

HIGH SPEED TEXT SEARCH SYSTEM

HSTS SOFTWARE
LISTINGS

VOL. 5 OF 5

Diagnostics
Part 2

STAT

NGA Review Complete

HSTS MASTER COMPUTER SOFTWARE LISTINGS

SL120100

VOLUME 5 of 5

Prepared for:

Central Intelligence Agency
Washington, DC 20505

R80-016

March 1980

STA

STA

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

.TITLE .LOADER.

HARDWARE QUERY RESOLVER LOADER.

LOAD MICROCODE INTO HQR (MRP AND CP)

LOAD DATA MEMORY FILES

MRP DATA MEMORY

CP DATA MEMORY

QEX WINDOW MEMORY

QEX LOCATION MEMORY

FAL POINTER MEMORY

FAL COUNTER MEMORY

QLB REFERENCE PAGE

QLB PAGE 0

QLB PAGE 1

QLB PAGE 2

QEX SUCCESS BIT MEMORY

SUBDOC REFERENCE PAGE MEMORY

SLB PAGE MEMORY

SUBREAD MEMORY

SUBID MEMORY 1

SUBID MEMORY 2

LOADER PROMPTS FOR COMMAND LINE INPUT IN THE FORM OF
A MEMORY MNEMONIC AND AN OPTIONAL FILE VERSION NUMBER.
LOADER OPENS A FILE WHOSE NAME IS IN THE FORM LDXX.DAT
WHERE XX IS REPLACED BY THE MEMORY MNEMONIC AND LOADS
THE HQR.

THE FORMATS OF THE INPUT FILES VARY.

THE MICROCODE INPUT FILES CONTAIN THE LOADABLE MICROCODE
IN COLUMNS. THE FIRST WORD OF THE FIRST BLOCK OF THE FILE
CONTAINS THE NUMBER OF WORDS IN ONE COLUMN. SEE THE
EXPLANATORY NOTES TO THE PROGRAM 'CONVRT' WHICH WRITES THE
FILES.

FOR ALL FRAME 2 MEMORIES, THE LDXX FILES CONTAIN ADDRESS/DATA
PAIRS. THE LOADER PRE-CLEARs FRAME 2 MEMORIES BEFORE LOADING
ONLY AT THE ADDRESSES ENCOUNTERED IN THE LDXX FILE. THE FIRST
WORD OF THE FIRST BLOCK OF THE FILE CONTAINS THE NUMBER OF
WORDS TO BE LOADED.

FOR THE FRAME 1 MEMORIES (MRP DATA MEMORY AND CP DATA MEMORY)
THE LDXX FILES CONTAIN DATA ONLY. THE FIRST WORD OF THE FIRST
BLOCK OF THE FILE CONTAINS THE NUMBER OF WORDS TO BE LOADED.
THE TWO FILES LDMD.DAT AND LDOD.DAT ARE DIFFERENT FROM THE
FRAME 2 FILES BECAUSE THEIR LOADING IS DONE BY DMA.

EXIT FROM THE PROGRAM TAKES PLACE IN RESPONSE TO THE
COMMAND 'EX' ENTERED FROM THE TERMINAL.

.MCALL QIOW\$,QID\$,EXIT\$,ABRT\$,GCML\$,GCMLB\$,FSRSZ\$,CLEF\$
.MCALL FDBDF\$,FDRC\$,FDBK\$,FDDP\$,NMBLK\$,OPEN\$
.MCALL OPEN\$,CLOSE\$,READ\$,WTSE\$,SETF\$,UTLO\$,RDAF\$

```

58 ;
59 ;
60 LUN.TT = 1 ;LUN FOR TT0
61 EFN.1 = 1 ;EVENT FLAG FOR TT0
62 EFN.3 = 3 ;EVENT FLAG FOR HOR INTERRUPTS
63 EFN.33 = 33 ;EVENT FLAG FOR COMMUNICATING WITH EXERCISER
64 CMILUN = 2 ;GCML LUN
65 INLUN = 3 ;LUN FOR LDXX.DAT FILES
66 FIRST = 1 ;FIRST BLOCK READ
67 ALL = 2 ;LOAD ALL FILES
68 LAST = 4 ;TRANSFER OF LAST BLOCK OF FRAME 1 DATA MEMORY
69 ;
70 ;
71 ;
72 .NLIST BEX
73 MYSELF : .RAD50 /LOADER/
74 TSKTCB : .WORD 0 ;MY TCB
75 OLDVEC : .WORD 0 ;OLD VECTOR ADDRESS AT 274
76 EFBUF : .BLKW 4 ;EVENT FLAG BUFFER
77 LHM : .RAD50 /LDMM/ ;MRP MICROPROGRAM MEMORY LOADABLE FILE
78 LMD : .RAD50 /LDMD/ ;MRP DATA MEMORY LOADABLE FILE
79 LCS : .RAD50 /LDCS/ ;CP CONTROL STORE LOADABLE FILE
80 LCD : .RAD50 /LDCC/ ;CP DATA MEMORY LOADABLE FILE
81 LQW : .RAD50 /LDQW/ ;QEX WINDOW MEMORY
82 LQL : .RAD50 /LDQL/ ;QEX LOCATION MEMORY
83 LFP : .RAD50 /LDFF/ ;FAL POINTER MEMORY
84 LFC : .RAD50 /LDFF/ ;FAL COUNTER MEMORY
85 LQR : .RAD50 /LDQR/ ;QLB REFERENCE PAGE
86 LQ0 : .RAD50 /LDQ0/ ;QLB PAGE 0
87 LQ1 : .RAD50 /LDQ1/ ;QLB PAGE 1
88 LQ2 : .RAD50 /LDQ2/ ;QLB PAGE 2
89 LSF : .RAD50 /LDSF/ ;SLB REFERENCE PAGE
90 LS0 : .RAD50 /LDS0/ ;SLB PAGE
91 LS1 : .RAD50 /LDS1/ ;SIDMEM 1
92 LS2 : .RAD50 /LDS2/ ;SIDMEM 2
93 LSR : .RAD50 /LDSR/ ;SUBREAD MEMORY
94 LQX : .RAD50 /LDQX/ ;QEX SUCCESS BIT MEMORY
95 ;
96 VIRT : .WORD 0 ;VIRTUAL BLOCK COUNTER
97 ; .WORD 1
98 STAT : .BLKW 2 ;ID STATUS
99 FVER : .WORD 0 ;FILE VERSION NUMBER
100 CODE : .WORD 0 ;MEMORY SELECT CODE
101 ERWORD : .WORD 0 ;INDEX VALUE FOR ERROR MESSAGE TABLE
102 BINWD : .WORD 0 ;TARGET FOR NUMERIC CONVERSIONS FROM ASCII
103 SELECT : .WORD 0 ;ALL PURPOSE FLAG
104 APLACE : .WORD 0 ;PLACE TO MESS WITH CSR1
105 LCOUNT : .WORD 0 ;NUMBER OF WORDS TO LOAD
106 WCOUNT : .WORD 0 ;WORKING COUNTER
107 SCOUNT : .WORD 0 ;MEMORY WORD SIZE INDEX
108 DATA1 : .WORD 0 ;DATA FOR MEMORY WRITE
109 ADDR : .WORD 0 ;ADDR FOR MEMORY WRITE
110 MSTR2 : .WORD 0 ;DMA POINTER
111 TRANSF : .WORD 0 ;DMA TRANSFER COUNT
112 INSAVE : .WORD 0 ;
113 ALLPT : .WORD 0 ;-> TABLE OF ALL MEMORY RTH ADDRESSES
114 ALLCT : .WORD 0 ;NUMBER OF MEMORIES

```

115 000176		GCMBUF: .BLKW	40.		: COMMAND LINE BUFFER
116 000316	000000	GCMLN: .WORD	0		: COMMAND LINE BUFFER LENGTH
117 000320	000000	GCMPNT: .WORD	0		: POINTER TO COMMAND LINE BUFFER
118 000322		INLINE: .BLKW	256.		: INPUT AREA FOR LDXX DAT BLOCKS
119		:			
120		:			
121		:			
122 001322	077777	:			
123 001324	076000	QXHIGH: .WORD	077777		:X'7FFF'
124 001326	007777	QXLOW: .WORD	076000		:X'7C00'
125 001330	000000	FAHIGH: .WORD	4095.		:X'FFF'
126 001332	003777	FALOW: .WORD	0		
127 001334	000000	LHHIGH: .WORD	003777		:X'3FF'
128 001336	000377	LHLOW: .WORD	0		
129 001340	000000	SOHIGH: .WORD	255.		:SUBREF, SUBQLB
130 001342	000377	SOLOW: .WORD	0		
131 001344	000000	SRHIGH: .WORD	255.		:SUBREAD
132 001346	007777	SRLOW: .WORD	0		
133 001350	000000	SDHIGH: .WORD	4095.		:SIDMEM, 1, 2
134		SDLOW: .WORD	0		
135		:			
136		:			
137		:			
138		:			
139		:			
140		:			
141 001352		:			
142 001352	101	114			
143 001354	003114'	115			
144 001356	115	115			
145 001360	003222'	104			
146 001362	115	104			
147 001364	003616'	123			
148 001366	103	123			
149 001370	003232'	104			
150 001372	103	104			
151 001374	003242'	127			
152 001376	121	127			
153 001400	004100'	114			
154 001402	121	114			
155 001404	004124'	120			
156 001406	106	120			
157 001410	004406'	103			
158 001412	106	103			
159 001414	004432'	122			
160 001416	121	122			
161 001420	004666'	060			
162 001422	121	060			
163 001424	005114'	061			
164 001426	121	061			
165 001430	005142'	062			
166 001432	121	062			
167 001434	005170'	130			
168 001436	121	130			
169 001440	003252'	122			
170 001442	123	122			
171 001444	003434'				

```

172 001446      123      106      .ASCII /SF/           ;SUBREF MEMORY.
173 001450 003320*      .WORD SF.
174 001452      123      060      .ASCII /S0/          ;SUBQLB MEMORY.
175 001454 003366*      .WORD S0.
176 001456      123      061      .ASCII /S1/          ;SIDMEM 1.
177 001460 003502*      .WORD S1.
178 001462      123      062      .ASCII /S2/          ;SIDMEM 2.
179 001464 003550*      .WORD S2.
180 001466      105      130      .ASCII /EX/          ;EXIT ROUTINES.
181 001470 005732*      .WORD EXIT.
182          000024      FNUM. = <.-FTBL/4>
183          ;
184          ;
185          ;
186 001472      ALLTBL:
187 001472 003222* 003232* 003616* .WORD MRPMMC,CPCSC,MRPMD,CPCDC,QU,QL,FP,FC,QR,Q0,Q1,Q2.
188          000014      ALLNUM. = <.-ALLTBL>/2.
189          ;
190          ;
191          ;
192          ;
193 001522      015      012      .PRINT: .BYTE 15,12. ;PRECEDE PRINT LINE WITH CRLF.
194 001524
195
196          000116      .NLIST MEB.
197          .REPT 78.
198          .BYTE 40
199          .ENDR.
200          ;
201          ;
202          ;
203          ;
204 001642      000
205 001643      015      012      015      .BYTE 0
206 001647      105      130      111      .BYTE 15,12,15,12.
207 001666      015      012      000      .ASCII /EXIT:HDR:LOADER/.
208 001671      015      012      015      .BYTE 15,12,0
209 001677      110      101      122      .BYTE 15,12,15,12,15,12.
210 001735      015      012      000      .ASCII /HARDWARE QUERY RESOLVER LOADER/.
211 001740      015      012
212 001742      111      116      126      .BYTE 15,12,0
213 001770      015      012      .ASCII /INVALID NUMERIC VALUE/.
214 001772      105      122      122      .BYTE 15,12.
215 002010      015      012      .ASCII /ERROR ON READ/.
216 002012      111      116      126      .BYTE 15,12.
217 002042      015      012      .ASCII /INVALID MEMORY MNEMONIC/.
218 002044      115      111      123      .BYTE 15,12.
219 002064      015      012      .ASCII /MISSING OPERAND/.
220 002066      111      116      103      .BYTE 15,12.
221 002120      015      012      .ASCII /INCORRECT CHARACTER COUNT/.
222 002122      123      105      114      .BYTE 15,12.
223 002150      377      ASCII: .BYTE 377
224          .EVEN.
225          .LIST BEX.
226          .NLIST CND.
227          ;
228          ;

```

```
229 ; COMMAND-LINE MACRO
230 ;
231 ;
232 GCMBLK: GCMLB$ 2,,GCMBUF,CMILUN
233 ;
234 ; INPUT-FILE-FDB
235 ;
236 INFDB::
237 FDBDF$
238 FDRCA$ FD:RUM
239 FDBK$A INLINE,512,,,STAT
240 FDBP$A INLUN,,INDNB
241 INDNB:: NMBLK$ ,DAT
242 FRSZ$ 1
```



```

244 ;
245 ;
246 ; ENTER HERE
247 ;
248 ;
249 ; START:
250 002656 016767 000000 175120 MOV #TKTCB,TSKTCB ;SAVE MY TCB
251 002664 013767 000274 175114 MOV @#274,OLDVEC ;SAVE VECTOR AT 274
252 002672 012737 005764 000274 MOV #BPTISR,@#274 ;MOVE IN MY ISR ADDRESS
253 ;
254 002700 CALL OUT1 ;ISSUE INFORMATION MESSAGE
255 002704 012746 177777 MOV #177777,-(SP) ;CLEAR EVERYTHING
256 002710 012746 000010 MOV #Q#RSET,-(SP) ;HQR RESET
257 002714 CALL CSR1
258 002720 012746 000010 MOV #Q#RSET,-(SP) ;CLEAR RESET
259 002724 012746 176000 MOV #<Q#NCLK>,-(SP) ;SET NO-CLOCKS
260 002730 CALL CSR1
261 ;
262 ;
263 ; TOP OF COMMAND LOOP
264 ;
265 ;
266 ; SELECT MEMORY OR EXIT PROGRAM
267 ;
268 ; IF 'LOAD ALL FILES' WAS PREVIOUSLY SELECTED, JUMP
269 ; DIRECTLY TO THE ALL FILES SEQUENCING ROUTINE
270 ;
271 002734 ; COM:
272 002734 032767 000002 175206 BIT #ALL,SELECT ;LOAD ALL MEMORIES
273 002742 001402 BEQ 10$ ;NO READ COMMAND LINE
274 002744 000167 000166 JMP ALL2 ;LET 'ALL' COMMAND CONTROL LOADING
275 ;
276 ; PROMPT FOR MEMORY SELECTION OR EXIT
277 ; VALIDATE THE SELECTION
278 ;
279 10$: CALL SELMEM ;PROMPT FOR MEMORY SELECTION
280 002754 CALL FIND ;FIND THE MEMORY MNEMONIC
281 002760 103003 BCC 1$ ;OK, CONTINUE
282 002762 CALL ERR2
283 002766 000762 BR COM
284 002770 022700 000002 1$: CMP #2,R0 ;COMMANDS ARE 2 CHARS
285 002774 001403 BEQ 2$
286 002776 CALL ERR1
287 003002 000754 BR COM ;TRY AGAIN
288 ;
289 ; MATCH THE MNEMONIC FROM THE COMMAND LINE AGAINST A
290 ; TABLE OF VALID MNEMONICS
291 ;
292 2$: MOV #FNUM,R0 ;R0 = NUMBER OF MEMORIES
293 003010 012702 001352 MOV #FTBL,R2 ;R2 -> TABLE OF MEMORY MNEMONICS
294 003014 CALL SCAN ;FIND MATCH IN TABLE
295 003020 103003 BCC 3$ ;OK, CONTINUE
296 003022 CALL ERR3 ;COMMAND NOT IN TABLE
297 003026 000742 BR COM ;TRY AGAIN
298 ;
299 ; CHECK FURTHER IN THE COMMAND LINE FOR A FILE VERSION
300 ; NUMBER. IF THERE IS ONE, CONVERT IT FROM ASCII OCTAL

```

```

301 ; TO BINARY. IF THERE IS NOT ONE, THE FILE OPENED WILL
302 ; BE THAT WITH THE HIGHEST VERSION NUMBER.
303 ;
304 003030 010146 3$: MOV R1, -(SP) ; SAVE ROUTINE ADDRESS.
305 003032 CALL FIND ; LOOK FOR FILE VERSION NUMBER.
306 003036 103415 BCS 5$ ; NOTHING THERE, NO OVERRIDE.
307 003040 CALL PACKO ; CONVERT OCTAL VALUE
308 003044 103004 BCC 4$ ; OK.
309 003046 CALL ERR6 ; ERROR ON CONVERSION
310 003052 005726 TST (SP)+ ; RESTORE SP.
311 003054 000727 BR COM ; START OVER.
312 ;
313 ; JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
314 ;
315 003056 116767 175064 177552 4$: MOVB BINWD, INDNB+N, FVER ; INSERT FILE VERSION NUMBER.
316 003064 016767 175056 175046 MOV BINWD, FVER ; SAVE VERSION NUMBER
317 003072 042767 000001 175050 5$: BIC #FIRST, SELECT ; CLEAR 'FIRST TIME THROUGH' FLAG.
318 003100 012767 000001 175024 MOV #1, VIRT+2 ; INIT VIRTUAL BLOCK TO 1
319 003106 012601 MOV (SP)+, R1 ; LOAD ROUTINE ADDRESS.
320 003110 000171 000000 JMP @R1 ; GO TO ROUTINE.

```

```

322. ;
323 ;
324 ; LOAD ALL MEMORIES.
325 ;
326 ; SET UP TO SEQUENCE THROUGH ALL LOAD ROUTINES.
327 ; NB. ALL LOAD ROUTINES RETURN TO THE TOP OF THE
328 ; COMMAND LOOP ('COM'). CONTROL WILL BE RE-ROUTED HERE
329 ; WITHOUT A PROMPT ONCE THE #ALL FLAG HAS BEEN SET.
330 ;
331 ; IF THE ORIGINAL COMMAND HAD A FILE VERSION NUMBER
332 ; FOLLOWING IT IN THE COMMAND LINE, THEN ALL INPUT
333 ; FILES WITH THAT VERSION NUMBER WILL BE LOADED:
334 ;
335 ; >AL 4
336 ;
337 ; OTHERWISE THE FILES WITH THE HIGHEST VERSION NUMBERS
338 ; WILL BE LOADED.
339 ;
340 003114 ;
341 003114 052767 000002 175026 ; AL:
342 003122 012767 001472 175042 ; BIS #ALL,SELECT ; SET FLAG FOR LOADING ALL
343 003130 012767 000014 175036 ; MOV #ALLTBL,ALLPT ; INITIALIZE POINTER
344 003136 ; MOV #ALLNUM,ALLCT ; INITIALIZE COUNT
345 003136 005767 175032 ; ALL2:
346 003142 001005 ; TST ALLCT ; FINISHED?
347 003144 042767 000002 174776 ; BNE 1$ ; NO, CONTINUE
348 003152 000167 177556 ; BIC #ALL,SELECT ; CLEAR ALL FLAG
349 ; JMP COM ; RETURN TO MAIN LOOP
350 003156 116767 174756 177452 1$:
351 003164 042767 000001 174756 ; MOV FVER,INDNB+N,FVER ; INSERT FILE VERSION NUMBER
352 003172 012767 000001 174732 ; BIC #FIRST,SELECT ; CLEAR FIRST TIME THROUGH FLAG
353 003200 005367 174770 ; MOV #1,VIRT+2 ; INIT VIRTUAL BLOCK TO 1
354 003204 016701 174762 ; DEC ALLCT ; SUB FROM MEMORY COUNT
355 003210 062767 000002 174754 ; MOV ALLPT,R1 ; R1 -> MEMORY RTH ADDRESS
356 003216 000171 000000 ; ADD #2,ALLPT ; POINT TO NEXT MEMORY ROUTINE
357 ; JMP @R1 ; JUMP TO MEMORY ROUTINE

```

```
358 ;
359 ; MRP MICROPROGRAM MEMORY
360 ;
361 003222 MRPMMC:
362 003222 CALL MRPMM
363 003226 000167 177502 JMP COM
364 ;
365 ;
366 ; GP CONTROL STORE
367 ;
368 ;
369 ;
370 003232 CPCSC:
371 003232 CALL CPCS
372 003236 000167 177472 JMP COM
373 ;
374 ;
375 ; GP DATA MEMORY
376 ;
377 ;
378 003242 CPCDC:
379 003242 CALL CD
380 003246 000167 177462 JMP COM
```

```

382. ;
383 ;
384 ; LOAD SUBDOCUMENT PROCESSOR MEMORIES.
385 ;
386 ; QEX SUCCESS BIT MEMORY
387 ;
388 003252. 012767 000004 174662. QX: MOV. #S$QX.CODE. ;SET MEMORY SELECT CODE.
389 003260 016767 176052 174666. MOV. SQHIGH,LCOUNT. ;SET MEMORY UPPER BOUND.
390 003266 012767 000001 174662. MOV. #1,WCOUNT. ;SET MEMORY WORD SIZE.
391 003274 016767 174624 177324. MOV. LQX,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
392 003302. 016767 174620 177320. MOV. LQX+2,INDNB+N.FNAM+2.
393 003310. CALL. SPLOAD. ;LOAD THE MEMORY.
394 003314 000167 177414. JMP. COM.
395 ;
396 ; SUBREF MEMORY.
397 ;
398 003320. SF:
399 003320 012767 000006 174614. MOV. #S$QR.CODE. ;SET MEMORY SELECT CODE.
400 003326 016767 176004 174620. MOV. SQHIGH,LCOUNT. ;SET MEMORY UPPER BOUND.
401 003334 012767 000001 174614. MOV. #1,WCOUNT. ;SET MEMORY WORD SIZE.
402 003342. 016767 174532 177256. MOV. LSF,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
403 003350 016767 174526 177252. MOV. LSF+2,INDNB+N.FNAM+2.
404 003356. CALL. SPLOAD. ;LOAD THE MEMORY.
405 003362. 000167 177346. JMP. COM.
406 ;
407 ; SUBOLB MEMORY.
408 ;
409 003366. S0:
410 003366 012767 000005 174546. MOV. #S$QB.CODE. ;SET MEMORY SELECT CODE.
411 003374 016767 175736 174552. MOV. SQHIGH,LCOUNT. ;SET MEMORY UPPER BOUND.
412 003402. 012767 000001 174546. MOV. #1,WCOUNT. ;SET MEMORY WORD SIZE.
413 003410 016767 174470 177210. MOV. LS0,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
414 003416 016767 174464 177204. MOV. LS0+2,INDNB+N.FNAM+2.
415 003424. CALL. SPLOAD. ;LOAD THE MEMORY.
416 003430 000167 177300. JMP. COM.
417 ;
418 ; SUBREAD MEMORY.
419 ;
420 003434. SR:
421 003434 012767 000007 174500. MOV. #S$SR.CODE. ;SET MEMORY SELECT CODE.
422 003442. 016767 175674 174504. MOV. SRHIGH,LCOUNT. ;SET MEMORY UPPER BOUND.
423 003450 012767 000001 174500. MOV. #1,WCOUNT. ;SET MEMORY WORD SIZE.
424 003456 016767 174436 177142. MOV. LSR,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
425 003464 016767 174432 177136. MOV. LSR+2,INDNB+N.FNAM+2.
426 003472. CALL. SPLOAD. ;LOAD THE MEMORY.
427 003476 000167 177232. JMP. COM.
428 ;
429 ; SIDMEM 1
430 003502. S1:
431 003502 012767 000010 174432. MOV. #S$S1.CODE. ;SET MEMORY SELECT CODE.
432 003510 016767 175632 174436. MOV. SDHIGH,LCOUNT. ;SET MEMORY UPPER BOUND.
433 003516 012767 000003 174432. MOV. #3,WCOUNT. ;SET MEMORY WORD SIZE.
434 003524 016767 174360 177074. MOV. LS1,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
435 003532 016767 174354 177070. MOV. LS1+2,INDNB+N.FNAM+2.
436 003540. CALL. SPLOAD. ;LOAD THE MEMORY.
437 003544 000167 177164. JMP. COM.
438 ;

```

439				:	SIDMEM.2.		
440				:			
441	003550			S2:			
442	003550	012767	000014	174364	MOV.	#S\$S2.CODE.	:SET MEMORY SELECT CODE.
443	003556	016767	175564	174370	MOV.	SDHIGH.LCOUNT.	:SET MEMORY UPPER BOUND.
444	003564	012767	000003	174364	MOV.	#3.WCOUNT.	:SET MEMORY WORD SIZE.
445	003572	016767	174316	177026	MOV.	LS2.INDNB+N.FNAM.	:PLACE FILE NAME INTO INPUT DNB.
446	003600	016767	174312	177022	MOV.	LS2+2.INDNB+N.FNAM+2.	
447	003606				CALL.	SPLoad.	:LOAD THE MEMORY.
448	003612	000167	177116		JMP.	COM.	

```

450 ;
451 ;
452 ; MRP DATA MEMORY
453 ;
454 ;
455 ;
456 ;
457 ;
458 003616 MRPMD:
459 003616 005067 174342 CLR MSTR2 ;CLEAR MRP DATA MEMORY ADDRESS
460 003622 016767 174176 176776 MOV LMD,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
461 003630 016767 174172 176772 MOV LMD+2,INDNB+N,FNAM+2
462 003636 OPEN#R *INFDB
463 ;
464 003654 CALL DMASET ;START UP DMA MICROCODE
465 ;
466 ;
467 ; GET FIRST RECORD (BLOCK) FROM THE FIRST WORD OF THIS
468 ; BLOCK SAVE THE NUMBER OF WORDS TO BE TRANSFERRED TO
469 ; THE MRP
470 003660 MDNEXT: CALL GET ;READ A RECORD
471 003664 103002 BCC 1$
472 003666 000167 000150 JMP MDDX ;ERROR, EXIT
473 003672 012705 000322 1$ MOV *INLINE,R5 ;POINT TO RECORD READ
474 003676 032767 000001 174244 BIT *FIRST,SELECT ;FIRST TIME THROUGH
475 003704 001014 BNE 3$ ;NO
476 003706 052767 000001 174234 BIS *FIRST,SELECT ;SET FLAG FOR FIRST TIME THROUGH
477 003714 012567 174234 MOV (R5)+,LCOUNT ;GET NUMBER OF DATA WORDS
478 003720 012767 177777 174240 MOV *-1,TRANSF ;START TRANSFER COUNT AT -1
479 003726 026727 174222 000400 CMP LCOUNT,*256 ;256 WORDS TO TRANSFER
480 003734 001413 BEQ 4$ ;YES, 2 TRANSFERS NEEDED (COUNT WORD)
481 ;
482 ;
483 ; TRANSFER A FULL 256-WORD BLOCK IF POSSIBLE. IF TRANSFER
484 ; COUNT IS LESS THAN OR EQUAL TO 256, THIS IS THE LAST
485 ; TRANSFER
486 003736 026727 174212 000400 3$ CMP LCOUNT,*256 ;MORE THAN A FULL BLOCK LEFT TO TRANSFER
487 003744 003007 BGT 4$ ;YES, TRANSFER 256 WORDS THIS TIME
488 003746 052767 000004 174174 BIS *LAST,SELECT ;LAST TRANSFER
489 003754 066767 174174 174204 ADD LCOUNT,TRANSF ;TRANSFER REMAINING WORDS
490 003762 000406 BR 5$ ;LOAD CD BY DMA
491 003764 062767 000400 174174 4$ ADD *256,TRANSF ;NUMBER OF WORDS TO TRANSFER
492 003772 166767 174170 174154 SUB TRANSF,LCOUNT ;SUB FROM TOTAL
493 ;
494 ;
495 ; SIGNAL MICROCODE TO ACCEPT DATA FOR MRP
496 004000 010567 174164 5$ MOV R5,INSAVE ;SAVE POINTER TO INPUT DATA
497 004004 012746 000004 MOV *0$LDMD,-(SP) ;SELECT MRP DATA MEMORY
498 004010 CALL DMA ;LOAD MRP DATA MEMORY
499 004014 032767 000004 174126 BIT *LAST,SELECT ;FINISHED?
500 004022 001007 BNE MDDX ;YES
501 004024 066767 174136 174132 ADD TRANSF,MSTR2 ;SET CD ADDRESS FOR NEXT TRANSFER
502 004032 005067 174130 CLR TRANSF
503 004036 000167 177616 JMP MDNEXT ;GET NEXT RECORD
504 ;
505 ;
506 ; EXIT

```

507	004042			MDDX:			
508	004042	005046			CLR	-(SP)	;CLEAR-NOTHING-IN-CSR1
509	004044	012746	176000		MOV	#0\$NCLK, -(SP)	;SET-NO-CLOCKS
510	004050				CALL	CSR1	
511	004054	005067	176422		CLR	QR\$CR2	;SET-LOAD-MODE
512				:			
513	004060				CLOSE\$	#INFDB	
514	004070	105067	176542		CLRB	INDNB+N,FVER	;RESET-FILE-VERSION-NUMBER
515	004074	000167	176634		JMP	COM	
516				:			


```

518 ;
519 ;
520 QEX WINDOW MEMORY.
521 QEX LOCATION MEMORY.
522 ;
523 ;
524 004100 012767 000042 174034 QW: MOV #Q$QW, CODE ;SET MEMORY CODE = WINDOW
525 004106 016767 173726 176512 MOV LQW, INDNB+N, FNAM ;PLACE FILE NAME INTO INPUT DNB
526 004114 016767 173722 176506 MOV LQW+2, INDNB+N, FNAM+2
527 004122 000411 BR QEX
528 004124 012767 000043 174010 QL: MOV #Q$QL, CODE ;SET MEMORY CODE = LOCATION
529 004132 016767 173706 176466 MOV LQL, INDNB+N, FNAM ;PLACE FILE NAME INTO INPUT DNB
530 004140 016767 173702 176462 MOV LQL+2, INDNB+N, FNAM+2
531 004146 QEX:
532 004146 OPEN$R #INFDB
533 004164 005067 173774 CLR MSTR2 ;CD TRANSFER START ADDRESS
534 ;
535 ; GET THE FIRST RECORD (BLOCK). THE SECOND WORD OF THE FIRST
536 ; RECORD CONTAINS THE NUMBER OF ADDRESS/DATA PAIRS TO BE LOADED
537 ; INTO QEX MEMORY. SAVE THIS VALUE.
538 ;
539 004170 QXNEXT:
540 004170 CALL GET ;READ A RECORD
541 004174 103002 BCC 1$
542 004176 000167 000154 JMP QXX ;ERROR, EXIT
543 004202 1$:
544 004202 012705 000322 MOV #INLINE, R5 ;POINT TO RECORD READ
545 004206 010567 173756 MOV R5, INSAVE ;SET CC DMA BUFFER ADDRESS
546 004212 012725 000001 MOV #1, (R5)+ ;SET WRITE FLAG IN CD ADDR 0
547 004216 032767 000001 173724 BIT #FIRST, SELECT ;FIRST TIME THROUGH
548 004224 001014 BNE 3$ ;NO
549 004226 052767 000001 173714 BIS #FIRST, SELECT ;SET FLAG FOR FIRST TIME THROUGH
550 004234 012504 MOV (R5)+, R4 ;GET COUNT OF ADDRESS/DATA PAIRS
551 004236 001447 BEQ QXX ;EXIT IF NONE
552 004240 006304 ASL R4 ;CONVERT TO WORD COUNT
553 004242 062704 000002 ADD #2, R4 ;ADD SPACE FOR CODE AND COUNT
554 004246 010467 173702 MOV R4, LCOUNT ;SAVE NUMBER OF DATA WORDS
555 004252 CALL DMASET ;START UP DMA MICROCODE
556 ;
557 ; TRANSFER A FULL 256-WORD BLOCK IF POSSIBLE. IF TRANSFER
558 ; COUNT IS LESS THAN OR EQUAL TO 256, THIS IS THE LAST
559 ; TRANSFER.
560 ;
561 004256 026727 173672 000400 3$: CMP LCOUNT, #256 ;MORE THAN A FULL BLOCK LEFT TO TRANSFER
562 004264 003007 BGT 4$ ;YES, TRANSFER 256 WORDS THIS TIME
563 004266 052767 000004 173654 BIS #LAST, SELECT ;LAST TRANSFER
564 004274 016767 173654 173664 MOV LCOUNT, TRANSF ;TRANSFER REMAINING WORDS
565 004302 000406 BR 5$ ;LOAD CD BY DMA
566 004304 012767 000400 173654 4$: MOV #256, TRANSF ;NUMBER OF WORDS TO TRANSFER
567 004312 166767 173650 173634 SUB TRANSF, LCOUNT ;SUB FROM TOTAL
568 ;
569 ; SIGNAL MICROCODE TO ACCEPT CP DATA MEMORY DATA
570 ;
571 004320 5$:
572 004320 012746 000003 MOV #Q$LDCD, -(SP) ;SELECT CP DATA MEMORY
573 004324 CALL DMA ;LOAD CP DATA MEMORY
574 004330 032767 000004 173612 BIT #LAST, SELECT ;FINISHED?

