET CHAR TO ECHO |
| 2114 | 007702' | 005244 | | | INC | -(R4) | | | ;ADD 1 TO CURR BUF ADR |
| 2115 | 007704' | 005344 | | | DEC | -(R4) | | | ;SUB 1 FROM BUF COUNT |
| 2116 | 007706' | 000456 | | | BR | CTCKCH | | | ;GO CHECK THIS CHARACTER |
| 2117 | 007710' | 012703 | 011060' | CTWRCD: | MOV | #CTWCNT,R3 | | | ;SET UP ADR OF DATA COUNT |
| 2118 | 007714' | 005713 | | | TST | (R3) | | | ;DATA CHAR COUNT ZERO? |
| 2119 | 007716' | 001036 | | | BNE | CTDOOT | | | ; (Y,N-CTDOOT) |
| 2120 | 007720' | 042742 | 000100 | CTWABT: | BIC | #100,-(R2) | | | ;RESET WRITE'S INT ENABLE |
| 2121 | 007724' | 032743 | 000040 | | BIT | #CTESY,-(R3) | | | ;WERE WE DOING ECHO'S? |
| 2122 | 007730' | 001403 | | | BEQ | 3\$ | | | ; (Y,N-3\$) |
| 2123 | 007732' | 004567 | 000770 | | JSR | R5,CTRTRM | | | ;GO TERMINATE THE RD FUNCTION |
| 2124 | 007736' | 000423 | | | BR | CTWREX | | | ;GO TO INT ROUT EXIT |
| 2125 | 007740' | 042713 | 000002 | 3\$: | BIC | #CTWBSY,(R3) | | | ;RESET WR BSY FLAG |
| 2126 | 007744' | 005713 | | | TST | (R3) | | | ;THIS A USER WRITE? |
| 2127 | 007746' | 100017 | | | BPL | CTWREX | | | ; (Y,N-CTWREX) |
| 2128 | 007750' | 004577 | 000002' | | JSR | R5,2UFWCLR | | | ;RESET USER'S WAIT FOR |
| 2129 | 007754' | 000020 | | | .WORD | CTPRIO | | | ;CONSOLE/PRINTER I/O FLAG |
| 2130 | 007756' | 001106 | | | .WORD | CTWUFW- | | | |
| 2131 | 007760' | 032777 | 040000 000012' | | BIT | #CRUDED,2CSYSFW | | | ;SUPPOSED TO RESET USER DEDICATED? |
| 2132 | 007766' | 001407 | | | BEQ | CTWREX | | | ;Y,N-CTWREX |
| 2133 | 007770' | 042777 | 060000 000012' | | BIC | #CTUDED+CRUDED,2CSYSFW | | | ;RESET SYSTEM DEDICATED FLAGS |
| 2134 | 007776' | 004577 | 000002' | | JSR | R5,2UFWCLR | | | ;RESET USER DEDICATED FLAG |
| 2135 | 010002' | 000040 | | | .WORD | SETDED | | | |
| 2136 | 010004' | 001060 | | | .WORD | CTWUFW- | | | |
| 2137 | 010006' | 004567 | 001026 | CTWREX: | JSR | R5,CRTREG | | | ;GO RESTORE REGISTERS |
| 2138 | 010012' | 000002 | | | RTI | | | | ;RETURN FROM INTERRUPT |
| 2139 | 010014' | 032743 | 001000 | CTDOOT: | BIT | #CTCREQ,-(R3) | | | ;CONSOLE REQUEST PENDING? |
| 2140 | 010020' | 001403 | | | BEQ | 5\$ | | | ; (Y,N-5\$) |
| 2141 | 010022' | 005723 | | | TST | (R3)+ | | | ;POINT AT WRITE COUNT |
| 2142 | 010024' | 005013 | | | CLR | (R3) | | | ;CLEAR WRITE DATA COUNT |
| 2143 | 010026' | 000734 | | | BR | CTWABT | | | ;GO ABORT THIS WRITE |
| 2144 | 010030' | 117300 | 000004 | 5\$: | MOVB | 24(R3),R0 | | | ;GET DATA CHARACTER |
| 2145 | 010034' | 005263 | 000004 | | INC | 4(R3) | | | ;INCREMENT DATA ADR |
| 2146 | 010040' | 005363 | 000002 | | DEC | 2(R3) | | | ;DECREMENT DATA COUNT |
| 2147 | 010044' | 062705 | 000004 | CTCKCH: | ADD | #4,R5 | | | ;ADJUST FILL/TAB TABLE ADR |
| 2148 | 010050' | 120027 | 000040 | | CMPB | R0,#40 | | | ;THIS A CONTROL/SPECIAL CHAR? |

| | | | | | | | | | |
|------|---------|--------|---------------|------------|---------------|--|--|--|------------------------------------|
| 2149 | 010054' | 103023 | | BHIS | 8\$ | | | | ;(Y,N-8\$) |
| 2150 | 010056' | 122700 | 000011 | CMPB | #011,RO | | | | ;IS CHAR A TAB? |
| 2151 | 010062' | 001010 | | BNE | 7\$ | | | | ;(Y,N-7\$) |
| 2152 | 010064' | 005315 | | DEC | (R5) | | | | ;DECR CURR TAB CNT BY 1 |
| 2153 | 010066' | 011565 | 177774 | MOV | (R5),-4(R5) | | | | ;MOVE CURR TAB CNT TO TAB/FILL CNT |
| 2154 | 010072' | 012715 | 000010 | MOV | #10,(R5) | | | | ;RESTORE CURR TAB COUNT |
| 2155 | 010076' | 012745 | 000040 | MOV | #40,-(R5) | | | | ;MOVE A SPACE TO FILL/SPACE CHAR |
| 2156 | 010102' | 000663 | | BR | CTWFOT | | | | ;GO BEGIN ISSUING SPACES |
| 2157 | 010104' | 122700 | 000015 | 7\$: CMPB | #015,RO | | | | ;IS IT A CARRIAGE RETURN? |
| 2158 | 010110' | 001407 | | BEQ | 9\$ | | | | ;(N,Y-9\$) |
| 2159 | 010112' | 122700 | 000033 | CMPB | #033,RO | | | | ;IS IT AN ALTMODE? |
| 2160 | 010116' | 001006 | | BNE | 11\$ | | | | ;(Y,N-11\$) |
| 2161 | 010120' | 112700 | 000044 | MOVB | #044,RO | | | | ;SET UP "S" CHARACTER |
| 2162 | 010124' | 005315 | | 8\$: DEC | (R5) | | | | ;DECR CURR TAB CNT BY 1 |
| 2163 | 010126' | 001002 | | BNE | 11\$ | | | | ;CNT = ZERO? (Y,N-11\$) |
| 2164 | 010130' | 012715 | 000010 | 9\$: MOV | #10,(R5) | | | | ;RESTORE CURR TAB CNT TO 8 |
| 2165 | 010134' | 120067 | 001020 | 11\$: CMPB | RO,CTFID | | | | ;FILLS AFTER THIS CHAR? |
| 2166 | 010140' | 001006 | | BNE | 13\$ | | | | ;(Y,N-13\$) |
| 2167 | 010142' | 016767 | 001006 000726 | MOV | CTFCNT,CTFCT | | | | ;SET UP FILL COUNT |
| 2168 | 010150' | 016767 | 001002 000722 | MOV | CTFCAR,CTFSCH | | | | ;SET UP FILL CHAR |
| 2169 | 010156' | 110012 | | 13\$: MOVB | RO,(R2) | | | | ;MOVE CHAR TO DEVICE BUF |
| 2170 | 010160' | 000712 | | BR | CTWREX | | | | ;GO TO INT ROUT EXIT |

;CONSOLE TERMINAL READ INTERRUPT ROUTINE

| | | | | | | | | | |
|------|---------|--------|----------------|--------------|-----------------|--|--|--|--------------------------------|
| 2171 | | | | | | | | | |
| 2172 | | | | | | | | | |
| 2173 | | | | | | | | | |
| 2174 | | | | | | | | | |
| 2175 | | | | | | | | | |
| 2176 | 010162' | 004567 | 000636 | CTRINT: JSR | R5,CSVREG | | | | ;GO SAVE ALL REGISTERS |
| 2177 | 010166' | 012704 | 011056' | MOV | #CTFLGW,R4 | | | | ;SET UP ADR OF CONS TERM FLGWD |
| 2178 | 010172' | 016702 | 000742 | MOV | CTRDRA,R2 | | | | ;SET UP DEV REG'S ADR |
| 2179 | 010176' | 016200 | 000002 | MOV | 2(R2),RO | | | | ;GET READ DATA CHARACTER |
| 2180 | 010202' | 042700 | 000200 | BIC | #200,RO | | | | ;RESET HIGH ORDER BIT |
| 2181 | 010206' | 120027 | 000141 | LOWCAS: CMPB | RO,#141 | | | | ;THIS A LOWER CASE ALPHA? |
| 2182 | 010212' | 103405 | | BLO | 10\$ | | | | ;Y,N-10\$ |
| 2183 | 010214' | 120027 | 000172 | CMPB | RO,#172 | | | | |
| 2184 | 010220' | 101002 | | BHI | 10\$ | | | | |
| 2185 | 010222' | 042700 | 000040 | BIC | #40,RO | | | | ;RESET LOWER CASE BIT |
| 2186 | 010226' | 120027 | 000023 | 10\$: CMPB | RO,#023 | | | | ;THIS A CNTRL/S? |
| 2187 | 010232' | 001003 | | BNE | 20\$ | | | | ;Y,N-20\$ |
| 2188 | 010234' | 052714 | 002000 | BIS | #CTWAIT,(R4) | | | | ;SET SOFTWARE "WAIT" FLAG |
| 2189 | 010240' | 000503 | | BR | CTRDEX | | | | ;GO TO INT EXIT |
| 2190 | 010242' | 032714 | 002000 | 20\$: BIT | #CTWAIT,(R4) | | | | ;SOFTWARE "WAIT" FLAG SET? |
| 2191 | 010246' | 001403 | | BEQ | 30\$ | | | | ;Y,N-30\$ |
| 2192 | 010250' | 042714 | 002000 | BIC | #CTWAIT,(R4) | | | | ;RESET THE "WAIT" FLAG |
| 2193 | 010254' | 000475 | | BR | CTRDEX | | | | ;GO TO INT EXIT |
| 2194 | 010256' | 120027 | 000017 | 30\$: CMPB | RO,#017 | | | | ;THIS A CNTRL/O? |
| 2195 | 010262' | 001004 | | BNE | 40\$ | | | | ;Y,N-40\$ |
| 2196 | 010264' | 052777 | 010000 000012' | BIS | #CNTRLO,@CSYSFW | | | | ;SET THE SUPPRESS TYPING FLAG |
| 2197 | 010272' | 000466 | | BR | CTRDEX | | | | ;GO TO INT EXIT |
| 2198 | 010274' | 032714 | 001000 | 40\$: BIT | #CTCREQ,(R4) | | | | ;IS CONSOLE REQUEST FLAG SET? |
| 2199 | 010300' | 001063 | | BNE | CTRDEX | | | | ;(N,Y-CTRDEX) |
| 2200 | 010302' | 120027 | 000003 | CMPB | RO,#003 | | | | ;THIS CHAR A CNTRL/C? |
| 2201 | 010306' | 001527 | | BEQ | CTRLC | | | | ;(N,Y-CTRLC) |
| 2202 | 010310' | 032714 | 000001 | BIT | #CTRBSY,(R4) | | | | ;IS RD BUSY FLAG SET? |
| 2203 | 010314' | 001455 | | BEQ | CTRDEX | | | | ;(Y,N-CTRDEX) |
| 2204 | 010316' | 012705 | 011070' | MOV | #CTRADR,R5 | | | | ;SET UP ADR OF READ INFO |

```

2205 010322' 120027 000025      CMPB    R0,#025      ;IS IT A CTRL/U?
2206 010326' 001502              BEQ     CTRCTU      ;(N,Y-CTRCTU)
2207 010330' 122700 000177      CMPB    #177,R0     ;RD DATA A RUBOUT?
2208 010334' 001450              BEQ     CTRRKY      ;(N,Y-CTRRKY)
2209 010336' 032714 000100      BIT     #CTRRUB,(R4) ;RUBOUT FLAG SET?
2210 010342' 001405              BEQ     50$        ;(Y,N-50$)
2211 010344' 004567 000376      JSR     R5,CTRPEB  ;PUT A BACKSLASH IN ECHO BUF
2212 010350' 134 000          .BYTE  134,0
2213 010352' 042714 000100      BIC     #CTRRUB,(R4) ;RESET RUBOUT FLAG
2214 010356' 110035              50$:   MOVB   R0,2(R5)+ ;MOVE RD DATA CHAR TO USER ADR
2215 010360' 005245              INC     -(R5)      ;INCR RD DATA ADR BY 1
2216 010362' 122700 000015      CMPB    #015,R0     ;IS THIS A CARR RET?
2217 010366' 001004              BNE     52$        ;(Y,N-52$)
2218 001                          .IF DF MIMIC
2219 .IFF
2220 010370' 004567 000352      51$:   JSR     R5,CTRPEB  ;PUT CR/LF IN ECHO BUF
2221 010374' 015 012          .BYTE  015,012
2222 .IFT
2223 51$:   MOV     #5015,R3 ;SET UP CR/LF DATA
2224 JSR     R5,CTWPEB ;PUT THEM IN THE ECHO BUF
2225 .ENDC
2226 010376' 000416              BR      54$        ;GO PROCESS RD END
2227 010400' 122700 000012      52$:   CMPB    #012,R0     ;IS IT A LINE FEED?
2228 010404' 001771              BEQ     51$        ;N,Y-51$
2229 010406' 110067 000004      MOVB   R0,53$     ;SET UP RD DATA FOR ECHO S/R
2230 010412' 004567 000330      JSR     R5,CTRPEB  ;PUT CHAR IN ECHO BUF
2231 010416' 000000      53$:   .WORD  XXXX
2232 010420' 005345              DEC     -(R5)      ;DECR DATA COUNT BY ONE
2233 010422' 001012              BNE     CTRDEX     ;DATA CNT = 0? (Y,N-CTRDEX)
2234 010424' 112775 000015 000002  MOVB   #015,2(R5) ;STORE CARR RET IN USER'S AREA
2235 001                          .IF DF MIMIC
2236 .IFF
2237 010432' 000756              BR      51$        ;GO ECHO CR/LF & TERMINATE READ
2238 .IFT
2239 NOP                               ;REPLACE CODE IS 000755
2240 .ENDC
2241 010434' 052714 000400      54$:   BIS     #CTRDON,(R4) ;SET READ DONE FLAG
2242 010440' 105714              TSTB   (R4)        ;IS ECHO FLAG SET?
2243 010442' 100402              BMI     CTRDEX     ;(N,Y-CTRDEX)
2244 010444' 004567 000256      JSR     R5,CTRTRM  ;GO TERMINATE THIS READ
2245 010450' 004567 000364      CTRDEX: JSR     R5,CRTREG ;RESTORE ALL REG'S
2246 010454' 000002              RTI
2247
2248 ;RUBOUT KEY PROCESSING
2249
2250 010456' 032714 000100      CTRRKY: BIT     #CTRRUB,(R4) ;RUBOUT FLAG ALREADY SET?
2251 010462' 001005              BNE     56$        ;(N,Y-56$)
2252 010464' 052714 000100      BIS     #CTRRUB,(R4) ;SET THE RUBOUT FLAG
2253 010470' 004567 000252      JSR     R5,CTRPEB  ;GO PUT BACKSLASH IN ECHO BUF
2254 010474' 134 000          .BYTE  134,0
2255 010476' 021565 000004      56$:   CMP     (R5),4(R5) ;CURR DATA ADR = ORG DATA ADR?
2256 010502' 001004              BNE     58$        ;(Y,N-58$)
2257 010504' 004567 000236      JSR     R5,CTRPEB  ;SET UP BACKSLASH FOR ECHO
2258 010510' 134 000          .BYTE  134,0
2259 010512' 000417              BR      CTRRECL    ;GO DO CARR RET & LINE FEED
2260 010514' 005325      58$:   DEC     (R5)+     ;DECR DATA ADR BY 1

```

```

2261 010516' 115567 000006          MOVB   2-(R5),70$          ;GET CHAR THAT IS BEING DELETED
2262 010522' 005245          INC    -(R5)             ;INCREMENT DATA COUNT BY 1
2263 010524' 004567 000216          JSR    R5,CTRPEB        ;GO SET UP DELETED CHAR FOR ECHO
2264 010530' 000000          70$:  .WORD 0
2265 010532' 000746          BR     CTRDEX           ;GO TO INT ROUT EXIT
2266
2267                                ;CONTROL 'U' PROCESSING
2268
2269 010534' 016515 000004          CTRCTU: MOV 4(R5),(R5)      ;SET UP ORG DATA ADR
2270 010540' 016545 000002          MOV 2(R5),-(R5)         ;SET UP ORG DATA COUNT
2271 010544' 004567 000176          JSR    R5,CTRPEB        ;PUT "↑" AND "U" IN ECHO BUF
2272 010550' 136 125          .BYTE 136,125
2273 010552' 004567 000170          CTRECL: JSR R5,CTRPEB    ;PUT "CR/LF" IN ECHO BUF
2274 010556' 015 012          .BYTE 015,012
2275 010560' 042714 000100          BIC   #CTRRUB,(R4)      ;RESET RUBOUT FLAG
2276 010564' 000731          BR     CTRDEX           ;GO TO INT ROUT EXIT
2277
2278                                ;CONTROL "C" PROCESSING
2279
2280 010566' 005777 000012'          CTRLC: TST 2CSYSFW       ;CONSOLE REQ BEING PROCESSED?
2281 010572' 100726          BMI   CTRDEX           ;(N,Y-CTRDEX)
2282 010574' 032777 002000 000012'  BIT   #CUSPGR,2CSYSFW   ;USER PROG RUNNING?
2283 010602' 001722          BEQ   CTRDEX           ;(Y,N-CTRDEX)
2284 010604' 032714 000001          BIT   #CTRBSY,(R4)     ;IS RD BUSY FLG SET?
2285 010610' 001404          BEQ   CTCRCQ           ;Y,N-CTRCQ
2286 010612' 052714 000400          BIS   #CTRDON,(R4)     ;SET READ DONE FLAG
2287 010616' 004567 000104          JSR   R5,CTRTRM        ;GO TERMINATE READ FUNCTION
2288
2289                                ;CONSOLE REQUEST RECEIVED
2290
2291 010622' 012703 041536          CTCRCQ: MOV #↑C R3       ;SET UP "↑" & "C" CHARS
2292 010624' 004567 000122          JSR   R5,CTWPEB        ;GO PUT THEM IN ECHO BUFFER
2293 010632' 052714 001000          BIS   #CTCREQ,(R4)     ;SET THE CONSOLE REQ FLAG
2294 010636' 012746 000022          MOV   #18,-(SP)        ;SET UP STK PNTR ADR
2295 010642' 060616          ADD   SP,(SP)          ;AT TIME OF INT
2296 010644' 032777 000001 000012'  BIT   #M1VER,2CSYSFW   ;RUNNING UNDER MEM MGMNT?
2297 010652' 001406          BEQ   72$              ;Y,N-72$
2298 010654' 032766 140000 000020  BIT   #140000,16.(SP)  ;WERE WE IN USER MODE?
2299 010662' 001402          BEQ   72$              ;Y,N-72$
2300 010664' 005726          TST   (SP)+            ;REMOVE OTHER ADR
2301 010666' 006506          MFPI  SP               ;GET USER'S STK PNTR
2302 010670' 005046          72$: CLR -(SP)           ;SET PROCESSOR PRIORITY TO ZERO
2303 010672' 012746 010700'          MOV   #74$,-(SP)
2304 010676' 000002          RTI
2305 010700' 032777 000100 000240  74$: BIT #100,2CTWDRA      ;CONS TERM WRITE INT ENABLE SET?
2306 010706' 001374          BNE   74$              ;(N,Y-74$)
2307 010710' 052777 100000 000012'  BIS   #CTRQBP,2CSYSFW  ;SET CONS REQ BEING PROC FLG
2308 010716' 042714 001000          BIC   #CTCREQ,(R4)     ;RESET CONSOLE REQ FLAG
2309 010722' 000177 000014'          JMP   2USRINT          ;GO TO MPG USER INT ENTRY
2310
2311
2312                                ;CONSOLE TERMINAL READ TERMINATE S/R
2313
2314
2315 010726' 032767 000400 000122  CTRTRM: BIT #CTRDON,CTFLGW ;IS THE READER DONE FLAG SET?
2316 010734' 001403          BEQ   1$               ;(Y,N-1$)

```

```

2317 010736' 042767 000441 000112      BIC      #CTRBSY+CTRDON+CTEBSY,CTFLGW      ;RESET READER FLAGS
2318 010744' 000205      1$:      RTS      R5      ;EXIT IN-LINE
2319
2320      ;CONSOLE TERMINAL PUT IN ECHO BUFFER S/R
2321
2322 010746' 012503      CTRPEB: MOV      (R5)+,R3      ;GET CHARACTERS TO STORE
2323 010750' 105714      TSTB      (R4)      ;IS ECHO FLAG SET?
2324 010752' 100023      BPL      CTRPEX      ;(Y,N-CTRPEX)
2325 010754' 010546      CTWPEB: MOV      R5,-(SP)      ;SAVE WORK REG
2326 010756' 012705 011104'      MOV      #CTEID,R5      ;SU ADR OF ECHO BUF CNTRL WDS
2327 010762' 021565 000006      70$:     CMP      (R5),6(R5)      ;INPUT ADR AT END OF BUF?
2328 010766' 001002      BNE      80$      ;(Y,N-80$)
2329 010770' 162715 000024      SUB      #24,(R5)      ;SET UP BUFFER START ADR
2330 010774' 110335      80$:     MOV      R3,(R5)+      ;MOVE CHAR TO ECHO BUF
2331 010776' 005215      INC      (R5)      ;ADD 1 TO BUFFER COUNT
2332 011000' 005245      INC      -(R5)      ;ADD 1 TO BUFFER ADR
2333 011002' 105003      CLRB      R3      ;CLEAR CHAR JUST STORED
2334 011004' 000303      SWAB      R3      ;SET UP FOR NEXT CHAR
2335 011006' 005703      TST      R3      ;ANOTHER CHAR TO STORE?
2336 011010' 001364      BNE      70$      ;(N,Y-70$)
2337 011012' 012605      MOV      (SP)+,R5      ;RESTORE WORK REG
2338 011014' 052777 000100 000124      BIS      #100,CTWDRA      ;SET WRITE INT ENABLE
2339 011022' 000205      CTRPEX: RTS      R5      ;EXIT IN-LINE
2340
2341
2342
2343
2344      ;INTERRUPT SUBROUTINES
2345
2346
2347      ;SAVE REGISTERS R0 THRU R5
2348
2349 011024' 010446      CSVREG: MOV      R4,-(SP)      ;SAVE REGISTERS 0 THRU 4
2350 011026' 010346      MOV      R3,-(SP)      ;R5 IS SAVED BY JSR TO
2351 011030' 010246      MOV      R2,-(SP)      ;THIS SUBROUTINE
2352 011032' 010146      MOV      R1,-(SP)
2353 011034' 010046      MOV      R0,-(SP)
2354 011036' 010507      MOV      R5,PC      ;EXIT
2355
2356
2357      ;RESTORE REGISTERS R0 THRU R5
2358
2359 011040' 005726      CRTREG: TST      (SP)+      ;TAKE THIS JSR'S R5 OFF STACK
2360 011042' 012600      MOV      (SP)+,R0      ;RELOAD REG'S
2361 011044' 012601      MOV      (SP)+,R1
2362 011046' 012602      MOV      (SP)+,R2
2363 011050' 012603      MOV      (SP)+,R3
2364 011052' 012604      MOV      (SP)+,R4
2365 011054' 000205      RTS      R5      ;RESTORE OLD R5 & EXIT