```

575	004336	001005			BNE QXLD	:YES
576	004340	066767	173622	173616	ADD TRANSF,MSTR2	:SET CD ADDRESS FOR NEXT TRANSFER
577	004346	000167	177616		JMP QXNEXT	:GET NEXT RECORD
578					:	
579	004352				QXLD:	
580	004352				CALL LOADQX	:LOAD QEX MEMORY VIA MICROCODE
581					:	
582					EXIT	
583					:	
584	004356				QXX:	
585	004356	012746	000040		MOV #0\$CLR,-(SP)	:CLEAR PPS
586	004362				CALL PPCR	:WRITE TO CONTROL REGISTER
587					:	
588	004366				CLOSE\$ #INFDB	
589	004376	105067	176234		CLRB INDNB+N,FVER	:RESET FILE VERSION NUMBER
590	004402	000167	176326		JMP COM	

```

592.      ;
593      ;
594      ;      FAL POINTER MEMORY
595      ;      FAL COUNTER MEMORY
596      ;
597      ;
598 004406 012767 000046 173526 FP:  MOV  #0$FP, CODE      ; SET MEMORY CODE = POINTER
599 004414 016767 173430 176204  MOV  LFP, INDNB+N, FNAM  ; PLACE FILE NAME INTO INPUT DNB
600 004422 016767 173424 176200  MOV  LFP+2, INDNB+N, FNAM+2
601 004430 000411      BR      FAL
602 004432 012767 000045 173502 FC:  MOV  #0$FC, CODE      ; SET MEMORY SELECT CODE = COUNTER
603 004440 016767 173410 176160  MOV  LFC, INDNB+N, FNAM  ; PLACE FILE NAME INTO INPUT DNB
604 004446 016767 173404 176154  MOV  LFC+2, INDNB+N, FNAM+2
605      ;
606 004454 005067 173504      FAL:  CLR  ADDR      ; START ADDRESS AT ZERO
607 004460 005067 173476      CLR  DATA      ; CLEAR DATA MEMORY TO ZERO
608 004464      1$:  CALL  ONEFA      ; WRITE 1 WORD OF FAL POINTER MEMORY
609 004470 005267 173470      INC  ADDR      ; BUMP TO NEXT ADDRESS
610 004474 026767 174626 173462  CMP  FAHIGH, ADDR      ; CLEARED ALL OF MEMORY?
611 004502 103370      BHS  1$      ; NO, DO NEXT LOCATION
612      ;
613 004504      OPEN$R #INFDB
614      ;
615      2$:  CALL  GET      ; READ A RECORD
616 004526 103437      BCS  FAX      ; ERROR, EXIT
617 004530 012705 000322*  MOV  #INLINE, R5      ; POINT TO RECORD READ
618 004534 012704 000400      MOV  #256, R4      ; NUMBER OF WORDS IN RECORD (MAX)
619 004540 032767 000001 173402  BIT  #FIRST, SELECT      ; FIRST TIME THROUGH
620 004546 001012      BNE  3$      ; NO
621 004550 052767 000001 173372  BIS  #FIRST, SELECT      ; SET FLAG FOR FIRST TIME THROUGH
622 004556 016567 000002 173370  MOV  2(R5), LCOUNT      ; GET NUMBER OF DATA WORDS
623 004564 062705 000004      ADD  #4, R5      ; BUMP PAST COUNT
624 004570 162704 000002      SUB  #2, R4      ; SUB FROM TOTAL NUMBER OF WORDS IN RECORD
625      ;
626 004574 012567 173364      3$:  MOV  (R5)+, ADDR      ; GET MEMORY ADDR FROM RECORD
627 004600 012567 173356      MOV  (R5)+, DATA      ; GET DATA FROM RECORD
628 004604      CALL  ONEFA      ; LOAD ONE MEMORY LOCATION
629 004610 005367 173340      DEC  LCOUNT      ; SUB FROM OVERALL COUNT
630 004614 001404      BEQ  FAX      ; FINISHED
631 004616 162704 000002      SUB  #2, R4      ; SUB FROM TOTAL WORDS
632 004622 003364      BGT  3$      ; DO NEXT
633 004624 000736      BR   2$      ; READ NEXT RECORD
634      ;
635 004626 012746 077777      FAX:  MOV  #077777, -(SP)      ; HALT CODE
636 004632      CALL  STOP      ; LOAD OCL POINTER
637 004636 012746 000040      MOV  #0$CLR, -(SP)      ; CLEAR PPS
638 004642      CALL  PPCR
639      ;
640 004646      CLOSE$ #INFDB
641 004656 185067 175754      CLR  INDNB+N, FVER      ; RESET FILE VERSION NUMBER
642 004662 000167 176046      JMP  COM

```

```

644 ;
645 ;
646 ; QLB REFERENCE PAGE
647 ;
648 ;
649 004666 ; QR:
650 004666 005067 173272 CLR ADDR ; START ADDRESS AT ZERO
651 004672 005067 173264 CLR DATA ; CLEAR DATA MEMORY TO ZERO
652 004676 1% CALL ONEQR ; WRITE 1 WORD OF QLB REF MEMORY
653 004702 005267 173256 INC ADDR ; BUMP TO NEXT ADDRESS
654 004706 026767 174420 173250 CMP LHHIGH, ADDR ; CLEARED ALL OF MEMORY?
655 004714 103370 BHS 1% ; NO, DO NEXT LOCATION
656 ;
657 004716 016767 173136 175702 MOV LQR, INDNB+N, FNAM ; PLACE FILE NAME INTO INPUT DNB
658 004724 016767 173132 175676 MOV LQR+2, INDNB+N, FNAM+2
659 004732 OPEN#R ; #INFDB
660 ;
661 004750 2% CALL GET ; READ A RECORD
662 004754 103437 BCS QRX ; ERROR, EXIT
663 004756 012705 000322 MOV #INLINE, R5 ; POINT TO RECORD READ
664 004762 012704 000400 MOV #256, R4 ; NUMBER OF WORDS IN RECORD (MAX)
665 004766 032767 000001 173154 BIT #FIRST, SELECT ; FIRST TIME THROUGH
666 004774 001012 BNE 3% ; NO
667 004776 052767 000001 173144 BIS #FIRST, SELECT ; SET FLAG FOR FIRST TIME THROUGH
668 005004 016567 000002 173142 MOV 2(R5), LCOUNT ; GET NUMBER OF DATA WORDS
669 005012 062705 000004 ADD #4, R5 ; BUMP PAST COUNT
670 005016 162704 000002 SUB #2, R4 ; SUB FROM TOTAL NUMBER OF WORDS IN RECORD
671 ;
672 005022 012567 173136 3% MOV (R5)+, ADDR ; GET MEMORY ADDR FROM RECORD
673 005026 012567 173130 MOV (R5)+, DATA ; GET DATA FROM RECORD
674 005032 CALL ONEQR ; LOAD ONE MEMORY LOCATION
675 005036 005367 173112 DEC LCOUNT ; SUB FROM OVERALL COUNT
676 005042 001404 BEQ QRX ; FINISHED
677 005044 162704 000002 SUB #2, R4 ; SUB FROM TOTAL WORDS
678 005050 003364 BGT 3% ; DO NEXT
679 005052 000736 BR 2% ; READ NEXT RECORD
680 ;
681 005054 QRX:
682 005054 012746 000013 MOV #QHLT, -(SP) ; HALT CODE
683 005060 CALL PPCR
684 005064 012746 000040 MOV #QCLR, -(SP) ; CLEAR PPS
685 005070 CALL PPCR
686 ;
687 005074 CLOSE# #INFDB
688 005104 105067 175526 CLR# INDNB+N, FVER ; RESET FILE VERSION NUMBER
689 005110 000167 175620 JMP COM

```

```

691 ;
692 ; QLB PAGE 0
693 ; QLB PAGE 1
694 ; QLB PAGE 2
695 ;
696 ;
697 005114 ; 00:
698 005114 012746 000000 MOV #0,-(SP) ;SELECT PAGE 0
699 005120 CALL SELPG ;SELECT A QLB PAGE
700 005124 016767 172734 175474 MOV LQ0,INDNB+N.FNAM ;PLACE FILE NAME INTO INPUT DNB
701 005132 016767 172730 175470 MOV LQ0+2,INDNB+N.FNAM+2
702 005140 000425 BR QLB
703 005142 ;
704 005142 012746 000001 Q1: MOV #1,-(SP) ;SELECT PAGE 1
705 005146 CALL SELPG
706 005152 016767 172712 175446 MOV LQ1,INDNB+N.FNAM ;PLACE FILE NAME INTO INPUT DNB
707 005160 016767 172706 175442 MOV LQ1+2,INDNB+N.FNAM+2
708 005166 000412 BR QLB
709 005170 ;
710 005170 012746 000002 Q2: MOV #2,-(SP) ;SELECT QLB PAGE 2
711 005174 CALL SELPG
712 005200 016767 172670 175420 MOV LQ2,INDNB+N.FNAM ;PLACE FILE NAME INTO INPUT DNB
713 005206 016767 172664 175414 MOV LQ2+2,INDNB+N.FNAM+2
714 ;
715 005214 005067 172744 QLB: CLR ADDR ;START ADDRESS AT ZERO
716 005220 005067 172736 CLR DATA ;CLEAR DATA MEMORY TO ZERO
717 005224 1$: CALL ONEQ ;WRITE 1 WORD OF QLB PAGE 0 MEMORY
718 005230 005267 172730 INC ADDR ;BUMP TO NEXT ADDRESS
719 005234 026767 174072 172722 CMP LHHIGH,ADDR ;CLEARED ALL OF MEMORY?
720 005242 103370 BHS 1$ ;NO, DO NEXT LOCATION
721 ;
722 005244 ; OPEN$R #INFDB
723 ;
724 005262 2$: CALL GET ;READ A RECORD
725 005266 103437 BCS QBX ;ERROR, EXIT
726 005270 012705 000322 MOV #INLINE,R5 ;POINT TO RECORD READ
727 005274 012704 000400 MOV #256,R4 ;NUMBER OF WORDS IN RECORD (MAX)
728 005300 032767 000001 172642 BIT #FIRST,SELECT ;FIRST TIME THROUGH
729 005306 001012 BNE 3$ ;NO
730 005310 052767 000001 172632 BIS #FIRST,SELECT ;SET FLAG FOR FIRST TIME THROUGH
731 005316 016567 000002 172630 MOV 2(R5),LCOUNT ;GET NUMBER OF DATA WORDS
732 005324 052705 000004 ADD #4,R5 ;BUMP PAST COUNT
733 005330 162704 000002 SUB #2,R4 ;SUB FROM TOTAL NUMBER OF WORDS IN RECORD
734 ;
735 005334 012567 172624 3$: MOV (R5)+,ADDR ;GET MEMORY ADDR FROM RECORD
736 005340 012567 172616 MOV (R5)+,DATA ;GET DATA FROM RECORD
737 005344 CALL ONEQ ;LOAD ONE MEMORY LOCATION
738 005350 005367 172600 DEC LCOUNT ;SUB FROM OVERALL COUNT
739 005354 001404 BEQ QBX ;FINISHED
740 005356 162704 000002 SUB #2,R4 ;SUB FROM TOTAL WORDS
741 005362 003364 BGT 3$ ;DO NEXT
742 005364 000736 BR 2$ ;READ NEXT RECORD
743 ;
744 005366 QBX:
745 005366 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE
746 005372 PPCR ;CLEAR PPS
747 005376 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS

```

748 005402.
749
750 005406
751 005416 105067 175214
752 005422 000167 175306

GALL- PPCR
CLOSE\$ #INFDB
CLRB- INDNB+N.FVER :RESET FILE VERSION NUMBER
JMP- COM-

```

754 ; SUBDOCUMENT PROCESSOR MEMORIES
755 ;
756 005426 ; SPLOAD:
757 005426 005067 172526 CLR SCOUNT ; INITIALIZE PAGE INDEX
758 ;
759 ; CLEAR MEMORY
760 ;
761 005432 ; SPCLR:
762 005432 012746 000001 MOV #S$LA, -(SP) ; SET ADDRESS SELECT
763 005436 CALL SPCR
764 005442 016746 172506 MOV LCOUNT, -(SP) ; SET ADDRESS
765 005446 CALL LBSP
766 005452 ;
767 005452 016746 172464 10$: MOV CODE, -(SP) ; SET MEMORY SELECT CODE
768 005456 066716 172476 ADD SCOUNT, @SP ; ADD PAGE INDEX IF APPLICABLE
769 005462 CALL SPCR
770 005466 005046 CLR -(SP) ; ZERO IS MEMORY VALUE
771 005470 CALL LBSP
772 005474 005267 172460 INC SCOUNT ; BUMP PAGE INDEX
773 005500 026767 172454 172450 CMP SCOUNT, WCOUNT ; ALL WORDS AT THIS ADDRESS CLEARED?
774 005506 002761 BLT 10$ ; BRANCH IF NOT
775 005510 005067 172444 CLR SCOUNT ; RESET PAGE INDEX
776 005514 005367 172434 DEC LCOUNT ; NEXT LOWER ADDRESS
777 005520 002344 BGE SPCLR ; ZERO IT
778 ;
779 005522 ; OPEN$R: #INFDB ; OPEN INPUT FILE
780 005540 016767 172412 172412 MOV WCOUNT, SCOUNT ; INITIALIZE SCOUNT
781 ;
782 ; GET NEXT INPUT RECORD
783 ;
784 005546 ; SPNEXT:
785 005546 CALL GET ; READ A RECORD
786 005552 103460 BCS SPX ; ERROR, EXIT
787 005554 012705 000322 MOV #INLINE, R5 ; POINT TO RECORD READ
788 005560 012704 000400 MOV #256, R4 ; NUMBER OF WORDS IN RECORD (MAX)
789 005564 032767 000001 172356 BIT #FIRST, SELECT ; FIRST TIME THROUGH?
790 005572 B0H SPLOOP ; NO
791 005574 052767 000001 172346 BIS #FIRST, SELECT ; SET FLAG FOR FIRST TIME THROUGH
792 005602 016567 000002 172344 MOV 2(R5), LCOUNT ; GET NUMBER OF DATA WORDS
793 005610 062705 000004 ADD #4, R5 ; BUMP PAST COUNT
794 005614 162704 000002 SUB #2, R4 ; SUBTRACT FROM RECORD WORD COUNT
795 ;
796 ; LOOP THROUGH RECORD AND EXTRACT ADDRESS AND DATA
797 ; WCOUNT CONTAINS NUMBER OF DATA WORDS PER ADDRESS VALUE
798 ; FIRST TIME THROUGH, SCOUNT IS INITIALIZED TO WCOUNT
799 ;
800 005620 ; SPLOOP:
801 005620 026767 172334 172330 CMP SCOUNT, WCOUNT ; ADDRESS OR DATA NEXT?
802 005626 001014 BNE 10$ ; BRANCH IF DATA
803 005630 005767 172320 TST LCOUNT ; ALL ADDRESSES BEEN LOADED?
804 005634 003427 BLE SPX ; BRANCH IF SO
805 005636 012746 000001 MOV #S$LA, -(SP) ; SET ADDRESS SELECT
806 005642 CALL SPCR
807 005646 005067 172306 CLR SCOUNT ; RESET WORD INDEX
808 005652 005367 172276 DEC LCOUNT ; BUMP ITEM COUNT IN CURRENT RECORD
809 005656 000410 BR 20$ ; GO TO ESTABLISH ADDRESS
810 005660

```

811	005660	016746	172256		MOV	CODE,-(SP)		;SET MEMORY SELECT CODE
812	005664	066716	172270		ADD	SCOUNT,@SP		;ADD PAGE INDEX IF APPLICABLE
813	005670				CALL	SPCR		
814	005674	005267	172260		INC	SCOUNT		;SET NEXT DATA WORD INDEX
815	005700			20\$:				
816	005700	012546			MOV	(R5)+,-(SP)		;FETCH WORD FROM INPUT RECORD
817	005702				CALL	LBSP		
818	005706	005304			DEC	R4		;DECREMENT RECORD WORD COUNT
819	005710	003343			BGT	SPLOOP		;PROCESS NEXT WORD IN RECORD
820	005712	000715			BR	SPNEXT		;READ NEXT RECORD
821	005714			SPX:				
822	005714				CLOSE\$	#INFDB		
823	005724	105067	174706		CLRB	INDNB+N,FVER		;RESET FILE VERSION NUMBER
824	005730				RETURN			


```
826 ;
827 ;
828 ; EXIT PROGRAM
829 ;
830 ;
831 005732 ; EXIT:
832 005732 016737 172050 000274 MOV OLDVEC,@#274 ;RESTORE ORIGINAL VECTOR CONTENTS
833 005740 CALL ENDTST ;PUT OUT EXIT MESSAGE
834 005744 SETF$S #EFN.33 ;SET GLOBAL EVENT FLAG
835 ;
836 005756 EXIT$S
```

```
838 ;
839 ;
840 ;
841 ; INTERRUPT SERVICE ROUTINE
842 ; TRAP INTERRUPTS FROM HQR THROUGH VECTOR ADDRESS 274
843 ; SET EVENT FLAG 3
844 ; GP DEBUGGING ROUTINES WILL READ CSR #2 AND DECODE THE INTERRUPT
845 ;
846 005764 BPTISR:
847 005764 ;
848 ;
849 006000 016705 172000 ; SAVE R0,R1,R2,R3,R4,R5
850 006004 012700 000003 ; MOV TSKTCB,R5 ;LOAD MY TCB
851 006010 ; MOV #EFN.3,R0 ;EVENT FLAG TO BE SET
852 006014 050011 ; CALL $CEFI
853 006016 ; BIS R0,(R1) ;SET LOCAL FLAG
854 ; CALL $DRDSE ;DECLARE SIGNIFICANT EVENT
855 006022 ;
856 006036 000002 ; RESTOR R0,R1,R2,R3,R4,R5
; RTI
```

```

858 ;
859 ;
860 ; LOAD ONE WORD OF FAL POINTER OR COUNTER MEMORY.
861 ;
862 006040 ; ONEFA:
863 006040 016746 172120 MOV. ADDR.-(SP) ;LOAD ADDR INTO QCL POINTER.
864 006044 CALL. STQP
865 006050 016746 172066 MOV. CODE.-(SP) ;SELECT MEMORY.
866 006054 CALL. PPCR ;WRITE SELECTION TO CONTROL REG.
867 006060 016746 172076 MOV. DATA.-(SP) ;SEND DATA WORD TO FAL MEMORY.
868 006064 CALL. LBPP
869 006070 012746 000040 MOV. #Q$CLR.-(SP)
870 006074 CALL. PPCR
871 006100 RETURN.
872 ;
873 ;
874 ; LOAD ONE WORD OF QLB REFERENCE PAGE.
875 ;
876 ;
877 006102 ; ONEQR:
878 006102 012746 000053 MOV. #Q$QLA.-(SP) ;ADDRESS SELECT FOR QLB PAGE.
879 006106 CALL. PPCR ;SEND TO PP CONTROL REG.
880 006112 016746 172046 MOV. ADDR.-(SP) ;ACTUAL ADDRESS.
881 006116 CALL. LBPP ;SEND TO PP.
882 006122 012746 000001 MOV. #Q$QLR.-(SP) ;SELECT QLB REF MEMORY.
883 006126 CALL. PPCR
884 006132 016746 172024 MOV. DATA.-(SP) ;DATA WORD FOR MEMORY.
885 006136 CALL. LBPP ;SEND DATA TO PPS.
886 006142 RETURN.
887 ;
888 ;
889 ; LOAD ONE WORD INTO QLB PAGES.
890 ;
891 ;
892 006144 ; ONEQ:
893 006144 012746 000053 MOV. #Q$QLA.-(SP) ;ADDRESS SELECT FOR QLB PAGE.
894 006150 CALL. PPCR ;SEND TO PP CONTROL REG.
895 006154 016746 172004 MOV. ADDR.-(SP) ;ACTUAL ADDRESS.
896 006160 CALL. LBPP ;SEND TO PP.
897 006164 012746 000054 MOV. #Q$QLB.-(SP) ;SELECT QLB REF MEMORY.
898 006170 CALL. PPCR
899 006174 016746 171762 MOV. DATA.-(SP) ;DATA WORD FOR MEMORY.
900 006200 CALL. LBPP ;SEND DATA TO PPS.
901 006204 RETURN.

```

```

903 ;
904 ;
905 READ A RECORD (BLOCK)
906 ;
907 FILE NAME BLOCK PRE-INITIALIZED
908 ;
909 OUTPUT:
910 C-BIT CLEAR - GOOD READ
911 C-BIT SET - ERROR ON READ
912 ;
913 ;
914 006206 GET::
915 006206 READ$ #INFDB,,,#VIRT,#EFN,1,#STAT
916 006254 103005 BCC 1$
917 006256 CALL ERR5
918 006262 CALL ERNAME ;TELL WHICH FILE WAS IN ERROR
919 006266 000421 BR GETSX
920 ;
921 006270 1$: WTSE$S #EFN,1
922 ;
923 006302 CLEF$S #EFN,1
924 006314 105767 171614 TSTB STAT
925 006320 003006 BGT GETCX ;GOOD COMPLETION
926 006322 CALL ERR5
927 006326 CALL ERNAME ;TELL WHICH FILE WAS IN ERROR
928 ;
929 006332 000261 GETSX: SEC
930 006334 000403 BR GETX
931 006336 005267 171570 GETCX: INC VIRT+2 ;INC BLOCK COUNTER
932 006342 000241 CLC
933 006344 GETX: RETURN

```



```

955 ;
956 ;
957 ; SCAN A TABLE FOR A VALID COMMAND/MNEMONIC.
958 ;
959 ; INPUT:
960 R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
961 R1 -> CHAR STRING IN GCMC COMMAND LINE.
962 R2 -> TOP OF COMMAND TABLE.
963 ;
964 ; OUTPUT:
965 R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
966 R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
967 R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
968 ;
969 ;
970 SCAN:
971 MOV R3, -(SP) ;SAVE R3
972 MOV R0, -(SP) ;SAVE # ENTRIES
973 MOV R1, -(SP) ;SAVE POINTER TO BEGINNING OF STRING
974 ;
975 FNOUT1: MOV (SP), R1 ;POINT TO NON-BLANK IN COMMAND LINE
976 MOV #2, R3 ;NUMBER OF CHARS IN NON-BLANK FIELD
977 FNIN1: CMPB (R1)+, (R2)+ ;DOES COMMAND LINE MATCH TABLE ENTRY
978 BNE FNOUT2 ;NO, TRY NEXT TABLE ENTRY
979 DEC R3 ;SUB FROM LOOP COUNT
980 BNE FNIN1
981 BR FNMTCH ;COMMAND FOUND IN TABLE
982 FNOUT2: ADD R3, R2 ;ADD # UNCOMPARED CHARS TO POINTER
983 INC R2 ;THEN ADJUST TO NEXT TABLE ENTRY
984 DEC R0 ;SUB FROM OUTER LOOP COUNT
985 BNE FNOUT1 ;TRY AGAIN
986 MOV (SP)+, R1 ;RESTORE POINTER TO COMMAND LINE
987 MOV (SP)+, R0 ;RELOAD R0
988 MOV (SP)+, R3 ;RESTORE R3
989 SEC ;COMMAND NOT IN TABLE
990 RETURN
991 ;
992 FNMTCH: MOV R2, R1 ;POINT R1 AT RTN ADDR IN TABLE
993 ADD #2, SP ;POINT TO INCOMING R0 ON STACK
994 MOV (SP)+, R2 ;GET TOTAL # TABLE ENTRIES
995 SUB R0, R2 ;GET POSITION OF MATCHED ENTRY
996 MOV R2, R0 ;PUT IN R0 FOR RETURN
997 MOV (SP)+, R3 ;RESTORE R3
998 CLC
999 RETURN

```

```

1001 ;
1002 ;
1003 ; FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER.
1004 ; THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
1005 ;
1006 ; INPUT:
1007 ; GCMLN - NUMBER OF UNPROCESSED BYTES IN COMMAND LINE.
1008 ; GCMPT - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE.
1009 ;
1010 ; OUTPUT:
1011 ; R1 -> STRING, R0 = LENGTH OF STRING.
1012 ; GCMLN, GCMPT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE.
1013 ;
1014 ; THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
1015 ; IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLN AND
1016 ; GCMPT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
1017 ; (SEE THE MESSAGE PRINTING/PROMPTING ROUTINES).
1018 ;
1019 ;
1020 006500 FIND:
1021 006500 010246 MOV R2, -(SP) ; SAVE R2.
1022 006502 016701 171610 MOV GCMLN, R1 ; # BYTES REMAINING IN COMMAND BUFFER.
1023 006506 001440 BEQ FSECC ; THERE ARE NONE.
1024 006510 016702 171604 MOV GCMPT, R2 ; LOAD CURRENT POINTER.
1025 006514 122712 000040 1$: CMPB #40, (R2) ; LOOK FOR A BLANK.
1026 006520 001403 BEQ 10$ ; OK, BUMP TO NEXT CHAR.
1027 006522 122712 000054 CMPB #'', (R2) ; COMMA IN COMMAND LINE.
1028 006526 001004 BNE 2$ ; TREAT COMMA AS BLANK.
1029 006530 005202 10$: INC R2 ; BUMP POINTER.
1030 006532 005301 DEC R1 ; SUB FROM REMAINING LENGTH.
1031 006534 001367 BNE 1$
1032 006536 000424 BR FSECC ; NO NON-BLANK FOUND.
1033 ;
1034 006540 010246 2$: MOV R2, -(SP) ; TEMP SAVE POINTER TO BEGINNING OF STRING.
1035 006542 005000 CLR R0 ; CLEAR CHAR COUNT.
1036 006544 122712 000040 3$: CMPB #40, (R2) ; LOOK FOR A BLANK.
1037 006550 001407 BEQ 4$ ; FOUND END OF STRING.
1038 006552 122712 000054 CMPB #'', (R2) ; TREAT COMMAS AS BLANKS.
1039 006556 001404 BEQ 4$
1040 006560 005202 INC R2 ; BUMP POINTER.
1041 006562 005200 INC R0 ; BUMP CHAR COUNT.
1042 006564 005301 DEC R1 ; SUB FROM BYTES REMAINING.
1043 006566 001366 BNE 3$
1044 ;
1045 006570 010267 171524 4$: MOV R2, GCMPT ; SAVE POINTER FOR NEXT TIME.
1046 006574 010167 171516 MOV R1, GCMLN ; SAVE BYTES REMAINING FOR NEXT TIME.
1047 006600 012601 MOV (SP)+, R1 ; POINTER TO BEGINNING OF STRING.
1048 006602 012602 MOV (SP)+, R2 ; RESTORE R2.
1049 006604 000241 CLC
1050 006606 RETURN.
1051 ;
1052 006610 012602 FSECC: MOV (SP)+, R2 ; RESTORE R2.
1053 006612 000261 SEC
1054 006614 RETURN.

```

```

1056 ;
1057 ;
1058 ; CONVERT AN OCTAL ASCII VALUE FROM THE COMMAND LINE.
1059 ;
1060 ; INPUT:
1061 ; R0 = ASCII OCTAL STRING CHARACTER COUNT.
1062 ; R1 -> ASCII OCTAL STRING.
1063 ;
1064 ; OUTPUT:
1065 ; BINWD CONVERTED VALUE.
1066 ; C-BIT CLEAR GOOD CONVERSION.
1067 ; C-BIT SET ERROR ON CONVERSION.
1068 ;
1069 ; R0, R1 DESTROYED.
1070 ;
1071 ;
1072 006616 ; PACK0:
1073 006616 022700 000006 CMP #6,R0 ;UPPER LIMIT ON OCTAL DIGITS.
1074 006622 002414 BLT PSEXC ;TOO MANY.
1075 006624 010146 MOV R1, -(SP) ;SAVE STRING POINTER TEMPORARILY.
1076 006626 060016 ADD R0, (SP) ;ADD CHAR COUNT.
1077 006630 010100 MOV R1,R0 ;GET STRING ADDR INTO R0 FOR SUBRTN.
1078 006632 CALL $COTB ;CONVERT ASCII OCTAL
1079 006636 005300 DEC R0 ;SUBRTN PUSHES R0 1 TOO FAR.
1080 006640 020026 CMP R0, (SP)+ ;FULL STRING CONVERTED.
1081 006642 001004 BNE PSEXC ;NO ERROR
1082 006644 010167 171276 MOV R1, BINWD ;SAVE CONVERTED VALUE.
1083 ;
1084 006650 000241 PCLCX: CLC
1085 006652 RETURN
1086 006654 000261 PSEXC: SEC
1087 006656 RETURN

```



```

1089 ;
1090 ;
1091 ;
1092 ; LOAD QCL POINTER
1093 ;
1094 ;
1095 006660 ; STOP;
1096 006660 016667 000002 176424 MOV 2(SP),QR#LBR ; MOVE POINTER WORD TO LOD BUS REG
1097 006666 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ; CLEAR DRIVE AND PULSE
1098 006672 052716 000360 BIS #Q$CSEL,(SP) ; CLEAR SELECTION BITS
1099 006676 012746 176000 MOV #Q$NCLK,-(SP) ; SET NO-CLOCKS
1100 006702 ; CALL CSR1
1101 ;
1102 006706 005046 CLR -(SP) ; CLEAR NOTHING
1103 006710 012746 001300 MOV #<Q$PP2+Q$LBD>,-(SP) ; SELECT PPS AND SET DRIVE
1104 006714 ; CALL CSR1
1105 ;
1106 ; SET FAL LOAD
1107 ;
1108 006720 012767 004800 176422 MOV #Q$FAL,QR#CR2 ; SET FAL LOAD
1109 ;
1110 ; EXTRA CLOCK FOR PPS
1111 ;
1112 006726 012746 000001 MOV #Q$LBP,-(SP) ; CLEAR PULSE
1113 006732 052716 000360 BIS #Q$RNC,(SP) ; CLEAR PPS NO-CLOCK
1114 006736 005046 CLR -(SP) ; SET NOTHING
1115 006740 ; CALL CSR1
1116 ;
1117 ; TURN OFF FAL LOAD
1118 ;
1119 006744 005067 176422 CLR QR#CR2
1120 ;
1121 ; DE-SELECTION
1122 ;
1123 006750 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ; CLEAR DRIVE AND PULSE
1124 006754 052716 000360 BIS #Q$CSEL,(SP) ; CLEAR SELECTION BITS
1125 006760 012746 176000 MOV #Q$NCLK,-(SP) ; SET NO-CLOCKS
1126 006764 ; CALL CSR1
1127 ;
1128 006770 011666 000002 MOV (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
1129 006774 005726 TST (SP)+ ; POINT TO RETURN ADDRESS
1130 006776 RETURN

```



```

1167 ;
1168 ;
1169 ; WRITE TO TT0 AND PROMPT.
1170 ;
1171 ;
1172 007206 005267 170732 ENDTST: INC ERWORD.
1173 007212 005267 170726 OUT1: INC ERWORD.
1174 007216 005267 170722 ERR6: INC ERWORD.
1175 007222 005267 170716 ERR5: INC ERWORD.
1176 007226 005267 170712 ERR3: INC ERWORD.
1177 007232 005267 170706 ERR2: INC ERWORD.
1178 007236 005267 170702 ERR1: INC ERWORD.
1179 ;
1180 007242 NESTOP:
1181 007242 005267 170676 SELMEM: INC ERWORD.
1182 000001 NEST: = <.-NESTOP>/4
1183 ;
1184 ; USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM
1185 ; OF THE MESSAGE TABLE. FIND THE END OF THE MESSAGE
1186 ; FIRST, THEN THE BEGINNING, THEN GET THE LENGTH.
1187 ;
1188 007246 016702 170672 MOV ERWORD,R2 ;LOAD LOOP COUNT
1189 007252 012701 002150* MOV #ASCIZ,R1 ;POINT TO END OF MESSAGE TABLE
1190 007256 105741 1$: TSTB -(R1) ;LOOK FOR END OF MESSAGE
1191 007260 001376 BNE 1$
1192 007262 005302 DEC R2 ;LOOP COUNT
1193 007264 001374 BNE 1$ ;BACK UP ANOTHER MESSAGE
1194 007266 010100 MOV R1,R0 ;SAVE POINTER TO END OF MESSAGE
1195 007270 105741 2$: TSTB -(R1) ;BACK UP TO BEGINNING OF MESSAGE
1196 007272 001376 BNE 2$
1197 007274 005201 INC R1 ;BUMP TO FIRST CHAR OF MESSAGE
1198 007276 160100 SUB R1,R0 ;R0 NOW = MESSAGE LENGTH
1199 ;
1200 007300 Q10W$S #10,WVB,#LUN,TT,#EFN,1, #STAT, <R1,R0>,ABEND
1201 ;
1202 007354 CLEF$S #EFN,1
1203 007366 105767 170542 TSTB STAT ;GOOD RETURN
1204 007372 003424 BLE ABEND ;NO
1205 ;
1206 ; ISSUE GCML
1207 ;
1208 007374 022767 000001 170542 CMP #NEST,ERWORD ;PROMPT WITH MESSAGE
1209 007402 002415 BLT TTX ;NO, JUST EXIT
1210 007404 GCML$ #GCMLBK
1211 007420 103411 BCS ABEND
1212 007422 016067 000146 170666 MOV G,CMLD(R0),GCMLN ;SAVE LENGTH
1213 007430 012767 000176* 170662 MOV #GCMBUF,GCMPNT ;INITIALIZE COMMAND BUFFER POINTER
1214 007436 005067 170502 CLR ERWORD ;CLEAR ERROR NUMBER INDICATOR
1215 007442 TTX: RETURN ;AND RETURN
1216 ;
1217 007444 ABEND: ABRT$S #MYSELF
1218 002656* .END START

```

ABEND 007444R.	BYTE3 = 000003	BYTE81 = 000121	ERNAME 007000R.	F.MBCT = 000054
ABEND2 007154RG.	BYTE30 = 000036	BYTE82 = 000122	ERR1 007236R.	F.MBC1 = 000055
ADDR 000164RG.	BYTE31 = 000037	BYTE83 = 000123	ERR2 007232RG.	F.MBFG = 000056
AL 003114R.	BYTE32 = 000040	BYTE84 = 000124	ERR3 007226R.	F.NRBD = 000024
ALL = 000002.	BYTE33 = 000041	BYTE85 = 000125	ERR5 007222R.	F.NREC = 000030
ALLCT 000174R.	BYTE34 = 000042	BYTE86 = 000126	ERR6 007216R.	F.OVBS = 000030
ALLNUM = 000014	BYTE35 = 000043	BYTE87 = 000127	ERWORD 000144R.	F.RACC = 000016
ALLPT 000172R.	BYTE36 = 000044	BYTE88 = 000130	EXIT 005732R.	F.RATT = 000001
ALLTBL 001472R.	BYTE37 = 000045	BYTE89 = 000131	FAHIGH 001326R.	F.RCNM = 000034
ALL2 003136R.	BYTE38 = 000046	BYTE9 = 000011	FAL 004454R.	F.RCTL = 000017
ALUCKE = 040000	BYTE39 = 000047	BYTE90 = 000132	FALOW 001330R.	F.RSIZ = 000002
ALUOE = 004000	BYTE4 = 000004	BYTE91 = 000133	FAX 004626R.	F.RTYP = 000000
APLACE 000152RG.	BYTE40 = 000050	BYTE92 = 000134	FC 004432R.	F.SEQN = 000100
ASCIZ 002150R.	BYTE41 = 000051	BYTE93 = 000135	FD.CCL = ***** GX.	F.SPDV = 000072
A01 = 010000	BYTE42 = 000052	BYTE94 = 000136	FD.REC = ***** GX.	F.SPUN = 000074
BINWD 000146RG.	BYTE43 = 000053	BYTE95 = 000137	FD.RUM = ***** GX.	F.STBK = 000036
BITVAL = 000000	BYTE44 = 000054	BYTE96 = 000140	FD.TTY = ***** GX.	F.UNIT = 000136
BIT0 = 000001	BYTE45 = 000055	BYTE97 = 000141	FIND 006500R.	F.URBD = 000020
BIT1 = 000002	BYTE46 = 000056	BYTE98 = 000142	FIRST = 000001 G.	F.VBN = 000064
BIT10 = 002000	BYTE47 = 000057	BYTE99 = 000143	FNIN1 006422R.	F.VBSZ = 000060
BIT11 = 004000	BYTE48 = 000060	BYTVAL = 000144	FNMTCH 006456R.	GCMBLK 002152R.
BIT12 = 010000	BYTE49 = 000061	CBKALL = 001000	FNOUT1 006414R.	GCMBUF 000176R.
BIT13 = 020000	BYTE5 = 000005	CBKCLK = 000400	FNOUT2 006434R.	GCMLEN 000316R.
BIT14 = 040000	BYTE50 = 000062	CD = ***** GX.	FNUM = 000024	GCMPTNT 000320R.
BIT15 = 100000	BYTE51 = 000063	CMILUN = 000002	FO.RD = ***** GX.	GET 006206RG.
BIT2 = 000004	BYTE52 = 000064	CNOBRE = 100000	FP 004406R.	GETCX 006336R.
BIT3 = 000010	BYTE53 = 000065	CODE 000142RG.	FSECX 006610R.	GETSX 006332R.
BIT4 = 000020	BYTE54 = 000066	COM 002734R.	FTBL 001352R.	GETX 006344R.
BIT5 = 000040	BYTE55 = 000067	CPCCEN = 010000	FVER 000140R.	GE.BIF = 177775
BIT6 = 000100	BYTE56 = 000070	CPCDC 003242R.	F.ACTL = 000076	GE.CLO = 000004
BIT7 = 000200	BYTE57 = 000071	CPCS = ***** GX.	F.ALOC = 000040	GE.COM = 000001
DIT0 = 000400	BYTE58 = 000072	CPCSC 003232R.	F.BBFS = 000062	GE.CON = 000020
DIT9 = 001000	BYTE59 = 000073	CPREAD = 040000	F.BDB = 000070	GE.EOF = 177765
JPTISR 005764R.	BYTE6 = 000006	CPURTE = 020000	F.BGBC = 000057	GE.IND = 000002
BYTE0 = 000000	BYTE60 = 000074	CSADR = 000004	F.BKDN = 000026	GE.IOR = 177777
BYTE1 = 000001	BYTE61 = 000075	CSEQCI = 100000	F.BKDS = 000020	GE.LC = 000010
BYTE10 = 000012	BYTE62 = 000076	CSOE = 000040	F.BKEF = 000050	F.BKEF = 177774
BYTE11 = 000013	BYTE63 = 000077	CSR1 006346RG.	F.BKPI = 000051	GE.OPR = 177776
BYTE12 = 000014	BYTE64 = 000100	CSURTE = 000100	F.BKST = 000024	GE.RBG = 177730
BYTE13 = 000015	BYTE65 = 000101	DATA 000162RG.	F.BKVB = 000064	GE.SIZ = 000040
BYTE14 = 000016	BYTE66 = 000102	DATA1 000162RG.	F.CHR = 000075	G.CMLD = 000146
BYTE15 = 000017	BYTE67 = 000103	DBR.RD = 000001	F.CNTG = 000034	G.DPRM = 000160
BYTE16 = 000020	BYTE68 = 000104	DB*CPP = 001457	F.DFNB = 000046	G.ERR = 000140
BYTE17 = 000021	BYTE69 = 000105	DB*SP = 000026	F.DSPT = 000044	G.ISIZ = 000020
BYTE18 = 000022	BYTE7 = 000007	DB*TPC = 000023	F.DVNM = 000134	G.LPDL = 000060
BYTE19 = 000023	BYTE70 = 000106	DISPSC = 100000	F.EFBK = 000010	G.MODE = 000141
BYTE2 = 000002	BYTE71 = 000107	DMA = ***** GX.	F.EFN = 000050	G.PSDS = 000142
BYTE20 = 000024	BYTE72 = 000110	DMAAWR = 000005	F.EOBB = 000032	G.SIZE = 000224
BYTE21 = 000025	BYTE73 = 000111	DMARRD = 000003	F.ERR = 000052	INDNB 002620RG.
BYTE22 = 000026	BYTE74 = 000112	DMARWR = 000004	F.FACC = 000043	INFDB 002460RG.
BYTE23 = 000027	BYTE75 = 000113	DMASET = ***** GX.	F.FFBY = 000014	INLINE 000322R.
BYTE24 = 000030	BYTE76 = 000114	EFBUF 000010RG.	F.FNAM = 000110	INLUN = 000003
BYTE25 = 000031	BYTE77 = 000115	EFN.1 = 000001	F.FNB = 000102	INSAVE 000170RG.
BYTE26 = 000032	BYTE78 = 000116	EFN.3 = 000003 G.	F.FTYP = 000116	IO.LVBS = ***** GX.
BYTE27 = 000033	BYTE79 = 000117	EFN.33 = 000041 G.	F.FVER = 000120	LAST = 000004 G.
BYTE28 = 000034	BYTE8 = 000008	ENBR = 010000	F.HIBK = 000004	LBPP = ***** GX.
BYTE29 = 000035	BYTE80 = 000120	ENDTST 007206R.	F.LUN = 000003	LBSP = ***** GX.