```

```

2367          .SBTTL  CONSOLE TERMINAL INTERRUPT CONSTANTS
2368
2369
2370 011056' 0C0000          CTFLGW: .WORD  0          ;CONSOLE TERMINAL FLAGWORD
2371
2372          000001          CTRBSY= 1          ; READ BUSY
2373          000002          CTWBSY= 2          ; WRITE BUSY
2374          000040          CTEBSY= 40         ; ECHO BUSY
2375          000100          CTRRUB= 100        ; READ RUBOUT STRING
2376          000200          CTRDEC= 200        ; READ WITH ECHO
2377          000400          CTRDON= 400        ; READ DONE
2378          001000          CTCREQ= 1000       ; CONSOLE REQUEST
2379          002000          CTWAIT= 2000      ; SOFTWARE "WAIT" FLAG
2380          100000          CTWUSR= 100000     ; USER WRITE
2381
2382          ;CTWCNT MUST FOLLOW CTFLGW
2383
2384 011060' 000000          CTWCNT: .WORD  0          ;CONSOLE TERMINAL WRITE COUNT
2385 011062' 000000          CTWADR: .WORD  0          ;CT WR DATA ADR
2386 011064' 000000          CTWUFW: .WORD  0          ;*;CT WRITE USER FLGMD ADR
2387
2388 011066' 000000          CTRCNT: .WORD  0          ;CT RD DATA COUNT (CURRENT)
2389 011070' 000000          CTRADR: .WORD  0          ;CT RD DATA ADR (CURRENT)
2390 011072' 000000          .WORD  0          ;CT RD DATA COUNT (ORIG)
2391 011074' 000000          .WORD  0          ;CT RD DATA ADR (ORIG)
2392
2393
2394 011076' 000000          CTFTCT: .WORD  0          ;CT WR FILL/TAB COUNT
2395 011100' 000000          CTFSCH: .WORD  0          ;CT WR FILL/SPACE CHAR
2396 011102' 000010          .WORD  10         ;CT WR CURR SOFTWARE TAB CNT
2397
2398
2399          000024          CTEBFL= 24
2400 011104' 011114'          CTEIAD: .WORD  CTEBUF          ;ECHO BUFFER INPUT ADR
2401 011106' 000000          CTECNT: .WORD  0          ;ECHO BUF COUNT
2402 011110' 011114'          .WORD  CTEBUF          ;ECHO BUF OUTPUT ADR
2403 011112' 011140'          .WORD  CTEBUF+CTEBFL      ;ECHO BUF END ADR
2404 011114' 000024          CTEBUF: .BLKB  CTEBFL          ;ECHO BUFFER
2405
2406
2407          ;CONSOLE TERMINAL CONSTANTS
2408
2409 011140' 177560          CTRDRA: .WORD  177560      ;CONSOLE TERMINAL RD DEV REG ADR
2410 011142' 000060          CTRVEC: .WORD  60         ;CT RD INT VECTOR ADR
2411 011144' 000200          CTRBR: .WORD  200        ;CT RD PRIORITY LEVEL (BR4)
2412 011146' 177564          CTWDRA: .WORD  177564      ;CT WR DEV REG ADR
2413 011150' 000064          CTWVEC: .WORD  64         ;CT WR INT VECTOR ADR
2414 011152' 000200          CTWBR: .WORD  200        ;CT WR PRIORITY LEVEL (BR4)
2415
2416 011154' 000002          CTFcnt: .WORD  2          ;CT WR FILL COUNT
2417 011156' 000000          CTFcar: .WORD  0          ;CT WR FILL CHAR
2418 011160' 000012          CTFID: .WORD  012        ;CT WR FILL I.D. CHAR
2419

```

.SBTTL PRINTER (LP11,LS11,LV11) INTERRUPT ROUTINE

```

*****
THIS ROUTINE PROCESSES INTERRUPTS FOR THE LP11, LS11, AND LV11
PRINTERS. AN ERROR CONDITION WILL CAUSE IMMEDIATE TERMINATION
OF THE I/O OPERATION. MAXIMUM LINE LENGTH WILL BE
80 CHARACTERS. ANY ABOVE THAT WILL BE TRUNCATED. TERMINATION
INCLUDES THE RESETTING OF USER 'WHIOT' & 'CTPRIO' FLAGS
IF A USER I/O.
*****

```

2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600

011162' 004567 177636
011166' 012701 011432'
011172' 016700 000250
011176' 005710
011200' 100457
011202' 106327
011204' 000000
011206' 001051
011210' 005711
011212' 001452
011214' 005311
011216' 115102
011220' 005221
011222' 120227 000040
011226' 103413
011230' 005327
011232' 000120
011234' 002765
011236' 106327
011240' 000001
011242' 001424
011244' 110260 000002
011250' 004567 177564
011254' 000002

011256' 120227 000011
011262' 001420
011264' 120227 000015
011270' 001406
011272' 120227 000012
011276' 001403
011300' 120227 000014
011304' 001341
011306' 012767 000120 177716
011314' 012767 000001 177716
011322' 000750
011324' 016767 177710 177652
011332' 012702 000040
011336' 000734

011340' 005767 000062
011344' 100017
011346' 004577 000002'

```

CPRINT: JSR      RS,CSVREG      ;GO SAVE ALL REGISTERS
        MOV      #CPCNT,R1     ;SET UP INFO STORAGE AREA ADR
        MOV      PRDRA,R0     ;GET DEV REG ADR
        TST      (R0)         ;PRINTER ERROR?
        BMI      CPRTRM       ;N,Y-CPRTRM
        ASLB     (PC)+        ;DOING A TAB?

CPRTFG: .WORD    0

CPRIGN: TST      CPRDOT       ;N,Y-CPRDOT
        BEQ      CPRTRM       ;CHAR CNT = 0?
        DEC      (R1)         ;N,Y-CPRTRM
        MOVB    2-(R1),R2     ;DECR CNT BY 1
        INC      (R1)+        ;GET DATA CHAR
        CMPB    R2,#40        ;INCR DATA ADR BY 1
        BLO     CPRCTC       ;THIS A CONTROL CHAR?
        DEC     (PC)+        ;N,Y-CPRCTC
        BLT     CPRIGN        ;DECR LINE CHAR COUNT

CPRMCK: .WORD    80
CPRLNC: .WORD    80

CPRTBC: .WORD    1
        BEQ     CPRRTC       ;ROOM LEFT? (Y,N-CPRIGN)
        BR     CPRSCH        ;UPDATE RUNNING TAB COUNT

CPRSCH: MOVB    R2,2(R0)     ;CNT EXHAUSTED? (N,Y-CPRRTC)
CPRX:   JSR      RS,CRTREG   ;SEND CHAR TO PRINTER
        RTI     ;RESTORE REGISTERS
        ;RETURN AFTER INTERRUPT

CPRCTC: CMPB    R2,#011     ;CHAR A TAB?
        BEQ     CPRTAB       ;N,Y-CPRTAB
        CMPB    R2,#015     ;IS IT A CARR RETURN?
        BEQ     CPRSTC       ;N,Y-CPRSTC
        CMPB    R2,#012     ;A LINE FEED?
        BEQ     CPRSTC       ;N,Y-CPRSTC
        CMPB    R2,#014     ;A FORM FEED?
        BNE     CPRIGN        ;Y,N-CPRIGN -- IGNORE IT
        MOV     #80,CPRLNC   ;INITIALIZE LINE CHAR COUNT
        MOV     #1,CPRTBC    ;INITIALIZE TAB COUNT
        BR     CPRSCH        ;GO SEND THE CHAR
        MOV     CPRTBC,CPRTFG ;SET UP CURR TAB COUNT
        MOV     #40,R2       ;SET UP SPACE CHAR
        BR     CPRMCK        ;GO CHECK FOR ROOM

CPRTRM: TST      CPRFWD       ;THIS A USER PRINT?
        BPL     10$         ;Y,N-10$
        JSR     RS,DUFWCLP   ;RESE* USER'S WAIT FOR

```

```

011352' 000020 .WORD CTPRIO ;CONSOLE/PRINTER I/O FLAG
011354' 000060 .WORD CPRUFW-
011356' 032777 040000 000012' BIT @CRUDED,@CSYSFW ;SUPPOSED TO RESET USER DEDICATED?
011364' 001731 BEQ CPREX ;Y,N-CPREX
011366' 042777 064000 000012' BIC @CRUDED+CTUDED+CHPINT,@CSYSFW ;RESET SYS'S DED. FLAGS
011374' 004577 000002' JSR RS,@UFWCLR ;RESET USER DEDICATED FLAG
011400' 000040 .WORD SETDED
011402' 000032 .WORD CPRUFW-
011404' 032777 004000 000012' 10$ BIT @CHPINT,@CSYSFW ;INT ENB BEING HELD SET?
011412' 001001 BNE 20$ ;N,Y-20$
011414' 005010 CLR (R0) ;RESET PRINTER'S INT ENABLE
011416' 042767 000001 000002 20$ BIC @CPRBSY,CPRFWD ;RESET PRINTER'S BUSY FLAG
011424' 000711 BR CPREX ;GO TO EXIT

011426' 000000 CPRFWD: .WORD 0 ;PRINTER FLAG WORD
000001 CPRBSY= 1 ; PRINTER IS BUSY
000200 CPRERR= 200 ; PRINT ERROR MSG ISSUED
100000 CPRUSR= 100000 ; USER PRINT FUNCTION

011430' 000000 CPRADR: .WORD 0 ;CURRENT DATA ADR
011432' 000000 CPRCNT: .WORD 0 ;CURRENT CHAR COUNT
011434' 000000 CPRUFW: .WORD 0 ;*;USER FLAGWORD ADR

011436' CLISTI: ;LIST DEVICE INFO STORAGE
011436' 000000 .WORD 0 ; DEFAULT MDL CODE WORD
011440' 000000 .WORD 0 ; DEFAULT UNIT #
011442' 000000 .WORD 0 ; CURRENT MODEL CODE WORD
011444' 000000 .WORD 0 ; CURRENT UNIT #
011446' 177514 PRTDRA: .WORD 177514 ; DEV REG ADR
011450' 000200 PRTIVA: .WORD 200 ; INT VECTOR ADR
011452' 000200 PRTPSW: .WORD 200 ; PS WORD
011454' 000000 .WORD 0

011456' 000000 JFWHLD: .WORD 0 ;*:USER FLGWC ADR HOLD