LCD	=	000034RG	N.FVER	=	000016	Q#HBF	=	000002	SPX	=	005714R	T#FSAB	=	000004
LCOUNT	=	000154RG	N.NEXT	=	000022	Q#ICP	=	000006	SQHIG	=	001336R	T#FSAC	=	000014
LCS	=	000030RG	N.STAT	=	000020	Q#IHB	=	000003	SOLOW	=	001340R	T#FSB2	=	000010
LFC	=	000054R	N.UNIT	=	000034	Q#IHL	=	000002	SR	=	003434R	T#IB	=	000026
LFP	=	000050R	OLDVEC	=	000006R	Q#IMRP	=	000007	SRHIGH	=	001342R	T#IBAR	=	000024
LHHIGH	=	001332R	ONEFA	=	006040R	Q#LBD	=	001000	SRLow	=	001344R	T#IBE	=	020000
LHLOW	=	001334R	ONEQ	=	006144R	Q#LBDP	=	001001	START	=	002656R	T#IBF	=	040000
LMD	=	000024RG	ONEQR	=	006102R	Q#LBP	=	000001	STAT	=	000134R	T#ICD	=	000040
LMM	=	000020RG	OUT1	=	007212R	Q#LDCD	=	000003	STOP	=	006660R	T#MODE	=	004000
LOADQX	=	***** GX	PACKO	=	006616R	Q#LDMD	=	000004	S#CLR	=	000000	T#OB	=	000036
LOC.EN	=	000100	PAR##	=	000027	Q#LDPP	=	002000	S#LA	=	000001	T#OBE	=	004000
LOC.WA	=	040000	PCLCX	=	006650R	Q#LHP	=	010000	S#OB	=	000005	T#OBF	=	010000
LOC.WB	=	100000	PLB	=	000010	Q#MNC	=	140000	S#OR	=	000006	T#OBRA	=	000034
LQL	=	000044R	PLC	=	000020	Q#MR	=	000052	S#OX	=	000004	T#OBWA	=	000032
LQR	=	000060R	PLD	=	000030	Q#MRP	=	000040	S#SR	=	000007	T#OUTA	=	100000
LQW	=	000040R	PLRWR	=	000200	Q#MRP2	=	000240	S#S1	=	000010	T#RBD0	=	000200
LQX	=	000124R	PLR.EN	=	000200	Q#MSC	=	040000	S#S2	=	000014	T#RNB	=	000040
LQ0	=	000064R	PPCR	=	***** GX	Q#MSET	=	000004	S.BFHD	=	000020	T#RSET	=	040000
LQ1	=	000070R	PRINT	=	001524RG	Q#MSP	=	100000	S.FATT	=	000016	T#RSC	=	000022
LQ2	=	000074R	PSECX	=	006654R	Q#NCLK	=	176000	S.FDB	=	000140	T#SCLK	=	020000
LSF	=	000100R	QBX	=	005366R	Q#PP	=	000100	S.FNAM	=	000006	T#SEG1	=	000000
LSR	=	000120R	QEX	=	004146R	Q#PPSW	=	000320	S.FNB	=	000036	T#SEG2	=	000001
LS0	=	000104R	QL	=	004124R	Q#PP2	=	000300	S.FNBW	=	000017	T#SEG3	=	000002
LS1	=	000110R	QLB	=	005214R	Q#QHLT	=	000013	S.FNTY	=	000004	T#SO	=	000001
LS2	=	000114R	QR	=	004666R	Q#QL	=	000043	S.FTYP	=	000002	T#UBUS	=	100000
LUN.TT	=	000001	QRX	=	005054R	Q#QLA	=	000053	S.NFEN	=	000020	T#ICLK	=	000400
MAREN1	=	000001	QR#CR1	=	176420	Q#QLB	=	000054	S0	=	003366R	T#8BEN	=	000020
MAREN2	=	004000	QR#CR2	=	176422	Q#QLR	=	000001	S1	=	003502R	UBD.IN	=	000020
MARLOD	=	010000	QR#LBR	=	176424	Q#QW	=	000042	S2	=	003500R	VIRT	=	000130RG
MAROUT	=	000002	QW	=	004100R	Q#RDCD	=	000005	TD#CTR	=	176370	WCOUNT	=	000156RG
MAR.LO	=	002000	QX	=	003252R	Q#RDMD	=	000006	TD#CTW	=	176360	WORD0	=	000000
MAR.OU	=	000040	QXHIGH	=	001322RG	Q#REBK	=	001000	TD#INL	=	004000	WORD1	=	000002
MBKALL	=	001000	QXLD	=	004352R	Q#RNC	=	006000	TD#MEM	=	000270	WORD10	=	000024
MBKCLK	=	000400	QXLOW	=	001324R	Q#RSC	=	004000	TD#0AR	=	176344	WORD11	=	000026
MDDX	=	004042R	QXNEXT	=	004170R	Q#RSET	=	000010	TD#0TR	=	176346	WORD12	=	000030
MDNEXT	=	003660R	QXX	=	004356R	Q#SM	=	100000	TD#QRD	=	000274	WORD13	=	000032
MMADR	=	000100	Q#ATTN	=	000100	Q#SP	=	000120	TD#SW	=	176376	WORD14	=	000034
MMLEFT	=	000002	Q#BCL	=	000001	Q#SP2	=	000340	TD#STAR	=	176372	WORD15	=	000036
MMOE	=	000004	Q#CCCP	=	000040	Q0	=	005114R	TD#TAW	=	176362	WORD16	=	000040
MMWRTE	=	000010	Q#CHB	=	000400	Q1	=	005142R	TD#TDR	=	176374	WORD17	=	000042
MNOBRE	=	100000	Q#CHRL	=	000200	Q2	=	005170R	TD#TDU	=	176364	WORD18	=	000044
MREN1	=	000001	Q#CLR	=	000040	RG0.EN	=	000200	TRANSF	=	000166RG	WORD19	=	000046
MREN2	=	020000	Q#CNC	=	030000	RG0.VA	=	020000	TSKTCB	=	00004R	WORD2	=	000004
MRPMD	=	003616R	Q#CP	=	000060	SCAN	=	006406R	TTX	=	007436R	WORD20	=	000050
MRPM1	=	***** GX	Q#CPCC	=	000010	SCOUNT	=	000160RG	T#AD	=	000020	WORD21	=	000052
MRPM2	=	003222R	Q#CP2	=	000260	SDHIGH	=	001346R	T#BA	=	000002	WORD22	=	000054
MSTR2	=	000164RG	Q#CSC	=	010000	SDLOW	=	001350R	T#BD	=	000010	WORD23	=	000056
MSYN	=	000040	Q#CSEL	=	000360	SELECT	=	000150RG	T#BSO	=	100000	WORD24	=	000060
MYSELF	=	000000R	Q#CSET	=	000002	SELMEM	=	007242R	T#BT	=	000020	WORD25	=	000062
N	=	0000144	Q#CSP	=	020000	SELPG	=	***** GX	T#BTAR	=	000030	WORD26	=	000064
NEST	=	000001	Q#DMA	=	000001	SEQ.CI	=	000010	T#BD	=	002000	WORD27	=	000066
NESTOP	=	007242R	Q#ENBK	=	040000	SF	=	003320R	T#CD	=	000100	WORD28	=	000070
N.DID	=	000024	Q#ENOP	=	020000	SPCLR	=	005432R	T#CLK	=	002000	WORD29	=	000072
N.DVNM	=	000032	Q#FAL	=	004000	SPCR	=	***** GX	T#D1SK	=	000200	WORD30	=	000074
N.FID	=	000000	Q#FC	=	000045	SPLoad	=	005426R	T#DRD	=	000002	WORD31	=	000076
N.FNAM	=	000006	Q#FD	=	000044	SPLoop	=	005620R	T\$EMEM	=	010000	WORD32	=	000100
N.FTYP	=	000014	Q#FF	=	000046	SPNEXT	=	005546R	T#FAA	=	000000			

WORD33 = 000102	WORD50 = 000144	WORD68 = 000210	WORD85 = 000252	\$CEF1 = ***** GX
WORD34 = 000104	WORD51 = 000146	WORD69 = 000212	WORD86 = 000254	\$COTB = ***** GX
WORD35 = 000106	WORD52 = 000150	WORD7 = 000016	WORD87 = 000256	\$CSTA = ***** GX
WORD36 = 000110	WORD53 = 000152	WORD70 = 000214	WORD88 = 000260	\$DRDSE = ***** GX
WORD37 = 000112	WORD54 = 000154	WORD71 = 000216	WORD89 = 000262	\$TKTCB = ***** GX
WORD38 = 000114	WORD55 = 000156	WORD72 = 000220	WORD9 = 000022	\$\$\$ = 002332R
WORD39 = 000116	WORD56 = 000160	WORD73 = 000222	WORD90 = 000264	\$\$\$ARG = 000002
WORD4 = 000010	WORD57 = 000162	WORD74 = 000224	WORD91 = 000266	\$\$\$T1 = 000067
WORD40 = 000120	WORD58 = 000164	WORD75 = 000226	WORD92 = 000270	\$\$\$T2 = 000027
WORD41 = 000122	WORD59 = 000166	WORD76 = 000230	WORD93 = 000272	.CLOSE = ***** G
WORD42 = 000124	WORD6 = 000014	WORD77 = 000232	WORD94 = 000274	.FSRCE = ***** G
WORD43 = 000126	WORD60 = 000170	WORD78 = 000234	WORD95 = 000276	.GCML1 = ***** G
WORD44 = 000130	WORD61 = 000172	WORD79 = 000236	WORD96 = 000300	.OPEN = ***** G
WORD45 = 000132	WORD62 = 000174	WORD8 = 000020	WORD97 = 000302	.READ = ***** G
WORD46 = 000134	WORD63 = 000176	WORD80 = 000240	WORD98 = 000304	...PC1 = 002460R
WORD47 = 000136	WORD64 = 000200	WORD81 = 000242	WORD99 = 000306	...PC2 = 002634R
WORD48 = 000140	WORD65 = 000202	WORD82 = 000244	WRDVAL = 000310	...PC3 = 002460R
WORD49 = 000142	WORD66 = 000204	WORD83 = 000246	XTREAD = 001000	...TPC = 000020
WORDS = 000012	WORD67 = 000206	WORD84 = 000250	XTWRTE = 000400	

.ABS: 000000 000
007476 001
\$\$\$FSR1 001020 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 9075 WORDS (.36 PAGES)
DYNAMIC MEMORY: 10196 WORDS (.39 PAGES)
ELAPSED TIME: 00:01:51
LOADER, LOADER / -SP=[20, 1]IM,[20, 1]LOADER.

```

1          .TITLE - GD..
2 000000   .PSECT - CD..
3          ;
4          ;
5          ;
6          ;
7          ;
8          ;
9          ;
10         ;
11         ;
12         ;
13         ;
14         ;
15         ;
16         ;
17         ;
18         ;
19         ;
20         ;
21         ;
22         ;
23         ;
24 000000   CD::
25 000000   042767 000000G.000000G.   BIC   #FIRST,SELECT      ;CLEAR FIRST TIME THROUGH FLAG
26 000000   012767 000001 000002G.   MOV   #1,VIRT+2        ;RE-INIT BLOCK COUNT
27 000000   005067 000000G.           CLR   ADDR             ;CLEAR CP DATA MEMORY ADDRESS
28 000026   016767 000000G.000000C.   MOV   LCD,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
29 000034   016767 000002G.000000C.   MOV   LCD+2,INDNB+N,FNAM+2
30         ;
31 000052   CALL   DMASET          ;START UP DMA MICROCODE
32         ;
33         ;
34         ;
35         ;
36         ;
37 000056   CDNEXT: CALL   GET          ;READ A RECORD
38 000062   103002.   BCC   1$              ;
39 000064   000167   000150   JMP   CPDX            ;ERROR, EXIT
40 000070   1$:
41 000070   016705   000000C.   MOV   INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ
42 000074   032767   000000G.000000G.   BIT   #FIRST,SELECT    ;FIRST TIME THROUGH
43 000102   001014   BNE   3$             ;NO
44 000104   052767   000000G.000000G.   BIS   #FIRST,SELECT    ;SET FLAG FOR FIRST TIME THROUGH
45 000112   012567   000000G.   MOV   (R5)+,LCOUNT     ;GET NUMBER OF DATA WORDS
46 000116   012767   177777 000000G.   MOV   #-1,TRANSF      ;START TRANSFER COUNT AT -1
47 000124   026727   000000G.000400   CMP   LCOUNT,#256     ;256 WORDS TO TRANSFER
48 000132   001413   BEQ   4$             ;YES, 2 TRANSFERS NEEDED (COUNT WORD)
49         ;
50         ;
51         ;
52         ;
53         ;
54 000134   026727   000000G.000400   3$:   CMP   LCOUNT,#256     ;MORE THAN A FULL BLOCK LEFT TO TRANSFER
55 000142   003007   BGT   4$             ;YES, TRANSFER 256 WORDS THIS TIME
56 000144   052767   000000G.000000G.   BIS   #LAST,SELECT    ;LAST TRANSFER
57 000152   066767   000000G.000000G.   ADD   LCOUNT,TRANSF   ;TRANSFER REMAINING WORDS

```

```

58 000160 000406
59 000162 062767 000400 000000G 4#:
60 000170 166767 000000G 000000G
61 ;
62 ;
63 ;
64 000176 010567 000000G 5#:
65 000202 012746 000003
66 000206
67 ;
68 ;
69 ;
70 ;
71 000212 032767 000000G 000000G
72 000220 001007
73 000222 066767 000000G 000000G
74 000230 005067 000000G
75 000234 000167 177616
76 ;
77 000240 CPDX:
78 000240 005046
79 000242 012746 176000
80 000246
81 000252 005067 176422
82 ;
83 000256
84 000266 105067 000000G
85 000272
86 ;
87 000001 .END

```

```

BR 5$ ;LOAD CD BY DMA
ADD #256, TRANSF ;NUMBER OF WORDS TO TRANSFER
SUB TRANSF, LCOUNT ;SUB FROM TOTAL
;
SIGNAL MICROCODE TO ACCEPT CP DATA MEMORY DATA
;
MOV RS, INSAVE ;SAVE POINTER TO INPUT DATA
MOV #0$LDCD, -(SP) ;MOVE ATTN CODE TO STACK
CALL DMA ;PERFORM DMA LOAD
;
IF NOT FINISHED WITH ALL DATA, GO BACK AND PREPARE TO
TRANSFER NEXT BLOCK
;
BIT #LAST, SELECT ;FINISHED?
BNE CPDX ;YES
ADD TRANSF, ADDR ;SET CD ADDRESS FOR NEXT TRANSFER
CLR TRANSF
JMP CDNEXT ;GET NEXT RECORD
;
CLR -(SP) ;CLEAR NOTHING IN CSR1
MOV #0$NCLK, -(SP) ;SET NO-CLOCKS
CALL CSR1
CLR OR$CR2 ;SET LOAD MODE
;
CLOSE$ #INFDB
CLRB INDNB+N, FVER ;RESET FILE VERSION NUMBER
RETURN
;

```


ADDR = ***** GX.	BYTE41 = 000051	BYTE93 = 000135	MMOE = 000004	Q\$MSP = 100000
ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MMWRITE = 000010	Q\$NCLK = 176000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	MNOBRE = 100000	Q\$PP = 000100
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	MREN1 = 000001	Q\$PPSW = 000320
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	MREN2 = 020000	Q\$PP2 = 000300
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	MSYN = 000040	Q\$CHLT = 000013
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	N = 000144	Q\$QL = 000043
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	N.FNAM = ***** GX.	Q\$QLA = 000053
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	N.FVER = ***** GX.	Q\$QLB = 000054
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	PAR\$\$\$ = 000027	Q\$QLR = 000001
BIT13 = 020000	BYTE50 = 000062	CD = 000000RG.	002.PLB = 000010	Q\$QW = 000042
BIT14 = 040000	BYTE51 = 000063	CDNEXT = 000056R.	002.PLC = 000020	Q\$RDOD = 000005
BIT15 = 100000	BYTE52 = 000064	CHOBRE = 100000	PLD = 000030	Q\$RDMD = 000006
BIT2 = 000004	BYTE53 = 000065	CPCCEN = 010000	PLRWR = 000200	Q\$REBK = 001000
BIT3 = 000010	BYTE54 = 000066	CPDX = 000240R.	002.PLR.EN = 000200	Q\$RNC = 006000
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	QR\$CR1 = 176420	Q\$RSC = 004000
BIT5 = 000040	BYTE56 = 000070	CPWRITE = 020000	QR\$CR2 = 176422	Q\$RSET = 000010
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	QR\$LBR = 176424	Q\$SM = 100000
BIT7 = 000200	BYTE58 = 000072	CSEQCI = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BIT8 = 000400	BYTE59 = 000073	CSOE = 000040	Q\$BCL = 000001	Q\$SP2 = 000340
BIT9 = 001000	BYTE6 = 000006	CSR1 = ***** GX.	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE0 = 000000	BYTE60 = 000074	CSURTE = 000100	Q\$CHB = 000400	RGQ.VA = 020000
BYTE1 = 000001	BYTE61 = 000075	DBR.RD = 000001	Q\$CHRL = 000200	SELECT = ***** GX.
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	Q\$CLR = 000040	SEQ.CI = 000010
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	Q\$CNC = 030000	S\$CLR = 000000
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	Q\$CP = 000060	S\$LA = 000001
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q\$CPC = 000010	S\$OB = 000005
BYTE14 = 000016	BYTE66 = 000102	DMA = ***** GX.	Q\$CP2 = 000260	S\$QR = 000006
BYTE15 = 000017	BYTE67 = 000103	DMAAWR = 000005	Q\$CSC = 010000	S\$QX = 000004
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q\$CSEL = 000360	S\$SR = 000007
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q\$CSET = 000002	S\$S1 = 000010
BYTE18 = 000022	BYTE7 = 000007	DMASET = ***** GX.	Q\$CSP = 020000	S\$S2 = 000014
BYTE19 = 000023	BYTE70 = 000106	ENBR = 010000	Q\$DMA = 000001	TD\$CTR = 176370
BYTE2 = 000002	BYTE71 = 000107	FIRST = ***** GX.	Q\$ENBK = 040000	TD\$CTW = 176360
BYTE20 = 000024	BYTE72 = 000110	FO.RD = ***** GX.	Q\$ENOP = 020000	TD\$INL = 004000
BYTE21 = 000025	BYTE73 = 000111	F.BKDS = ***** GX.	Q\$FAL = 004000	TD\$MEM = 000270
BYTE22 = 000026	BYTE74 = 000112	F.FACC = ***** GX.	Q\$FC = 000045	TD\$OAR = 176344
BYTE23 = 000027	BYTE75 = 000113	GET = ***** GX.	Q\$FO = 000044	TD\$OTR = 176346
BYTE24 = 000030	BYTE76 = 000114	INDNB = ***** GX.	Q\$FP = 000046	TD\$ORD = 000274
BYTE25 = 000031	BYTE77 = 000115	INFDB = ***** GX.	Q\$HBF = 000002	TD\$SW = 176376
BYTE26 = 000032	BYTE78 = 000116	INSAVE = ***** GX.	Q\$ICP = 000006	TD\$STAR = 176372
BYTE27 = 000033	BYTE79 = 000117	LAST = ***** GX.	Q\$IHB = 000003	TD\$TAW = 176362
BYTE28 = 000034	BYTE8 = 000010	LCD = ***** GX.	Q\$IHRL = 000002	TD\$TDR = 176374
BYTE29 = 000035	BYTE80 = 000120	LCCOUNT = ***** GX.	Q\$IHRLP = 000007	TD\$TDW = 176364
BYTE3 = 000003	BYTE81 = 000121	LOC.EN = 000100	Q\$LBD = 001000	TRANSF = ***** GX.
BYTE30 = 000036	BYTE82 = 000122	LOC.WA = 040000	Q\$LBDP = 001001	T\$AD = 000020
BYTE31 = 000037	BYTE83 = 000123	LOC.WB = 100000	Q\$LBP = 000001	T\$BA = 000002
BYTE32 = 000040	BYTE84 = 000124	MAREN1 = 000001	Q\$LDCD = 000003	T\$BD = 000010
BYTE33 = 000041	BYTE85 = 000125	MAREN2 = 004000	Q\$LDMD = 000004	T\$B50 = 100000
BYTE34 = 000042	BYTE86 = 000126	MARLOD = 010000	Q\$LDPP = 002000	T\$BT = 000020
BYTE35 = 000043	BYTE87 = 000127	MAROUT = 000002	Q\$LHP = 010000	T\$BTAR = 000030
BYTE36 = 000044	BYTE88 = 000130	MAR.LO = 002000	Q\$MNC = 140000	T\$BDT = 002000
BYTE37 = 000045	BYTE89 = 000131	MAR.OU = 000040	Q\$MR = 000052	T\$CD = 000100
BYTE38 = 000046	BYTE9 = 000011	MBKALL = 001000	Q\$MRP = 000040	T\$CLK = 002000
BYTE39 = 000047	BYTE90 = 000132	MBKCLK = 000400	Q\$MRP2 = 000240	T\$DISK = 000200
BYTE4 = 000004	BYTE91 = 000133	MMADRD = 000100	Q\$MSC = 040000	T\$DRD = 000004
BYTE40 = 000050	BYTE92 = 000134	MMLEFT = 000002	Q\$MSET = 000004	T\$EMEM = 010000

T#FSAA = 000000	T#BBEN = 000020	WORD30 = 000074	WORD55 = 000156	WORD8 = 000020
T#FSAB = 000004	UBD, IN = 000020	WORD31 = 000076	WORD56 = 000160	WORD80 = 000240
T#FSAC = 000014	VIRT = ***** GX	WORD32 = 000100	WORD57 = 000162	WORD81 = 000242
T#FSB2 = 000010	WORD0 = 000000	WORD33 = 000102	WORD58 = 000164	WORD82 = 000244
T#IB = 000026	WORD1 = 000002	WORD34 = 000104	WORD59 = 000166	WORD83 = 000246
T#IBAR = 000024	WORD10 = 000024	WORD35 = 000106	WORD6 = 000014	WORD84 = 000250
T#IBE = 020000	WORD11 = 000026	WORD36 = 000110	WORD60 = 000170	WORD85 = 000252
T#IBF = 040000	WORD12 = 000030	WORD37 = 000112	WORD61 = 000172	WORD86 = 000254
T#ICD = 000040	WORD13 = 000032	WORD38 = 000114	WORD62 = 000174	WORD87 = 000256
T#MODE = 004000	WORD14 = 000034	WORD39 = 000116	WORD63 = 000176	WORD88 = 000260
T#OB = 000036	WORD15 = 000036	WORD4 = 000010	WORD64 = 000200	WORD89 = 000262
T#OBE = 004000	WORD16 = 000040	WORD40 = 000120	WORD65 = 000202	WORD9 = 000022
T#OBF = 010000	WORD17 = 000042	WORD41 = 000122	WORD66 = 000204	WORD90 = 000264
T#OBRA = 000034	WORD18 = 000044	WORD42 = 000124	WORD67 = 000206	WORD91 = 000266
T#OBWA = 000032	WORD19 = 000046	WORD43 = 000126	WORD68 = 000210	WORD92 = 000270
T#OUTA = 100000	WORD2 = 000004	WORD44 = 000130	WORD69 = 000212	WORD93 = 000272
T#RBD0 = 000200	WORD20 = 000050	WORD45 = 000132	WORD7 = 000016	WORD94 = 000274
T#RNB = 000040	WORD21 = 000052	WORD46 = 000134	WORD70 = 000214	WORD95 = 000276
T#RSET = 040000	WORD22 = 000054	WORD47 = 000136	WORD71 = 000216	WORD96 = 000300
T#SC = 000022	WORD23 = 000056	WORD48 = 000140	WORD72 = 000220	WORD97 = 000302
T#SCLK = 020000	WORD24 = 000060	WORD49 = 000142	WORD73 = 000222	WORD98 = 000304
T#SEG1 = 000000	WORD25 = 000062	WORDS = 000012	WORD74 = 000224	WORD99 = 000306
T#SEG2 = 000001	WORD26 = 000064	WORD50 = 000144	WORD75 = 000226	WRDVAL = 000310
T#SEG3 = 000002	WORD27 = 000066	WORD51 = 000146	WORD76 = 000230	XTREAD = 001000
T#SD = 000001	WORD28 = 000070	WORD52 = 000150	WORD77 = 000232	XTWRITE = 000400
T#UBUS = 100000	WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	.CLOSE = *****
T#1CLK = 000400	WORD3 = 000006	WORD54 = 000154	WORD79 = 000236	.OPEN = *****

. ABS: 000000 000
000000 001
CD: 000274 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3981 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:44
CD, CD/←SP=C20, 1JIM, C20, 1JCD


```

13 ; START-UP-DMA-LOAD-MICROCODE-IN-CP:MRP-DOES-NOTHING.
14 ;
15 ; DMASET:
16 000000 012746 000377 MOV #377,-(SP) ;SET-MRP-MICRO-ADDRESS=-X'FF' (JUMP-SELF)
17 000004 CALL SEQMM:
18
19 000010 005046 CLR -(SP) ;RESET-BR-INHIBIT
20 000012 CALL MRPCR
21 000016 005046 CLR -(SP) ;START-MICROCODE-AT-0
22 000020 CALL SEQCS
23 000024 005046 CLR -(SP) ;RESET-BR-INHIBIT
24 000026 CALL CPCR
25 000032 012767 001000 176422 MOV #Q$REBK,QR$CR2 ;RE-ARM-INTERRUPTS
26 000040 012767 120000 176422 MOV #<Q$SM+Q$ENDP>,QR$CR2 ;SET-SEARCH-MODE-+ENABLE-INTERRUPTS
27 000046 012746 000360 MOV #Q$CSEL,-(SP) ;CLEAR-ALL-SELECTIONS
28 000052 052716 001001 BIS #<Q$LBD+Q$LBP>,(SP) ;CLEAR-DRIVE-AND-PULSE
29 000056 052716 170000 BIS #<Q$MNC+Q$CNC>,(SP) ;CLEAR-CP-NO-CLOCK
30 000062 005046 CLR -(SP) ;SET-NOTHING
31 000064 CALL CSR1
32 ;
33 000070 RETURN

```

```

35 ;
36 ;
37 ;
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 000072. ; DMA:
45 000072. 016667 000002 176424 MOV 2(SP),QR$LBR ; MOVE ATTN CODE TO LOD BUS REG
46 000100 012767 120100 176422. MOV #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ; SET ATTN CODE READY
47 000106 016701 176422 6$: MOV QR$CR2,R1 ; READ CSR2
48 000112 032701 000100 BIT #Q$ATTN,R1 ; ATTN CLEAR
49 000116 001373 BNE 6$ ; NO, READ AGAIN
50 ;
51 000120 016767 000000 176424 MOV MSTR2,QR$LBR ; CD MEMORY START ADDRESS
52 000126 012767 120040 176422. MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ; SET CC TO CP
53 000134 016701 176422 7$: MOV QR$CR2,R1 ; READ CSR2
54 000140 032701 000040 BIT #Q$CCCP,R1 ; IS CC TO CP CLEAR
55 000144 001373 BNE 7$ ; NO, READ AGAIN
56 ;
57 000146 016767 000000 176424 MOV TRANSF,QR$LBR ; TRANSFER COUNT
58 000154 012767 120040 176422. MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ; SET CC TO CP
59 000162 016701 176422 8$: MOV QR$CR2,R1 ; READ CSR2
60 000166 032701 000040 BIT #Q$CCCP,R1 ; IS CC TO CP CLEAR
61 000172 001373 BNE 8$ ; NO, READ AGAIN
62 ;
63 000174 016767 000000 176424 MOV INSAVE,QR$LBR ; CC MEMORY DATA BUFFER
64 000202 012767 120040 176422. MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ; SET CC TO CP
65 ;
66 ;
67 ;
68 000210 ;
69 9$: RDAF$S #EFBUF ; READ EVENT FLAGS
70 000222 032767 000004 000000. BIT #BIT2,EFBUF ; WAS EF#3 SET
71 000230 001767 BEQ 9$ ; NO, READ AGAIN
72 ;
73 000232 ;
74 ;
75 ;
76 ;
77 000244 012767 100400 176422. MOV #<Q$SM+Q$CHB>,QR$CR2 ; CLEAR INTERRUPT (USE HIT BUFFER INT)
78 000252 012767 101000 176422. MOV #<Q$SM+Q$REBK>,QR$CR2 ; RE-ARM
79 000260 012767 160000 176422. MOV #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ; ENABLE
80 ;
81 000266 011666 000002 MOV (SP),2(SP) ; MOVE RETURN ADDRESS DOWN STACK
82 000272 005726 TST (SP)+ ; POINT TO RETURN ADDRESS
83 000274 RETURN
84 000001 .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLD = 000030	Q\$RDMD = 000006
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLRWR = 000200	Q\$REBK = 001000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLR,EN = 000200	Q\$RNC = 006000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$RCR1 = 176420	Q\$RSC = 004000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$RCR2 = 176422	Q\$RSET = 000010
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$RLBR = 176424	Q\$SM = 100000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$ATTN = 000100	Q\$SP = 000120
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CCCP = 000040	RGD,EN = 000200
BIT13 = 020000	BYTE50 = 000062	CHOBRE = 100000	Q\$CHB = 000400	RGD,VA = 020000
BIT14 = 040000	BYTE51 = 000063	CPCCEN = 010000	Q\$CHRL = 000200	SEQCS = ***** GX
BIT15 = 100000	BYTE52 = 000064	CPCR = ***** GX	Q\$CLR = 000040	SEOMM = ***** GX
BIT2 = 000004	BYTE53 = 000065	CPREAD = 040000	Q\$CNC = 030000	SEQ,CI = 000010
BIT3 = 000010	BYTE54 = 000066	CPURTE = 020000	Q\$CP = 000060	S\$CLR = 000000
BIT4 = 000020	BYTE55 = 000067	CSADRD = 000004	Q\$CPCC = 000010	S\$LA = 000001
BIT5 = 000040	BYTE56 = 000070	CSEQCI = 100000	Q\$CP2 = 000260	S\$OB = 000005
BIT6 = 000100	BYTE57 = 000071	CSOE = 000040	Q\$CSC = 010000	S\$OR = 000006
BIT7 = 000200	BYTE58 = 000072	CSR1 = ***** GX	Q\$CSEL = 000360	S\$OX = 000004
BIT8 = 000400	BYTE59 = 000073	CSURTE = 000100	Q\$CSET = 000002	S\$SR = 000007
BIT9 = 001000	BYTE6 = 000006	DBR,RD = 000001	Q\$CSP = 020000	S\$S1 = 000010
BYTE0 = 000000	BYTE60 = 000074	DB\$CPP = 001457	Q\$DMA = 000001	S\$S2 = 000014
BYTE1 = 000001	BYTE61 = 000075	DB\$SPT = 000026	Q\$ENBK = 040000	TD\$CTR = 176370
BYTE10 = 000012	BYTE62 = 000076	DB\$TPC = 000023	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$FAL = 004000	TD\$INL = 004000
BYTE12 = 000014	BYTE64 = 000100	DMA = 000072RG	002 Q\$FC = 000045	TD\$MEM = 000270
BYTE13 = 000015	BYTE65 = 000101	DMAWR = 000005	Q\$FD = 000044	TD\$OAR = 176344
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$FP = 000046	TD\$QTR = 176346
BYTE15 = 000017	BYTE67 = 000103	DMARWR = 000004	Q\$HBF = 000002	TD\$QRD = 000274
BYTE16 = 000020	BYTE68 = 000104	DMASET = 000000RG	002 Q\$ICP = 000006	TD\$SW = 176376
BYTE17 = 000021	BYTE69 = 000105	EFBUF = ***** GX	Q\$IHB = 000003	TD\$TAR = 176372
BYTE18 = 000022	BYTE7 = 000007	EFN,3 = ***** GX	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE19 = 000023	BYTE70 = 000106	ENBR = 010000	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE2 = 000002	BYTE71 = 000107	INSAVE = ***** GX	Q\$LBD = 001000	TD\$TDW = 176364
BYTE20 = 000024	BYTE72 = 000110	LOC,EN = 000100	Q\$LBDP = 001001	TRANSF = ***** GX
BYTE21 = 000025	BYTE73 = 000111	LOC,WA = 040000	Q\$LBP = 000001	T\$AD = 000020
BYTE22 = 000026	BYTE74 = 000112	LOC,WB = 100000	Q\$LCD = 000003	T\$BA = 000002
BYTE23 = 000027	BYTE75 = 000113	MAREN1 = 000001	Q\$LMD = 000004	T\$BD = 000010
BYTE24 = 000030	BYTE76 = 000114	MAREN2 = 004000	Q\$LDPP = 002000	T\$BSQ = 100000
BYTE25 = 000031	BYTE77 = 000115	MARLOD = 010000	Q\$LHP = 010000	T\$BT = 000020
BYTE26 = 000032	BYTE78 = 000116	MAROUT = 000002	Q\$MNC = 140000	T\$BTAR = 000030
BYTE27 = 000033	BYTE79 = 000117	MAR,LO = 002000	Q\$MR = 000052	T\$BTID = 002000
BYTE28 = 000034	BYTE8 = 000010	MAR,OU = 000040	Q\$MRP = 000040	T\$CD = 000100
BYTE29 = 000035	BYTE80 = 000120	MBKALL = 001000	Q\$MRP2 = 000240	T\$CLK = 002000
BYTE3 = 000003	BYTE81 = 000121	MBKCLK = 000400	Q\$MSC = 040000	T\$DISK = 000200
BYTE30 = 000036	BYTE82 = 000122	MMADR = 000100	Q\$MSET = 000004	T\$DRD = 000004
BYTE31 = 000037	BYTE83 = 000123	MMLEFT = 000002	Q\$MSP = 100000	T\$MEM1 = 010000
BYTE32 = 000040	BYTE84 = 000124	MNDE = 000004	Q\$NCLK = 176000	T\$FSAA = 000000
BYTE33 = 000041	BYTE85 = 000125	MMURTE = 000010	Q\$PP = 000100	T\$FSAB = 000004
BYTE34 = 000042	BYTE86 = 000126	MNOBRE = 100000	Q\$PPSW = 000320	T\$FSAC = 000014
BYTE35 = 000043	BYTE87 = 000127	MREN1 = 000001	Q\$PP2 = 000300	T\$FSB2 = 000010
BYTE36 = 000044	BYTE88 = 000130	MREN2 = 020000	Q\$QHLT = 000013	T\$IB = 000026
BYTE37 = 000045	BYTE89 = 000131	MRPCR = ***** GX	Q\$QL = 000043	T\$IBAR = 000024
BYTE38 = 000046	BYTE9 = 000011	MSTR2 = ***** GX	Q\$QLA = 000053	T\$IBE = 020000
BYTE39 = 000047	BYTE90 = 000132	MSYN = 000040	Q\$QLB = 000054	T\$IBF = 040000
BYTE4 = 000004	BYTE91 = 000133	N = 000144	Q\$QLR = 000001	T\$ICD = 000040
BYTE40 = 000050	BYTE92 = 000134	PLB = 000010	Q\$QL = 000042	T\$MODE = 004000
BYTE41 = 000051	BYTE93 = 000135	PLC = 000020	Q\$Q = 000005	T\$OB = 000036

T\$DBE = 004000	WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240
T\$DBF = 010000	WORD16 = 000040	WORD38 = 000114	WORD60 = 000170	WORD81 = 000242
T\$DBRA = 000034	WORD17 = 000042	WORD39 = 000116	WORD61 = 000172	WORD82 = 000244
T\$DBWA = 000032	WORD18 = 000044	WORD40 = 000118	WORD62 = 000174	WORD83 = 000246
T\$DUTA = 100000	WORD19 = 000046	WORD41 = 000120	WORD63 = 000176	WORD84 = 000250
T\$RBD0 = 000200	WORD20 = 000048	WORD42 = 000124	WORD64 = 000200	WORD85 = 000252
T\$RNB = 000040	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD86 = 000254
T\$RSET = 040000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD87 = 000256
T\$SC = 000022	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD88 = 000260
T\$SCLK = 020000	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD89 = 000262
T\$SEG1 = 000000	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T\$SEG2 = 000001	WORD26 = 000064	WORD48 = 000140	WORD70 = 000016	WORD91 = 000266
T\$SEG3 = 000002	WORD27 = 000066	WORD49 = 000142	WORD71 = 000214	WORD92 = 000270
T\$SO = 000001	WORD28 = 000070	WORD50 = 000144	WORD72 = 000216	WORD93 = 000272
T\$UBUS = 100000	WORD29 = 000072	WORD51 = 000146	WORD73 = 000220	WORD94 = 000274
T\$1CLK = 000400	WORD30 = 000074	WORD52 = 000150	WORD74 = 000222	WORD95 = 000276
T\$BEN = 000020	WORD31 = 000076	WORD53 = 000152	WORD75 = 000224	WORD96 = 000300
UBD.IN = 000020	WORD32 = 000100	WORD54 = 000154	WORD76 = 000226	WORD97 = 000302
WORD0 = 000000	WORD33 = 000102	WORD55 = 000156	WORD77 = 000230	WORD98 = 000304
WORD1 = 000002	WORD34 = 000104	WORD56 = 000160	WORD78 = 000232	WORD99 = 000306
WORD10 = 000024	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WORDVAL = 000310
WORD11 = 000026	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	XTREAD = 001000
WORD12 = 000030				XTWRITE = 000400
WORD13 = 000032				
WORD14 = 000034				

. ABS: 000000 000
000000 001
DMASUB: 000276 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3288 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
DMASUB, DMASUB /-SP=C20,111M,C20,11DMASUB

```
1                                     .TITLE- MICRO-
2 000000                             .PSECT: MICRO-
3                                     ;
4                                     ;
5                                     ;
6                                     ;
7                                     ;
8                                     ;
9                                     ;
10                                    ;
11                                    ;
12                                    ;
13                                    ;
14 000000                             .MCALL: OPEN#R,CLOSE#,FINIT$
15 000000                             MICRO::
16 000004                             FINIT$
17 000010 042767 000000G:000000G     CALL: MRPMM:           ;LOAD:MRP:MICROCODE-
18 000016 012767 000001 000002G     BIC:  #FIRST,SELECT:  ;CLEAR:FIRST-TIME-THROUGH-FLAG
19 000024                             MOV:  #1,VIRT+2:     ;RE-INIT:BLOCK-COUNT
20 000030                             CALL:  CPCS           ;LOAD:CP:MICROCODE-
                                     RETURN-
```



```

22. ;
23. ; MRP MICROPROGRAM MEMORY.
24. ;
25. ;
26. ;
27. ;
28. 000032. ; MRPMM::
29. 000032. 016767 000000G.000000C. MOV. LMM, INDNB+N, FNAM. ; PLACE FILE NAME INTO INPUT DNB.
30. 000040. 016767 000002G.000000C. MOV. LMM+2, INDNB+N, FNAM+2.
31. 000046. OPEN$R. #INFDB.
32. ;
33. ;
34. ; SET MRP SEQUENCER TO ZERO.
35. ; GET FIRST RECORD. THE FIRST WORD OF THE FIRST RECORD
36. ; CONTAINS THE NUMBER OF WORDS TO BE LOADED INTO AN
37. ; MRP COLUMN (SEE BELOW). SAVE THIS VALUE.
38. 000064. 005067 000000G. CLR. MSTR2. ; INIT SEQUENCER = 0
39. 000070. 1$; CALL. GET. ; READ A RECORD.
40. 000074. 103511 BCS. MRPMM. ; ERROR, EXIT.
41. 000076. 016705 000000C. MOV. INFDB+F, BKDS+2, R5. ; POINT TO RECORD READ.
42. 000102. 012704 000400 MOV. #256., R4. ; NUMBER OF WORDS IN RECORD (MAX)
43. 000106. 032767 000000G.000000G. BIT. #FIRST, SELECT. ; FIRST TIME THROUGH.
44. 000114. 001011 BNE. 2$. ; NO.
45. 000116. 052767 000000G.000000G. BIS. #FIRST, SELECT. ; SET FLAG FOR FIRST TIME THROUGH.
46. 000124. 012567 000000G. MOV. (R5)+, LCOUNT. ; SAVE NUMBER OF WORDS IN COLUMN.
47. 000130. 016767 000000G.000000G. MOV. LCOUNT, WCOUNT. ; INITIALIZE WORKING COUNTER.
48. 000136. 005304 DEC. R4. ; SUB FROM NUMBER OF WORDS IN RECORD.
49. ;
50. ; EACH LOCATION IN MRP MICROPROGRAM MEMORY CONSISTS OF TWO
51. ; WORDS, A LEFT WORD AND A RIGHT WORDS. IN LOADING, ALL LEFT
52. ; WORDS ARE LOADED FIRST (IE, A COLUMN) THEN ALL RIGHT WORDS.
53. ; THE PROGRAM 'CONVRT' HAS WRITTEN THE FILE LDMM.DAT TO CONTAIN
54. ; MRP MICROPGM MEMORY DATA IN COLUMNS.
55. ;
56. ;
57. ;
58. ;
59. 000140. 2$;
60. 000144. 016746 000000G. MOV. MSTR2, -(SP) ; INITIALIZE SEQUENCER ADDRESS.
61. 000150. 012746 000012 CALL. SEQMM. ; SET ADDRESS.
62. 000154. MOV. #<MMURTEN+MMLEFT>, -(SP)
63. 000160. 012546 CALL. MRPCRA. ; DIRECT CONTROL WORD TO MRP.
64. 000162. MOV. (R5)+, -(SP) ; GET READY TO MOVE DATA TO MRP
65. 000166. 005046 CALL. LBMSC. ; DO IT.
66. 000170. CLR. -(SP) ; CLEAR CONTROL REG.
67. CALL. MRPCR.
68. 000174. 005367 000000G. DEC. WCOUNT. ; SUB FROM # WORDS IN A COLUMN.
69. 000200. 001405 BEQ. MRPRGT. ; DO RIGHT HAND COLUMN.
70. 000202. 005267 000000G. INC. MSTR2. ; ADVANCE SEQUENCER ADDRESS.
71. 000206. 005304 DEC. R4. ; FINISHED WITH THIS RECORD.
72. 000210. 001727 BEQ. 1$. ; YES, GET NEXT.
73. 000212. 000752 BR. 2$. ; NO, LOAD NEXT WORD.
74. ;
75. ;
76. ;
77. 000214. MRPRTG:
78. 000214. 016767 000000G.000000G. MOV. LCOUNT, WCOUNT. ; REINITIALIZE WORKING COUNTER.

```

```

79 000222 005067 000000G
80 000226 005304
81 000230 001007
82 000232
83 000236 103430
84 000240 016705 000000C
85 000244 012704 000400
86
87 000250
88 000250 016746 000000G
89 000254
90 000260 012746 000010
91 000264
92 000270 012546
93 000272
94 000276 005046
95 000300
96
97 000304 005367 000000G
98 000310 001403
99 000312 005267 000000G
100 000316 000743
101
102 000320
103 000320
104 000330 105067 000000C
105 000334

```

```

1$: CLR MSTR2 ; INIT SEQUENCER = 0
DEC R4 ; FINISHED WITH THIS RECORD
BNE 2$ ; NO, CONTINUE
CALL GET ; READ NEXT
BCS MRPMX ; ERROR, EXIT
MOV INFDB+FBKDS+2,R5 ; POINT TO RECORD READ
MOV #256,R4 ; R4 = NUMBER OF WORDS IN RECORD
;
2$: MOV MSTR2,-(SP) ; INITIALIZE SEQUENCER ADDRESS
CALL SEQMM ; SET ADDRESS
MOV #<MMWRTE>,-(SP)
CALL MRPCRA ; DIRECT CONTROL WORD TO MRP
MOV (R5)+,-(SP) ; GET READY TO MOVE DATA TO MRP
CALL LBMSC ; DO IT
CLR -(SP) ; CLEAR CONTROL REG
CALL MRPCR
;
DEC WCOUNT ; FINISHED WITH THIS COLUMN
BEQ MRPMX ; YES, DONE
INC MSTR2 ; NO, ADVANCE SEQUENCER ADDRESS
BR 1$ ; SET IT
;
MRPMX: CLOSE# #INFDB ;
CLRB INDNB+N,FVER ; RESET FILE VERSION NUMBER
RETURN

```

```

107      ;
108      ;
109      ;       CP CONTROL STORE SUBROUTINE.
110      ;
111      ;
112      ;       CPCS::
113      ;       FILL IN FILE NAME BLOCK FOR LDCS.DAT.
114      ;       OPEN FILE.
115      000336 016767 000000G.000000C.  MOV.   LCS,INDNB+N,FNAM.           ;PLACE FILE NAME INTO INPUT DNB.
116      000344 016767 000002G.000000C.  MOV.   LCS+2,INDNB+N,FNAM+2.
117      000352.  OPEN$R. #INFDB.
118      ;
119      ;
120      ;       RESET CP, SET CP SEQUENCER TO ZERO.
121      ;       GET FIRST RECORD, THE FIRST WORD OF THE FIRST RECORD
122      ;       CONTAINS THE NUMBER OF WORDS TO BE LOADED INTO AN
123      ;       CP COLUMN (SEE BELOW), SAVE THIS VALUE.
124      000370 005046  CLR.   -(SP)                       ;CLEAR NOTHING.
125      000372 012746 000002  MOV.   #0$CSET,-(SP)              ;CP RESET.
126      000376  CALL.  CSR1
127      000402 012746 000002  MOV.   #0$CSET,-(SP)              ;CLEAR RESET.
128      000406 005046  CLR.   -(SP)                       ;SET NOTHING.
129      000410  CALL.  CSR1
130      ;
131      000414 005067 000000G.  CLR.   MSTR2.                     ;SET SEQUENCER TO ZERO.
132      000420 1$:  CALL.  GET.
133      000424 103002  BCC.  15$
134      000426 000167 000432  JMP.  CPCSX.                       ;BRANCH IF OK.
135      000432. 15$:
136      000432 016705 000000C.  MOV.   INFDB+F,BKDS+2,R5          ;POINT TO RECORD READ.
137      000436 012704 000400  MOV.   #256,,R4                  ;NUMBER OF WORDS IN RECORD (MAX)
138      000442 032767 000000G.000000G. BIT.   #FIRST,SELECT.            ;FIRST TIME THROUGH.
139      000450 001011  BNE.  2$
140      000452 052767 000000G.000000G. BIS.   #FIRST,SELECT.            ;NO.
141      000460 012567 000000G.  MOV.   (R5)+,LCOUNT.              ;SET FLAG FOR FIRST TIME THROUGH.
142      000464 016767 000000G.000000G. MOV.   LCOUNT,WCOUNT.            ;SAVE NUMBER OF WORDS IN COLUMN.
143      000472 005304  DEC.   R4                          ;INITIALIZE WORKING COUNTER.
144      ;
145      ;
146      ;       EACH LOCATION IN CP CONTROL STORE CONSISTS OF FOUR WORDS,
147      ;       SECTION 'A', SECTION 'B', SECTION 'C', SECTION 'D'. IN
148      ;       WORDS ARE LOADED FIRST (IE, A COLUMN) THEN ALL RIGHT WORDS.
149      ;       LOADING, ALL OF SECTION 'A' IS LOADED FIRST, THEN SECTION
150      ;       'B', 'C', 'D', THE PROGRAM 'CONVRT' HAS WRITTEN THE FILE
151      ;       LDCS.DAT TO CONTAIN CP CONTROL STORE IN COLUMNS.
152      ;
153      ;       FILL SECTION 'A' OF CP CONTROL STORE.
154      000474 2$:
155      000474 016746 000000G.  MOV.   MSTR2,-(SP)                ;INITIALIZE SEQUENCER ADDRESS.
156      000500  CALL.  SEQCS.                     ;SET ADDRESS.
157      000504 012746 000100  MOV.   #CSWRTEN,-(SP)            ;SELECT SECTION A.
158      000510  CALL.  CPCRA.                     ;DIRECT CONTROL WORD TO CP.
159      000514 012546  MOV.   (R5)+,-(SP)              ;GET READY TO MOVE DATA TO MRP.
160      000516  CALL.  LBCSC.                     ;DO IT.
161      000522 005046  CLR.   -(SP)                       ;CLEAR CP CONTROL REG.
162      000524  CALL.  CPCR.
163      ;

```

```

164 000530 005367 000000G..... DEC.... WCOUNT..... ;SUB FROM # WORDS IN A COLUMN
165 000534 001405 BEQ... CPB... ;DO SECT B
166 000536 005267 000000G. INC... MSTR2... ;ADVANCE SEQUENCER ADDRESS
167 000542 005304 DEC... R4 ;FINISHED WITH THIS RECORD
168 000544 001725 BEQ... 1$ ;YES, GET NEXT
169 000546 000752 BR... 2$ ;NO, RESET SEQUENCER ADDRESS
170 ;
171 ;
172 ; LOAD SECTION 'B'
173 000550 ; CPB:
174 000550 016767 000000G.000000G. MOV... LCOUNT,WCOUNT... ;REINIT WORKING COUNTER
175 000556 005067 000000G. CLR... MSTR2... ;SET SEQUENCER TO ZERO
176 000562 005304 1$: DEC... R4 ;FINISHED WITH THIS RECORD
177 000564 001007 BNE... 2$ ;NO, CONTINUE
178 000566 CALL... GET... ;READ NEXT
179 000572 103534 BCS... CPCSX... ;ERROR, EXIT
180 ;
181 000574 016705 000000G. MOV... INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ
182 000600 012704 000400 MOV... #256.,R4 ;R4 = NUMBER OF WORDS IN RECORD
183 ;
184 000604 2$:
185 000604 016746 000000G. MOV... MSTR2,-(SP) ;INITIALIZE SEQUENCER ADDRESS
186 000610 CALL... SEQCS... ;SET ADDRESS
187 000614 012746 000110 MOV... #<CSWRTEN+PLB>,-(SP) ;SELECT SECTION B
188 000620 CALL... CPCRA... ;DIRECT CONTROL WORD TO CP
189 000624 012546 MOV... (R5)+,-(SP) ;GET READY TO MOVE DATA TO MRP
190 000626 CALL... LBCSC... ;DO IT
191 000632 005046 CLR... -(SP) ;CLEAR CP CONTROL REG
192 000634 CALL... CPCR
193 ;
194 000640 005367 000000G. DEC... WCOUNT... ;FINISHED WITH THIS COLUMN
195 000644 001403 BEQ... CPC... ;YES, GET NEXT
196 000646 005267 000000G. INC... MSTR2... ;NO, ADVANCE SEQUENCER ADDRESS
197 000652 000743 BR... 1$ ;SET IT
198 ;
199 ;
200 ; LOAD SECTION 'C'
201 000654 ; CPC:
202 000654 016767 000000G.000000G. MOV... LCOUNT,WCOUNT... ;REINITIALIZE WORKING COUNTER
203 000662 005067 000000G. CLR... MSTR2... ;INIT SEQUENCER = 0
204 000666 005304 1$: DEC... R4 ;FINISHED WITH THIS RECORD
205 000670 001007 BNE... 2$ ;NO, CONTINUE
206 000672 CALL... GET... ;READ NEXT
207 000676 103472 BCS... CPCSX... ;ERROR, EXIT
208 000700 016705 000000G. MOV... INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ
209 000704 012704 000400 MOV... #256.,R4 ;R4 = NUMBER OF WORDS IN RECORD
210 ;
211 000710 2$:
212 000710 016746 000000G. MOV... MSTR2,-(SP) ;INITIALIZE SEQUENCER ADDRESS
213 000714 CALL... SEQCS... ;SET ADDRESS
214 000720 012746 000120 MOV... #<CSWRTEN+PLC>,-(SP) ;SELECT SECTION C
215 000724 CALL... CPCRA... ;DIRECT CONTROL WORD TO CP
216 000730 012546 MOV... (R5)+,-(SP) ;GET READY TO MOVE DATA TO MRP
217 000732 CALL... LBCSC... ;DO IT
218 000736 005046 CLR... -(SP) ;CLEAR CP CONTROL REG
219 000740 CALL... CPCR
220 ;

```

```

221 000744 005367 000000G.          DEC.   WCOUNT.          ;FINISHED WITH THIS COLUMN.
222 000750 001403                    BEQ.   CPD.              ;YES, GET NEXT.
223 000752 005267 000000G.          INC.   MSTR2.           ;NO, ADVANCE SEQUENCER ADDRESS
224 000756 000743                    BR.    1$.              ;SET IT.
225                                     ;
226                                     ;
227                                     ;
228 000760                    CPD:
229 000760 016767 000000G.000000G.  MOV.   LCOUNT,WCOUNT. ;REINITIALIZE WORKING COUNTER.
230 000766 005067 000000G.          CLR.   MSTR2.           ;INIT SEQUENCER = 0
231 000772 005304                    1$:   DEC.   R4.            ;FINISHED WITH THIS RECORD.
232 000774 001007                    BNE.   2$.              ;NO, CONTINUE.
233 000776                    CALL.  GET.              ;READ NEXT
234 001002 103430                    BCS.   CPCSX.           ;ERROR, EXIT.
235 001004 016705 000000C.          MOV.   INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ.
236 001010 012704 000400          MOV.   #256.,R4.        ;R4 = NUMBER OF WORDS IN RECORD.
237                                     ;
238 001014                    2$:
239 001014 016746 000000G.          MOV.   MSTR2,-(SP)      ;INITIALIZE SEQUENCER ADDRESS.
240 001020                    CALL.  SEQCS.           ;SET ADDRESS.
241 001024 012746 000130          MOV.   #<<CSWRTEN+PLD>,-(SP) ;SELECT SECTION D.
242 001030                    CALL.  CPCRA.           ;DIRECT CONTROL WORD TO CP.
243 001034 012546                    MOV.   (R5)+,-(SP)      ;GET READY TO MOVE DATA TO MRP
244 001036                    CALL.  LBCSC.           ;DO IT.
245 001042 005046                    CLR.   -(SP)            ;CLEAR CP CONTROL REG.
246 001044                    CALL.  CPRC.
247                                     ;
248 001050 005367 000000G.          DEC.   WCOUNT.          ;FINISHED WITH THIS COLUMN.
249 001054 001403                    BEQ.   CPCSX.           ;YES, ALL DONE.
250 001056 005267 000000G.          INC.   MSTR2.           ;NO, ADVANCE SEQUENCER ADDRESS
251 001062 000743                    BR.    1$.              ;SET IT.
252                                     ;
253 001064                    CPCSX:
254 001064                    CLOSE# #INFDB.
255 001074 105067 000000C.          CLRB.  INDNB+N,FVER.    ;RESET FILE VERSION NUMBER.
256 001100                    RETURN.
257 001102                    RETURN.
258                                     ;
259 000001                    .END.

```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CCKLCK = 000400
CNOBRE = 100000
CPB = 000550R
CPC = 000654R
CPCCEN = 010000
CPCR = 000000 GX
CPCRA = 000000 GX
CPCS = 000336RG
CPCSX = 001064R
CPD = 000760R
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = 000000 GX
CSURTE = 000100
DBR.RD = 000001
DB\$CPP = 001457
DB\$SPT = 000026
DB\$TPC = 000023
DISPSS = 100000
DMAAWR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
F IRST = 000000 GX
FO.RD = 000112
F.BKDS = 000000 GX
F.FACC = 000000 GX
GET = 000000 GX
INDNB = 000000 GX
INFD = 000117
LBCSC = 000000 GX
LBMSC = 000000 GX
LCOUNT = 000000 GX
LCS = 000000 GX
LMM = 000000 GX
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.DU = 000040
MBKALL = 001000
MBKCLK = 000400
MICRO = 000000RG
MMADR = 000100
MMLEFT = 000002
MMDE = 000004
MMURTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MRPCR = 000000 GX
MRPCRA = 000000 GX
002.MRPM1 = 00032RG
002.MRPMX = 000320R
MRPRGT = 000214R
MSTR2 = 000000 GX
MSYN = 000040
002.N = 000144
002.N.FNAM = 000000 GX
002.N.FVER = 000000 GX
PAR\$\$\$ = 000027
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000050
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HFB = 000002
Q\$ICP = 000006
Q\$IH = 000003
Q\$IHL = 000002
Q\$IHPR = 000007
Q\$LBD = 001000
Q\$LDPP = 002000
Q\$LBP = 000001
Q\$LCD = 000003
002.Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
002.Q\$NCLK = 176000
002.Q\$PP = 000100
002.Q\$PPSW = 000320
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 000000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGO.EN = 000200
RGO.VA = 020000
SELECT = 000000 GX
SEQCS = 000000 GX
SEQMM = 000000 GX
SEQ.CI = 000010
Q\$CLR = 000000
Q\$LA = 000001
Q\$OB = 000005
Q\$OR = 000006
Q\$QX = 000004
Q\$SR = 000007
Q\$S1 = 000010
Q\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$OTR = 176346
TD\$ORD = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDR = 176364
T\$AD = 000020
T\$BA = 000002

T#BD = 000010	T#SCLK = 020000	WORD26 = 000064	WORDS2 = 000150	WORD79 = 000236
T#BSO = 100000	T#SEG1 = 000000	WORD27 = 000066	WORDS3 = 000152	WORD8 = 000020
T#BT = 000020	T#SEG2 = 000001	WORD28 = 000070	WORDS4 = 000154	WORD80 = 000240
T#BTAR = 000030	T#SEG3 = 000002	WORD29 = 000072	WORDS5 = 000156	WORD81 = 000242
T#BTD = 000000	T#SO = 000001	WORD3 = 000006	WORDS6 = 000160	WORD82 = 000244
T#CD = 000100	T#UBUS = 100000	WORD30 = 000074	WORDS7 = 000162	WORD83 = 000246
T#CLK = 000000	T#1CLK = 000400	WORD31 = 000076	WORDS8 = 000164	WORD84 = 000250
T#DISK = 000200	T#BBEN = 000020	WORD32 = 000100	WORDS9 = 000166	WORD85 = 000252
T#DRD = 000004	UBD:IN = 000020	WORD33 = 000102	WORD6 = 000014	WORD86 = 000254
T#EMEM = 010000	VIRT = ***** GX	WORD34 = 000104	WORD61 = 000170	WORD87 = 000256
T#FSAA = 000000	WCOUNT = ***** GX	WORD35 = 000106	WORD62 = 000174	WORD88 = 000260
T#FSAB = 000004	WORD0 = 000000	WORD36 = 000110	WORD63 = 000176	WORD89 = 000262
T#FSAC = 000014	WORD1 = 000002	WORD37 = 000112	WORD64 = 000200	WORD9 = 000022
T#FSB2 = 000010	WORD10 = 000024	WORD38 = 000114	WORD65 = 000202	WORD90 = 000264
T#IB = 000026	WORD11 = 000026	WORD39 = 000116	WORD66 = 000204	WORD91 = 000266
T#IBAR = 000024	WORD12 = 000030	WORD4 = 000010	WORD67 = 000206	WORD92 = 000270
T#IBE = 020000	WORD13 = 000032	WORD40 = 000120	WORD68 = 000210	WORD93 = 000272
T#IBF = 040000	WORD14 = 000034	WORD41 = 000122	WORD69 = 000212	WORD94 = 000274
T#ICD = 000040	WORD15 = 000036	WORD42 = 000124	WORD7 = 000016	WORD95 = 000276
T#MODE = 004000	WORD16 = 000040	WORD43 = 000126	WORD70 = 000214	WORD96 = 000300
T#OB = 000036	WORD17 = 000042	WORD44 = 000130	WORD71 = 000216	WORD97 = 000302
T#OBE = 004000	WORD18 = 000044	WORD45 = 000132	WORD72 = 000220	WORD98 = 000304
T#OBF = 010000	WORD19 = 000046	WORD46 = 000134	WORD73 = 000222	WORD99 = 000306
T#OBRA = 000034	WORD2 = 000004	WORD47 = 000136	WORD74 = 000224	WORDVAL = 000310
T#OBWA = 000032	WORD20 = 000050	WORD48 = 000140	WORD75 = 000226	XTREAD = 001000
T#OUTA = 100000	WORD21 = 000052	WORD49 = 000142	WORD76 = 000230	XTWRITE = 000400
T#RBD0 = 000200	WORD22 = 000054	WORDS = 000012	WORD77 = 000232	.CLOSE = ***** G
T#RNB = 000040	WORD23 = 000056	WORDS0 = 000144	WORD78 = 000234	.FINIT = ***** G
T#RSET = 040000	WORD24 = 000060	WORDS1 = 000146		.OPEN = ***** G
T#SC = 000022	WORD25 = 000062			

. ABS: 000000 000
000000 001
MICRO: 001104 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 4015 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:51
MICRO, MICRO / - SP [20, 1] JM, [20, 1] MICRO