```

```

                .SBTTL PAPER TAPE HANDLER PARAMETER TABLE & ROUTINES
2514
2515
2516
2517 011460' 011532'          PTPARM: .WORD PTBOOT          ;PAPER TAPE VECTOR TABLE
2518 011462' 011532'          .WORD PTDIR
2519 011464' 011532'          .WORD PTZERO
2520 011466' 011532'          .WORD PTDELT
2521 011470' 011556'          .WORD PTCLOS
2522 011472' 011550'          .WORD PTENT
2523 011474' 011536'          .WORD PTLOOK
2524 011476' 011550'          .WORD PTALLC
2525 011500' 011604'          .WORD PTDVR
2526
2527 011502' 000000          PTUNIT: .WORD 0          ;UNIT #
2528 011504' 000000          PTCREG: .WORD 0          ;PC11 CMND REG ADR
2529 011506' 000776          PTW CNT: .WORD 510.      ;BYTE COUNT
2530 011510' 001146'          PTBAOR: .WORD BUF+2     ;BUS ADR
2531 011512' 000000          PTBLK: .WORD 0          ;BLOCK #
2532 011514' 000000          PTCMND: .WORD 0          ;COMMAND
2533 011516' 111221          PTRD: .WORD 111221      ;READ COMMAND--MOVB (R2),(R1)+
2534 011520' 112112          PTWR: .WORD 112112      ;WRITE COMMAND--MOVB (R1)+,(R2)
2535 011522' 000000          PTRBCT: .WORD 0          ;REQUESTED BLOCK COUNT
2536 011524' 011530'          PTDPTR: .WORD PTDBLK    ;POINTS TO 1ST DIR BLOCK
2537 011526' 000000          PTLBLK: .WORD 0          ;LAST BLOCK # ALLOCATED
2538 011530'
2539 011530' 000000          PTPEND:
          PTDBLK: .WORD 0          ;0 INDICATES NON-DIRECTORY DEVICE
2540
2541
2542
2543
2544
2545
2546
2547
2548 011532'
2549 011532'
2550 011532'
2551 011532' 000167 174660          PTBOOT:
2552 011536' 012765 000001 177744  PTDIR:
2553 011544' 062716 000002          PTZERO:
2554 011550' 012765 000001 177750  PTDELT: JMP INVCMD          ;ISSUE INVALID CMND ERROR MSG
2555 011556' 000207          PTLOOK: MOV #1,X1STBK(R5) ;SET UP DUMMY BLK # IN JOB
2556
2557 011560' 012767 177550 177716  PTCLOS: ADD #2,(SP)      ;SET UP SUCCESSFUL EXIT
2558 011566' 000403          PTENT:
2559 011570' 012767 177554 177706  PTCMND: MOV #1,XLSTBK(R5) ;FAKE A BLOCK #
2560 011576' 012700 011460'          PTDPTR: MOV PC          ;EXIT IN-LINE
2561 011602' 000207
                ;DEV ADDRESS SET UP ROUTINES
2557 011560' 012767 177550 177716  SETPR: MOV #177550,PTCREG ;SET UP PTR DEV REG ADR
2558 011566' 000403          BR PTCOM                ;GO TO COMMON POINT
2559 011570' 012767 177554 177706  SETPP: MOV #177554,PTCREG ;SET UP PTP DEV REG ADR
2560 011576' 012700 011460'          PTCOM: MOV #PTPARM,RO     ;SET UP PT PARAM TBL ADR
2561 011602' 000207          RTS PC                  ;EXIT IN-LINE

```

```

2563 .SBTTL PAPER TAPE HANDLER DRIVER
2564
2565
2566 011604' 005027 PTDVR: CLR (PC)+ ;CLEAR DATA XFER FLAG
2567 011606' 000000 PTNOUT: .WORD 0 ;DATA TRANSFER FLAG
2568 011610' 016503 000000 MOV XCM(R5),R3 ;GET STATUS REG ADR
2569 011614' 010302 MOV R3,R2 ;SET UP DATA REG ADR
2570 011616' 005722 TST (R2)+ ;ERROR BIT SET?
2571 011620' 100470 BMI PTERR ;(N,Y-PTERR)
2572 011622' 012701 001146' MOV #BUF+2,R1 ;SET UP BUFFER ADR
2573 011626' 012700 000776 MOV #510,R0 ;SET UP BYTE COUNT
2574 011632' 016567 000010 000006 MOV XCO(R5),PT2 ;SET UP MOV8 INST
2575 011640' 005213 PT1: INC (R3) ;SET THE "GO" BIT IF THERE
2576 011642' 004767 000062 JSR PC,PTIMER ;GO WAIT FOR DONE
2577 011646' 000240 PT2: NOP ;CHANGED TO EITHER 111221 OR 112112
2578 011650' 001014 BNE PT3 ;BR IF DATA IS NON-ZERO
2579 011652' 005767 177730 TST PTNOUT ;ANY DATA TRANSFERRED?
2580 011656' 001011 BNE PT3 ;BR IF YES
2581 011660' 005765 177716 TST XWCTR(R5) ;NO, THIS 1ST BLK?
2582 011664' 001006 BNE PT3 ;BR IF NOT
2583 011666' 026567 000010 177622 CMP XCO(R5),PTR0 ;THIS A READ OPERATION?
2584 011674' 001002 BNE PT3 ;BR IF NOT
2585 011676' 005301 DEC R1 ;BKUP XFER ADR, IGNORE LEADING 0'S
2586 011700' 000757 BR PT1 ;GO KEEP I/O GOING
2587 011702' 005267 177700 PT3: INC PTNOUT ;SET DATA XFERRED FLAG
2588 011706' 001405 BEQ PTDONE
2589 011710' 005300 DEC R0 ;DECR BYTE COUNT
2590 011712' 003352 BGT PT1 ;CNT = 0? (Y,N-PT1)
2591 011714' 012767 000001 167222 MOV #1,BUF ;SET UP BLOCK #
2592 011722' 005265 177716 PTDONE: INC XWCTR(R5) ;UP BLKS PROCESSED CNT
2593 011726' 000207 RTS PC ;EXIT IN-LINE
2594
2595 011730' 005027 PTIMER: CLR (PC)+ ;RESET DELAY COUNTER
2596 011732' 000000 PTCNT: .WORD 0 ;DELAY COUNTER
2597 011734' 032713 100200 PTIME1: BIT #100200,(R3) ;TEST FOR ERROR OR DONE
2598 011740' 100415 BMI PTOUT ;ERROR SET? (N,Y-PTOUT)
2599 011742' 003016 BGT PTIMEX ;DONE SET? (N,Y-PTIMEX)
2600 011744' 026767 177762 177760 CMP PTCNT,PTCNT ;DELAY A BIT
2601 011752' 026767 177754 177752 CMP PTCNT,PTCNT
2602 011760' 005267 177746 INC PTCNT ;INCR DELAY COUNTER
2603 011764' 001363 BNE PTIME1 ;TIMED OUT? (Y,N-PTIME1)
2604 011766' 005767 177614 TST PTNOUT ;DONE ANY OUTPUT?
2605 011772' 001403 BEQ PTERR ;Y,N-PTERR
2606 011774' 005067 167144 PTOUT: CLR BUF ;EOF ON THE DEVICE
2607 012000' 000207 PTIMEX: RTS PC ;EXIT IN-LINE FROM S/R
2608
2609 012002' 000167 174316 PTERR: JMP DEVERR ;GO REPORT DEVICE ERROR
2610 001 .IF DF MIMIC
2611
2612
2613 ALPT1= .-LOC600
2614 ALFCT1= ALPT1&000077+8./2
2615 002 .IF NE ALFCT1
2616 ALQUN1= 31.-ALFCT1
2617 .BLKW ALQUN1 ;ALIGN TO NEXT 32 WORD BNDRY - 8
2618 .WORD 0

```

MAINTDEC-11-DTECA-B COMMON SECTION FOR ALL MPG EXECUTIVES
DTECAB.P11 PAPER TAPE HANDLER DRIVER

G05
MACY11 27(732) 24-SEP-76 13:56 PAGE 24-2

SEQ 0310

2619
2620

001
000

.ENDC
.ENDC

2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650

.SBTTL LOAD MEDIA HANDLER

012006' 007016'
012010' 011560'
012012' 007014'
012014' 011560'

EMDLTB: .WORD 7016 ; PAPER TAPE (PC11) MDL CODE
PTSUAD: .WORD SETPR ; PAPER TAPE RDR (PC11) SU ADR
.WORD 7014 ; PAPER TAPE RDR (PR11) MDL CODE
.WORD SETPR ; SU ADR

; THE REST OF THIS TABLE IS CONTAINED IN
; THE FIRST PART OF THE LOAD MEDIA HANDLER.

000000'

.CSECT HANDLR

: THE LOAD MEDIA DEVICE HANDLER WILL RESIDE IN THIS
: AREA OF MEMORY AND WILL BE FOLLOWED IN MEMORY
: BY THE ONE TIME HOUSEKEEPING ROUTINE.

: THE LISTINGS THAT CONTAIN THE VARIOUS HANDLERS WHICH
: OCCUPY THIS AREA ARE:

- : TCMPG = DTE0A-? - TC11 HANDLER
- : RKMPG = DTE1A-? - RK11 HANDLER
- : TMMPG = DTE2A-? - TM11 HANDLER
- : THMPG = DTE3A-? - TM02 HANDLER
- : RXMPG = DTE4A-? - RX11 HANDLER
- : RBMPG = DTE5A-? - RPO4 HANDLER

```

2652 .SBTTL MPG EXEC TEMPORARY ONE TIME HOUSEKEEPING
2653
2654 000000' .CSECT EXHSKP
2655
2656
2657 000000' HKLOCZ:
2658 001 .IF NDF MIMIC
2659
2660
2661 000000' 000000 UFWSET: .WORD XXXX
2662 000002' 000000 UFWCLR: .WORD XXXX
2663 000004' 000000 UFWTST: .WORD XXXX
2664 000006' 000000 BINASC: .WORD XXXX
2665 000010' 000000 BTASLB: .WORD XXXX
2666 000012' 000014' CSYSFW: .WORD USRINT
2667 000014' 000000 USRINT: .WORD XXXX
2668
2669 000016' MPGBGN =. ;MPG WILL BE LOADED STARTING HERE OR THE
2670 000 ;NEXT 32 WORD BOUNDARY IF USING MEM MGMNT.
2671 000 .ENDC
2672
2673 ;THE FOLLOWING CODE IS EXECUTED ONE TIME
2674 ;WHEN THE MPG EXEC IS FIRST LOADED. THIS CODE
2675 ;WILL BE DESTROYED WHEN THE MPG PROGRAM
2676 ;IS LOADED INTO THIS AREA.
2677
2678
2679 ;FORMAT VECTOR LOCATIONS 0 THRU 276.
2680 ;SET THE PSW TO 0.
2681 ;TEST IF CONSOLE TERMINAL IS PRESENT.
2682 ;INITIALIZE CONSOLE TERMINAL ADDRESSES.
2683
2684 000016' 005000 E1^HKP: CLR R0 ;SET LOCATIONS 0 & 2
2685 000020' 005020 CLR (R0)+ ;TO HALT'S
2686 000022' 005020 CLR (R0)+
2687 000024' 010001 MOV R0,R1 ;SET LOCATIONS 4 THRU 276 TO
2688 000026' 005721 TST (R1)+ ;+2 AND HALT.
2689 000030' 010120 10$: MOV R1,(R0)+ ;DO NOT CHANGE LOCATIONS
2690 000032' 005020 CLR (R0)+ ;40 & 42
2691 000034' 062701 000004 20$: ADD #4,R1
2692 000040' 020027 000300 30$: CMP R0,#300
2693 000044' 001406 BEQ 40$
2694 000046' 020027 000040 CMP R0,#40
2695 000052' 001366 BNE 10$
2696 000054' 062700 000004 ADD #4,R0
2697 000060' 000765 BR 20$
2698 000062' 012737 000002 000102 40$: MOV #2,R#102 ;INIT LSI-11 CLK VECTOR
2699 000070' 005046 CLR -(SP) ;SET UP PSW OF 0
2700 000072' 012746 000100' MOV #TTYTST,-(SP) ;SET UP ADR OF INST AFTER RTI
2701 000076' 000002 RTI ;POP PSW & PC OFF STACK
2702 000100' 012737 000124' 000004 TTYTST: MOV #TTYTRP,R#4 ;SET UP BUS TIMEOUT TRAP
2703 000106' 105777 011146' TSTB @CTWDRA ;IS A TERMINAL THERE FOR WRITE?
2704 000112' 100407 BMI FMTVCT ;N,Y-FMTVCT
2705 000114' 012700 011140' TTYERR: MOV #CTRDRA,R0 ;SET CONS TERM INFO ADR IN R0
2706 000120' 000000 HALT ;STOP; CONSOLE TERM NOT THERE.
2707 ;ALLOW USER TO CHANGE CONS