```

1          .TITLE .LOADQX.
2 000000   .PSECT .LOADQX.
3          ;
4          ;
5          ;
6          ;
7          ;
8          ;
9 000000   .MCALL .WTSE$S,CLEF$S.
10          LOADQX::
11 000000   005046   CLR . -(SP) ;CLEAR NOTHING IN CSR1
12 000002   012746   176000   MOV . #Q$NCLK,-(SP) ;SET NO-CLOCKS
13 000012   005067   176422   CALL . CSR1
14          ;
15          ;
16          ;
17          ;
18 000016   016746   000000G   SELECT MEMORY TO BE LOADED (WINDOW OR LOCATION).
19 000022   .          .          START THE MICROCODE THAT DOES THE LOADING OF THE QEX.
20 000026   012746   001760   MOV . CODE,-(SP) ;SELECT WINDOW OR LOCATION MEMORY.
21 000032   .          .          CALL . PPCR
22          .          .          MOV . #1760,-(SP) ;START CP MICROCODE AT X'3F0'
23 000036   005046   .          .          CALL . SEQCS
24 000040   .          .          CLR . -(SP) ;RESET BR INHIBIT
25 000044   012746   000377   CALL . CPCR
26 000050   .          .          MOV . #377,-(SP) ;SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
27 000054   005046   .          .          CALL . SEQM1
28 000056   .          .          CLR . -(SP) ;RESET BR INHIBIT
29 000062   012767   001000   176422   CALL . MRPCR
30 000070   012767   022000   176422   MOV . #Q$REBK,QR$CR2 ;RE-ARM INTERRUPTS
31 000076   012746   000360   MOV . #Q$LDPP+Q$ENOP>,QR$CR2 ;SET LOAD PPS MODE + ENABLE INTERRUPTS
32 000102   052716   001001   MOV . #Q$CSEL,-(SP) ;CLEAR ALL SELECTIONS
33 000106   052716   176000   BIS . #Q$LBD+Q$LBP>,(SP) ;CLEAR DRIVE AND PULSE
34 000112   005046   .          .          BIS . #Q$NCLK,(SP) ;CLEAR ALL NO-CLOCKS
35 000114   .          .          CLR . -(SP) ;SET NOTHING
36          .          .          CALL . CSR1
37          ;
38          ;
39          ;
40 000120   .          .          WAIT FOR INTERRUPT FROM CP.
41          .          .          USE DMA INTERRUPT (SEE CP MICROCODE FOR QEX)
42 000132   .          .          WTSE$S . #EFN,3
43          .          .          GLEF$S . #EFN,3
44          ;
45          ;
46 000144   012767   100400   176422   RE-ARM INTERRUPTS.
47 000152   012767   101000   176422   MOV . #Q$SM+Q$CHB>,QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
48 000160   012767   160000   176422   MOV . #Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
49          .          .          MOV . #Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
50 000166   005046   .          .          ;
51 000170   012746   176000   1$: CLR . -(SP) ;CLEAR NOTHING IN CSR1
52 000174   .          .          MOV . #Q$NCLK,-(SP) ;SET NO-CLOCKS
53 000200   005067   176422   CALL . CSR1
54 000204   .          .          CLR . QR$CR2 ;SET LOAD MODE
55          .          .          RETURN
56          .          .          .END

```


ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PPCR = ***** GX	Q\$RNC = 006000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR\$CR1 = 176420	Q\$RSC = 004000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR\$CR2 = 176422	Q\$RSET = 000010
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	QR\$LBR = 176424	Q\$SM = 100000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$ATTN = 000100	Q\$SP = 000120
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$BCL = 000001	Q\$SP2 = 000340
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CCCP = 000040	RGQ.EN = 000200
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CHB = 000400	RGQ.VA = 020000
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$CHRL = 000200	SEOC5 = ***** GX
BIT13 = 020000	BYTE51 = 000063	CNOBRE = 100000	Q\$CLR = 000040	SEQMM = ***** GX
BIT14 = 040000	BYTE52 = 000064	CODE = ***** GX	Q\$CNC = 030000	SEQ.CI = 000010
BIT15 = 100000	BYTE53 = 000065	CPCCEN = 010000	Q\$CP = 000060	S\$CLR = 000000
BIT2 = 000004	BYTE54 = 000066	CPCR = ***** GX	Q\$CPCC = 000010	S\$LA = 000001
BIT3 = 000010	BYTE55 = 000067	CPREAD = 040000	Q\$CP2 = 000260	S\$OB = 000005
BIT4 = 000020	BYTE56 = 000070	CPWRTE = 020000	Q\$CSC = 010000	S\$OR = 000006
BIT5 = 000040	BYTE57 = 000071	CSADRD = 000004	Q\$CSEL = 000360	S\$QX = 000004
BIT6 = 000100	BYTE58 = 000072	CSEDCI = 100000	Q\$CSET = 000002	S\$SR = 000007
BIT7 = 000200	BYTE59 = 000073	C\$OE = 000040	Q\$CSP = 020000	S\$S1 = 000010
BIT8 = 000400	BYTE60 = 000074	CSR1 = ***** GX	Q\$DMA = 000001	S\$S2 = 000014
BIT9 = 001000	BYTE61 = 000075	CSURTE = 000100	Q\$ENBK = 040000	TD\$CTR = 176370
BYTE0 = 000000	BYTE62 = 000076	DBR.RD = 000001	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE1 = 000001	BYTE63 = 000077	DB\$CPP = 001457	Q\$FAL = 004000	TD\$INL = 004000
BYTE10 = 000012	BYTE64 = 000100	DB\$SPT = 000026	Q\$FC = 000045	TD\$MEM = 000270
BYTE11 = 000013	BYTE65 = 000101	DB\$TPC = 000023	Q\$FO = 000044	TD\$NTR = 176344
BYTE12 = 000014	BYTE66 = 000102	DISPGS = 100000	Q\$FP = 000046	TD\$OTR = 176346
BYTE13 = 000015	BYTE67 = 000103	DMAAWR = 000005	Q\$HBF = 000002	TD\$QDR = 000274
BYTE14 = 000016	BYTE68 = 000104	DMARRD = 000003	Q\$ICP = 000006	TD\$SW = 176376
BYTE15 = 000017	BYTE69 = 000105	DMARWR = 000004	Q\$IHB = 000003	TD\$TAR = 176372
BYTE16 = 000020	BYTE70 = 000106	EFN3 = ***** GX	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE17 = 000021	BYTE71 = 000107	ENBR = 010000	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE18 = 000022	BYTE72 = 000110	LOADQX = 000000RG	Q\$LBD = 001000	TD\$TDW = 176364
BYTE19 = 000023	BYTE73 = 000111	LOC.EN = 000100	Q\$LBDP = 001001	T\$AD = 000020
BYTE2 = 000002	BYTE74 = 000112	LOC.WA = 040000	Q\$LBP = 000001	T\$BA = 000002
BYTE20 = 000024	BYTE75 = 000113	LOC.WB = 100000	Q\$LDCD = 000003	T\$BD = 000010
BYTE21 = 000025	BYTE76 = 000114	MAREN1 = 000001	Q\$LDMD = 000004	T\$BSO = 100000
BYTE22 = 000026	BYTE77 = 000115	MAREN2 = 004000	Q\$LDPP = 002000	T\$BT = 000020
BYTE23 = 000027	BYTE78 = 000116	MARL0D = 010000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE24 = 000030	BYTE79 = 000117	MAROUT = 000002	Q\$MNC = 140000	T\$BD = 002000
BYTE25 = 000031	BYTE80 = 000120	MAR.LO = 002000	Q\$MR = 000052	T\$CD = 000100
BYTE26 = 000032	BYTE81 = 000121	MAR.OU = 000040	Q\$MRP = 000040	T\$CLK = 002000
BYTE27 = 000033	BYTE82 = 000122	MBKALL = 001000	Q\$MRP2 = 000240	T\$D ISK = 000200
BYTE28 = 000034	BYTE83 = 000123	MBKCLK = 000400	Q\$MSC = 040000	T\$DRD = 000004
BYTE29 = 000035	BYTE84 = 000124	MMADR = 000100	Q\$MSET = 000004	T\$EMEM = 010000
BYTE3 = 000003	BYTE85 = 000125	MMLLEFT = 000002	Q\$MSP = 100000	T\$FSAA = 000000
BYTE30 = 000036	BYTE86 = 000126	MNODE = 000004	Q\$NCLK = 176000	T\$FSAB = 000004
BYTE31 = 000037	BYTE87 = 000127	MNRWTE = 000010	Q\$PP = 000100	T\$FSAC = 000014
BYTE32 = 000040	BYTE88 = 000130	MNOBRE = 100000	Q\$PPSW = 000320	T\$FSB2 = 000010
BYTE33 = 000041	BYTE89 = 000131	MREN1 = 000001	Q\$PP2 = 000300	T\$IB = 000026
BYTE34 = 000042	BYTE90 = 000132	MREN2 = 020000	Q\$QHLT = 000013	T\$IBAR = 000024
BYTE35 = 000043	BYTE91 = 000133	MRPCR = ***** GX	Q\$QL = 000043	T\$IBE = 020000
BYTE36 = 000044	BYTE92 = 000134	MSYN = 000040	Q\$QLA = 000053	T\$IBF = 040000
BYTE37 = 000045	BYTE93 = 000135	N = 000144	Q\$QLB = 000054	T\$ICD = 000040
BYTE38 = 000046		PLB = 000010	Q\$QLR = 000001	T\$MODE = 004000
BYTE39 = 000047		PLC = 000020	Q\$QW = 000042	T\$OB = 000036
BYTE4 = 000004		PLD = 000030	Q\$RDCD = 000005	T\$OBE = 004000
BYTE40 = 000050		PLRWR = 000200	Q\$RDMD = 000006	T\$OBFA = 010000
BYTE41 = 000051		PLR.EN = 000200	Q\$REBK = 001000	T\$OBRA = 000034

T#0BWA = 000032	WORD17 = 000042	WORD39 = 000116	WORD60 = 000170	WORD81 = 000242
T#OUTA = 100000	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD82 = 000244
T#RBD0 = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD83 = 000246
T#RNB = 000040	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD84 = 000250
T#RSET = 040000	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD85 = 000252
T#SC = 000022	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD86 = 000254
T#SCLK = 020000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD87 = 000256
T#SEG1 = 000000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD88 = 000260
T#SEG2 = 000001	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD89 = 000262
T#SEG3 = 000002	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD9 = 000022
T#SO = 000001	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD90 = 000264
T#UBUS = 100000	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD91 = 000266
T#1CLK = 000400	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD92 = 000270
T#BEN = 000020	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD93 = 000272
UBD:IN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD94 = 000274
WORD0 = 000000	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD95 = 000276
WORD1 = 000002	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD96 = 000300
WORD10 = 000024	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD97 = 000302
WORD11 = 000026	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD98 = 000304
WORD12 = 000030	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORD99 = 000306
WORD13 = 000032	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WRDVAL = 000310
WORD14 = 000034	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	XTREAD = 001000
WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240	XTWRITE = 000400
WORD16 = 000040	WORD38 = 000114	WORD6 = 000014		

. ABS: 000000 000
000000 001
LOADQX: 000206 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3258 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
LOADQX, LOADQX, -SP=[20, 1]IM, [20, 1]LOADQX

LOADER.TSK MEMORY ALLOCATION: MOP, TKB
27-MAR-80 16:15

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

TASK NAME : LOADER
PARTITION NAME : GEN
IDENTIFICATION : 08
TASK UIC : [7.5]
TASK PRIORITY : 100
STACK LIMITS : 040212-041211 001000 00512
PRG XFR ADDRESS : 044070
TASK ATTRIBUTES : AL,CP
TOTAL ADDRESS WINDOWS : 2
TASK IMAGE SIZE : 6432 WORDS
TASK ADDRESS LIMITS : 040000 071067
R-W DISK BLK LIMITS : 000042-000073 000032-00026

*** ROOT SEGMENT: LOADER

R/W MEM LIMITS : 040000 071067 031070 12856
DISK BLK LIMITS : 000042-000073 000032-00026

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK : (RW, I, LCL, REL, CON)	041212-014334	06364	
CD : (RW, I, LCL, REL, CON)	041212-007476	03902	LOADER
CPSUB : (RW, I, LCL, REL, CON)	055546-000274	00188	CD
DMASUB : (RW, I, LCL, REL, CON)	056042-000502	00322	CPSUB
LOADQX : (RW, I, LCL, REL, CON)	056544-000276	00190	DMASUB
MICRO : (RW, I, LCL, REL, CON)	057042-000206	00134	LOADQX
MRPSUB : (RW, I, LCL, REL, CON)	057250-001104	00580	MICRO
PPSUB : (RW, I, LCL, REL, CON)	060354-000506	00326	MRPSUB
SPSUB : (RW, I, LCL, REL, CON)	061062-000766	00502	PPSUB
\$\$\$FSR1 : (RW, D, GBL, REL, OVR)	061062-000766	00502	SPSUB
\$\$\$FSR2 : (RW, D, GBL, REL, CON)	062050-000324	00212	\$\$\$FSR1
\$\$\$RESL : (RW, I, LCL, REL, CON)	062050-000324	00212	\$\$\$FSR2
\$\$\$RESM : (RW, I, LCL, REL, CON)	062374-001020	00520	\$\$\$RESL
	062374-001020	00520	LOADER
	063414-000104	00068	
	063520-005346	02790	
	132000-007656	04014	

GLOBAL SYMBOLS:

ABEND2-050366-R BINWD-041360-R CPCR-056454-R CPLB-056404-R DATA1-041374-R EFBUF-041222-R ERR2-050444-R
ADDR-041376-R CD-055546-R CPCRA-056462-R CSR1-047560-R DMA-056636-R EFN.3-000003 FIRST-000001
APLACE-041364-R CODE-041354-R CPCS-057606-R DATA-041374-R DMASET-056544-R EFN.33-000041 GET-047420-R

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

LOADER.TSK:5 MEMORY ALLOCATION MAP. TKB
LOADER 27-MAR-80 16:10

PAGE 2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

INDNB	044032-R	LBMRP	060526-R	LCOUNT	041366-R	MRPCRA	061000-R	QXHIGH	042534-R	SPLB	062244-R	\$CSTA	004226
INFDB	043672-R	LBMSC	060624-R	LCS	041242-R	MRPLB	060722-R	SCOUNT	041372-R	TRANSF	041400-R	\$DRDSE	017134
INSAVE	041402-R	LBPP	061062-R	LMD	041236-R	MRPMM	057302-R	SELECT	041362-R	VIRT	041342-R	\$TKTCB	004026
ID:UVB	011000	LBPSC	061160-R	LMM	041232-R	MSTR2	041376-R	SELPG	061410-R	WCCOUNT	041370-R		
LAST	000004	LBSP	062050-R	LOADOX	057042-R	PPCR	061326-R	SEDCS	056042-R	WRTCS	056130-R		
LBCP	056210-R	LBSSC	062146-R	MICRO	057250-R	PPLB	061256-R	SEOMM	060354-R	WRTMM	060442-R		
LBCSC	056306-R	LCD	041246-R	MRPCR	060772-R	PRINT	042736-R	SPCR	062312-R	\$CEFI	005174		

*** TASK-BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 17496.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 6634. WORDS (25. PAGES)
SIZE OF WORK FILE: 2560. WORDS (10. PAGES)

ELAPSED TIME: 00:00:19

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

.TITLE..QMT.

QUERY RESOLVER DIAGNOSTICS
MAIN MODULE

THIS MODULE EXECUTES TEST CYCLES. ALL CONTROL INFORMATION FOR A TEST CYCLE IS OBTAINED FROM THE COMMAND LINE. QMT BUILDS TABLES AND SETS FLAGS IN ORDER THAT THERE NEED BE NO USER INTERVENTION DURING THE EXECUTION OF THE CYCLE. THE PROMPTS FOR COMMAND LINE INPUT AND GENERAL QMT ACTION ARE:

1. PROMPT FOR ALL TESTS, ALL MEMORIES, FULL RANGE. IF THE ANSWER IS 'Y', QMT BUILDS A COMPLETE MEMORY TEST CYCLING TABLE, THEN PROCEEDS TO 7. (FOR THE STRUCTURE OF THE TABLE, SEE BELOW REF. TABLE - CURRENT JUMP TABLE).
- IF THE ANSWER IS 'N', PROCEED TO 2 FOR MORE SELECTIVE PROMPTS.
2. PROMPT FOR REGISTERS TO BE TESTED. QMT SCANS THE COMMAND LINE RESPONSE AND SETS A FLAG FOR EACH REGISTER MNEMONIC IT FINDS.
3. PROMPT FOR MEMORIES TO BE TESTED. QMT SCANS THE COMMAND LINE RESPONSE AND SETS A FLAG FOR EACH MEMORY MNEMONIC IT FINDS.
4. PROMPT FOR MEMORY LIMITS IF ANY MEMORIES WERE SELECTED. IF A MEMORY WAS SELECTED IN 3 (IE. ITS FLAG WAS SET), QMT PROMPTS FOR NUMERICAL VALUES WHICH DEFINE THAT PORTION OF A MEMORY ON WHICH THE TESTS ARE TO BE RUN. QMT PUTS THE MEMORY LIMITS FROM THE COMMAND LINE RESPONSE INTO A TABLE.
5. PROMPT FOR REGISTER TESTS IF ANY REGISTERS WERE SELECTED. QMT SCANS THE COMMAND LINE RESPONSE FOR TEST NUMBERS. THEN, FOR EACH WHOSE FLAG IS UP, QMT BUILDS AN ENTRY IN THE REGISTER CURRENT JUMP TABLE (SEE BELOW).
6. PROMPT FOR MEMORY TESTS IF ANY MEMORIES WERE SELECTED. QMT SCANS THE COMMAND LINE RESPONSE FOR TEST NUMBERS. THEN, FOR EACH WHOSE FLAG IS UP, QMT BUILDS AN ENTRY IN THE MEMORY CURRENT JUMP TABLE (SEE BELOW).
7. PROMPT FOR LOOP ON TEST. QMT SETS A FLAG THAT DETERMINES WHETHER A TEST CYCLE WILL BE EXECUTED ONCE OR EXECUTED REPEATEDLY.
7. PROMPT FOR ERROR OPTIONS. QMT SETS FLAGS WHICH DETERMINE WHAT ACTION WILL BE TAKEN IN THE EVENT OF AN ERROR.
8. PUT OUT DIRECTIONS FOR STOPPING TEST. QMT ALLOWS THE TEST CYCLE TO BE INTERRUPTED BY AN UNSOLICITED CHARACTER INTERRUPT FROM THE TERMINAL.
9. TEST CYCLE BEGINS. NO MORE PROMPTING.

REFERENCE TABLE - CURRENT JUMP TABLE

THE ACTUAL EXECUTION OF A TEST CYCLE DEPENDS UPON THE CONTENTS OF THE CURRENT JUMP TABLE. THERE ARE TWO REF.

```

58      ;      TABLES AND TWO CURRENT JUMP TABLES, ONE EACH FOR REGISTERS
59      ;      AND ONE EACH FOR MEMORIES. SINCE BOTH CURRENT JUMP TABLES
60      ;      ARE BUILT IN A SIMILAR FASHION, THE FOLLOWING EXPLANATION
61      ;      WILL CHOOSE THE MEMORY TABLES FOR ILLUSTRATION. FOR THE
62      ;      EXECUTION OF A TEST CYCLE, THE REGISTER AND MEMORY TABLES
63      ;      ARE LOGICALLY COMBINED. THE METHOD FOR DOING THIS IS DESCRIBED
64      ;      BELOW (SEE 'MTSET').
65      ;
66      ;
67      ;      EACH MEMORY TEST 1 - 13 HAS ITS OWN CONTROL ROUTINE IN QMT.
68      ;      THE REFERENCE TABLE ENTRIES ARE THE ADDRESSES OF THESE
69      ;      CONTROL ROUTINES. FOR EACH TEST, THE REFERENCE TABLE
70      ;      CONTAINS IN CONTIGUOUS POSITIONS THE CONTROL ROUTINE
71      ;      ADDRESS REPLICATED A NUMBER OF TIMES. THE NUMBER OF
72      ;      REPLICATIONS IS EQUAL TO THE NUMBER OF MEMORIES THAT
73      ;      CAN BE TESTED (IE, 12, THE VALUE OF THE EQUATE 'NMEMS').
74      ;      SO, FOR EXAMPLE, SINCE THERE ARE 12 MEMORIES, TEST 1'S
75      ;      CONTROL ROUTINE ADDRESS WILL BE REPEATED IN THE REFERENCE
76      ;      TABLE 12 TIMES:
77      ;
78      ;      .WORD  T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1
79      ;
80      ;      EACH ADDRESS HERE IS A PLACE-HOLDER FOR A MEMORY. THE
81      ;      FOLLOWING IS A TABLE OF PLACE-HOLDER VALUES (POSITIONS OF
82      ;      ADDRESSES IN THE ABOVE LINE) AND THE MEMORIES TO WHICH THEY
83      ;      CORRESPOND:
84      ;
85      ;      0      MRP MICROPGM MEMORY.
86      ;      1      MRP DATA MEMORY.
87      ;      2      QEX WINDOW MEMORY.
88      ;      3      QEX LOCATION MEMORY.
89      ;      4      CP CONTROL STORE.
90      ;      5      CP DATA MEMORY.
91      ;      6      FAL POINTER MEMORY.
92      ;      7      FAL COUNTER MEMORY.
93      ;      8      QLB REFERENCE PAGE.
94      ;      9      QLB PAGE 0.
95      ;      10     QLB PAGE 1.
96      ;      11     QLB PAGE 2.
97      ;
98      ;      THE IDEA HERE IS THAT RATHER THAN HAVING SEPARATE CONTROL
99      ;      ROUTINES FOR EACH MEMORY FOR EACH TEST, QMT CAN MAKE
100     ;      USE OF THE POSITIONS OF ADDRESSES IN THE TABLE.
101     ;
102     ;      THE FILLING OF THE CURRENT JUMP TABLE TAKES PLACE AS
103     ;      FOLLOWS: A UNIQUE FLAG IS SET IN A FLAG WORD FOR EACH
104     ;      MEMORY WHOSE MNEMONIC QMT ENCOUNTERS IN THE COMMAND
105     ;      LINE RESPONSE. THE THE PROMPT 'SELECT MEMORIES', EACH
106     ;      POSITION IN THE FLAG WORD (0 - 11) CORRESPONDS TO A
107     ;      MEMORY PLACE HOLDER POSITION IN THE REF TABLE AND
108     ;      CURRENT JUMP TABLE. EG. THE FLAG FOR QLB PAGE 0 IS IN
109     ;      POSITION 9 IN THE FLAG WORD AND THE PLACE-HOLDER
110     ;      POSITION FOR QLB PAGE 0 IS 9 (SEE ABOVE). THEN FOR EACH
111     ;      TEST NUMBER QMT ENCOUNTERS IN THE COMMAND LINE RESPONSE TO
112     ;      THE PROMPT 'SELECT MEMORY TEST(S)', QMT MOVES THE ADDRESS
113     ;      OF THAT TEST'S CONTROL ROUTINE FROM THE REF TABLE TO THE
114     ;      CURRENT JUMP TABLE DEPENDING UPON THE MEMORY FLAG SETTINGS.
115     ;      IE, QMT SCANS THE MEMORY FLAG WORD AND FOR EVERY BIT SET MOVES

```

```

115      ; AN ADDRESS FROM THE REF TABLE TO THE CURRENT JUMP TABLE.
116      ;
117      ; EXAMPLE:
118      ; IF QMT ENCOUNTERS A '1' IN THE COMMAND LINE RESPONSE TO
119      ; 'SELECT MEMORY TEST(S)' AND THE FLAGS FOR MRP DATA MEMORY
120      ; AND QLB PAGE 1 HAVE BEEN PREVIOUSLY SET, QMT WILL MOVE
121      ; TEST 1'S CONTROL ROUTINE ADDRESS FROM THE REF TABLE TO THE
122      ; CURRENT JUMP TABLE IN POSITIONS 1 AND 10 FOR TEST 1.
123      ;
124      ; REF TABLE:
125      ; .WORD  T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1
126      ; .WORD  T2,T2,T2,T2,T2,T2,T2,T2,T2,T2,T2,T2
127      ;
128      ; CURRENT JUMP TABLE:
129      ; .WORD  0,T1,0,0,0,0,0,0,0,T1,0
130      ; .WORD  0,0,0,0,0,0,0,0,0,0,0,0
131      ;
132      ; TEST CYCLING
133      ; CYCLING IS DESCRIBED HERE AS IF ONLY MEMORIES ARE
134      ; BEING TESTED. THE ADJUSTMENTS NECESSARY IF EITHER
135      ; ONLY REGISTERS OR BOTH REGISTERS AND MEMORIES ARE
136      ; BEING TESTED ARE DESCRIBED BELOW AT 'MTSET' AND
137      ; 'JMPMT'.
138      ;
139      ; THE CYCLE CONTROL ROUTINE MAINTAINS A POINTER TO THE
140      ; CURRENT JUMP TABLE. THE CONTROL ROUTINE SCANS THE TABLE
141      ; UNTIL IT FINDS A NON-ZERO ENTRY. IT DERIVES THE MEMORY
142      ; TO BE TESTED FROM THE PLACE-HOLDING CHARACTERISTICS OF
143      ; THE TABLE:
144      ;
145      ; 1. GETS THE POSITION OF THE CURRENT NON-ZERO ENTRY
146      ;    RELATIVE TO THE BEGINNING OF THE TABLE.
147      ; 2. DIVIDES THIS NUMBER BY THE NUMBER OF MEMORIES. THE
148      ;    REMAINDER GIVES THE PLACE-HOLDER VALUE.
149      ;
150      ; FOR EXAMPLE, IF 'QEX WINDOW' AND 'TEST 2' HAVE BEEN
151      ; SELECTED, THE CURRENT JUMP TABLE WILL CONTAIN THE
152      ; FOLLOWING INFORMATION:
153      ;
154      ; .WORD  0,0,0,0,0,0,0,0,0,0,0,0
155      ; .WORD  0,0,0,T2,0,0,0,0,0,0,0,0
156      ;
157      ; THE FIRST NON-ZERO ENTRY IN THE TABLE IS AT OFFSET 15.
158      ; THE REMAINDER FROM THE DIVISION OF 15 BY 12 (12 = NUMBER
159      ; OF MEMORIES) IS 3. THE VALUE 3 IS THE PLACE-HOLDER
160      ; VALUE FOR THE QEX WINDOW MEMORY.
161      ;
162      ; THE CONTROL ROUTINE PASSES CONTROL TO THE TEST CONTROL
163      ; ROUTINE WHOSE ADDRESS IS THE CURRENT NON-ZERO ENTRY IN
164      ; THE CURRENT JUMP TABLE. THE CYCLE CONTROL ROUTINE PASSES
165      ; THE REMAINDER FROM THE ABOVE DIVISION IN R0. THE TEST
166      ; CONTROL ROUTINE USES THE CONTENTS OF R0 AS AN INDEX
167      ; INTO A TABLE OF QMT SUB-MODULE MEMORY TEST ADDRESSES.
168      ; THE TEST CONTROL ROUTINE IN TURN PASSES CONTROL TO THE
169      ; ROUTINE IN THE SUB-MODULE THAT WILL EXECUTE THE TEST ON
170      ; THE CORRECT MEMORY.
171      ;

```

```

172.      ;
173.      ;
174.      ; THE SUB-MODULES OF QMT ARE:
175.      ; MMTEST.      TEST MRP MICROPGM MEMORY.
176.      ; MDTEST.      TEST MRP DATA MEMORY.
177.      ; QXTEST.      TEST QEX MEMORIES.
178.      ; CSTEST.      TEST CP CONTROL STORE.
179.      ; CDTEST.      TEST CP DATA MEMORY.
180.      ; FATEST.      TEST FAL MEMORIES.
181.      ; QRTEST.      TEST QLB REFERENCE PAGE.
182.      ; QBTEST.      TEST QLB MEMORIES.
183.      ;
184.      ; THE MODULE RTEST CONTAINS THE REGISTER TESTS.
185.      ; THERE ARE MODULES WHICH CONTAIN SUBROUTINES FOR THE
186.      ; QMT SUB-MODULES. THESE SUBROUTINE MODULES ARE
187.      ; MRPSUB, CPSUB, AND PPSUB. ROUTINES IN THESE MODULES
188.      ; ARE GLOBAL, ALLOWING CROSS-USAGE.
189.      ;
190.      ; ALL ROUTINES IN THE QMT SUB-MODULES RETURN TO THE TEST
191.      ; CONTROL ROUTINE THAT CALLED THEM. THE TEST CONTROL ROUTINES
192.      ; RETURN TO THE CYCLING ROUTINE WHICH SCANS THE CURRENT JUMP
193.      ; TABLE FOR THE NEXT NON-ZERO ENTRY.
194.      ;
195.      ;
196.      ; EXIT FROM THE PROGRAM DEPENDS UPON THE STATUS OF THE CURRENT
197.      ; JUMP TABLE, LOOP OPTIONS, HALT OPTIONS, OR TERMINAL INPUT.
198.      ;
199.      ; JUMP TABLE EMPTY.      - EXIT.
200.      ; LOOP OPTION OFF.      - EXECUTE ONE TEST CYCLE.
201.      ; LOOP COUNT.            - EXECUTE A NUMBER OF CYCLES EQUAL TO
202.      ;                        THE LOOP COUNT.
203.      ; HALT OPTION ON.        - HALT AFTER ONE ERROR.
204.      ; COUNT + 'H'.          - PRINT A NUMBER OF MESSAGES EQUAL TO
205.      ;                        THE COUNT AND HALT.
206.      ;
207.      ; WHILE THE TESTS ARE RUNNING, THE ENTERING FROM THE TERMINAL
208.      ; OF ANY CHARACTER OTHER THAN W, C, P, OR T (THESE HAVE SPECIAL
209.      ; MEANINGS - SEE THE ROUTINE 'AST') STOPS THE TESTS IMMEDIATELY.
210.      ;
211.      ;
212.      ; REGISTER TESTS.
213.      ;
214.      ; REGISTERS:
215.      ; MRP MEMORY ADDRESS REGISTER.
216.      ; CP MEMORY ADDRESS REGISTER.
217.      ;
218.      ; TESTS:
219.      ; TEST 01 WRITE ZEROS.
220.      ; TEST 02 WRITE ONES.
221.      ; TEST 03 WRITE USER SUPPLIED TEST PATTERN.
222.      ; TEST 04 INCREMENT MAR.
223.      ;
224.      ;
225.      ; MEMORY TESTS
226.      ;
227.      ; MEMORIES:
228.      ; MRP MICROPROGRAM MEMORY.
229.      ; MRP DATA MEMORY.

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

229      ; QEX WINDOW MEMORY.
230      ; QEX LOCATION MEMORY.
231      ; CP CONTROL STORE.
232      ; CP DATA MEMORY.
233      ; FAL POINTER MEMORY.
234      ; FAL COUNTER MEMORY.
235      ; QLB REFERENCE PAGE.
236      ; QLB PAGE 0
237      ; QLB PAGE 1
238      ; QLB PAGE 2.
239      ;
240      ;
241      ; TESTS:
242      ; TEST 01 WRITE ADDRESS INTO ITSELF.
243      ; TEST 02 WRITE ZEROS.
244      ; TEST 03 WRITE ONES.
245      ; TEST 04 WRITE AAAA.
246      ; TEST 05 WRITE CCCC AND 3333
247      ; TEST 06 CROSS-TALK TEST.
248      ; TEST 07 WRITE ADDRESS COMPLEMENT INTO ADDRESS.
249      ; TEST 08 WRITE 00FF AND FF00
250      ; TEST 09 SHIFT-BIT TEST.
251      ; TEST 10 WRITE USER SUPPLIED TEST PATTERN.
252      ; TEST 11 BIT MARCH TEST.
253      ; TEST 12 ADDRESSING TEST.
254      ; TEST 13 FILE COMPARE TEST.
255      ;
256      ;
257      ; ASSEMBLY: FROM [5.3].
258      ; MCR>MAC QMT,LP=IM04.QMT.
259      ;
260      ; TASK BUILD: ON HQR PACK.
261      ; QMT/DA,QMT=QMT,RTTEST,MMTEST,MDTEST,QXTEST,CSSTEST,CDTEST,
262      ; FATEST,QRTEST,QBTEST,MRPSUB,CPSUB,PPSUB.
263      ; /
264      ; PAR=PAR14K.
265      ; TASK=...QMT.
266      ; ASG=TT0:1
267      ; ASG=TT0:2.
268      ; /
269      ; //

```

```

270 ;
271 ;
272 LOCAL DATA AREAS
273 ;
274 ;
275 .MCALL Q10W$, Q10$, EXIT$, ABRT$, GCML$, GCMLB$, FSRSZ$, CLEF$, ASTX$,
276 .MCALL FDBDF$, FDRC$, FDBK$, FDFP$, NMBLK$, OPEN$, F INIT$, CLOSE$, READ$
277 .MCALL WTSE$
278 ;
279 ;
280 LUN.TT = 1 ;LUN FOR TERMINAL
281 EFN.1 = 1 ;EVENT FLAG FOR TERMINAL
282 EFN.3 = 3 ;EVENT FLAG FOR HDR INTERRUPTS
283 CMILUN = 2 ;LUN FOR GCML
284 INLUN = 3 ;LUN FOR LDXX DAT FILES
285 ;
286 ;
287 SETTINGS FOR FLAG 'BASE'
288 ;
289 LOOP = 1 ;LOOP ON TEST
290 TEST3 = 2 ;USER PATTERN REGISTER TEST SELECTED
291 TEST6 = 4 ;CROSS-TALK TEST SELECTED
292 TEST10 = 10 ;USER PATTERN MEMORY TEST SELECTED
293 REGSTR = 20 ;REGISTERS SELECTED
294 MEMORY = 40 ;MEMORIES SELECTED
295 ALLTST = 100 ;ALL MEMORY TESTS, ALL MEMORIES
296 HALT = 200 ;HALT FLAG
297 ERROR = 400 ;ERROR ENCOUNTERED
298 FIRST = 1000 ;FIRST RECORD READ IN LDXX FILE
299 ;
300 SELECTION FLAGS
301 TO BE USED WITH FLAG WORDS 'RSEL' (REGISTERS) AND 'MSEL' (MEMORIES)
302 ;
303 MA = 1 ;MRP MAR
304 CA = 2 ;CP MAR
305 MM = 1 ;MRP MICROPGM MEMORY
306 MD = 2 ;MRP DATA MEMORY
307 QW = 4 ;QEX WINDOW MEMORY
308 QL = 10 ;QEX LOCATION MEMORY
309 CS = 20 ;CP CONTROL STORE
310 CD = 40 ;CP DATA MEMORY
311 FP = 100 ;FAL POINTER MEMORY
312 FC = 200 ;FAL COUNTER MEMORY
313 QR = 400 ;QLB REFERENCE PAGE
314 Q0 = 1000 ;QLB PAGE 0
315 Q1 = 2000 ;QLB PAGE 1
316 Q2 = 4000 ;QLB PAGE 2
317 ;
318 .NLIST BEX
319 MYSELF : .RAD50 /...QMT/
320 TSKTCB : .WORD 0 ;TCB OF MY TASK
321 OLDVEC : .WORD 0 ;OLD VECTOR AT 274
322 ASTWRD : .WORD 0 ;UNSOLICITED CHAR FROM TERMINAL
323 STAT : .BLKW 2 ;IO STATUS
324 ERWORD : .WORD 0 ;MESSAGE INDEX
325 ERLIM : .WORD 0 ;ERROR MESSAGE COUNT
326 BINWD : .WORD 0 ;VALUE CONVERTED FROM COMMAND LINE
327 LOWER : .WORD 0 ;LOWER MEMORY LIMITS HOLD AREA

```

```

327 000026 000000          UPPER: .WORD 0          ;UPPER MEMORY LIMITS HOLD AREA
328 000030 000000          BASE: .WORD 0          ;ALL-PURPOSE FLAG WORD
329 000032 000000          APLACE: .WORD 0       ;WORK AREA FOR CSR1 ROUTINE
330 000034 000000          DATA1: .WORD 0
331 000036 000000 000001  VIRT: .WORD 0,1       ;RELATIVE BLOCK IN FILE
332 000042 000000          FVER: .WORD 0         ;FILE VERSION NO.
333 000044 000000          LCOUNT: .WORD 0     ;WORD COUNT IN RECORD
334 000046 000000          WCOUNT: .WORD 0    ;WORKING COUNTER
335 000050 000000          QXCODE: .WORD 0     ;QEX MEMORY SELECT
336 000052 000000          FACODE: .WORD 0     ;FAL MEMORY SELECT
337 000054 000000          QBPAGE: .WORD 0    ;QB PAGE SELECT
338 000056 000000          RT3: .WORD 0       ;USER PATTERN REGISTER TEST
339 000060 000000          MT10: .WORD 0      ;USER PATTERN MEMORY TEST
340 000062          GCMBUF: .BLKW 41   ;TERMINAL INPUT
341 000204 000000          GCMLEN: .WORD 0    ;COMMAND LINE LENGTH
342 000206 000000          GCMPT: .WORD 0    ;POINTER TO COMMAND LINE
343 000210 000000          PASSH: .WORD 0    ;PASS COUNT UPPER WORD
344 000212 000001          PASS: .WORD 1     ;PASS COUNT LOWER WORD
345 000214 000000          LOOPCT: .WORD 0   ;LOOP COUNT
346 000216 000000          MSEL: .WORD 0    ;MEMORY SELECT FLAG WORD
347 000220 000000          RSEL: .WORD 0    ;REGISTER SELECT FLAG WORD
348 000222 000000          MTPNT: .WORD 0   ;CURRENT JUMP TABLE POINTER
349 000224 000000          MTCNT: .WORD 0   ;CURRENT JUMP TABLE COUNT
350 000226 000000          NXTPNT: .WORD 0  ;CURRENT JUMP TABLE POINTER - REFRESH MTPNT
351 000230 000000          NXTCNT: .WORD 0  ;CURRENT JUMP TABLE COUNT - REFRESH MTCNT
352          ;
353          ;
354          ;
355 000232          ;
356          ;
357 000313          ;
358 000432          ;
359          ;
360          ;
361          ;
362 000432          ;
363          ;
364 000512          ;
365 000533          ;
366 000533          ;
367 000632          ;
368          ;
369          ;
370          ;
371 000632          ;
372          ;
373          ;
374          ;
375          ;
376 000652          ;
377 000005          ;
378 000657          ;
379 000027          ;
380          ;
381          ;
382          ;
383          ;

```

```

384 ; VALID REGISTER MNEMONICS AND REGISTER SELECT FLAG SETTINGS.
385 ; USED IN SETTING FLAG WORD 'RSEL' FOR REGISTERS TO BE TESTED.
386 ;
387 000706 RTBL:
388 000706 115 101 .ASCII /MA/ ;MRP-MAR.
389 000710 000001 .WORD MA.
390 000712 103 101 .ASCII /CA/ ;CP-MAR.
391 000714 000002 .WORD CA.
392 000004 RTBLN: = <.-RTBL/2>
393 ;
394 ;
395 ; MEMORY TABLE.
396 ; VALID MEMORY MNEMONICS AND MEMORY SELECT FLAG SETTINGS.
397 ; USED IN SETTING FLAG WORD 'MSEL' FOR MEMORIES TO BE TESTED.
398 ;
398 000716 MTBL:
399 000716 115 115 .ASCII /MM/ ;MRP-MICROPGM-MEMORY
400 000720 000001 .WORD MM.
401 000722 115 104 .ASCII /MD/ ;MRP-DATA-MEMORY.
402 000724 000002 .WORD MD.
403 000726 121 127 .ASCII /QW/ ;QEX-WINDOW-MEMORY.
404 000730 000004 .WORD QW.
405 000732 121 114 .ASCII /QL/ ;QEX-LOCATION-MEMORY
406 000734 000010 .WORD QL.
407 000736 103 123 .ASCII /CS/ ;CP-CONTROL-STORE.
408 000740 000020 .WORD CS.
409 000742 103 104 .ASCII /CD/ ;CP-DATA-MEMORY.
410 000744 000040 .WORD CD.
411 000746 106 120 .ASCII /FP/ ;FAL-POINTER-MEMORY.
412 000750 000100 .WORD FP.
413 000752 106 103 .ASCII /FC/ ;FAL-COUNTER-MEMORY.
414 000754 000200 .WORD FC.
415 000756 121 122 .ASCII /QR/ ;QLB-REFERENCE-PAGE.
416 000760 000400 .WORD QR.
417 000762 121 060 .ASCII /Q0/ ;QLB-PAGE-0
418 000764 001000 .WORD Q0.
419 000766 121 061 .ASCII /Q1/ ;QLB-PAGE-1
420 000770 002000 .WORD Q1.
421 000772 121 062 .ASCII /Q2/ ;QLB-PAGE-2.
422 000774 004000 .WORD Q2.
423 000030 MTBLN: = <.-MTBL/2>
424 ;
425 ;
426 ; REGISTER AND MEMORY TEST REFERENCE TABLES.
427 ;
428 ;
429 ; REGISTER TEST CONTROL ROUTINE ADDRESSES (REFERENCE)
430 ; PLACE-HOLDER VALUES FOR REGISTERS ARE:
431 ;
432 ; 0 MRP-MAR.
433 ; 1 CP-MAR.
434 ;
435 000776 RGRF:
436 000776 011524' 011524' .WORD T1R,T1R.
437 001002 011540' 011540' .WORD T2R,T2R.
438 001006 011556' 011556' .WORD T3R,T3R.
439 001012 011574' 011574' .WORD T4R,T4R.
440 ;

```

```

441
442
443 001016
444 001016 011610* 011610* 011610*
445 001046 011620* 011620* 011620*
446 001076 011634* 011634* 011634*
447 001126 011652* 011652* 011652*
448 001156 011670* 011670* 011670*
449 001206 011720* 011720* 011720*
450 001236 011760* 011760* 011760*
451 001266 011770* 011770* 011770*
452 001316 012020* 012020* 012020*
453 001346 012054* 012054* 012054*
454 001376 012072* 012072* 012072*
455 001426 012216* 012216* 012216*
456 001456 012322* 012322* 012322*
457
458
459
460 001506
461 001506 000377
462 001510 000000
463 001512 007777
464 001514 000000
465 001516 077777
466 001520 076000
467 001522 077777
468 001524 076000
469 001526 001777
470 001530 000000
471 001532 007777
472 001534 000000
473 001536 007777
474 001540 000000
475 001542 007777
476 001544 000000
477 001546 001777
478 001550 000000
479 001552 001777
480 001554 000000
481 001556 001777
482 001560 000000
483 001562 001777
484 001564 000000
485 000030
486
487
488
489 000004
490 000015
491 000002
492 000014
493
494 001566
495 001566
496 001606
497 001606

```

MEMORY TEST CONTROL ROUTINE ADDRESSES (REFERENCE)

MTREF:

```

.WORD T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1
.WORD T2,T2,T2,T2,T2,T2,T2,T2,T2,T2,T2,T2
.WORD T3,T3,T3,T3,T3,T3,T3,T3,T3,T3,T3,T3
.WORD T4,T4,T4,T4,T4,T4,T4,T4,T4,T4,T4,T4
.WORD T5,T5,T5,T5,T5,T5,T5,T5,T5,T5,T5,T5
.WORD T6,T6,T6,T6,T6,T6,T6,T6,T6,T6,T6,T6
.WORD T7,T7,T7,T7,T7,T7,T7,T7,T7,T7,T7,T7
.WORD T8,T8,T8,T8,T8,T8,T8,T8,T8,T8,T8,T8
.WORD T9,T9,T9,T9,T9,T9,T9,T9,T9,T9,T9,T9
.WORD TA,TA,TA,TA,TA,TA,TA,TA,TA,TA,TA,TA
.WORD TB,TB,TB,TB,TB,TB,TB,TB,TB,TB,TB,TB
.WORD TC,TC,TC,TC,TC,TC,TC,TC,TC,TC,TC,TC
.WORD TD,TD,TD,TD,TD,TD,TD,TD,TD,TD,TD,TD

```

LOWER AND UPPER MEMORY LIMITS (REFERENCE)

LIMREF:

```

.WORD 255. ;MRP MICROPGM MEMORY UPPER LIMIT
.WORD 0 ;MRP MICROPGM MEMORY LOWER LIMIT
.WORD 4095. ;MRP DATA MEMORY UPPER LIMIT
.WORD 0 ;MRP DATA MEMORY LOWER LIMIT
.WORD 077777 ;QEX WINDOW MEMORY UPPER LIMIT =X'7FFF
.WORD 076000 ;LOWER LIMIT =X'7C00
.WORD 077777 ;QEX LOCATION MEMORY UPPER LIMIT =X'7FFF
.WORD 076000 ;LOWER LIMIT =X'7C00
.WORD 1023. ;CP CONTROL STORE UPPER LIMIT
.WORD 0 ;CP CONTROL STORE LOWER LIMIT
.WORD 4095. ;CP DATA MEMORY UPPER LIMIT
.WORD 0 ;CP DATA MEMORY LOWER LIMIT
.WORD 4095. ;FAL POINTER MEMORY UPPER LIMIT
.WORD 0 ;FAL POINTER MEMORY LOWER LIMIT
.WORD 4095. ;FAL COUNTER MEMORY UPPER LIMIT
.WORD 0 ;FAL COUNTER MEMORY LOWER LIMIT
.WORD 1023. ;QLB REFERENCE PAGE UPPER LIMIT
.WORD 0 ;QLB REFERENCE PAGE LOWER LIMIT
.WORD 1023. ;QLB PAGE 0 UPPER LIMIT
.WORD 0 ;QLB PAGE 0 LOWER LIMIT
.WORD 1023. ;QLB PAGE 1 UPPER LIMIT
.WORD 0 ;QLB PAGE 1 LOWER LIMIT
.WORD 1023. ;QLB PAGE 2 UPPER LIMIT
.WORD 0 ;QLB PAGE 2 LOWER LIMIT

```

LIMNUM = <.-LIMREF>/2.

REGISTER AND MEMORY TEST ROUTINE ADDRESSES (CURRENT JUMP TABLES)

```

RT = 4 ;NUMBER OF REGISTER TESTS
MT = 13 ;NUMBER OF MEMORY TESTS
NREGS = 2 ;NUMBER OF REGISTERS
NMEMS = 12 ;NUMBER OF MEMORIES

```

RGSUB: .BLKW <RT*NREGS>

MTSUB: .BLKW <MT*NMEMS>

```

498 ;
499 ; LOWER AND UPPER MEMORY LIMITS (CURRENT TEST)
500 ;
501 002276 CURLIM: ;
502 002276 ; .BLKW LIMNUM
503 ;
504 ;
505 ; SUB-MODULE TEST ROUTINE ADDRESSES
506 ;
507 ;
508 ;
509 ; REGISTERS
510 002356 000000G 000000G STRADD: .WORD STMA,STCA
511 002362 000000G 000000G RT4ADD: .WORD IMA,ICA
512 ;
513 ; MEMORIES
514 ;
515 002366 000000G 000000G 000000G STADDR: .WORD STUFMM,STUFMD,STUFQX,STUFQX,STUFCS,STUFCD
516 002402 000000G 000000G 000000G STUFFA,STUFFA,STUFQR,STUFQB,STUFQB,STUFQB
517 002416 000000G 000000G 000000G T1ADDR: .WORD T1M1,T1MD,T1QX,T1QX,T1CS,T1CD
518 002432 000000G 000000G 000000G T1FA,T1FA,T1QR,T1QB,T1QB,T1QB
519 002446 000000G 000000G 000000G T6ADDR: .WORD T6M1,T6MD,T6QX,T6QX,T6CS,T6CD
520 002462 000000G 000000G 000000G T6FA,T6FA,T6QR,T6QB,T6QB,T6QB
521 002476 000000G 000000G 000000G T7ADDR: .WORD T7M1,T7MD,T7QX,T7QX,T7CS,T7CD
522 002512 000000G 000000G 000000G T7FA,T7FA,T7QR,T7QB,T7QB,T7QB
523 002526 000000G 000000G 000000G TCDADD: .WORD TCMMD,TCMDD,TCQXD,TCQXD,TCCSD,TCCDD
524 002542 000000G 000000G 000000G TCFAD,TCFAD,TCQRD,TCQBD,TCQBD,TCQBD
525 002556 000000G 000000G 000000G TCUADD: .WORD TCMMU,TCMDU,TCQXU,TCQXU,TCCSU,TCCDU
526 002572 000000G 000000G 000000G TCFAU,TCFAU,TCQRU,TCQBU,TCQBU,TCQBU
527 002606 000000G 012332' 012332' TDADDR: .WORD TDM1,TDNUL,TDNUL,TDNUL,TDNS,TDNUL
528 002622 012332' 012332' 012332' .WORD TDNUL,TDNUL,TDNUL,TDNUL,TDNUL,TDNUL
529 ;
530 ; ERROR ROUTINE WORK AREAS
531 ;
532 002636 000000 CKDATA: .WORD 0 ; TEST PATTERN
533 002640 000000 CK2: .WORD 0 ; TEST 12 READ TEST PATTERN
534 002642 000000 CK3: .WORD 0 ; TEST 12 WRITE TEST PATTERN
535 002644 000000 PREADD: .WORD 0 ; CURRENT MEMORY ADDRESS
536 002646 000000 ERRADD: .WORD 0 ; ADDRESS AT ERROR
537 002650 000000 ERRCT: .WORD 0 ; NUMBER OF ERROR WORDS TO PRINT
538 002652 000000 ERW1: .WORD 0 ; ERRONEOUS DATA FROM MEMORY -- 1
539 002654 000000 ERW2: .WORD 0 ; - 2
540 002656 000000 ERW3: .WORD 0 ; - 3
541 002660 000000 ERW4: .WORD 0 ; - 4
542 ;
543 ;
544 002662 045655 050500 LMM: .RAD50 /LDMM/
545 002666 045643 073300 LCS: .RAD50 /LDCS/
546 002672 040 040 040 ASWRK: .ASCII / /
547 002677 124 105 123 TMSG: .ASCII /TEST/
548 002704 120 101 123 PMSG: .ASCII /PASS/
549 002711 101 104 104 AMSG: .ASCII /ADDRESS/
550 002722 105 130 120 EMSG: .ASCII /EXPECTED/
551 002734 122 105 103 RMSG: .ASCII /RECEIVED/
552 002746 052 040 052 UNMSG: .ASCII /* */
553 ;
554 ; REGISTER AND MEMORY NAMES

```

```

555
556 002753
557 002753      115      122      120
558 002762      103      120      040
559
560 002771
561 002771      115      122      120
562 003014      115      122      120
563 003037      121      105      130
564 003062      121      105      130
565 003105      103      120      040
566 003130      103      120      040
567 003153      106      101      114
568 003176      106      101      114
569 003221      121      114      102
570 003244      121      114      102
571 003267      121      114      102
572 003312      121      114      102
573
574
575
576
577 003335      015      012
578 003337
579
580
581
582
583
584
585
586
587 003455      000
588 003456      015      012      015
589 003462      124      105      123
590 003500      015      012      015
591 003504      124      105      123
592 003523      015      012
593 003525      105      116      124
594 003571      015      012      015
595 003575      124      105      123
596 003615      015      012      015
597 003623      121      125      105
598 003655      015      012      000
599 003660      015      012
600 003662      124      105      123
601 003752      015      012
602 003754      105      122      122
603 004007      015      012      000
604 004012      015      012
605 004014      105      122      122
606 004032      015      012
607 004034      111      116      126
608 004061      015      012
609 004063      111      116      126
610 004107      015      012
611 004111      111      116      126

```

RFTBL:

ASCII: /MRP MAR/

ASCII: /CP MAR/

MFTBL:

ASCII: /MRP MICROPGM MEMORY/

ASCII: /MRP DATA MEMORY/

ASCII: /QEX WINDOW MEMORY/

ASCII: /QEX LOCATION MEMORY/

ASCII: /CP CONTROL STORE/

ASCII: /CP DATA MEMORY/

ASCII: /FAL POINTER MEMORY/

ASCII: /FAL COUNTER MEMORY/

ASCII: /QLB REFERENCE PAGE/

ASCII: /QLB PAGE 0/

ASCII: /QLB PAGE 1/

ASCII: /QLB PAGE 2/

PRINT-LINE

PRINT::

.BYTE 15,12

.REPT 78

.BYTE 40

.ENDR

TABLE OF MESSAGES

.BYTE 0

.BYTE 15,12,15,12

.ASCII /TEST(S) ENDED/

.BYTE 15,12,15,12

.ASCII /TEST(S) HALTED/

.BYTE 15,12

.ASCII /ENTER ANY CHARACTER TO STOP TEST(S)/

.BYTE 15,12,15,12

.ASCII /TEST(S) STARTED/

.BYTE 15,12,15,12,15,12

.ASCII /QUERY RESOLVER DIAGNOSTICS/

.BYTE 15,12,0

.BYTE 15,12

.ASCII /TEST 6 INCOMPATABLE WITH MEMORY LIMITS, TEST DISCARDED/

.BYTE 15,12

.ASCII /ERROR: NO SELECTIONS, EXIT,/

.BYTE 15,12,0

.BYTE 15,12

.ASCII /ERROR ON READ/

.BYTE 15,12

.ASCII /INVALID ERROR OPTION/

.BYTE 15,12

.ASCII /INVALID LOOP OPTION/

.BYTE 15,12

.ASCII /INVALID TEST PATTERN/

.BYTE 15,12

.ASCII /INVALID TEST PATTERN/

.BYTE 15,12

612	004136	015	012		.BYTE	15,12
613	004140	111	116	126	.ASCIZ	/INVALID TEST NUMBER/
614	004164	015	012		.BYTE	15,12
615	004166	111	114	114	.ASCIZ	/ILLEGAL ODD ADDRESS/
616	004212	015	012		.BYTE	15,12
617	004214	111	116	126	.ASCIZ	/INVALID UPPER MEMORY LIMITS/
618	004250	015	012		.BYTE	15,12
619	004252	111	116	126	.ASCIZ	/INVALID LOWER MEMORY LIMITS/
620	004306	015	012		.BYTE	15,12
621	004310	111	116	126	.ASCIZ	/INVALID MEMORY MNEMONIC/
622	004340	015	012		.BYTE	15,12
623	004342	111	116	126	.ASCIZ	/INVALID REGISTER MNEMONIC/
624	004374	015	012		.BYTE	15,12
625	004376	111	116	126	.ASCIZ	/INVALID RESPONSE/
626	004417	015	012		.BYTE	15,12
627	004421	105	116	124	.ASCIZ	/ENTER ERROR CONTROL/
628	004445	015	012		.BYTE	15,12
629	004447	114	117	117	.ASCIZ	/LOOP ON TEST(S)?/
630	004470	015	012		.BYTE	15,12
631	004472	105	116	124	.ASCIZ	/ENTER PATTERN FOR TEST 10/
632	004524	015	012		.BYTE	15,12
633	004526	105	116	124	.ASCIZ	/ENTER PATTERN FOR TEST 3/
634	004557	015	012		.BYTE	15,12
635	004561	123	105	114	.ASCIZ	/SELECT MEMORY TEST(S)/
636	004607	015	012		.BYTE	15,12
637	004611	123	105	114	.ASCIZ	/SELECT REGISTER TEST(S)/
638	004641	015	012		.BYTE	15,12
639	004643	105	116	124	.ASCIZ	/ENTER LIMITS FOR QLB PAGE 2/
640	004677	015	012		.BYTE	15,12
641	004701	105	116	124	.ASCIZ	/ENTER LIMITS FOR QLB PAGE 1/
642	004735	015	012		.BYTE	15,12
643	004737	105	116	124	.ASCIZ	/ENTER LIMITS FOR QLB PAGE 0/
644	004773	015	012		.BYTE	15,12
645	004775	105	116	124	.ASCIZ	/ENTER LIMITS FOR QLB REFERENCE PAGE/
646	005041	015	012		.BYTE	15,12
647	005043	105	116	124	.ASCIZ	/ENTER LIMITS FOR FAL COUNTER MEMORY/
648	005107	015	012		.BYTE	15,12
649	005111	105	116	124	.ASCIZ	/ENTER LIMITS FOR FAL POINTER MEMORY/
650	005155	015	012		.BYTE	15,12
651	005157	105	116	124	.ASCIZ	/ENTER LIMITS FOR CP DATA MEMORY/
652	005217	015	012		.BYTE	15,12
653	005221	105	116	124	.ASCIZ	/ENTER LIMITS FOR CP CONTROL STORE/
654	005263	015	012		.BYTE	15,12
655	005265	105	116	124	.ASCIZ	/ENTER LIMITS FOR QEX LOCATION MEMORY/
656	005332	015	012		.BYTE	15,12
657	005334	105	116	124	.ASCIZ	/ENTER LIMITS FOR QEX WINDOW MEMORY/
658	005377	015	012		.BYTE	15,12
659	005401	105	116	124	.ASCIZ	/ENTER LIMITS FOR MRP DATA MEMORY/
660	005442	015	012		.BYTE	15,12
661	005444	105	116	124	.ASCIZ	/ENTER LIMITS FOR MRP MICROPGM MEMORY/
662	005511	015	012		.BYTE	15,12
663	005513	123	105	114	.ASCIZ	/SELECT MEMORIES/
664	005533	015	012		.BYTE	15,12
665	005535	123	105	114	.ASCIZ	/SELECT REGISTER(S)/
666	005550	015	012		.BYTE	15,12
667	005562	101	114	114	.ASCIZ	/ALL TESTS, ALL MEMORIES, FULL RANGE?/
668	005630	377			.ASCIZ	.BYTE 377


```

669 ;
670 005631 ; PMSG2:
671 005631 120 101 123 .ASCII /PASS-NUMBER./
672 000014 PM2LN. =. .-PMSG2.
673 ;
674 005645 105 116 104 ENDOF: .ASCII /END OF PASS./
675 000014 ENDLN. =. .-ENDOF.
676 .LIST BEX.
677 .NLIST CND.
678 .EVEN.
679 ;
680 ;
681 ; COMMAND-LINE MACRO.
682 ;
683 ;
684 005662 GCMBLK: GCMLB$ 2,,GCMBUF,CMILUN.
685 ;
686 ; INPUT-FILE-FDB.
687 ;
688 006170 INFDB::
689 006170 FDBDF$
690 006330 FDRCA$ FD,RUM.
691 006330 FDBK$A HRL0,512,,,STAT.
692 006330 FDOF$A INLUN,,INDNB.
693 ;
694 006330 INDNB::
695 006330 NMBLK$ .DAT
696 006366 FSRSZ$ 1

```

```

698 ;
699 ;
700 ; ENTER HERE
701 ;
702 ;
703 ; INITIALIZE HOR
704 ;
705 ; START:
706 006366 CALL OUT1 ; ISSUE INFORMATION MESSAGE
707 006372 016767 0000006 171404 MOV $TKTCB,TSKTCB ; SAVE MY TCB
708 006400 013767 000274 171400 MOV @#274,OLDVEC ; SAVE VECTOR AT 274
709 006406 012737 012344' 000274 MOV #BPTISR,@#274 ; MOVE IN MY INTERRUPT HANDLER ADDR
710 006414 F INIT$
711 ;
712 ; PERFORM MASTER RESET AND NO CLOCKS
713 ;
714 006420 012746 177777 MOV #177777,-(SP) ; CLEAR CSR1
715 006424 012746 000010 MOV #0$RSET,-(SP) ; SET MASTER RESET
716 006430 CALL CSR1 ; DO IT
717 006434 012746 000010 MOV #0$RSET,-(SP) ; CLEAR RESET
718 006440 012746 176000 MOV #<0$NCLK>,-(SP) ; SET NO-CLOCKS
719 006444 CALL CSR1
720 ;
721 ; INITIALIZE PPS
722 ;
723 006450 012746 000053 MOV #0$QLA,-(SP) ; ADDRESS SELECT FOR QLB PAGES
724 006454 CALL PPCR
725 006460 012746 002000 MOV #2000,-(SP) ; SEND ADDRESS X'400' (ILLEGAL)
726 006464 CALL LBPP
727 ;
728 ; RESET MRP AND CP
729 ;
730 006470 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
731 006472 012746 000004 MOV #0$MSET,-(SP) ; SET RESET
732 006476 CALL CSR1
733 006502 012746 000004 MOV #0$MSET,-(SP) ; CLEAR RESET
734 006506 005046 CLR -(SP) ; SET NOTHING
735 006510 CALL CSR1
736 ;
737 006514 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
738 006516 012746 000002 MOV #0$CSET,-(SP) ; SET RESET
739 006522 CALL CSR1
740 006526 012746 000002 MOV #0$CSET,-(SP) ; CLEAR RESET
741 006532 005046 CLR -(SP) ; SET NOTHING
742 006534 CALL CSR1
743 ;
744 006540 012746 000100 MOV #100,-(SP) ; SET 'QLB ERASE'
745 006544 CALL PPCR

```

```

747 ;
748 ;
749 ; PROMPT FOR ALL TESTS, ALL MEMORIES, FULL RANGE.
750 ;
751 ;
752 ALL:
753 CALL ALLSEL ; ISSUE PROMPT.
754 BCC 1$ ; NEED A RESPONSE.
755 CALL ERR2 ; INVALID RESPONSE.
756 BR ALL ; PROMPT AGAIN.
757 ;
758 1$: CALL FIND ; LOCATE RESPONSE IN COMMAND LINE.
759 BCC 2$ ; OK, VALIDATE RESPONSE.
760 CALL ERR2
761 BR ALL
762 ;
763 ; PARSE RESPONSE.
764 ;
765 2$: CMPB #'N',(R1) ; N = NO.
766 BNE 3$ ; TRY YES.
767 JMP RGSEL ; PROMPT FOR REGISTER TESTS.
768 3$: CMPB #'Y',(R1) ; Y = YES.
769 BEQ MOVE ; OK, SET UP FOR ALL.
770 CALL ERR2 ; MUST BE Y OR N.
771 BR ALL ; PROMPT AGAIN.
772 ;
773 ; SET UP MEMORY LIMITS TABLE FOR CURRENT TEST.
774 ; COPY REFERENCE TABLE TO CURRENT TABLE (IE, TEST
775 ; MEMORIES OVER THEIR FULL RANGE).
776 ;
777 MOVE: MOV #LIMREF,R0 ; POINT TO REF TABLE.
778 MOV #CURLIM,R1 ; POINT TO CURRENT TABLE.
779 MOV #LIMNUM,R2 ; NUMBER OF WORDS TO MOVE.
780 1$: MOV (R0)+(R1)+
781 DEC R2
782 BNE 1$
783 ;
784 ; MOVE ALL OF MEMORY ROUTINE REFERENCE TABLE TO CURRENT JUMP
785 ; TABLE.
786 ;
787 MOV #MTRF,R0 ; POINT TO REF TABLE.
788 MOV #MTSUB,R1 ; POINT TO CURRENT TABLE.
789 MOV #<MT*NMEMS>,R2 ; NUMBER OF WORDS.
790 2$: MOV (R0)+(R1)+
791 DEC R2
792 BNE 2$
793 ;
794 ; JUMP TO LOOP SELECTION
795 ;
796 B1$ #<MEMORY+ALLTST>.BASE ; SET FLAG FOR MEMORIES AND ALL TESTS.
797 JMP LPRMPT
    
```

```

799      ;
800      ;
801      ;
802      ;
803      ;
804      ;
805      ;
806 006704      ;
807 006704      MTRTN:
808 006704 005067 171310      RGSEL:
809 006710      CLR      RSEL      ;CLEAR SELECT FLAG
810 006714      CALL     REGSEL   ;PROMPT FOR SELECTION
811 006720 103005      CALL     FIND      ;FIND REG MNEMONIC IN COMMAND LINE
812 006722 052767 000003 171270      BCC     1$        ;NON-BLANK WAS FOUND
813 006730 000167 000314      BIS     #<MA+CA>,RSEL ;SELECT ALL
814      ;
815 006734 122711 000116      1$:      CMPB    #'N,(R1)   ;NO REGISTER TESTS
816 006740 001430      BEQ     MMSEL     ;SKIP TO MEM TESTS
817 006742 052767 000020 171060      BIS     #REGSTR,BASE ;SET FLAG FOR REG TESTS SELECTED
818      ;
819      ;
820      ;
821      ;
822      ;
823      ;
824 006750      ;
825 006750 022700 000002      REGTOP:  CMP     #2,R0      ;CORRECT NUMBER OF CHARS
826 006754 001403      BEQ     2$        ;YES, CONTINUE
827 006756      CALL     ERR20    ;WRITE ERROR MESSAGE
828 006762 000750      BR      RGSEL     ;AND START OVER
829      ;
830      ;
831      ;
832      ;
833      ;
834 006764 012700 000004      2$:      MOV     #RTBL,R0   ;LENGTH OF SCAN TABLE
835 006770 012702 000706      MOV     #RTBL,R2   ;POINT TO SCAN TABLE
836 006774      CALL     SCAN      ;MATCH COMMAND LINE AGAINST TABLE
837 007000 103003      BCC     3$        ;MATCH WAS MADE
838 007002      CALL     ERR20    ;
839 007006 000736      BR      RGSEL     ;START OVER
840      ;
841      ;
842      ;
843      ;
844      ;
845      ;
846 007010 051167 171204      3$:      BIS     (R1),RSEL  ;SET FLAG FOR REGISTER SELECTED
847 007014      CALL     FIND      ;ANYTHING ELSE IN COMMAND LINE
848 007020 103353      BCC     REGTOP    ;YES

```

```

850 ;
851 ;
852 ; PROMPT FOR MEMORY SELECTIONS.
853 ; IF RESPONSE IS <CR>, SELECT ALL MEMORIES.
854 ; IF RESPONSE IS 'N', SKIP MEMORY TESTS.
855 ;
856 ;
857 007022. ; MMSEL:
858 007022. 005067 171170 CLR MSEL ; CLEAR MEMORY SELECT FLAG.
859 007026 CALL MEMSEL ; PROMPT FOR SELECTION.
860 007032. CALL FIND ; FIND MEM MNEMONIC IN COMMAND LINE.
861 007036 103004 BCC 1$ ; NON-BLANK WAS FOUND
862 007040 052767 007777 171150 BIS #<MM+MD+QW+QL+CS+CD+FP+FC+QR+00+Q1+Q2>,MSEL ;
863 007046 000435 BR LIMM1 ; PROMPT FOR LIMITS.
864 ;
865 007050 122711 000116 1$: CMPB #'N,(R1) ; NO MEMORY TESTS.
866 007054 001002. BNE 2$ ; THERE ARE MEMORY TESTS.
867 007056 000167 000166 JMP ENDLIM ; BUILD CURRENT JUMP TABLE.
868 007062. 052767 000040 170740 2$: BIS #MEMORY,BASE ; FLAG MEMORY TESTS SELECTED.
869 ;
870 ;
871 ;
872 ; LOOP TO PROCESS MEMORY MNEMONICS. THERE MAY BE MORE THAN
873 ; ONE IN THE COMMAND LINE. EG:
874 ; >MM QL FC Q2
875 007070 MEMTOP:
876 007070 022700 000002 CMP #2,R0 ; CORRECT NUMBER OF CHARS.
877 007074 001403 BEQ 2$ ; YES, CONTINUE.
878 007076 CALL ERR3 ; WRITE ERROR MESSAGE
879 007102. 000747 BR MMSEL ; AND START OVER.
880 ;
881 ;
882 ; MATCH MEM MNEMONIC FROM THE COMMAND LINE AGAINST
883 ; A TABLE OF VALID MNEMONICS AND THEIR ASSOCIATED
884 ; MEMORY SELECT FLAG SETTINGS.
885 007104 012700 000030 2$: MOV #MTBL,R0 ; LENGTH OF SCAN TABLE.
886 007110 012702. 000716* MOV #MTBL,R2 ; POINT TO SCAN TABLE
887 007114 CALL SCAN ; MATCH COMMAND LINE AGAINST TABLE.
888 007120 103003 BCC 3$ ; MATCH WAS MADE.
889 007122. CALL ERR3
890 007126 000735 BR MMSEL ; START OVER.
891 ;
892 ;
893 ; SUBROUTINE SCAN SETS R1 -> FLAG.
894 ; LOOK FOR THE NEXT MNEMONIC IN THE COMMAND LINE. IF
895 ; THERE IS ONE, PROCESS IT, ELSE FALL THROUGH TO PROMPT.
896 ; FOR MEMORY LIMITS.
897 007130 051167 171062 3$: BIS (R1),MSEL ; SET FLAG FOR MEMORY SELECTED.
898 007134 CALL FIND ; ANYTHING ELSE IN COMMAND LINE
899 007140 103353 BCC MEMTOP ; YES.

```

```

901 ;
902 ;
903 ; PROMPT FOR MEMORY LIMITS.
904 ; READ AND VERIFY THEM.
905 ;
906 ;
907 ;
908 ; SCAN THE MEMORY SELECT FLAG WORD FROM POSITION 0
909 ; TO POSITION 11. FOR EVERY MEMORY WHOSE FLAG IS SET,
910 ; CALL SUBROUTINE 'LIMITS'.
911 ;
912 007142. ; LIMM1:
913 007142. 012700 000001 MOV #MM,R0 ; START TESTING WITH MM.
914 007146. 012701 016544* MOV #PMTM,R1 ; POINT TO FIRST PROMPT MESSAGE
915 007152. 012702 001506* MOV #LIMREF,R2 ; POINT TO REFERENCE LIMITS TABLE
916 007156. 012703 002276* MOV #CURLIM,R3 ; POINT TO CURRENT LIMITS TABLE
917 007162. 012704 000014 MOV #NMEMS,R4 ; LOOP COUNT = NUMBER OF MEMORIES
918 007166. 030067 171024 ;
919 007172. 001414 1$: BIT R0,MSEL ; WAS MEMORY SELECTED
920 007174. 010046 BEQ 2$ ; NO. BUMP POINTERS
921 007176. 010146 MOV R0,-(SP) ; SAVE TEST BIT
922 007200. 011246 MOV R1,-(SP) ; ADDR OF PROMPT MESSAGE
923 007202. 016246 000002 MOV (R2),-(SP) ; MOVE UPPER REF LIMITS
924 007206. CALL LIMITS ; MOVE LOWER REF LIMITS
925 007212. 012663 000002 MOV (SP)+,2(R3) ; MOVE IN CURRENT LOWER LIMITS
926 007216. 012613 MOV (SP)+,(R3) ; MOVE IN CURRENT UPPER LIMITS
927 007220. 012601 MOV (SP)+,R1 ; RESTORE POINTER TO PROMPTS
928 007222. 012600 MOV (SP)+,R0 ; RESTORE TEST BIT
929 ;
930 007224. 005304 2$: DEC R4 ; FINISHED?
931 007226. 001410 BEQ ENDLIM ; YES
932 007230. 006300 ASL R0 ; SHIFT TO TEST NEXT BIT
933 007232. 162701 000004 SUB #4,R1 ; BACK UP PROMPT ADDR POINTER
934 007236. 062702 000004 ADD #4,R2 ; BUMP REF POINTER
935 007242. 062703 000004 ADD #4,R3 ; BUMP CURRENT POINTER
936 007246. 000747 BR 1$ ; TEST NEXT

```

```

938 ;
939 ;
940 ; SELECT REGISTER TESTS
941 ;
942 ;
943 ; PROMPT FOR TEST NUMBERS. IF THE RESPONSE IS <CR>
944 ; (CARRIAGE RETURN ONLY), MOVE A PSEUDO COMMAND LINE
945 ; INTO THE COMMAND LINE BUFFER. THIS PSEUDO LINE
946 ; CONSISTS OF THE TEST NUMBERS FOR ALL TESTS EXCEPT
947 ; TEST 3 (USER PATTERN). PROCEED TO PROCESS THIS
948 ; LINE AS THOUGH IT WAS ENTERED FROM THE TERMINAL.
949 ;
950 ; ENDLIM:
951 007250 BIT #REGSTR,BASE ;ANY REGS SELECTED
952 007256 BNE 10$ ;YES, PROMPT
953 007260 JMP #MEMS ;PROMPT MEMORIES
954 ;
955 007264 10$: CALL SELRT ;PROMPT FOR REGISTER SELECTION
956 007270 CALL FIND ;SCAN COMMAND LINE FOR A TEST NUMBER
957 007274 103021 BCC RTSL ;SOMETHING WAS THERE
958 007276 012700 MOV #ALLREG,R0 ;POINT TO STRING OF ALL REG TESTS
959 007302 012701 MOV #GCMBUF,R1 ;POINT TO COMMAND LINE BUFFER
960 007306 012702 MOV #STRREG,R2 ;LENGTH OF STRING
961 007312 112021 1$: MOVB (R0)+(R1)+ ;MOVE STRING TO COMMAND BUFFER
962 007314 005302 DEC R2
963 007316 001375 BNE 1$
964 007320 012767 000005 170656 MOV #STRREG,GCMLN ;PRETEND LINE HAS BEEN READ IN FROM CONSOLE
965 007326 012767 000062* 170652 MOV #GCMBUF,GCMPNT ;INIT COMMAND LINE POINTER
966 007334 CALL FIND ;LOCATE FIRST TEST NUMBER IN PSEUDO LINE
967 ;
968 ; PROCESS ONE TEST NUMBER AT A TIME. FIRST VALIDATE THE
969 ; NUMBER. TRANSLATE SINGLE DIGIT FROM ASCII DECIMAL INTO
970 ; BINARY. IF A ZERO VALUE IS RETURNED FROM THE TRANSLATION,
971 ; THE ASCII CHARACTER IS INVALID. THE BINARY VALUE
972 ; ALSO CANNOT BE GREATER THAN 4 (HIGHEST TEST NUMBER).
973 ;
974 007340 RTSL:
975 007340 022700 000001 CMP #1,R0 ;TEST NUMBER MUST BE 1 DIGIT
976 007344 001011 BNE RTERR ;NO GOOD
977 007346 111103 MOVB (R1),R3 ;LOAD ASCII TEST NUMBER
978 007350 012704 000232* MOV #TROCT,R4 ;POINT TO TRANSLATE TABLE
979 007354 060304 ADD R3,R4 ;INDEX TO BINARY VALUE
980 007356 111403 MOVB (R4),R3 ;LOAD BINARY VALUE
981 007360 001403 BEQ RTERR ;ZERO VALUE, ERROR
982 007362 122703 000004 CMPB #4,R3 ;TEST NUMBER > 4
983 007366 002012 BGE ;NO, CONTINUE
984 ;
985 ; INVALID TEST NUMBER. PUT OUT A MESSAGE, CLEAR THE
986 ; REGISTER CURRENT JUMP TABLE, GO BACK AND PROMPT
987 ; AGAIN.
988 ;
989 007370 RTERR: CALL ERR6 ;WRITE ERROR MESSAGE
990 007374 012700 001566* MOV #RGSUB,R0 ;POINT TO JUMP TABLE
991 007400 012701 000010 MOV #<RT*HREGS>,R1 ;LOAD NUMBER OF WORDS IN TABLE
992 007404 005020 1$: CLR (R0)+ ;ERASE TABLE
993 007406 005301 DEC R1
994 007410 001375 BNE 1$

```

```

995 007412 000716          BR      ENDLIM          ;TRY AGAIN
996                          ;
997                          ; IF TEST NUMBER = 3, SET A FLAG FOR LATER ACTION.
998                          ;
999 007414          RZREL:
1000 007414 122703 000003    CMPB   #3,R3          ;TEST 3
1001 007420 001003          BNE   1$             ;NO, NO PROMPT LATER
1002 007422 052767 000002 170400 BIS   #TEST3.BASE   ;SET FLAG FOR PROMPT
1003                          ;
1004                          ;
1005                          ; MAKE TEST NUMBER ZERO-RELATIVE, MULTIPLY THE ZERO-
1006                          ; RELATIVE TEST NUMBER BY THE NUMBER OF REGISTERS X 2.
1007                          ; TO GET A BYTE OFFSET INTO THE REG REFERENCE TABLE AND
1008                          ; CURRENT JUMP TABLE. FOR EXAMPLE, IF THE ASCII TEST NUMBER
1009                          ; WAS 2, THE ZERO-RELATIVE NUMBER IS 1, THIS NUMBER IS
1010                          ; MULTIPLIED BY 4 TO GET A BYTE OFFSET = 4.
1011                          ;
1012                          ; ADD THE PRODUCT TO THE START ADDRESS OF THE CURRENT JUMP
1013                          ; TABLE AND PUT THE RESULT IN R0, ADD THE SAME PRODUCT TO
1014                          ; THE START ADDRESS OF THE REF TABLE AND PUT THE RESULT IN
1015                          ; R1. THE RESULTS ARE:
1016                          ;
1017                          ; REG CURRENT JUMP TABLE (ASSUMING TEST 2 SELECTED)
1018                          ; .WORD 0,0
1019                          ; .WORD 0,0
1020                          ;
1021                          ; REGISTER REFERENCE TABLE
1022                          ; .WORD T1R,T1R
1023                          ; .WORD T2R,T2R
1024                          ;
1025                          ; R0 -> FIRST 0 IN THE SECOND LINE UNDER 'REG CURRENT JUMP TABLE'
1026                          ; R1 -> FIRST T2R IN THE REFERENCE TABLE.
1027                          ;
1028                          ;
1029 007430 005303          1$: DEC   R3             ;MAKE TEST NUMBER ZERO REL
1030 007432 010301          MOV   R3,R1          ;SYSTEM EXPECTS MULTIPLICAND IN R1
1031 007434 012700 000004          MOV   #<NREGS*2>,R0 ;AND MULTIPLIER IN R0
1032 007440          CALL  #MUL          ;GET OFFSET INTO TABLE OF WORDS
1033 007444 010103          MOV   R1,R3          ;LOAD PRODUCT INTO R3
1034 007446 012700 001566*          MOV   #RGSUB,R0     ;R0 -> TOP OF REG TABLE
1035 007452 060300          ADD  R3,R0          ;TEST ADDRESSES GO HERE
1036 007454 012701 000776*          MOV   #RGREF,R1     ;R1 -> REG TEST REF TABLE
1037 007460 060301          ADD  R3,R1          ;TEST ADDRESSES COME FROM HERE
1038                          ;
1039                          ;
1040                          ; DEPENDING UPON WHICH REGISTERS HAVE BEEN SELECTED FOR TESTING,
1041                          ; MOVE ADDRESSES OF ROUTINES THAT GOVERN THE TESTS FROM THE
1042                          ; REFERENCE TABLE TO THE REGISTER CURRENT JUMP TABLE (RGSUB).
1043                          ; START TESTING THE REGISTER SELECT FLAG AT POSITION 0 (MRP
1044                          ; MAR).
1045                          ;
1046                          ; PROCEEDING WITH THE ABOVE EXAMPLE ASSUMING IN ADDITION THAT
1047                          ; CP MAR WAS THE REGISTER SELECTED, THIS ROUTINE WOULD FILL
1048                          ; THE REG CURRENT JUMP TABLE IN THE FOLLOWING MANNER:
1049                          ;
1050                          ; .WORD 0,0
1051                          ; .WORD 0,T2R

```



```

1052 007462 012702 000001      MOV  #MA,R2      ;START WITH MRP MAR
1053 007466 012703 000002      MOV  #NREGS,R3   ;LOOP COUNT = NUMBER OF REGISTERS
1054 007472 030267 170522      2$: BIT  R2,RSEL   ;WAS HDR REG SELECTED
1055 007476 001401      BEQ  3$          ;NO
1056 007500 011110      MOV  (R1),(R0)   ;MOVE FROM REF TO JUMP
1057 007502 022120      3$: CMP  (R1+),(R0+) ;INCR POINTERS
1058 007504 006302      ASL  R2          ;SHIFT TO TEST NEXT BIT
1059 007506 005303      DEC  R3          ;FINISHED?
1060 007510 001370      BNE  2$
1061      ;
1062      ; TRANSFERS BETWEEN THE REG REF TABLE ANF CURRENT JUMP
1063      ; TABLE ARE COMPLETE FOR ONE TEST NUMBER; NOW SCAN THE
1064      ; COMMAND LINE FOR THE NEXT TEST NUMBER.
1065      ;
1066 007512      CALL  FIND       ;SCAN COMMAND LINE FOR MORE
1067 007516 103310      BCC  RTSL       ;PROCESS THE VALUE
1068      ;
1069      ; CHECK FOR TEST 3 (USER SUPPLIED PATTERN)
1070      ;
1071 007520 032767 000002 170302      BIT  #TEST3,BASE ;IS PROMPT FLAG ON
1072 007526 001421      BEQ  MEMS       ;NO, SKIP ALL THIS
1073 007530      PMPT3: CALL  PMT3      ;PROMPT
1074 007534      CALL  FIND     ;FIND A NON-BLANK IN COMMAND LINE
1075 007540 103003      BCC  1$        ;OK, CONTINUE
1076 007542      CALL  ERR60
1077 007546 000770      BR   PMPT3
1078 007550      1$: CALL  PACK    ;TRY AGAIN
1079 007554 103003      BCC  2$        ;CONVERT TEST PATTERN TO BINARY
1080 007556      CALL  ERR60   ;CONVERSION OK
1081 007562 000762      BR   PMPT3
1082 007564 016767 170232 170264 2$: MOV  BINWD,RT3  ;PUT PATTERN IN A SAFE PLACE
    
```

```

1084 ;
1085 ;
1086 ;       SELECT MEMORY TESTS.
1087 ;
1088 ;
1089 ;
1090 ;       PROMPT FOR TEST NUMBERS. IF THE RESPONSE IS <CR>
1091 ;       (CARRIAGE RETURN ONLY), MOVE A PSEUDO-COMMAND LINE
1092 ;       INTO THE COMMAND LINE BUFFER. THIS PSEUDO-LINE
1093 ;       CONSISTS OF THE TEST NUMBERS FOR ALL TESTS EXCEPT
1094 ;       TEST 10 (USER PATTERN). PROCEED TO PROCESS THIS
1095 ;       LINE AS THOUGH IT WAS ENTERED FROM THE TERMINAL.
1096 007572. MEMS:
1097 007572. 032767 000040 170230 BIT #MEMORY, BASE ; MEMORIES SELECTED.
1098 007600 001002 BNE 10$ ; YES, PROMPT.
1099 007602 000167 000506 JMP CHECK0 ; CHECK CURRENT JUMP TABLE.
1100 ;
1101 007606 10$: CALL SELMT ; PROMPT FOR MEMORY SELECTION.
1102 007612. CALL FIND ; LOOK FOR TEST NUMBER IN COMMAND LINE.
1103 007616 103024 BCC MTSL ; SOMETHING WAS THERE.
1104 007620 052767 000100 170202. BIS #ALLTST, BASE ; SET FLAG FOR ALL TESTS (REPORT PASSES)
1105 007626 012700 000657. MOV #ALLMEM, R0 ; POINT TO STRING OF ALL MEM TESTS.
1106 007632. 012701 000062. MOV #GCMBUF, R1 ; POINT TO COMMAND LINE BUFFER.
1107 007636 012702. 000027 MOV #STRMEM, R2 ; LENGTH OF STRING.
1108 007642. 112021 1$: MOVB (R0)+, (R1)+ ; MOVE STRING TO COMMAND BUFFER.
1109 007644 005302. DEC R2.
1110 007646 001375 BNE 1$
1111 007650 012767 000027 170326 MOV #STRMEM, GCMLN ; PRETEND LINE HAS BEEN READ IN FROM CONSOLE.
1112 007656 012767 000062. 170322. MOV #GCMBUF, GCMPT ; INIT COMMAND LINE POINTER.
1113 007664 CALL FIND ; LOCATE FIRST TEST NUMBER IN PSEUDO-LINE
1114 ;
1115 ;       PROCESS ONE TEST NUMBER AT A TIME. FIRST VALIDATE THE
1116 ;       NUMBER.
1117 ;
1118 007670 MTSL:
1119 007670 022700 000002 CMP #2, R0 ; TEST NUMBERS ARE 1 OR 2 BYTES
1120 007674 002416 BLT MTERR ; NO GOOD (TOO MANY)
1121 007676 003007 BGT 1$ ; SINGLE DIGIT, PROCESS IT.
1122 007700 122721 000061 CMPEB #1, (R1)+ ; TENS DIGIT?
1123 007704 001012. BNE MTERR ; NO, ERROR
1124 007706 111103 MOVSB (R1), R3 ; LOAD ASCII TEST NUMBER
1125 007710 062703 000012 ADD #10, R3 ; ADD VALUE OF TENS DIGIT.
1126 007714 000401 BR MTRT ; AND CONTINUE.
1127 ;
1128 ;       TRANSLATE SINGLE DIGIT FROM ASCII DECIMAL INTO BINARY.
1129 ;       IF A ZERO VALUE IS RETURNED FROM THE TRANSLATION,
1130 ;       THE ASCII CHARACTER IS INVALID.
1131 ;
1132 007716 111103 1$: MOVSB (R1), R3 ; LOAD ASCII TEST NUMBER.
1133 007720 012704 000232. MTRT: MOV #TROCT, R4 ; POINT TO TRANSLATE TABLE.
1134 007724 060304 ADD R3, R4 ; INDEX TO BINARY VALUE.
1135 007726 111403 MOVSB (R4), R3 ; LOAD BINARY VALUE.
1136 007730 001012. BNE MZREL ; NON-ZERO VALUE, CONTINUE.
1137 ;
1138 ;       INVALID TEST NUMBER. PUT OUT MESSAGE, CLEAR THE MEMORY,
1139 ;       CURRENT JUMP TABLE AND GO BACK TO PROMPT.
1140 ;

```

```

1141 007732.          MTERR: CALL  ERR6          ;WRITE ERROR MESSAGE
1142 007736 012700 001606' MOV  #MTSUB,R0      ;POINT TO JUMP TABLE
1143 007742 012701 000234 MOV  #<MT*NMEMS>,R1 ;LOAD NUMBER OF WORDS IN TABLE
1144 007746 005020 1$: CLR  (R0)+      ;RESET TABLE
1145 007750 005301 DEC  R1             ;
1146 007752 001375 BNE  1$            ;
1147 007754 000706 BR   MEMS          ;TRY AGAIN
1148 ;
1149 ; IF TEST NUMBER = 6 OR 10, SET A FLAG FOR
1150 ; LATER ACTION.
1151 ;
1152 007756          MZREL:
1153 007756 122703 000006 CMPB #6,R3          ;TEST 6
1154 007762 001003 BNE  10$          ;NO, DO NOT SET FLAG
1155 007764 052767 000004 170036 BIS  #TEST6,BASE  ;SET FLAG FOR TEST 6 SELECTED
1156 007772 122703 000012 10$: CMPB #10,R3       ;TEST 10
1157 007776 001003 BNE  1$            ;NO, NO PROMPT LATER
1158 010000 052767 000010 170022 BIS  #TEST10,BASE ;SET FLAG FOR PROMPT
1159 ;
1160 ;
1161 ; MAKE TEST NUMBER ZERO-RELATIVE. MULTIPLY THE ZERO-
1162 ; RELATIVE TEST NUMBER BY THE NUMBER OF MEMORIES * 2.
1163 ; TO GET A BYTE OFFSET INTO THE REFERENCE TABLE AND
1164 ; CURRENT JUMP TABLE. FOR EXAMPLE, IF THE ASCII TEST NUMBER
1165 ; WAS 2, THE ZERO-RELATIVE NUMBER IS 1. THIS NUMBER IS
1166 ; MULTIPLIED BY 24 TO GET A BYTE OFFSET = 24.
1167 ;
1168 ; ADD THE PRODUCT TO THE START ADDRESS OF THE CURRENT JUMP
1169 ; TABLE AND PUT THE RESULT IN R0. ADD THE SAME PRODUCT TO
1170 ; THE START ADDRESS OF THE REF TABLE AND PUT THE RESULT IN
1171 ; R1. THE RESULTS ARE:
1172 ;
1173 ; CURRENT JUMP TABLE (ASSUMING TEST 2 SELECTED)
1174 ; .WORD 0,0,0,0,0,0,0,0,0,0,0,0
1175 ; .WORD 0,0,0,0,0,0,0,0,0,0,0,0
1176 ;
1177 ; REFERENCE TABLE
1178 ; .WORD T1,T1,T1,T1,T1,T1,T1,T1,T1,T1
1179 ; .WORD T2,T2,T2,T2,T2,T2,T2,T2,T2,T2
1180 ;
1181 ; R0 -> FIRST 0 IN THE SECOND LINE FOLLOWING *CURRENT JUM TABLE*
1182 ; R1 -> FIRST T2 IN THE REFERENCE TABLE
1183 ;
1184 ;
1185 010006 005303 1$: DEC  R3          ;MAKE TEST NUMBER ZERO REL
1186 010010 010301 MOV  R3,R1        ;SYSTEM EXPECTS MULTIPLICAND IN R1
1187 010012 012700 000030 MOV  #<NMEMS*2>,R0 ;AND MULTIPLIER IN R0
1188 010016 CALL #MUL         ;GET OFFSET INTO TABLE OF WORDS
1189 010022 010103 MOV  R1,R3        ;LOAD PRODUCT INTO R3
1190 010024 012700 001606' MOV  #MTSUB,R0    ;R0 -> TOP OF MEM TABLE
1191 010030 060300 ADD  R3,R0        ;TEST ADDRESSES GO HERE
1192 010032 012701 001016' MOV  #MTRF,R1     ;R1 -> MEM TEST REF TABLE
1193 010036 060301 ADD  R3,R1        ;TEST ADDRESSES COME FROM HERE
1194 ;
1195 ; DEPENDING UPON WHICH MEMORIES HAVE BEEN SELECTED FOR TESTING
1196 ; MOVE ADDRESSES OF ROUTINES THAT GOVERN THE TESTS FROM THE
1197 ; REFERENCE TABLE TO THE CURRENT JUMP TABLE (MTSUB). START TESTING

```

```

1198 ; THE MEMORY FLAG AT POSITION 0 (MRP MICROPGM MEMORY).
1199 ;
1200 ; PROCEEDING WITH THE ABOVE EXAMPLE ASSUMING IN ADDITION THAT
1201 ; QEX WINDOW WAS THE MEMORY SELECTED, THIS ROUTINE WOULD FILL
1202 ; THE MEMORY CURRENT JUMP TABLE IN THE FOLLOWING MANNER:
1203 ;
1204 ; .WORD 0.0.0.0.0.0.0.0.0.0
1205 ; .WORD 0.0.T2.0.0.0.0.0.0.0
1206 ;
1207 010040 012702 000001 MOV #M1,R2 ;START WITH MRP MICROPGM MEMORY
1208 010044 012703 000014 MOV #NMEMS,R3 ;LOOP COUNT = NUMBER OF MEMORIES
1209 010050 030267 170142 2$: BIT R2,MSEL ;WAS MEMORY SELECTED
1210 010054 001401 BEQ 3$ ;NO, BUMP TO NEXT
1211 010056 011110 MOV (R1),(R0) ;MOVE FROM REF TO JUMP
1212 010060 022120 3$: CMP (R1)+,(R0)+ ;INCR POINTERS
1213 010062 006302 ASL R2 ;SHIFT TO TEST NEXT BIT
1214 010064 005303 DEC R3 ;FINISHED?
1215 010066 001370 BNE 2$ ;NO, CONTINUE
1216 ;
1217 ; TRANSFERS BETWEEN REF TABLE AND CURRENT JUMP TABLE ARE
1218 ; COMPLETE FOR ONE TEST NUMBER, NOW SCAN THE COMMAND LINE
1219 ; FOR THE NEXT TEST NUMBER.
1220 ;
1221 010070 CALL FIND ;ANYTHING ELSE IN COMMAND LINE?
1222 010074 103402 BCS MTPMT ;NO
1223 010076 000167 177566 JMP MTSL ;YES, PROCESS IT

```



```

1282.      ;      FFFF...
1283.      ;
1284.      ;      BUMP LOWER LIMIT TO 1
1285.      ;      CLEAR LOCATIONS 1 AND 2
1286.      ;      WRITE 1'S INTO LOCATION 1
1287.      ;      READ ZEROS FROM LOCATION 2
1288.      ;      FFFF
1289.      ;      FFFF
1290.      ;      000000
1291.      ;
1292.      ; *****
1293.      ;
1294.      ;
1295.      ;      CHECK EACH NON-ZERO TEST 6 ENTRY IN THE CURRENT JUMP TABLE.
1296.      ;
1297.      ;      CHECK:  TST      (R0)          ;TEST 6 SELECTED FOR THIS MEMORY
1298.      ;      BEQ      2$             ;NO, SKIP CHECK
1299.      ;      MOV      R1,R3          ;SHIFT IN ANOTHER REG
1300.      ;      ASL      R3             ;
1301.      ;      ASL      R3             ;SHIFT FOR DOUBLE WORD OFFSET
1302.      ;      MOV      CURLIM+2(R3),R4 ;GET LOWER LIMIT
1303.      ;      ADD      *2,R4          ;UPPER LIMIT MUST BE AT LEAST 2 GT LOWER
1304.      ;      CMP      CURLIM(R3),R4 ;IS UPPER LIMIT OK FOR TEST 6
1305.      ;      BHS      2$             ;YES, CONTINUE
1306.      ;
1307.      ;
1308.      ;      FAILED CHECK. CLEAR THE ENTRY IN THE CURRENT JUMP TABLE
1309.      ;      AND REPORT TO THE CONSOLE. R1 = PLACEHOLDER VALUE. THIS
1310.      ;      VALUE IS USED AS AN INDEX INTO A TABLE OF MEMORY NAMES
1311.      ;      (AFTER MULTIPLYING THE VALUE BY 19, THE LENGTH OF EACH
1312.      ;      NAME).
1313.      ;
1314.      ;      CLR      (R0)          ;CLEAR TEST 6 ADDRESS FROM CURRENT TABLE
1315.      ;      MOV      R0,-(SP)       ;SAVE POINTER
1316.      ;      MOV      R1,-(SP)       ;SAVE OFFSET
1317.      ;      CALL   ERR10          ;PRINT GENERAL ERROR MESSAGE
1318.      ;      MOV      #19,R0        ;LENGTH OF MEMORY NAMES
1319.      ;      CALL   #MUL           ;GET OFFSET INTO MEMORY NAME TABLE (R0XR1)
1320.      ;      MOV      #MFTBL,R0     ;POINT TO MEMORY NAME TABLE
1321.      ;      MOV      #19,R1        ;NUMBER OF CHARS IN NAME
1322.      ;      MOV      #PRINT,R5    ;POINT TO PRINT LINE
1323.      ;      MOVVB  (R0)+,(R5)+    ;MOVE NAME TO PRINT LINE
1324.      ;      DEC     R1             ;
1325.      ;      BNE     1$             ;
1326.      ;      CALL   CONSOLE        ;WRITE MEMORY IN ERROR
1327.      ;      MOV      (SP)+,R1     ;
1328.      ;      MOV      (SP)+,R0     ;
1329.      ;
1330.      ;
1331.      ;      PREPARE TO CHECK NEXT TEST 6 ENTRY.
1332.      ;
1333.      ;      2$:  INC      R1             ;BUMP MEMORY OFFSET
1334.      ;      ADD      *2,R0          ;POINT TO NEXT TEST 6 ADDRESS
1335.      ;      DEC     R2             ;SUB FROM LOOP COUNT
1336.      ;      BNE     CHECK        ;
1337.      ;
1338.      ;      MAKE SURE THAT THERE IS AT LEAST ONE NON-ZERO ENTRY
1339.      ;      IN EITHER THE REGISTER OR MEMORY CURRENT JUMP TABLE.

```

```
1339 ; SINCE THE TWO TABLES ARE CONTIGUOUS, THEY CAN BE
1340 ; SCANNED IN ONE OPERATION.
1341 ;
1342 ;
1343 010314 CHECK0: MOV #RGSUB,R0 ;POINT TO TOP OF BOTH TABLES
1344 010314 012700 001566* MOV #(<<RT*NREGS>>+<MT*NMEMS>),R1 ;NUMBER OF TABLE ENTRIES
1345 010320 012701 000244 1$: TST (R0)+ ;IS A TABLE ENTRY PRESENT
1346 010324 005720 BNE LPRMPT ;YES, EXIT THIS ROUTINE
1347 010326 001007 DEC R1 ;SUB FROM ROUTINE COUNT
1348 010330 005301 BNE 1$ ;TRY NEXT POSITION
1349 010332 001374 CALL ERR9 ;EXECUTION IMPOSSIBLE
1350 010334 ;
1351 ;
1352 010340 EXIT$S
```

```

1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365 010346
1366 010346
1367 010352
1368 010356 103004
1369 010360 052767 000001 167442
1370 010366 000442
1371
1372 010370 122711 000131
1373 010374 001004
1374 010376 052767 000001 167424
1375 010404 000433
1376 010406 122711 000116
1377 010412 001004
1378 010414 042767 000001 167406
1379 010422 000424
1380
1381
1382
1383
1384 010424 060100
1385 010426 005200
1386 010430 010046
1387 010432 010100
1388 010434
1389 010440 020026
1390 010442 001403
1391 010444
1392 010450 000736
1393 010452 010167 167536
1394 010456 001003
1395 010460
1396 010464 000730
1397 010466 052767 000001 167334

```

```

;
;
; PROMPT FOR LOOP ON TEST
;
;
; RESPONSES:
; <CR> - CARRIAGE RETURN. LOOP ON TESTS
; Y - YES. LOOP ON TESTS
; N - NO. ONE MEMORY TEST CYCLE ONLY
; NUMERIC VALUE - NUMBER OF CYCLES TO EXECUTE
;
LPRMPT:
CALL LPTST ; PROMPT
CALL FIND ; FIND RESPONSE IN COMMAND LINE
BCC 1$ ; OK. RESPONSE FOUND
BIS #LOOP,BASE ; CR RESPONSE MEANS LOOP
BR ERPRMT ; AND CONTINUE
;
1$: CMPB #'Y,(R1) ; YES - LOOP ON TESTS
BNE 2$ ; TRY 'N'
BIS #LOOP,BASE ; SET FLAG FOR LOOP
BR ERPRMT ; PROMPT FOR ERROR OPTIONS
;
2$: CMPB #'N,(R1) ; NO - DO NOT LOOP ON TESTS
BNE 3$ ; NO. TEST FOR LOOP COUNT
BIC #LOOP,BASE ; CLEAR LOOP FLAG
BR ERPRMT
;
; ASSUME THAT THERE IS AN ASCII DECIMAL VALUE IN THE
; COMMAND LINE. CONVERT IT TO BINARY AND STORE.
;
3$: ADD R1,R0 ; POINT 1 PAST STRING
INC R0 ; BUMP FOR STUPID SYSTEM SUBRTN
MOV R0,(SP) ; SAVE FOR LATER COMPARISON
MOV R1,R0 ; MOVE POINTER TO R0 FOR SYSTEM SUBRTN
CALL #CDB ; CONVERT DECIMAL TO BINARY
CMP R0,(SP)+ ; WHOLE STRING CONVERTED
BEQ 4$ ; YES, CONTINUE
CALL ERR7
BR LPRMPT ; PROMPT AGAIN
MOV R1,LOOPCT ; SAVE LOOP COUNT
BNE 5$
CALL ERR7
BR LPRMPT
BIS #LOOP,BASE ; SET LOOP FLAG

```



```

1399 ;
1400 ;
1401 ;
1402 ; PROMPT FOR ERROR OPTIONS.
1403 ;
1404 ;
1405 ; RESPONSES:
1406 <CR> - CARRIAGE RETURN. DISPLAY ALL ERRORS.
1407 H. - HALT. STOP TESTS AFTER 1ST ERROR.
1408 NUMERIC VALUE. - PRINT ONLY THIS NUMBER OF ERROR MESSAGES.
1409 ; BUT CONTINUE TESTS.
1410 N. VALUE + H - PRINT THIS NUMBER OF ERROR MESSAGES AND HALT.
1411 ;
1412 ; ERPRMT:
1413 CALL EROPT ;PROMPT FOR OPTIONS.
1414 CALL FIND ;FIND RESPONSE.
1415 BCS MTSET ;DEFAULT TO "DISPLAY"
1416 ;
1417 CMPB #'H,(R1) ;HALT AFTER 1 ERROR.
1418 BNE 1$ ;NO.
1419 BIS #HALT,BASE ;SET FLAG FOR HALT.
1420 MOV #2,ERLIM ;PRINT ONLY 1 ERROR MESSAGE.
1421 BR MTSET.
1422 ;
1423 ; ASSUME THAT THERE IS AN ASCII DECIMAL VALUE IN THE
1424 ; COMMAND LINE. CONVERT IT TO BINARY AND STORE.
1425 1$: ADD R1,R0 ;POINT PAST STRING IN COMMAND LINE.
1426 INC R0 ;BUMP FOR STUPID SYSTEM SUBRTN
1427 MOV R0,-(SP) ;SAVE FOR LATER COMPARISON.
1428 MOV R1,R0 ;PREPARE TO CONVERT.
1429 CALL %CDB.
1430 CMP R0,(SP)+ ;ENTIRE STRING CONVERTED.
1431 BEQ 2$ ;YES.
1432 CALL ERR8
1433 BR ERPRMT. ;TRY AGAIN
1434 ;
1435 2$: MOV R1,ERLIM ;NUMBER OF ERROR MSGS TO PRINT
1436 INC ERLIM ;ADJUST FOR PRE-DECREMENT.
1437 CALL FIND ;SCAN COMMAND LINE.
1438 BCS MTSET ;NOTHING ELSE THERE.
1439 CMPB #'H,(R1) ;HALT AFTER MESSAGE COUNT EXHAUSTED.
1440 BEQ 4$ ;YES.
1441 CALL ERR8 ;BAD OPTION.
1442 BR ERPRMT. ;TRY AGAIN
1443 ;
1444 4$: BIS #HALT,BASE ;SET HALT FLAG.

```

```

1446 ;
1447 ;
1448 ;
1449 ;
1450 ;
1451 ;
1452 ;
1453 ;
1454 ;
1455 ;
1456 ;
1457 ;
1458 ;
1459 ;
1460 010620 032767 000020 167202 MTSET: BIT #REGSTR,BASE ;REGISTERS SELECTED FOR TEST
1461 010626 001007 BNE 1$ ;YES, INIT MAIN LOOP POINTER/COUNTER
1462 010630 012767 001606* 167364 MOV #MTSUB,MPNT ;POINT AT MEMORY TEST RTNS
1463 010636 012767 000234 167360 MOV #<MT*NMEMS>,MTCNT ;NUMBER OF MEM RTNS
1464 010644 000417 BR 2$ ;CLOSE UP TABLE
1465 ;
1466 ;
1467 ;
1468 ;
1469 ;
1470 010646 012767 001566* 167346 1$: MOV #RGSUB,MPNT ;START POINTER AT TOP OF REG TBL
1471 010654 012767 000010 167342 MOV #<RT*NREGS>,MTCNT ;START COUNT WITH # REG TESTS
1472 ;
1473 ;
1474 ;
1475 ;
1476 ;
1477 ;
1478 010662 032767 000040 167140 BIT #MEMORY,BASE ;MEMORY TESTS SELECTED
1479 010670 001405 BEQ 2$ ;NO, REGS ONLY
1480 010672 062767 000234 167324 ADD #<MT*NMEMS>,MTCNT ;ADD # MEMORY TESTS
1481 010700 012700 002276* MOV #MTSUB+<2*<MT*NMEMS>>,R0 ;POINT PAST MEM TABLE
1482 ;
1483 ;
1484 ;
1485 ;
1486 ;
1487 010704 005267 167314 2$: INC MTCNT ;ADJUST COUNT FOR PRE-DECREMENT
1488 010710 162767 000002 167304 SUB #2,MPNT ;ADJUST POINTER FOR PRE-INCREMENT
1489 010716 016767 167302 167304 MOV MTCNT,NXTCNT ;SAVE FOR FUTURE PASSES
1490 010724 016767 167272 167274 MOV MTPNT,NXTPNT ;SAME
1491 ;
1492 ;
1493 ;
1494 ;
1495 ;
1496 ;
1497 010732: CALL BEGTT ;PUT OUT START TEST MESSAGE
1498 010736 CALL STOP ;GIVE DIRECTION FOR STOPPING TEST
1499 010742: QID$S #IO,ATA,#LUN,TT,*,*,*,*,<#AST>

```

```

1501 ;
1502 ;
1503 ; MAIN LOOP
1504 ;
1505 ;
1506 ;
1507 ; CHECK FOR THE HALT OPTION. IF THE OPTION IS
1508 ; ON, CHECK TO SEE WHETHER ANY ERRORS HAVE OCCURRED.
1509 ; IF THEY HAVE (AND THE HALT COUNT IS EXHAUSTED),
1510 ; TERMINATE THE TESTS.
1511 ;
1512 011004 MTMAIN: BIT #HALT,BASE ;HALT AFTER ERROR
1513 011012 001413 BEQ 1$ ;NO
1514 011014 032767 000400 167006 BIT #ERROR,BASE ;HAS AN ERROR OCCURRED
1515 011022 001407 BEQ 1$ ;NO
1516 011024 026727 166770 000001 CMP ERLIM,#1 ;IS REMAINING PRINTOUT COUNT 1 OR LESS
1517 011032 003003 BGT 1$ ;NO, CONTINUE
1518 011034 CALL HLTTST ;PUT OUT HALT MESSAGE
1519 011040 000464 BR 8$ ;AND EXIT
1520 ;
1521 ;
1522 ; FIND A NON-ZERO ENTRY IN THE 'CURRENT JUMP TABLE'.
1523 ; (A NON-ZERO ENTRY IS THE ADDRESS OF A TEST CONTROL
1524 ; ROUTINE). IF NO NON-ZERO ENTRIES ARE FOUND BEFORE
1525 ; THE END OF THE TABLE IS REACHED, THEN ONE MEMORY TEST
1526 ; CYCLE OR 'PASS' IS COMPLETE.
1527 011042 016701 167154 1$: MOV MTPNT,R1 ;POINT TO JUMP TABLE
1528 011046 005367 167152 2$: DEC MTCNT ;FIRST SUB FROM # RTNS LEFT
1529 011052 001406 BEQ 3$ ;ALL DONE, TEST LOOP FLAG
1530 011054 005721 TST (R1)+ ;ADVANCE POINTER
1531 011056 005711 TST (R1) ;IS THERE AN ADDRESS IN THE TABLE
1532 011060 001772 BEQ 2$ ;NO, BUMP TO NEXT
1533 011062 010167 167134 MOV R1,MTPNT ;SAVE JUMP TABLE POINTER
1534 011066 000457 BR JMPNT ;AND JUMP TO ROUTINE
1535 ;
1536 ;
1537 ;
1538 ; PASS FINISHED
1539 ;
1540 ;
1541 ; IF THE LOOP FLAG IS NOT ON, EXIT QMT.
1542 ; IF THE LOOP FLAG IS ON AND THE LOOP COUNT IS
1543 ; EXHAUSTED, EXIT QMT.
1544 011070 032767 000001 166732 3$: BIT #LOOP,BASE ;IS LOOP FLAG ON
1545 011076 001443 BEQ 7$ ;NO, GET OUT
1546 011100 005767 167110 TST LOOPCT ;IS LOOP COUNT BEING USED
1547 011104 001403 BEQ 4$ ;NO, JUST KEEP LOOPING
1548 011106 005367 167102 DEC LOOPCT ;SUB FROM LOOP COUNT
1549 011112 001435 BEQ 7$ ;FINISHED
1550 ;
1551 ;
1552 ; IF ALL TESTS, PRINT 'END OF PASS NNNN'.
1553 ; REINITIALIZE FOR NEXT PASS.
1554 011114 032767 000100 166706 4$: BIT #ALLTST,BASE ;ALL TESTS
1555 011122 001415 BEQ 6$ ;NO
1556 011124 012702 005645 MOV #ENDOF,R2 ;POINT TO MESSAGE
1557 011130 012703 000014 MOV #ENDLN,R3 ;LENGTH OF MESSAGE

```

```

1558 011134 012705 003337'
1559 011140 112225
1560 011142 005303
1561 011144 001375
1562 011146
1563 011152
1564
1565 011156 062767 000001 167026 6$: MOV #PRINT,R5 ;POINT TO PRINT LINE
1566 011164 005567 167020 6$: MOV# (R2)+,(R5)+ ;MOVE MESSAGE TO PRINT LINE
1567 011170 016767 167032 167024 DEC R3
1568 011176 016767 167026 167020 BNE 5$
1569 011204 000677 CALL PASSC ;ADD # PASSES TO PRINT LINE
1570 CALL CONSOL ;PRINT MESSAGE
1571
1572
1573 011206 7$: ADD #1,PASS ;COUNT NUMBER OF PASSES
1574 011212 8$: ADC PASSH ;CARRY TO HIGH WORD
1575 011212 016737 166570 000274 MOV NXTPNT,MTPT ;RESET POINTER
1576 011220 MOV NXTCNT,MTCNT ;RESET COUNT
1577 BR MTMAIN ;AND ENTER LOOP
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591 011226 JMPMT:
1592 011226 022767 001606' 166766 CMP #MTSUB,MTPT ;DOES TABLE POINTER -> MEMORY SECTION
1593 011234 003422 BLE MEMOFF ;YES
1594 011236 012700 000010 MOV *(<RT*NR>REGS),R0 ;NUMBER OF TABLE ENTRIES
1595 011242 016701 166756 MOV MTCNT,R1 ;LOAD CURRENT COUNT
1596
1597
1598
1599
1600
1601 011246 032767 000040 166554 IF MEMORIES ARE BEING TESTED ON THIS RUN, BACK OUT
1602 011254 001402 THE NUMBER OF MEMORY TEST CURRENT JUMP TABLE ENTRIES
1603 011256 162701 000234 FROM THE COUNT.
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
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
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
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
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
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
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

```

1615 ; 0 MRP-MAR
1616 ; 1 CP-MAR
1617 ;
1618 ;
1619 ; PUT THE REMAINDER (SHIFTED FOR WORD OFFSET) INTO
1620 ; R0 FOR USE BY THE REGISTER TEST CONTROL ROUTINE.
1621 ;
1622 011262 160100 200$: SUB R1,R0 ;GET CURRENT POSITION
1623 011264 012701 000002 MOV #NREGS,R1 ;NUMBER OF REGSITERS
1624 011270 CALL $DIV
1625 011274 006301 ASL R1 ;SHIFT FOR WORD OFFSET
1626 011276 010100 MOV R1,R0 ;TEST CONTROL ROUTINES EXPECT R0
1627 011300 000505 BR LASTJ ;JUMP TO CONTROL ROUTINE
1628 ;
1629 ;
1630 ; IF THE CURRENT JUMP TABLE POINTER -> MEMORY CURRENT
1631 ; JUMP TABLE SET UP TO FIND OUT THE MEMORY PLACE-HOLDER
1632 ; VALUE.
1633 ;
1634 ; LOAD R0 = NUMBER OF MEMORY TESTS. LOAD R1 = CURRENT
1635 ; JUMP TABLE COUNT (IE. THE NUMBER OF CURRENT JUMP TABLE
1636 ; ENTRIES, INCLUDING THE CURRENT NON-ZERO ENRY, REMAINING
1637 ; TO BE PROCESSED BY MTMAIN).
1638 011302 MEMOFF:
1639 011302 012700 000234 MOV #<MT*NMEMS>,R0 ;TOTAL NUMBER OF MEMORY ROUTINES
1640 011306 016701 166712 MOV MTCNT,R1 ;LOAD CURRENT COUNT
1641 ;
1642 ;
1643 ; IF REGISTERS ARE BEING TESTED ON THIS RUN, BACK OUT
1644 ; THE NUMBER OF REGISTER TEST CURRENT JUMP TABLE ENTRIES
1645 ; FROM THE COUNT.
1646 011312 032767 000020 166510 BIT #REGSTR,BASE ;REGISTER TESTS SELECTED
1647 011320 001402 BEQ 100$ ;NO
1648 011322 162701 000010 SUB #<RT*NREGS>,R1 ;SUBTRACT OUT NUMBER OF REG TESTS
1649 ;
1650 ;
1651 ; SUBTRACT THE CURRENT COUNT FROM THE TOTAL NUMBER OF
1652 ; ENTRIES IN THE CURRENT JUMP TABLE (COMBINED OR MEM
1653 ; ONLY) TO GET POSITION RELATIVE TO THE TOP OF THE
1654 ; MEMORY TABLE OF THE CURRENT NON-ZERO ENTRY.
1655 ;
1656 ; DIVIDE BY THE NUMBER OF MEMORIES. THE QUOTIENT
1657 ; WILL BE THE MEMORY TEST NUMBER (IN ZERO-RELATIVE
1658 ; FORM). THE REMAINDER WILL BE THE PLACE-HOLDER VALUE
1659 ; FOR THE MEMORY TO BE TESTED. THESE VALUES ARE:
1660 ;
1661 ; 0 MRP MICROPGM MEMORY
1662 ; 1 MRP DATA MEMORY
1663 ; 2 QEX WINDOW MEMORY
1664 ; 3 QEX LOCATION MEMORY
1665 ; 4 CP CONTROL STORE
1666 ; 5 CP DATA MEMORY
1667 ; 6 FAL POINTER MEMORY
1668 ; 7 FAL COUNTER MEMORY
1669 ; 8 QLB REFERENCE PAGE
1670 ; 9 QLB PAGE 0
1671 ; 10 QLB PAGE 1
1671 ; 11 QLB PAGE 2

```

```

1672. ;
1673 011326 160100 ; 100$: SUB. R1,R0 ;GET CURRENT POSITION.
1674 011330 012701 000014 MOV. #NMEMS,R1 ;NUMBER OF MEMORIES.
1675 011334 CALL. $DIV ;
1676 ;
1677 ;
1678 ; CHECK FOR TEST ON QEX OR FAL MEMORIES MEMORIES.
1679 ; IF SO, PLACE MEMORY SELECT VALUE IN PROPER MODULE.
1680 ; CHECK FOR TEST ON QLB PAGES 0,1,2.
1681 ; IF SO, PLACE PAGE NUMBER IN QLB MODULE.
1682 011340 022701 000002 ;
1683 011344 001004 ;
1684 011346 012767 000042 166474 CMP. #2,R1 ;QEX WINDOW.
1685 011354 000450 BNE. 1$ ;
1686 ; MOV. #Q$QW,QXCODE. ;MOVE WINDOW MEMORY CODE TO QEX MODULE.
1687 011356 022701 000003 1$: ; PUT MEMORY LIMITS ON STACK.
1688 011362 001004 BNE. 2$ ;QEX LOCATION.
1689 011364 012767 000043 166456 MOV. #Q$QL,QXCODE. ;
1690 011372 000441 BR. LSTACK. ;MOVE LOCATION MEMORY CODE TO QEX MODULE
1691 ;
1692 011374 022701 000006 2$: CMP. #6,R1 ;FAL POINTER MEMORY.
1693 011400 001004 BNE. 3$ ;
1694 011402 012767 000046 166442 MOV. #Q$FP,FACODE. ;MOVE POINTER MEMORY CODE TO FAL MODULE.
1695 011410 000432 BR. LSTACK. ;
1696 ;
1697 011412 022701 000007 3$: CMP. #7,R1 ;FAL COUNTER MEMORY.
1698 011416 001004 BNE. 4$ ;
1699 011420 012767 000045 166424 MOV. #Q$FC,FACODE. ;MOVE COUNTER MEMORY CODE TO FAL MODULE.
1700 011426 000423 BR. LSTACK. ;
1701 ;
1702 011430 022701 000011 4$: CMP. #9,R1 ;QLB PAGE 0
1703 011434 001003 BNE. 5$ ;
1704 011436 005067 166412 CLR. QBPAGE. ;SET PAGE 0
1705 011442 000415 BR. LSTACK. ;
1706 ;
1707 011444 022701 000012 5$: CMP. #10,R1 ;QLB PAGE 1
1708 011450 001004 BNE. 6$ ;NO. MUST BE PAGE 2.
1709 011452 012767 000001 166374 MOV. #1,QBPAGE. ;SET PAGE 1
1710 011460 000406 BR. LSTACK. ;
1711 ;
1712 011462 022701 000013 6$: CMP. #11,R1 ;QLB PAGE 2.
1713 011466 001003 BNE. LSTACK. ;NONE OF THE ABOVE.
1714 011470 012767 000002 166356 MOV. #2,QBPAGE. ;SET PAGE 2.
1715 ;
1716 ;
1717 ; USE THE REMAINDER FROM THE ABOVE DIVISION (X4) AS AN
1718 ; OFFSET INTO THE DOUBLE-WORD MEMORY LIMITS TABLE. MOVE
1719 ; THE "CURRENT" MEMORY LIMITS TO THE STACK.
1720 ;
1721 ;
1722 ; STORE THE REMAINDER (X2 FOR WORD OFFSET) INTO R0 FOR
1723 ; USE BY THE MEMORY TEST CONTROL ROUTINE.
1723 011476 006301 LSTACK: ASL. R1 ;SHIFT FOR WORD OFFSET
1724 011500 010100 MOV. R1,R0 ;SAVE WORD OFFSET FOR INDEXING
1725 011502 006301 ASL. R1 ;SHIFT FOR DOUBLE WORD OFFSET
1726 011504 016146 002276 MOV. CURLIM(R1),-(SP) ;MOVE UPPER LIMIT TO STACK
1727 011510 016146 002300 MOV. CURLIM+2(R1),-(SP) ;MOVE LOWER LIMIT TO STACK
1728 ;

```

QMT.....MACRO:M1110 27-MAR-80 15:19 PAGE 18-4

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

1729 011514 016701 166502
1730 011520 000171 000000

LASTJ: MOV: MTPNT,R1
JMP: @R1)

:POINT TO TEST CONTROL ROUTINE
:GO THERE

```

1732.      ;
1733      ;
1734      ; REGISTER CONTROL ROUTINES.
1735      ;
1736      ; TESTS ARE FULLY DESCRIBED IN THE QMT SUB-MODULE RTEST.
1737      ;
1738      ;
1739      ; TEST-01
1740      ;
1741 011524      T1R:
1742 011524 005067 171106      CLR      CKDATA      ;SET TEST PATTERN TO ZERO.
1743 011530      CALL     @STRADD(R0)
1744 011534 000167 000044      JMP      RGJUMP      ;RETURN TO TOP OF MAIN LOOP.
1745      ;
1746      ;
1747      ; TEST-02.
1748      ;
1749 011540      T2R:
1750 011540 012767 177777 171070      MOV      #-1,CKDATA      ;SET TEST PATTERN TO ALL ONE'S
1751 011546      CALL     @STRADD(R0)
1752 011552 000167 000026      JMP      RGJUMP      ;RETURN TO TOP OF MAIN LOOP.
1753      ;
1754      ;
1755      ; TEST-03
1756      ;
1757 011556      T3R:
1758 011556 016767 166274 171052      MOV      RT3,CKDATA      ;SET USER SUPPLIED TEST PATTERN.
1759 011564      CALL     @STRADD(R0)
1760 011570 000167 000010      JMP      RGJUMP      ;RETURN TO TOP OF MAIN LOOP.
1761      ;
1762      ;
1763      ; TEST-04
1764      ;
1765 011574      T4R:
1766 011574      CALL     @RT4ADD(R0)
1767 011600 000167 000000      JMP      RGJUMP
1768      ;
1769      ;
1770 011604      RGJUMP:
1771 011604 000167 177174      JMP      MTMAIN.