```

```

2708 ; TERM PRESET INFO
2709 000122' 000766 ; GO TRY WITH NEW DATA
2710 000124' 012716 000114' TTYTRP: MOV #TTYERR, (SP) ; SET UP ERR RET ADR
2711 000130' 000002 RTI ; EXIT FROM TRAP
2712 000132' 012700 011142' FMTVCT: MOV #CTRVEC, R0 ; SET UP CONS TERM RD & WR
2713 000136' 012001 MOV (R0)+, R1 ; INT VECTOR LOCATIONS
2714 000140' 012721 010162' MOV #CTRIINT, (R1)+ ; STORE RD INT ROUT ADR
2715 000144' 012021 MOV (R0)+, (R1)+ ; STORE RD PS WORD
2716 000146' 005720 TST (R0)+ ; BY PASS WR DEV REG ADR
2717 000150' 012001 MOV (R0)+, R1 ; GET WR INT VECT ADR
2718 000152' 012721 007626' MOV #CTWIINT, (R1)+ ; STORE WRITE INT ROUT ADR
2719 000156' 012021 MOV (R0)+, (R1)+ ; STORE WR PS WORD
2720
2721
2722 ; ISSUE EXECUTIVE'S TITLE MESSAGE
2723
2724 000160' 004567 005654' JSR R5, MES ; ISSUE PORTION OF MSG THAT
2725 000164' 000000G .WORD EXCID ; IS IN THE LOADER OR HANDLER
2726 000166' 004567 005654' JSR R5, MES ; ISSUE TRAILING PORTION
2727 000172' 002040' .WORD EXCIDT ; OF THE MSG
2728
2729
2730 ; CLEAR MEM & DETERMINE MEM SIZE
2731
2732 000174' 012737 000236' 000004 EHKMEM: MOV #EHKBAT, @#4 ; SET BUS ADR TRAP ROUT ADR
2733 000202' 016700 001136 MOV LIMIT+2, R0 ; SET UP END OF EXEC ADR
2734 000206' 005010 10$: CLR (R0) ; CLEAR WORD OF MEMORY
2735 000210' 005720 TST (R0)+ ; POINT TO NEXT LOCATION
2736 000212' 020027 160000 CMP R0, #160000 ; REACHED END OF MAX MEM?
2737 000216' 001373 BNE 10$ ; Y, N-10$
2738 000220' 020027 100000 EMFND: CMP R0, #100000 ; MEM 16K WORDS OR LARGER?
2739 000224' 103407 BLO EINSFM ; Y, N-EINSFM
2740 000226' 005740 TST -(R0) ; SET UP LAST WORD ADR
2741 000230' 010067 000466' MOV R0, ENOMEM ; SAVE IT
2742 000234' 000410 BR EGTDAT ; GO TO GET DATE ROUTINE
2743
2744 000236' 012716 000220' EHKBAT: MOV #EMFND, (SP) ; CHANGE RETURN ADR ON STACK
2745 000242' 000002 RTI ; RETURN AFTER OUT OF MEM TRAP
2746
2747 000244' 004567 005654' EINSFM: JSR R5, MES ; ISSUE INSUFFICIENT MEMORY
2748 000250' 000136' .WORD ISUFM ; ERROR MSG
2749 000252' 000000 HALT ; INSUF MEM HALT
2750 000254' 000747 BR EHKMEM ; GO CK MEM AGAIN
2751
2752
2753 ; GET TODAY'S DATE & PACK INTO DOS COMPATIBLE FORMAT
2754
2755 000256' 004567 005654' EGTDAT: JSR R5, MES ; ISSUE "ENTER DATE" MSG
2756 000262' 001454' .WORD EDMSG
2757 000264' 004567 007124' JSR R5, CTRD ; ISSUE READ FOR THE REPLY
2758 000270' 001024 .WORD EK$BUF-.
2759 000272' 000025 .WORD 21.
2760 000274' 012767 001314' 001010 MOV #EK$BUF, EKBPTR ; INITIALIZE REPLY DATA POINTER
2761 000302' 004767 001144' JSR PC, ATOI ; GO SCAN FOR DAY VALUE
2762 000306' 005702 TST R2 ; IS IT 0?
2763 000310' 001527 BEQ EDATER ; N, Y-EDATER

```

| | | | | | | | | |
|------|---------|--------|---------------|----------|------------|----------------|-------------------------------------|------------------------------|
| 2764 | 000312' | 010204 | | MOV | R2,R4 | | ;SAVE DAY VALUE | |
| 2765 | 000314' | 010267 | 000072 | MOV | R2,33\$ | | | |
| 2766 | 000320' | 004767 | 000212 | JSR | PC,EKBD\$K | | ;GO CHECK FOR DATA TO BE SKIPPED | |
| 2767 | 000324' | 012701 | 006214' | MOV | #EMONTB,R1 | | ;SET UP MONTH TABLE START ADR | |
| 2768 | 000330' | 012702 | 000002 | MOV | #2,R2 | | ;SET UP COMPARE CNT -1 | |
| 2769 | 000334' | 122021 | | 10\$: | MOV | (R0)+,(R1)+ | ;MONTH CHAR'S COMPARE? | |
| 2770 | 000336' | 001413 | | 20\$: | CMPB | | ;N,Y-30\$ | |
| 2771 | 000340' | 060201 | | | BEQ | 30\$ | ;INCR PAST REST OF CHAR'S | |
| 2772 | 000342' | 112103 | | | ADD | R2,R1 | ;GET THIS MONTH'S # OF DAYS | |
| 2773 | 000344' | 060304 | | | MOV | (R1)+,R3 | ;ADD IT INTO RUNNING TOTAL | |
| 2774 | 000346' | 020127 | 006274' | | ADD | R3,R4 | ;CHECKED ALL MONTHS? | |
| 2775 | 000352' | 001506 | | | CMP | R1,#EMONTB+48. | ;N,Y-EDATER | |
| 2776 | 000354' | 012703 | 000003 | | BEQ | EDATER | ;BACKUP KYBD DATA PNTR | |
| 2777 | 000360' | 160203 | | | MOV | #3,R3 | | |
| 2778 | 000362' | 160300 | | | SUB | R2,R3 | | |
| 2779 | 000364' | 000761 | | | SUB | R3,R0 | | |
| 2780 | 000366' | 005302 | | | BR | 10\$ | ;GO CK NXT MONTH | |
| 2781 | 000370' | 100361 | | 30\$: | DEC | R2 | ;DECR COMPARE CNT | |
| 2782 | 000372' | 010067 | 000714 | | BPL | 20\$ | ;COMPARED 3 CHAR'S? (Y,N-20\$) | |
| 2783 | 000376' | 004767 | 000134 | | MOV | R0,EKBPTR | ;STORE NEW DATA PNTR | |
| 2784 | 000402' | 004767 | 001144' | | JSR | PC,EKBD\$K | ;DO SELECTED DATA SKIP | |
| 2785 | 000406' | 111103 | | | JSR | PC,ATOI | ;SCAN & CONVERT YEAR VALUE | |
| 2786 | 000410' | 022703 | | | MOV | (R1),R3 | ;GET # OF DAYS IN THIS MONTH | |
| 2787 | 000412' | 000000 | | | CMP | (PC)+,R3 | ;# OF DAYS ENTERED TOO LARGE? | |
| 2788 | 000414' | 101412 | | 33\$: | .WORD | XXXX | | |
| 2789 | 000416' | 032702 | 000003 | | BLOS | 36\$ | ;Y,N-36\$ | |
| 2790 | 000422' | 001062 | | | BIT | #3,R2 | ;THIS A LEAP YEAR? | |
| 2791 | 000424' | 020127 | 006223' | | BNE | EDATER | ;Y,N-EDATER | |
| 2792 | 000430' | 001057 | | | CMP | R1,#EMONTB+7 | ;IS THIS FEBRUARY? | |
| 2793 | 000432' | 026727 | 177754 000035 | | BNE | EDATER | ;Y,N-EDATER | |
| 2794 | 000440' | 001053 | | | CMP | 33\$,#29. | ;IS DAY = 29? | |
| 2795 | 000442' | 032702 | 000003 | | BNE | EDATER | ;Y,N-EDATER | |
| 2796 | 000446' | 001004 | | 36\$: | BIT | #3,R2 | ;THIS A LEAP YEAR? | |
| 2797 | 000450' | 020127 | 006224' | | BNE | 40\$ | ;Y,N-40\$ | |
| 2798 | 000454' | 103401 | | | CMP | R1,#EMONTB+8. | ;IS MONTH MARCH OR LATER? | |
| 2799 | 000456' | 005204 | | | BLO | 40\$ | ;Y,N-40\$ | |
| 2800 | 000460' | 020227 | 000143 | | INC | R4 | ;ADD TO RUN TOT FOR FEB 29 | |
| 2801 | 000464' | 101041 | | 40\$: | CMP | R2,#99. | ;YEAR ENTERED > 99? | |
| 2802 | 000466' | 162702 | 000106 | | BHI | EDATER | ;N,Y-EDATER | |
| 2803 | 000472' | 100436 | | | SUB | #70.,R2 | ;SUB BASE VALUE FROM YEAR # ENTERED | |
| 2804 | 000474' | 001404 | | | BMI | EDATER | ;LESS THAN BASE? (N,Y-DATER) | |
| 2805 | 000476' | 062704 | 001750 | | 50\$: | BEQ | 60\$ | ;AT BASE YEAR? (N,Y-60\$) |
| 2806 | 000502' | 005302 | | | ADD | #1000.,R4 | ;ADD YEAR INCR VALUE TO RUN TOT | |
| 2807 | 000504' | 000773 | | | DEC | R2 | ;DECR YEAR VALUE | |
| 2808 | 000506' | 010467 | 000470' | | BR | 50\$ | ;GO CK FOR BASE | |
| 2809 | 000512' | 004767 | 000020 | | 60\$: | MOV | R4,TODAY | ;STORE RUN TOT AS M/D/Y CODE |
| 2810 | 000516' | 005067 | 000336 | | JSR | PC,EKBD\$K | ;DO SELECTED DATA SKIP | |
| 2811 | 000522' | 121027 | 000012 | | CLR | DATELF | ;RESET LF TERMINATOR FLAG | |
| 2812 | 000526' | 001002 | | | CMPB | (R0),#012 | ;LF TERMINATE DATE REPLY? | |
| 2813 | 000530' | 005267 | 000324 | | BNE | 70\$ | ;Y,N-70\$ | |
| 2814 | 000534' | 000421 | | | 70\$: | INC | DATELF | ;SET LF TERMINATOR REPLY FLG |
| 2815 | | | | | BR | ESYDSU | ;GO SET UP FOR LOAD DEV | |
| 2816 | 000536' | 016700 | 000550 | EKBD\$K: | MOV | EKBPTR,R0 | ;GET CURR DATA PNTR | |
| 2817 | 000542' | 121027 | 000040 | 100\$: | CMPB | (R0),#40 | ;DATA CHAR A SPACE? | |
| 2818 | 000546' | 001403 | | | BEQ | 110\$ | ;N,Y-110\$ | |
| 2819 | 000550' | 121027 | 000055 | | CMPB | (R0),#'- | ;A MINUS SIGN? | |

```

2820 000554' 001002          BNE      120$          ;Y,N-120$
2821 000556' 005200          110$:  INC      RO          ;ADD 1 TO DATA ADR
2822 000560' 000770          BR       100$          ;GO CK NXT CHAR
2823 000562' 010067 000524  120$:  MOV      RO,EKBPTR ;STORE NEW PNTR
2824 000566' 000207          RTS      PC          ;EXIT IN-LINE
2825
2826 000570' 004567 005654'  EDATER: JSR      R5,MES    ;ISSUE 'INV DATE' MSG
2827 000574' 001504'        .WORD   EDERMG
2828 000576' 000627        BR       EGTDAT      ;GO TRY AGAIN
2829
2830
2831          ;INITIALIZE FOR SYSTEM LOAD DEVICE
2832
2833 000600' 005003          ESYDSU: CLR      R3          ;INITIALIZE FOR NO LINE FEED
2834 000602' 012700          MOV      (PC)+,RO      ;GET LOAD DEV UNIT #
2835 000604' 177777          LDUNUM: .WORD   -1       ;LOAD DEV UNIT # STORAGE
2836 000606' 100020          BPL      70$          ;IS THERE ONE? (N,Y-70$)
2837 000610' 004567 005654'  60$:  JSR      R5,MES    ;ISSUE MSG TO REQUEST ONE
2838 000614' 001745'        .WORD   LDUREQ
2839 000616' 004767 001240'  JSR      PC,OATOB     ;GET REPLY & CONVERT TO BINARY
2840 000622' 000772          BR       60$          ;DATA ERROR? (N,Y-60$)
2841 000624' 005700          TST      RO          ;ANY DATA RECEIVED?
2842 000626' 001410          BEQ      70$          ;Y,N-70$
2843 000630' 010100          MOV      R1,RO        ;MOVE DATA RC'VD TO CORRECT REG
2844 000632' 020027 000010  CMP      RO,#8.       ;UNIT # TOO LARGE?
2845 000636' 103404          BLO      70$          ;Y,N-70$
2846 000640' 004567 005654'  JSR      R5,MES    ;ISSUE ERROR MSG
2847 000644' 001725'        .WORD   INVDMG
2848 000646' 000760          BR       60$          ;GO ASK AGAIN
2849 000650' 010067 000002G  70$:  MOV      RO,CFTCHI+2 ;STORE AS LOAD DEV DEFAULT UNIT #
2850 000654' 010067 000002G  MOV      RO,CSAVEI+2
2851 000660' 010067 000006G  MOV      RO,CFTCHI+6 ;STORE AS CURRENT UNIT #
2852 000664' 010067 000006G  MOV      RO,CSAVEI+6
2853 000670' 122703 000012  CMPB    #012,R3      ;LINE FEED TERMINATOR?
2854 000674' 001002          BNE      CK4MM        ;Y,N-CK4MM
2855 000676' 005267 000156  INC      DATELF      ;SET LF TERM FLG
2856
2857
2858          ;DETERMINE IF MEMORY MANAGEMENT IS ON THE SYSTEM AND
2859          ;IF IT IS, ASK IF THEY WANT TO USE IT.
2860          ;IF USING IT, ALIGN MPG START ADR TO A 32 WORD BOUNDARY.
2861
2862 000702' 012737 001050' 000004  CK4MM: MOV      #MMTRAP,#4    ;SET UP BUS ADR TRAP ROUT ADR
2863 000710' 005737 177572          TST      #MMMRO      ;TEST IF MMMRO REG IS THERE
2864
2865          ;IF NO TRAP, WE DO THE FOLLOWING
2866
2867 000714' 004567 005654'          80$:  JSR      R5,MES    ;ASK IF THEY WANT TO USE MEM
2868 000720' 002006'        .WORD   USEMM        ;MGMT
2869 000722' 004567 007124'  JSR      R5,CTRD     ;ISSUE READ FOR THE REPLY
2870 000726' 000366          .WORD   EK'BUF-
2871 000730' 000025          .WORD   21.
2872 000732' 004767 001376'  JSR      PC,CKRPLY   ;CHECK FOR A REPLY
2873 000736' 000766          BR       80$          ;BR IF INV DATA
2874 000740' 000401          BR       85$          ;BR IF Y, <CR>, OR <LF>
2875 000742' 000424          BR       MMCKRT     ;BR IF N
    
```

```

2876          001          .IF DF MIMIC
2877          .IFT
2878          85$: MOV      #MPGBGN,RO          ;GET CURRENT MPG START ADR
2879          .IFF
2880 000744' 012767 040515 000034' 85$: MOV      #'MA,MPGID          ;CHANGE MPG'S FILENAME
2881 000752' 012700 000016'      MOV      #MPGBGN,RO          ;GET CURRENT MPG START ADR
2882          000          .ENDC
2883          001          .IF DF DEBUG
2884          .IFF
2885 000756' 032700 000077      BIT      #77,RO          ;IS IT ON A 32 WD BNDRY?
2886          .IFT
2887          BIT      #7777,RO          ;IS IT ON A 2048 WD BNDRY?
2888          000          .ENDC
2889 000762' 001414      BEQ      MMCKRT          ;N,Y-MMCKRT
2890 000764' 010001      MOV      RO,R1          ;SAVE CURR ADR
2891          001          .IF DF DEBUG
2892          .IFF
2893 000766' 042700 000077      BIC      #77,RO          ;ALIGN ADR TO NXT 32 WD BNDRY
2894 000772' 062700 000100      ADD      #100,RO
2895          .IFT
2896          BIC      #7777,RO          ;ALIGN ADR TO NEXT 2048 WD BNDRY
2897          ADD      #10000,RO
2898          .ENDC
2899 000776' 160100      SUB      R1,RO          ;GET POSITION ADJ FACTOR
2900 001000' 010067 000120'      MOV      RO,POSADJ          ;STORE IT FOR MPG'S 1 TIME HSKP
2901 001004' 060067 000472'      ADD      RO,MPGST          ;UPDATE MPG'S START ADR
2902          001          .IF NDF MIMIC
2903 001010' 060067 000046'      ADD      RO,MPGADR          ;UPDATE MPG'S LOAD ADR
2904          000          .ENDC
2905 001014' 120327 000040      MMCKRT: CMPB     R3,#040          ;REPLY BYTE A LINE TERMINATOR?
2906 001020' 103402      BLO     90$          ;N,Y-90$
2907 001022' 112203      MOVB    (R2)+,R3          ;GET NEXT REPLY BYTE
2908 001024' 000773      BR      MMCKRT          ;GO CK IT
2909 001026' 120327 000012      90$:  CMPB     R3,#012          ;IS IT A LINE FEED?
2910 001032' 001002      BNE     MMCKEX          ;Y,N-MMCKEX
2911 001034' 005267 000020      INC     DATELF          ;SET LF TERM FLAG
2912 001040' 012737 000006 000004 MMCKEX: MOV      #6,2#4          ;RESTORE TRAP VECTOR
2913 001046' 000403      BR      CTCONS          ;GO TO NEXT HSKP FUNCTION
2914
2915 001050' 012716 001040'      MMTRAP: MOV      #MMCKEX,(SP)          ;BYPASS REST OF THIS FUNCTION
2916 001054' 000002      RTI          ;RETURN AFTER THIS TRAP
2917
2918
2919          ;REQUEST CONSOLE TERMINAL CONSTANTS CHANGE
2920
2921 001056' 005727      CTCONS: TST      (PC)+          ;LF TERMINATOR ON DATE REPLY?
2922 001060' 000000      DATELF: .WORD    0          ;DATE LF TERMINATOR FLAG
2923 001062' 031072      BNE     RELGBL          ;N,Y-RELGBL
2924 001064' 004567 00565+'      5$:  JSR      RS,MES          ;ISSUE TITLE & FIRST REQ MSG
2925 001070' 001522'      .WORD    CONIMG
2926 001072' 004767 001240'      JSR      PC,OATOB          ;GET REPLY & CONVERT TO BINARY
2927 001076' 000772      BR      5$          ;ERROR? (N,Y-5$)
2928 001100' 005700      TST      RO          ;DATA RECEIVED?
2929 001102' 001402      BEQ     10$          ;Y,N-10$
2930 001104' 010167 011160'      MOV      R1,CTFID          ;STORE NEW I.D. CHAR
2931 001110' 122703 000012      10$:  CMPB     #012,R3          ;LINE FEED TERMINATOR?

```

```

2932 001114' 001455          BEQ      RELGBL          ;N,Y-RELGBL
2933 001116' 004567 005654' 15$:     JSR      R5,MES        ;ISSUE SECOND MSG
2934 001122' 001605'        .WORD   CON2MG
2935 001124' 004767 001240' JSR      PC,OATOB      ;GET REPLY & CONVERT
2936 001130' 000772        BR       15$           ;ERROR? (N,Y-15$)
2937 001132' 005700        TST     RO            ;DATA RECEIVED?
2938 001134' 001402        BEQ     20$           ;Y,N-20$
2939 001136' 010167 011156' MOV     R1,CTFCAR     ;STORE NEW FILL CHAR
2940 001142' 122703 000012 20$:     CMPB   #012,R3       ;LINE FEED TERMINATOR?
2941 001146' 001440        BEQ     RELGBL       ;N,Y-RELGBL
2942 001150' 004567 005654' 25$:     JSR      R5,MES        ;ISSUE THIRD MSG
2943 001154' 001632'        .WORD   CON3MG
2944 001156' 004767 001240' JSR      PC,OATOB      ;GET REPLY & CONVERT
2945 001162' 000772        BR       25$           ;ERROR? (N,Y-25$)
2946 001164' 005700        TST     RO            ;DATA RECEIVED?
2947 001166' 001402        BEQ     30$           ;Y,N-30$
2948 001170' 010167 011154' MOV     R1,CTFCNT     ;STORE NEW FILL CNT
2949 001174' 122703 000012 30$:     CMPB   #012,R3       ;LINE FEED TERMINATOR?
2950 001200' 001423        BEQ     RELGBL       ;N,Y-RELGBL
2951 001202' 004567 005654' 35$:     JSR      R5,MES        ;ISSUE LOWER CASE MSG
2952 001206' 001657'        .WORD   CON4MG
2953 001210' 004567 007124' JSR      R5,CTRD      ;ISSUE RD FOR REPLY
2954 001214' 000100        .WORD   EK$BUF-
2955 001216' 000025        .WORD   21
2956 001220' 004767 001376' JSR      PC,CKRPLY     ;GO CHECK THE REPLY
2957 001224' 000766        BR       35$           ;BR IF INV DATA
2958 001226' 000410        BR       RELGBL      ;BR IF Y, <CR> OR <LF>
2959 001230' 012702 010206' MOV     #LOWCAS,R2    ;NO SET UP ADR OF INST'S TO WIPE OUT
2960 001234' 012703 000010 50$:     MOV     #8,R3         ;SET UP # OF WORDS
2961 001240' 012722 000240 50$:     MOV     #240,(R2)+   ;CLEAR INST'S TO NOP'S
2962 001244' 005303        DEC     R3
2963 001246' 001374        BNE    50$
2964
2965
2966          ;RELOCATE EXEC INTERFACE ADR'S
2967
2968 001250' 012700 001424' RELGBL: MOV     #EGBLTE,R0    ;GET END ADR OF INTERFACE ADR'S TBL
2969 001254' 016701 000466' MOV     ENDMEM,R1     ;GET END OF MEM ADR
2970 001260' 012702 000027' MOV     #EGBLTE-EGBLTS/2,R2 ;GET # OF ADDRESSES
2971 001264' 010267 000075' MOV     R2,ESGCNT     ;STORE THIS CNT FOR HSKP EXTENSION
2972 001270' 014041 85$:     MOV     -(R0),-(R1)   ;STORE ADR AT TOP OF MEM
2973 001272' 005302        DEC     R2           ;DECR ADDRESS CNT
2974 001274' 001375        BNE    85$           ;STORED ALL? (Y,N-85$)
2975 001276' 010167 000062' MOV     R1,ESGADR     ;STORE ADR OF STORED ADR'S
2976
2977
2978          .IF DF MIMIC
2979          ;DISPLAY SLEEP BREAKPOINT ADDRESS
2980
2981 DSBKPT: MOV     #SLPBKP,R1 ;GET ADR OF BREAKPOINT INST
2982          MOV     #SBKPAD,R2 ;GET ADR OF MSG AREA
2983          JSR     PC,OBTOA ;CONVERT ADR TO BIN & PUT IN MSG
2984          JSR     R5,MES ;ISSUE SLP BKPT ADR MSG
2985          .WORD   SBKPMG
2986
2987
001

```

```

2998          000          .ENDC
2999          .HOUSEKEEPING EXIT
3000          EXHKEK: MOV      #240,-(R5)          ;SET CALLING JSR TO NOP'S
3001          MOV      (R5),-(R5)
3002          RTS      R5          ;EXIT IN-LINE
3003
3004          001312' 001314'  EKBPTR: .WORD    EKYBUF          ;KEYBOARD DATA BUFFER POINTER
3005          001314' 000025'  EKYBUF: .BLKB   21          ;21 BYTE KEYBOARD BUFFER
3006          001341'      015   .BYTE    015
3007
3008          001342' 000000 000000  LIMIT: .LIMIT          ;EXEC'S START & END ADR --
3009          .IF DF MIMIC          ;TAILORED BY THE LNKY11 PROGRAM
3010
3011          SBKPMG: .ASCII  <015><012>/SLEEP BKPT = /
3012          SBKPAD: .ASCIZ  /XXXXXX/<015><012>
3013          .EVEN
3014          .PAGE
3015          .SBTTL  ROUTINE THAT CREATES DISK FOR MIMIC DEBUGGING
3016
3017          RKER= 177402
3018          RKCS= 177404
3019          RKWC= 177406
3020          RKBA= 177410
3021          RKDA= 177412
3022
3023          DISKWR: MOV      #37744,R6          ;INITIALIZE STACK POINTER
3024          MOV      LIMIT+2,R0          ;GET END OF EXEC+MPG ADR
3025          ASR      R0          ;MAKE IT A WORD COUNT
3026          NEG      R0          ;MAKE IT NEGATIVE
3027          MOV      R0,(PC)+          ;STORE IT AS EXEC'S WORD COUNT
3028          ESIZE: .WORD    0
3029          MOV      #-256,@#RKWC          ;SET WORD COUNT TO 1 BLOCK
3030          CLR      @#RKBA          ;SET BUSS ADR TO 0
3031          CLR      @#RKDA          ;SET BLOCK # TO 0
3032          MOV      #3,@#RKCS          ;ISSUE WRITE DATA CMND
3033          10$: TST      @#RKCS          ;ERROR BIT SET?
3034          BMI      40$          ;N,Y-40$
3035          TSTB     @#RKCS          ;DONE BIT SET?
3036          BPL      10$          ;Y,N-10$
3037          MOV      ESIZE,@#RKWC          ;SET UP EXEC'S WORD COUNT
3038          CLR      @#RKBA          ;SET BUSS ADR TO 0
3039          MOV      #40,@#RKDA          ;SET BLOCK # TO 30
3040          MOV      #3,@#RKCS          ;ISSUE WRITE DATA CMND
3041          20$: TST      @#RKCS          ;ERROR BIT SET?
3042          BMI      50$          ;N,Y-50$
3043          TSTB     @#RKCS          ;DONE BIT SET?
3044          BPL      20$          ;Y,N-20$
3045          RESET
3046          HALT          ;CLEAR EVERYTHING
3047          ;FINISHED CREATING DISK
3048
3049          ;ERROR ENCOUNTERED
    
```

3044
3045
3046
3047
3048
3049
3050
3051
3052
3053

JOC

```

405: CLR R2 ;SET FLAG TO THIS ENTRY POINT
      BR 605
505: MOV @-1,R2 ;SET IT TO THIS ENTRY POINT
605: MOV @MARKCS,R0 ;SAVE CONTROL STATUS
      MOV @MARKER,R1 ;SAVE ERROR REG
      MOV @1,@MARKCS ;ISSUE CONTROLLER RESET CMND
      HALT ;ERROR HALT
      BR DISKWR ;GO RETRY
      .ENDC

```

.SBTTL MPG/EXEC INTERFACE ADDRESSES TABLE

3055
3056
3057 001346'
3058 001346' 003656'
3059 001350' 001144'
3060 001352' 000000G
3061 001354' 011436'
3062 001356' 003136'
3063 001360' 007256'
3064 001362' 003112'
3065 001364' 000000G
3066 001366' 011056'
3067 001370' 007124'
3068 001372' 010450'
3069 001374' 007104'
3070 001376' 006646'
3071 001400' 006750'
3072 001402' 007306'
3073 001404' 003514'
3074 001406' 002412'
3075 001410' 003576'
3076 001412' 003244'
3077 001414' 003006'
3078 001416' 003142'
3079 001420' 003542'
3080 001422' 000536'
3081 001424'
3082
3083
3084
3085 001
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104 002
3105
3106
3107
3108 001
3109 00C
3110

EGBLTS:
.WORD BOOT
.WORD BUF
.WORD CFTCHI
.WORD CLISTI
.WORD CLOSE
.WORD CPRT
.WORD CREATE
.WORD CSAVEI
.WORD CTFLGW
.WORD CTRO
.WORD CTRDEX
.WORD CTRDNE
.WORD CTUSWR
.WORD CTWR
.WORD CUSPRT
.WORD DELETE
.WORD GET
.WORD LIST
.WORD LOADVR
.WORD OPENL
.WORD PUT
.WORD ZERO
.WORD REALNM

EGBLTE =.

.IF DF MIMIC

.BLKW 1.

:MODIFIABLE FILL AREA USED
:TO FORCE MPG START ADR
:OF 20000 WHEN UNDER MIMIC
:AND MEM MGMT.

UFWSET: .WORD XXXX
UFWCLR: .WORD XXXX
UFWTST: .WORD XXXX
BINASC: .WORD XXXX
BTASLB: .WORD XXXX
CSYSFW: .WORD USRINT
USRINT: .WORD XXXX

MPGBGN =.

ALPT2= .-HKLOCZ
ALFCT2= ALPT2&000077/2
.IF NE ALFCT2
ALQUN2= 31.-ALFCT2
.BLKW ALQUN2
.WORD 0
.ENDC
.ENDC

:ALIGN TO NEXT 32 WORD BNDRY

MAINTDEC-11-DTECA-B COMMON SECTION FOR ALL MPG EXECUTIVES
DTECRB.P11 MPG EXEC INTERFACE ADDRESSES TABLE

E06

MACY11 27(732) 24-SEP-76 13:56 PAGE 26-9

SEQ 0321

3111
3112
3113

001424'
000000'

EXCEND =.
.END EXECST

| | | | | | | | | | | |
|----------------|-------|--|----------------|-------|-----------------|-------|----------------|-----|-----------------|-------|
| ALC = 177772 | | | CPRTFG 011204R | 003 | CTRTRM 010726R | 003 | EGETWD 002714R | 003 | EUSRSA 003000R | 003 |
| ATOI 001144R | 003 | | CPRTRM 011340R | 003 | CTRVEC 011142R | 003 | EGTDAI 000256R | 005 | EWRTIL 004534R | 003 |
| BCDCV 005730R | G 003 | | CPRUFW 011434R | 003 | CTUDED= 020000 | | EHKBAT 000236R | 005 | EXCEND= 001424R | 005 |
| BCLEAR 005000R | G 003 | | CPRUSR= 100000 | | CTUSWR 006646R | 003 | EHKMEM 000174R | 005 | EXCID = ***** | G |
| BINASC 000006R | 005 | | CREATE 003112R | 003 | CTWABT 007720R | 003 | EINSFM 000244R | 005 | EXCIDT 002040R | 003 |
| BKROD 004516R | 003 | | CREGSV 004172R | 003 | CTWADR 011062R | 003 | EINTOL 004312R | 003 | EXDRET 003066R | 003 |
| BKREAO 004504R | G 003 | | CRLF 005546R | G 003 | CTWAIT= 002000 | | EJNFND 003106R | 003 | EXECST 000000R | G 003 |
| BKWRTD 004652R | 003 | | CRPMAP 005010R | G 003 | CTWBR 011152R | 003 | EJNRET 003536R | 003 | EXERET 003072R | 003 |
| BKWRTI 004676R | 003 | | CRTREG 011040R | 003 | CTWBSY= 000002 | | EKBOSK 000536R | 005 | EXERM 006604R | 003 |
| BLKWRT 004644R | G 003 | | CRUDED= 040000 | | CTWCNT 011060R | 003 | EKBPTR 001312R | 005 | EXERPT 006532R | 003 |
| BMOVE 004744R | G 003 | | CSAVEI= ***** | G | CTWCOM 007012R | 003 | EKYBUF 001314R | 005 | EXMKEX 001302R | 005 |
| BOOT 003656R | 003 | | CSVREG 011024R | 003 | CTWDRA 011146R | 003 | EMCOM 004752R | 003 | EXNRET 003056R | 003 |
| BTASLB 000010R | 005 | | CSYSFW 000012R | 005 | CTWFOT 007652R | 003 | EMDLTB 012006R | 003 | EYBK 004666R | 003 |
| BUF 001144R | G 003 | | CTBRET 006730R | 003 | CTWFOT 007626R | 003 | EMSEEX 005722R | 003 | EITHKP 000016R | 005 |
| BUFEND 002144R | 003 | | CTCKCH 010044R | 003 | CTWPEB 010754R | 003 | EMFND 000220R | 005 | FCLDSE 004704R | 003 |
| CFTCHI= ***** | G | | CTCONS 001056R | 005 | CTWR 006750R | 003 | EMINST 004764R | 003 | FLMOTF 006356R | G 003 |
| CHKSUM 000520R | 003 | | CTCREQ= 001000 | | CTWRCD 007710R | 003 | EMONMG 006206R | 003 | FMTVCT 000132R | 005 |
| CHPINT= 004000 | | | CTDODT 010014R | 003 | CTWRCE 007656R | 003 | EMONTB 006214R | 003 | FSTM00 000515R | G 003 |
| CHR0UT 005512R | G 003 | | CTDUMR 006740R | 003 | CTWREX 010006R | 003 | EMSADR 002774R | 003 | GET 002412R | 003 |
| CKRPLY 001376R | 003 | | CTEBFL= 000024 | | CTWUFW 011064R | 003 | ENDMEM 000466R | 003 | HKLOCZ 000000R | 005 |
| CKSMER 006444R | 003 | | CTEBSY= 000040 | | CTWUSR= 100000 | | ENMRET 004432R | 003 | IBKLG 002200R | 003 |
| CK4MM 000702R | 005 | | CTEBUF 011114R | 003 | CTWVEC 011150R | 003 | ENXDTA 002776R | 003 | IDHSU 002146R | 003 |
| CLIST 007240R | 003 | | CTECNT 011106R | 003 | CUSPGR= 002000 | | EOFMSG 006626R | 003 | IFLCNT 002154R | 003 |
| CLISTI 011436R | 003 | | CTEAD 011104R | 003 | CUSPRT 007306R | 003 | EPARBK 003004R | 003 | IFLMOO 002152R | 003 |
| CLOSE 003136R | 003 | | CTFCAR 011156R | 003 | DATLFL 001060R | 005 | EPKCNT 005200R | 003 | IFNAM 002256R | 003 |
| CLABUF 005024R | G 003 | | CTFCNT 011154R | 003 | DATUPK 006036R | G 003 | EPKCOM 005202R | 003 | ILSTBK 002202R | 003 |
| CLS = 177764 | | | CTFID 011160R | 003 | DOBEND 002412R | 003 | EPKEX 005320R | 003 | INALC 002224R | 003 |
| CMFNAM 005040R | G 003 | | CTFLGM 011056R | 003 | DOBSTR 002150R | 003 | EPKPEX 005314R | 003 | INBA 002236R | 003 |
| CNTRLO= 010000 | | | CTFSCH 011100R | 003 | DELAY 005124R | G 003 | ERDEOF 003005R | 003 | INBKCT 002250R | 003 |
| CONIMG 001522R | 003 | | CTFTCT 011076R | 003 | DELERR 006274R | G 003 | ERDERR 004440R | 003 | INBLK 002240R | 003 |
| CON2MG 001605R | 003 | | CTPRIO= 000020 | | DELETE 003514R | 003 | ERDLNK 004400R | 003 | INBOOT 002206R | 003 |
| CON3MG 001632R | 003 | | CTRADR 011070R | 003 | DELFIL 006310R | G 003 | ERET 004436R | 003 | INCLS 002216R | 003 |
| CON4MG 001657R | 003 | | CTRBR 011144R | 003 | DEVERR 006324R | G 003 | ERMFND 006613R | 003 | INCM 002232R | 003 |
| CPRADR 011430R | 003 | | CTRBSY= 000001 | | DEVFUL 006340R | G 003 | ERRCOM 006462R | 003 | INCOM 002242R | 003 |
| CPRBSY= 000001 | | | CTRcnt 011066R | 003 | DIRBLK= ***** | G | ESGADR 000062R | 003 | INDEND= 002270R | 003 |
| CPRCNT 011432R | 003 | | CTRCOM 007142R | 003 | DLT = 177762 | | ESGCNT 000076R | 003 | INDEV 002232R | 003 |
| CPRCOM 007372R | 003 | | CTRCRQ 010622R | 003 | DRT = 177756 | | ESUCOM 004126R | 003 | INDIR 002252R | 003 |
| CPRCTC 011256R | 003 | | CTRCTU 010534R | 003 | EBCDWA 005776R | 003 | ESUFNM 004052R | 003 | INDLT 002214R | 003 |
| CPROOT 011332R | 003 | | CTR0 007124R | 003 | ECFORC 006474R | 003 | ESUFTH 004030R | 003 | INDRT 002210R | 003 |
| CPRERR= 000200 | | | CTRDEC= 000200 | | ECOEFF 005504R | 003 | ESUFTI 004034R | 003 | INDRV 002230R | 003 |
| CPREX 011250R | 003 | | CTRDEX 010450R | 003 | ECOMEX 004772R | 003 | ESUIF 004104R | 003 | INDT 002240R | 003 |
| CPRFMD 011426R | 003 | | CTRONE 007104R | 003 | ECOMTY 005516R | 003 | ESUIOC 004230R | 003 | INETR 002220R | 003 |
| CPRIGN 011210R | 003 | | CTR0N= 000400 | | EDATER 000570R | 005 | ESUIPT 004216R | 003 | INNBK 002254R | 003 |
| CPRINT 011162R | 003 | | CTRORA 011140R | 003 | EDERMG 001504R | 003 | ESUOPT 004224R | 003 | INPRC 002244R | 003 |
| CPRLNC 011232R | 003 | | CTRECL 010552R | 003 | EDMSG 001454R | 003 | ESUOS 004116R | 003 | INPWC 002246R | 003 |
| CPRMCK 011230R | 003 | | CTRINT 010162R | 003 | EDTACT 002772R | 003 | ESUSAV 003762R | 003 | INSRH 002222R | 003 |
| CPRRTC 011314R | 003 | | CTRLC 010566R | 003 | EFDEYN 006642R | 003 | ESYDSU 000600R | 005 | INSRV 002226R | 003 |
| CPRSCH 011244R | 003 | | CTRPEB 010746R | 003 | EFMSZ 000050R | 003 | ETMP 002722R | 003 | INVCMO 006416R | 003 |
| CPRSTC 011306R | 003 | | CTRPEX 011022R | 003 | EGBLTE= 001424R | 005 | ETR = 177766 | | INVDMG 001725R | 003 |
| CPRT 007256R | 003 | | CTRQBP= 100000 | | EGBLTS 001346R | 005 | ETYEX 006030R | 003 | INWC 002234R | 003 |
| CPRTAB 011324R | 003 | | CTRKY 010456R | 003 | EGERET 002754R | 003 | ETYPWK 005534R | 003 | INZER 002212R | 003 |
| CPRTBC 011240R | 003 | | CTRPUB= 000100 | | EGETBY 002742R | 003 | EUPKCM 005372R | 003 | ISUFM 000136R | 003 |
| CPRTER 007623R | 003 | | CTRTRB 007222R | 003 | EGETKG 002762R | 003 | EUSRCT 003002R | 003 | ISVBLK 002162R | 003 |

| | | | | | | | | | | | | | | | | |
|---------|---------|-----|---------|---------|-----|--------|---------|-----|---------|---------|-----|---------|---------|---------|-----|-----|
| ISVCNT | 002160R | 003 | MMVER = | 000001 | 003 | OUZER | 002334R | 003 | PTWR | 011520R | 003 | WRTLC | 004632R | G | 003 | |
| ISVDAT | 002172R | 003 | MPGADR | 000046R | 003 | OWCTR | 002272R | 003 | PTZERO | 011532R | 003 | XBA = | 000004 | | | |
| ISVEXT | 002170R | 003 | MPGBGM= | 000016R | 005 | O1STBK | 002320R | 003 | PT1 | 011640R | 003 | XBC = | 000016 | | | |
| ISVMAP | 002156R | 003 | MPGID | 000034R | 003 | PACK | 005164R | 003 | PT2 | 011646R | 003 | XBKLG= | 177746 | | | |
| ISVNAM | 002164R | 003 | MPGST | 000472R | 003 | PACK1 | 005174R | 003 | PT3 | 011702R | 003 | XBT = | 177754 | | | |
| ISVUPT | 002204R | 003 | NFNAME | 006364R | 003 | PAKNAM | 005136R | G | PUT | 003142R | 003 | XCM = | 000000 | | | |
| ISVXX | 002174R | 003 | NXTBLK | 004452R | G | PC = | %000007 | | READBK | 004524R | G | XCO = | 000010 | | | |
| ITOA | 006004R | G | OATOB | 001240R | 003 | PIPFLG | 000522R | G | REALNM | 000536R | 003 | XDHSU = | 177714 | | | |
| IWCTR | 002150R | 003 | OBKLG | 002322R | 003 | POSAOJ | 000120R | 003 | RELBL | 001250R | 005 | XDN = | 177776 | | | |
| I1STBK | 002176R | 003 | ODHSU | 002270R | 003 | PRTDRA | 011446R | 003 | RSTO4 | 004730R | G | XDR = | 000020 | | | |
| LDEVER | 003474R | 003 | OFLCNT | 002276R | 003 | PRTFLG | 000516R | 003 | R0 = | %000000 | | XDT = | 000006 | | | |
| LDRBYC | 003252R | 003 | OFLMOD | 002274R | 003 | PRTIVA | 011450R | 003 | R1 = | %000001 | | XFLCNT= | 177722 | | | |
| LDRDTA | 003354R | 003 | OFNAM | 002400R | 003 | PRTPSW | 011452R | 003 | R2 = | %000002 | | XFLMOD= | 177720 | | | |
| LDRDTC | 003356R | 003 | OLSTBK | 002324R | 003 | PTALLC | 011550R | 003 | R3 = | %000003 | | XLSTBK= | 177750 | | | |
| LDREXC | 003432R | 003 | OPENL | 003006R | 003 | PTBADR | 011510R | 003 | R4 = | %000004 | | XNB = | 000022 | | | |
| LDRGET | 003346R | 003 | OSVBLK | 002304R | 003 | PTBLK | 011512R | 003 | R5 = | %000005 | | XRD = | 000012 | | | |
| LDRISM | 003504R | 003 | OSVCNT | 002302R | 003 | PTBOOT | 011532R | 003 | R6 = | %000006 | | XSV = | 177774 | | | |
| LDRMST | 003246R | 003 | OSVDAT | 002314R | 003 | PTCLOS | 011556R | 003 | SAVO4 | 004716R | G | XSVBLK= | 177730 | | | |
| LDRNEX | 003424R | 003 | OSVEXT | 002312R | 003 | PTCMND | 011514R | 003 | SETDED= | 000040 | | XSVCNT= | 177726 | | | |
| LDRPN | 003012R | 003 | OSVMAP | 002300R | 003 | PTCNT | 011732R | 003 | SETPP | 011570R | 003 | XSVDAT= | 177740 | | | |
| LDUNUM | 000604R | G | OSVNAM | 002306R | 003 | PTCOM | 011576R | 003 | SETPR | 011560R | 003 | XSVEXT= | 177736 | | | |
| LDUREQ | 001745R | 003 | OSVUPT | 002326R | 003 | PTCREG | 011504R | 003 | SP = | %000006 | | XSVMAP= | 177724 | | | |
| LIGHTS | 000400R | 003 | OSVXX | 002316R | 003 | PTDBLK | 011530R | 003 | SRH = | 177770 | | XSVNAM= | 177732 | | | |
| LIMIT | 001342R | 005 | OUALC | 002346R | 003 | PTDELT | 011532R | 003 | STACK = | 001200 | | XSVUPT= | 177752 | | | |
| LIST | 003576R | 003 | OUBKCT | 002372R | 003 | PTDIR | 011532R | 003 | STKADR | 000512R | 003 | XSVXX = | 177742 | | | |
| LMPGDE | 000132R | 003 | OUBOOT | 002330R | 003 | PTDONE | 011722R | 003 | TAB | 005564R | G | XWC = | 000002 | | | |
| LMPGIM | 000124R | 003 | OUCLS | 002340R | 003 | PTOPTR | 011524R | 003 | TOOAY | 000470R | 003 | XWCTR = | 177716 | | | |
| LMPGIN | 000060R | 003 | ODULT | 002336R | 003 | PTDVR | 011604R | 003 | TTYERR | 000114R | 005 | XWT = | 000014 | | | |
| LOADVR | 003244R | 003 | ODURT | 002332R | 003 | PTENT | 011550R | 003 | TTYTRP | 000124R | 005 | XXNAM = | 000024 | | | |
| LOC600 | 000000R | 003 | OUETR | 002342R | 003 | PTERR | 012002R | 003 | TTYTST | 000100R | 005 | XXX = | 000000 | | | |
| LOWCAS | 010206R | 003 | OUPRC | 002366R | 003 | PTIMER | 011730R | 003 | TXEXT | 000532R | G | X1STBK= | 177744 | | | |
| LSTFLG | 000514R | G | OUPWC | 002370R | 003 | PTIMEX | 012000R | 003 | TXNAM | 000524R | G | ZER = | 177760 | | | |
| LTSEX | 000464R | 003 | OUSRH | 002344R | 003 | PTIME1 | 011734R | 003 | UFWCLR | 000002R | 005 | ZERO | 003542R | | 003 | |
| LTSPAT | 000402R | 003 | OUTBA | 002360R | 003 | PTLCLK | 011526R | 003 | UFWILD | 011456R | 003 | \$BUF | 000474R | G | 003 | |
| L2MCNT | 000424R | 003 | OUTBLK | 002362R | 003 | PTLOOK | 011536R | 003 | UFWSET | 000000R | 005 | \$BUF10 | 000500R | G | 003 | |
| MAPBUF | 000544R | G | OUTCM | 002354R | 003 | PTNOUT | 011606R | 003 | UFWTST | 000004R | 005 | \$BUF2 | 000476R | G | 003 | |
| MBUF | 000744R | G | OUTCOM | 002364R | 003 | PTOUT | 011774R | 003 | UNIMAP= | 000040 | | \$MAPBF | 000502R | G | 003 | |
| MES | 005654R | G | OUTDEV | 002354R | 003 | PTPARM | 011460R | 003 | UNPACK | 005354R | 003 | \$MBUF | 000504R | G | 003 | |
| MMAP | 000754R | G | OUTDIR | 002374R | 003 | PTPEND | 011530R | 003 | UPACK1 | 005364R | 003 | \$MMAP | 000506R | G | 003 | |
| MMCKEX | 001040R | 005 | OUTDRV | 002352R | 003 | PTRBCT | 011522R | 003 | UPKNAM | 005326R | G | \$TPNM1 | 005600R | G | 003 | |
| MMCKRT | 001014R | 005 | OUTDT | 002362R | 003 | PTRD | 011516R | 003 | USEMM | 002006R | 003 | \$TXNAM | 000510R | G | 003 | |
| MMMR0 = | 177572 | | OUTNBK | 002376R | 003 | PTSUAD | 012010R | 003 | USRINT | 000014R | 005 | \$TYPNM | 005574R | G | 003 | |
| MMMR3 = | 172516 | | OUTSRV | 002350R | 003 | PTUNIT | 011502R | 003 | WRTBLK | 004660R | G | . | = | 001424R | | 005 |
| . ABS. | 000000 | 000 | | | | | | | | | | | | | | |
| | 000000 | 001 | | | | | | | | | | | | | | |
| LOADER | 000000 | 002 | | | | | | | | | | | | | | |
| EXEC | 012016 | 003 | | | | | | | | | | | | | | |
| HANDLR | 000000 | 004 | | | | | | | | | | | | | | |
| EXHASKP | 001424 | 005 | | | | | | | | | | | | | | |

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*, DTECAB/NL: TOC/DOC=DTECAB.P11

MAINTDEC-11-DTECA-B COMMON SECTION FOR ALL MPG EXECUTIVES
DTECAB.P11 SYMBOL TABLE

H06
MACY11 27(732) 24-SEP-76 13:56 PAGE 27-2

SEQ 0324

RUN-TIME: 9 19 1 SECONDS
RUN-TIME RATIO: 54/30=1.7
CORE USED: 6K (11 PAGES)

DOCUMENT PAGES: 72