```



```

1773 ;
1774 ;
1775 ; MEMORY TEST CONTROL ROUTINES.
1776 ;
1777 ; TESTS ARE FULLY DESCRIBED IN THE QMT MEMORY TEST SUB-MODULES.
1778 ;
1779 ;
1780 ; TEST-01
1781 ;
1782 011610 T1:
1783 011610 CALL @TIADDR(R0)
1784 011614 000167 000514 JMP MTJUMP
1785 ;
1786 ;
1787 ; TEST-02
1788 ;
1789 011620 T2:
1790 011620 005067 171012 CLR CKDATA ; TEST PATTERN = 0
1791 011624 CALL @STADDR(R0)
1792 011630 000167 000500 JMP MTJUMP
1793 ;
1794 ;
1795 ; TEST-03
1796 ;
1797 011634 T3:
1798 011634 012767 177777 170774 MOV #-1,CKDATA ; TEST PATTERN = -1
1799 011642 CALL @STADDR(R0)
1800 011646 000167 000462 JMP MTJUMP
1801 ;
1802 ;
1803 ; TEST-04
1804 ;
1805 011652 T4:
1806 011652 012767 125252 170756 MOV #125252,CKDATA ; TEST PATTERN = 'X'AAAA'
1807 011660 CALL @STADDR(R0)
1808 011664 000167 000444 JMP MTJUMP
1809 ;
1810 ;
1811 ; TEST-05
1812 ;
1813 011670 T5:
1814 011670 012767 146314 170740 MOV #146314,CKDATA ; TEST PATTERN = 'X'CCCC'
1815 011676 CALL @STADDR(R0)
1816 011702 012767 031463 170726 MOV #031463,CKDATA ; TEST PATTERN = '3333'
1817 011710 CALL @STADDR(R0)
1818 011714 000167 000414 JMP MTJUMP
1819 ;
1820 ;
1821 ; TEST-06
1822 ;
1823 011720 T6:
1824 011720 005067 170712 CLR CKDATA ; SET TEST PATTERN TO ZERO
1825 011724 CALL @STADDR(R0)
1826 011730 CALL @T6ADDR(R0) ; PERFORM CROSS-TALK TEST
1827 011734 005216 INC (SP) ; PERFORM AT NEXT ADDRESS
1828 011736 005067 170674 CLR CKDATA ; RESET TEST PATTERN
1829 011742 CALL @STADDR(R0)

```

```

1830 011746          CALL @T6ADDR(R0)
1831 011752 005316    DEC (SP)
1832 011754 000167 000354    JMP MTJUMP          ;RESTORE ORIGINAL LOWER LIMITS
1833                ;
1834                ;
1835                ;
1836                ;
1837 011760          TEST 07
1838 011760          T7:
1839 011764 000167 000344    CALL @T7ADDR(R0)
1840                ;
1841                ;
1842                ;
1843                ;
1844 011770          TEST 08
1845 011770 012767 000377 170640    MOV #377,CKDATA
1846 011776          CALL @STADDR(R0)          ;SET TEST PATTERN TO X'00FF'
1847 012002 012767 177400 170626    MOV #177400,CKDATA
1848 012010          CALL @STADDR(R0)          ;SET TEST PATTERN TO X'FF00'
1849 012014 000167 000314    JMP MTJUMP
1850                ;
1851                ;
1852                ;
1853                ;
1854 012020          TEST 09
1855 012020 012767 000001 170610    MOV #1,CKDATA
1856 012026          CALL @STADDR(R0)          ;START TEST PATTERN AT 1
1857 012032 016702 170600    MOV CKDATA,R2
1858 012036 006302          ASL R2
1859 012040 010267 170572          MOV R2,CKDATA
1860 012044 005702          TST R2
1861 012046 001367          BNE 1$
1862 012050 000167 000260    JMP MTJUMP          ;NEXT TEST PATTERN
1863                ;
1864                ;
1865                ;
1866                ;
1867 012054          TEST 0A
1868 012054 016767 166000 170554    MOV MT10,CKDATA
1869 012062          CALL @STADDR(R0)          ;USER PATTERN
1870 012066 000167 000242    JMP MTJUMP
1871                ;
1872                ;
1873                ;
1874                ;
1875 012072          TEST 0B
1876 012072 012767 100001 170536    MOV #100001,CKDATA
1877 012100          CALL @STADDR(R0)          ;=B'1000000000000001'
1878 012104 012767 040002 170524    MOV #040002,CKDATA
1879 012112          CALL @STADDR(R0)          ;=B'010000000000010'
1880 012116 012767 020004 170512    MOV #020004,CKDATA
1881 012124          CALL @STADDR(R0)          ;=B'001000000000100'
1882 012130 012767 010010 170500    MOV #010010,CKDATA
1883 012136          CALL @STADDR(R0)          ;=B'000100000001000'
1884 012142 012767 004020 170466    MOV #004020,CKDATA
1885 012150          CALL @STADDR(R0)          ;=B'0000100000010000'
1886 012154 012767 002040 170454    MOV #002040,CKDATA

```

```

1887 012162.
1888 012166 012767 001100 170442. CALL @STADDR(R0)
1889 012174 MOV #001100,CKDATA ;=B'0000001001000000'
1890 012200 012767 000600 170430 CALL @STADDR(R0)
1891 012206 MOV #000600,CKDATA ;=B'0000000110000000'
1892 012212 000167 000116 CALL @STADDR(R0)
1893 JMP MTJUMP.
1894 ;
1895 ;
1896 ; TEST-0C.
1897 012216 ; TC:
1898 012216 005067 170414 CLR CKDATA ;SET TEST PATTERN = 0
1899 012222 CALL @STADDR(R0)
1900 012226 005067 170406 CLR CK2 ;SET READ PATTERN = 0
1901 012232 012767 177777 170402. MOV #-1,CK3 ;SET WRITE PATTERN = X'FFFF'
1902 012240 CALL @TCDADD(R0) ;READ X'0000', WRITE X'FFFF'
1903 012244 012767 177777 170366 MOV #-1,CK2 ;SET READ PATTERN = X'FFFF'
1904 012252 005067 170364 CLR CK3 ;SET WRITE PATTERN = 0
1905 012256 CALL @TCDADD(R0) ;READ X'FFFF', WRITE X'0000'
1906 ;
1907 012262 005067 170352 CLR CK2 ;SET READ PATTERN = 0
1908 012266 012767 177777 170346 MOV #-1,CK3 ;SET WRITE PATTERN = X'FFFF'
1909 012274 CALL @TCUADD(R0) ;READ/WRITE FROM BOTTOM UP
1910 012300 012767 177777 170332. MOV #-1,CK2 ;SET READ PATTERN = X'FFFF'
1911 012306 005067 170330 CLR CK3 ;CLEAR READ PATTERN
1912 012312 CALL @TCUADD(R0)
1913 012316 000167 000012 JMP MTJUMP.
1914 ;
1915 ;
1916 ; TEST-0D.
1917 ; TD:
1918 012322.
1919 012322. CALL @TDADDR(R0)
1920 012326 000167 000002 JMP MTJUMP.
1921 ;
1922 012332. TDNUL:
1923 012332. RETURN.
1924 ;
1925 ;
1926 012334 MTJUMP:
1927 012334 062706 000004 ADD #4,SP.
1928 012340 000167 176440 JMP MTMAIN.

```


Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

1951 ;
1952 ;
1953 ;
1954 ;
1955 ;
1956 ;
1957 ;
1958 ;
1959 ;
1960 ;
1961 ;
1962 ;
1963 ;
1964 ;
1965 012420 ;
1966 012420 016767 176420 165404 ;
1967 012426 046667 000004 165376 ;
1968 012434 056667 000002 165370 ;
1969 012442 016767 165364 176420 ;
1970 012450 011666 000004 ;
1971 012454 022626 ;
1972 012456 ;

ROUTINE TO PLACE A VALUE INTO CONTROL/STATUS REGISTER
NUMBER 1.

READ THE CURRENT VALUE OF THE CSR INTO A WORK AREA,
CLEAR THE BITS AT 4(SP), SET THE BITS AT 2(SP),
REWRITE CSR1 FROM THE WORK AREA.

INPUT:
2(SP) BITS TO BE SET IN CSR1
4(SP) BITS TO BE CLEARED IN CSR1

CSR1::
MOV QR$CR1,APLACE ;GET THE CURRENT VALUE.
BIC 4(SP),APLACE ;CLEAR FIRST.
BIS 2(SP),APLACE ;THEN SET.
MOV APLACE,QR$CR1 ;NOW RETURN IT.
MOV (SP),4(SP) ;MOVE RETURN ADDR TO TOP OF STACK.
CMP (SP)+,(SP)+ ;BUMP STACK POINTER PAST ARGS.
RETURN ;SPLIT.

```

```

1974 ;
1975 ;
1976 ; REGISTER TEST ERROR ROUTINE.
1977 ;
1978 ;
1979 012460 REGERR:
1980 012460 ;
1981 ;
1982 ; SAVE R0,R1,R2,R3,R4,R5
1983 ;
1984 ; SET FLAG FOR ERROR ENCOUNTERED. AN ERROR MESSAGE LIMIT
1985 ; COUNT OF ZERO MEANS THAT THE COUNT IS NOT BEING USED.
1986 ; A COUNT OF -1 MEANS THAT THE LIMIT HAS BEEN REACHED
1987 ; (NO MORE ERROR MESSAGES ARE TO BE PRINTED).
1988 012474 052767 000400 165326 BIS #ERROR,BASE ;SET FLAG FOR ERROR ENCOUNTERED
1989 012502 005767 165312 TST ERLIM ;IS ERROR COUNT BEING USED
1990 012506 001412 BEQ 2$ ;NO
1991 012510 003002 BGT 1$ ;DECREMENT COUNT
1992 012512 000167 000334 JMP REGX
1993 ;
1994 ; DECREMENT ERROR LIMIT COUNT. IF IT GOES ZERO HERE, SET
1995 ; IT TO -1.
1996 012516 005367 165276 1$: DEC ERLIM
1997 012522 001004 BNE 2$
1998 012524 005367 165270 DEC ERLIM ;SET ERROR LIMIT FIELD TO -1
1999 012530 000167 000316 JMP REGX
2000 ;
2001 ; PRINT TEST NUMBER
2002 ;
2003 012534 012705 003337 2$: MOV #PRINT,R5 ;POINT AT PRINT LINE
2004 012540 012700 002677 MOV #TMMSG,R0 ;POINT AT TEST
2005 012544 012701 000005 MOV #5,R1 ;NUMBER OF BYTES IN STRING
2006 012550 112025 3$: MOVB (R0)+,(R5)+ ;MOVE LABEL TO PRINT LINE
2007 012552 005301 DEC R1
2008 012554 001375 BNE 3$
2009 ;
2010 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
2011 ; REGISTER TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
2012 ; TABLE.
2013 ;
2014 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE. DIVIDE
2015 ; BY THE NUMBER OF REGISTERS IN THE TABLE. THE QUOTIENT
2016 ; IS THE TEST NUMBER (ZERO RELATIVE). THE REMAINDER IS THE
2017 ; REGISTER PLACEHOLDER VALUE. SAVE IT.
2018 ;
2019 ; BACK OUT THE NUMBER OF MEMORY TEST ENTRIES FROM THE
2020 ; CURRENT COUNT IF NECESSARY. FOR A COMPLETE DISCUSSION
2021 ; OF HOW THIS IS DONE, SEE ABOVE AT 'JMPNT'.
2022 ;
2023 012556 012700 000010 MOV #<RT*NREGS>,R0 ;TOTAL NUMBER OF ROUTINES
2024 012562 016701 165436 MOV MTCNT,R1 ;LOAD CURRENT COUNT
2025 012566 032767 000040 BIT #MEMORY,BASE ;MEMORY TESTS SELECTED
2026 012574 001402 BEQ 4$ ;NO, SKIP ADJUSTMENT
2027 012576 162701 000234 SUB #<MT*NMEMS>,R1 ;SUBTRACT OUT NUMBER OF MEM TESTS
2028 012602 160100 4$: SUB R1,R0 ;GET CURRENT POSITION
2029 012604 012701 000002 MOV #NREGS,R1 ;NUMBER OF REGISTERS
2030 012610 CALL $DIV ;DIVIDE FOR TEST NUMBER

```

```

2031 012614 010146          MOV.    R1, -(SP)          ;SAVE REMAINDER.
2032 012616 010001          MOV.    R0, R1           ;PREPARE FOR CONVERSION.
2033 012620 005201          INC.    R1              ;ADJUST ZERO RELATIVE NUMBER.
2034 012622 012700 002672'  MOV.    #ASWRK, R0      ;POINT TO CONVERSION WORK AREA
2035 012626 012702 000001  MOV.    #1, R2          ;KEEP ZEROS.
2036 012632                CALL.   $CBDSG          ;CONVERT TEST NUMBER TO ASCII.
2037                                ;
2038 012636 116725 170033  MOV.    ASWRK+3, (R5)+   ;MOVE A DIGIT OF TEST NUMBER.
2039 012642 116725 170030  MOV.    ASWRK+4, (R5)+   ;
2040 012646 062705 000002  ADD.    #2, R5          ;BUMP PRINT LINE POINTER.
2041                                ;
2042                                ;
2043                                ;
2044 012652 012700 002704'  MOV.    #PMSG, R0       ;POINT AT "PASS."
2045 012656 012701 000005  MOV.    #5, R1          ;NUMBER OF BYTES IN STRING.
2046 012662 112025 5$:    MOV.    (R0)+, (R5)+   ;MOVE LABEL TO PREIN LINE.
2047 012664 005301          DEC.    R1              ;
2048 012666 001375          BNE.    5$             ;
2049 012670                CALL.   PASC           ;CONVERT NUMBER OF PASSES.
2050 012674 062705 000003  ADD.    #3, R5          ;ADVANCE PRINT LINE POINTER.
2051                                ;
2052                                ;
2053                                ;
2054                                ;
2055 012700 012601          MOV.    (SP)+, R1       ;GET REMAINDER.
2056 012702 012700 000007  MOV.    #7, R0         ;LENGTH OF MESSAGE STRING IN TABLE.
2057 012706                CALL.   $MUL           ;
2058 012712 012700 002753'  MOV.    #RFTBL, R0     ;POINT TO TABLE OF MESSAGES.
2059 012716 060100          ADD.    R1, R0         ;POINT TO REG THAT FAILED.
2060 012720 012701 000007  MOV.    #7, R1         ;NUMBER OF CHARS IN NAME.
2061 012724 112025 6$:    MOV.    (R0)+, (R5)+   ;MOVE NAME TO PRINT LINE.
2062 012726 005301          DEC.    R1              ;
2063 012730 001375          BNE.    6$             ;
2064 012732                CALL.   CONSOL        ;WRITE TO CONSOLE.
2065                                ;
2066                                ;
2067                                ;
2068                                ;
2069 012736 012705 003337'  MOV.    #PRINT, R5     ;POINT TO START OF PRINT LINE.
2070 012742 012700 002722'  MOV.    #EMSG, R0      ;POINT TO "EXPECTED."
2071 012746 012701 000012  MOV.    #10, R1        ;LOAD NUMBER OF CHARS.
2072 012752 112025 7$:    MOV.    (R0)+, (R5)+   ;MOVE LABEL.
2073 012754 005301          DEC.    R1              ;
2074 012756 001375          BNE.    7$             ;
2075                                ;
2076 012760 016701 167652  MOV.    CKDATA, R1     ;LOAD MEMORY TEST PATTERN.
2077 012764                CALL.   UNPK           ;CONVERT.
2078 012770 005205          INC.    R5             ;BUMP PRINT LINE POINTER.
2079                                ;
2080 012772 012700 002734'  MOV.    #RMSG, R0      ;POINT TO "RECEIVED."
2081 012776 012701 000012  MOV.    #10, R1        ;LOAD NUMBER OF CHARS.
2082 013002 112025 8$:    MOV.    (R0)+, (R5)+   ;MOVE LABEL.
2083 013004 005301          DEC.    R1              ;
2084 013006 001375          BNE.    8$             ;
2085                                ;
2086                                ;
2087                                ;
COUNT OF NUMBER OF WORDS TO PRINT = 0 SIGNALS THAT
THE RESULTS OF SEVERAL REGISTER READS WERE UNEXPECTED.

```

```

2088 ; PRINT: * * *
2089 ;
2090 013010 016700 167634 MOV. ERRCT,R0 ;LOAD: NUMBER OF WORDS TO CONVERT.
2091 013014 003010 BGT. 10$ ;PRINT: WORDS.
2092 013016 012700 002746' MOV. #UNMSG,R0 ;* * *
2093 013022 012701 000005 MOV. #5,R1 ;LENGTH OF MESSAGE.
2094 013026 112025 9$: MOVB. (R0)+,(R5)+
2095 013030 005301 DEC. R1
2096 013032 001375 BNE. 9$
2097 013034 000404 BR. 11$
2098 ;
2099 013036 016701 167610 10$: MOV. ERW1,R1 ;LOAD: ERROR WORD FOR SUBRTN.
2100 013042. CALL. UNPK ;CONVERT.
2101 013046 11$: CALL. CONSOL. ;ELSE: WRITE TO CONSOLE.
2102. ;
2103 013052. REGX: RESTOR. R0,R1,R2,R3,R4,R5
2104 013052. RETURN.
2105 013066

```



```

2107 ;
2108 ;
2109 ; MEMORY TEST ERROR ROUTINE.
2110 ;
2111 013070 MEMERR::
2112 013070 SAVE R0,R1,R2,R3,R4,R5
2113 ;
2114 ; SET FLAG FOR ERROR ENCOUNTERED; AN ERROR MESSAGE LIMIT
2115 ; COUNT OF ZERO MEANS THAT THE COUNT IS NOT BEING USED.
2116 ; A COUNT OF -1 MEANS THAT THE LIMIT HAS BEEN REACHED.
2117 ; (NO MORE ERROR MESSAGES ARE TO BE PRINTED).
2118 ;
2119 013104 052767 000400 164716 BIS #ERROR,BASE ;SET FLAG FOR ERROR ENCOUNTERED.
2120 013112 005767 164702 TST ERLIM ;IS ERROR COUNT BEING USED.
2121 013116 001412 BEQ 2$
2122 013120 003002 BGT 1$
2123 013122 000167 000372 JMP MEMX
2124 ;
2125 ; DECREMENT ERROR LIMIT COUNT. IF IT GOES ZERO HERE, SET
2126 ; IT TO -1.
2127 ;
2128 013126 005367 164666 1$: DEC ERLIM.
2129 013132 001004 BNE 2$
2130 013134 005367 164660 DEC ERLIM. ;SET ERROR LIMIT FIELD TO -1.
2131 013140 000167 000354 JMP MEMX
2132 ;
2133 ; PRINT TEST NUMBER.
2134 ;
2135 013144 012705 003337 2$: MOV #PRINT,R5 ;POINT AT PRINT LINE.
2136 013150 012700 002677 MOV #TMSG,R0 ;POINT AT TEST.
2137 013154 012701 000005 MOV #5,R1 ;NUMBER OF BYTES IN STRING.
2138 013160 112025 3$: MOVB (R0)+(R5)+ ;MOVE LABEL TO PRINT LINE.
2139 013162 005301 DEC R1
2140 013164 001375 BNE 3$
2141 ;
2142 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
2143 ; MEMORY TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
2144 ; TABLE.
2145 ;
2146 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE. DIVIDE
2147 ; BY THE NUMBER OF MEMORIES IN THE TABLE. THE QUOTIENT
2148 ; IS THE TEST NUMBER (ZERO-RELATIVE). THE REMAINDER IS THE
2149 ; MEMORY PLACE-HOLDER VALUE. SAVE IT.
2150 ;
2151 ;
2152 ; BACK OUT THE NUMBER OF REGISTER TEST ENTRIES FROM THE
2153 ; CURRENT COUNT IF NECESSARY. FOR A COMPLETE DISCUSSION
2154 ; OF HOW THIS IS DONE, SEE ABOVE AT 'MEMOFF'.
2155 ;
2155 013166 012700 000234 MOV #<MT*NMEMS>,R0 ;TOTAL NUMBER OF ROUTINES.
2156 013172 016701 165026 MOV MTCNT,R1 ;LOAD CURRENT COUNT.
2157 013176 032767 000020 164624 BIT #REGSTR,BASE ;REGISTER TESTS SELECTED.
2158 013204 001402 BEQ 4$ ;NO SKIP ADJUSTMENT
2159 013206 162701 000010 SUB #<RT*NREGS>,R1 ;SUBTRACT OUT NUMBER OF REG TESTS.
2160 013212 160100 4$: SUB R1,R0 ;GET CURRENT POSITION.
2161 013214 012701 000014 MOV #NMEMS,R1 ;NUMBER OF MEMORIES.
2162 013220 CALL $DIV ;DIVIDE FOR TEST NUMBER.
2163 013224 010146 MOV R1,-(SP) ;SAVE REMAINDER.

```

```

2164 013226 010001      MOV.   R0,R1      ;PREPARE FOR CONVERSION.
2165 013230 005201      INC.   R1         ;ADJUST ZERO RELATIVE NUMBER.
2166 013232 012700 002672' MOV.   #ASURK,R0  ;POINT TO CONVERSION WORK AREA
2167 013236 012702 000001 MOV.   #1,R2     ;KEEP ZEROS.
2168 013242          CALL.  $CBDSG    ;CONVERT TEST NUMBER TO ASCII.
2169          ;
2170 013246 116725 167423 MOVVB. ASURK+3.(R5)+ ;MOVE A DIGIT OF TEST NUMBER.
2171 013252 116725 167420 MOVVB. ASURK+4.(R5)+
2172 013256 062705 000002 ADD.   #2,R5     ;BUMP PRINT LINE POINTER.
2173          ;
2174          ;
2175          ;
2176 013262 012700 002704' MOV.   #PMSG,R0  ;POINT AT "PASS."
2177 013266 012701 000005 MOV.   #5,R1     ;NUMBER OF BYTES IN STRING.
2178 013272 112025          MOVVB. (R0)+.(R5)+ ;MOVE LABEL TO PRINT LINE.
2179 013274 005301      DEC.   R1
2180 013276 001375      BNE.   5$
2181 013300          CALL.  PASSC    ;CONVERT NUMBER OF PASSES.
2182 013304 062705 000003 ADD.   #3,R5     ;ADVANCE PRINT LINE POINTER.
2183          ;
2184          ;
2185          ;
2186          ;
2187 013310 012601      MOV.   (SP)+,R1  ;GET REMAINDER.
2188 013312 012700 000023 MOV.   #19,R0   ;LENGTH OF MESSAGE STRING IN TABLE.
2189 013316          CALL.  $MUL     ;
2190 013322 012700 002771' MOV.   #MFTBL,R0 ;POINT TO TABLE OF MESSAGES.
2191 013326 060100      ADD.   R1,R0    ;POINT TO REG THAT FAILED.
2192 013330 012701 000023 MOV.   #19,R1   ;NUMBER OF CHARS IN NAME.
2193 013334 112025          MOVVB. (R0)+.(R5)+ ;MOVE NAME TO PRINT LINE.
2194 013336 005301      DEC.   R1
2195 013340 001375      BNE.   6$
2196 013342          CALL.  CONSOL  ;WRITE TO CONSOLE.
2197          ;
2198          ;
2199          ;
2200          ;
2201 013346 012705 003337' MOV.   #PRINT,R5 ;POINT TO PRINT LINE
2202 013352 012700 002711' MOV.   #AMSG,R0  ;POINT TO "ADDRESS."
2203 013356 012701 000011 MOV.   #9,R1     ;LOAD LENGTH OF STRING.
2204 013362 112025          MOVVB. (R0)+.(R5)+ ;MOVE LABEL.
2205 013364 005301      DEC.   R1
2206 013366 001375      BNE.   7$
2207          ;
2208 013370 016701 167252 MOV.   ERRADD,R1 ;LOAD ERROR ADDRESS.
2209 013374          CALL.  UNPK    ;CONVERT TO PRINTABLE CHARS.
2210 013400 005205      INC.   R5       ;BUMP PRINT LINE POINTER.
2211          ;
2212 013402 012700 002722' MOV.   #EMSG,R0  ;POINT TO "EXPECTED."
2213 013406 012701 000012 MOV.   #10,R1   ;LOAD NUMBER OF CHARS.
2214 013412 112025          MOVVB. (R0)+.(R5)+ ;MOVE LABEL.
2215 013414 005301      DEC.   R1
2216 013416 001375      BNE.   8$
2217          ;
2218 013420 016701 167212 MOV.   CKDATA,R1 ;LOAD MEMORY TEST PATTERN.
2219 013424          CALL.  UNPK    ;CONVERT.
2220 013430 005205      INC.   R5       ;BUMP PRINT LINE POINTER.

```

```

2221
2222 013432 012700 002734'
2223 013436 012701 000012
2224 013442 112025
2225 013444 005301
2226 013446 001375
2227
2228
2229
2230
2231
2232 013450 016700 167174
2233 013454 003010
2234 013456 012700 002746'
2235 013462 012701 000005
2236 013466 112025
2237 013470 005301
2238 013472 001375
2239 013474 000407
2240
2241 013476 012702 002652'
2242 013502 012201
2243 013504
2244 013510 005300
2245 013512 001373
2246 013514
2247
2248 013520
2249 013520
2250 013534

;
MOV #RMSG,R0 ;POINT TO 'RECEIVED'
MOV #10,R1 ;LOAD NUMBER OF CHARS
9$: MOVB (R0)+,(R5)+ ;MOVE LABEL
DEC R1
BNE 9$

;
COUNT OF NUMBER OF WORDS TO PRINT = 0 SIGNALS THAT
; THE RESULTS OF SEVERAL MEMORY READS WERE UNPREDICTABLE.
; PRINT * * *
;
MOV ERCT,R0 ;LOAD NUMBER OF WORDS TO CONVERT
BGT 11$ ;PRINT WORDS
MOV #UNMSG,R0 ;* * *
MOV #5,R1 ;LENGTH OF MESSAGE
10$: MOVB (R0)+,(R5)+
DEC R1
BNE 10$
BR 13$

;
11$: MOV #ERW1,R2 ;POINT TO FIRST OF THEM
12$: MOV (R2)+,R1 ;LOAD THE WORD ITSELF
CALL UNPK
DEC R0
BNE 12$
13$: CALL CONSOL ;ELSE WRITE TO CONSOLE
;
MEMX:
RESTOR R0,R1,R2,R3,R4,R5
RETURN
    
```

```

2252. ;
2253. ;
2254. ; AST.
2255. ;
2256. ;
2257. ; INTERCEPT CHARACTER FROM CONSOLE
2258. ;
2259. ; ON 'W' PRINT TEST NUMBER, PASS COUNT, MEMORY NAME, ADDRESS,
2260. ; AND TEST PATTERN (MEMORIES ONLY) OR TEST NUMBER, PASS COUNT,
2261. ; REGISTER NAME, AND TEST PATTERN (REGISTERS ONLY)
2262. ; ON 'C' PRINT TEST PATTERN ('CKDATA')
2263. ; ON 'P' PRINT NUMBER OF PASSES
2264. ; ON 'T' PRINT TEST NUMBER
2265. ; EXIT ON ANY OTHER CHARACTER
2266. ;
2267. ;
2268. 013536 AST:
2269. 013536 012667 164246 MOV. (SP)+,ASTWRD. ;GET CHAR.
2270. 013542. SAVE. R0,R1,R2,R3,R4,R5
2271. ;
2272. 013556 122767 000120 164224 CMPB. #'P,ASTWRD. ;PRINT NUMBER OF PASSES AND CONTINUE
2273. 013564 001002. BNE. 1$ ;NO. NEXT OPTION
2274. 013566 000167 000652 JMP. PAST
2275. 013572. 122767 000103 164210 1$: CMPB. #'C,ASTWRD. ;PRINT OUT TEST PATTERN
2276. 013600 001002. BNE. 2$ ;NO. NEXT OPTION
2277. 013602. 000167 000416 JMP. CAST
2278. 013606 122767 000124 164174 2$: CMPB. #'T,ASTWRD. ;PRINT OUT TEST NUMBER
2279. 013614 001002. BNE. 3$ ;NO. NEXT OPTION
2280. 013616 000167 000424 JMP. TAST
2281. 013622. 122767 000127 164160 3$: CMPB. #'W,ASTWRD. ;PRINT WHERE
2282. 013630 001402. BEQ. WAST
2283. 013632. 000167 000572 JMP. EAST ;PRINT NUMBER OF PASSES AND EXIT
2284. ;
2285. ;
2286. ; ROUTINE FOR 'W'
2287. ;
2288. ;
2289. 013636 WAST:
2290. 013636 026727 164360 001606' CMP. MTPNT,MTSUB. ;REGISTER OR MEMORY ?
2291. 013644 103402. BLO. 1$ ;REGISTER
2292. 013646 000167 000160 JMP. MAST ;MEMORY
2293. ;
2294. ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
2295. ; REGISTER TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
2296. ; TABLE.
2297. ;
2298. ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE. DIVIDE
2299. ; BY THE NUMBER OF REGISTERS IN THE TABLE. THE QUOTIENT
2300. ; IS THE TEST NUMBER (ZERO RELATIVE). THE REMAINDER IS THE
2301. ; REGISTER PLACE HOLDER VALUE. SAVE IT.
2302. ;
2303. ; BACK OUT THE NUMBER OF MEMORY TEST ENTRIES FROM THE
2304. ; CURRENT COUNT IF NECESSARY. FOR A COMPLETE DISCUSSION
2305. ; OF HOW THIS IS DONE, SEE ABOVE AT 'JMPMT'.
2306. ;
2307. 013652. 012700 000010 1$: MOV. #CRT*NREGS>,R0 ;TOTAL NUMBER OF ROUTINES
2308. 013656 016701 164342 MOV. MTCNT,R1 ;LOAD CURRENT COUNT

```



```

2423 ; TEST-PATTERN
2424 ;
2425 014204 016701 166426 ; MOV CKDATA,R1 ;LOAD TEST PATTERN
2426 014210 ; CALL UNPK ;CONVERT
2427 014214 ; CALL CONSOL ;WRITE LINE TO CONSOL
2428 014220 000167 000272 ; JMP REST ;EXIT-AST
2429 ;
2430 ;
2431 ;
2432 ;
2433 ; PRINT TEST PATTERN ALONE
2434 ;
2435 014224 ; CAST:
2436 014224 012705 003337* ; MOV #PRINT,R5
2437 014230 016701 166402 ; MOV CKDATA,R1 ;LOAD TEST PATTERN
2438 014234 ; CALL UNPK ;CONVERT TO ASCII
2439 014240 ; CALL CONSOL ;WRITE TO TT0
2440 014244 000524 ; BR REST
2441 ;
2442 ;
2443 ;
2444 ;
2445 ; PRINT OUT TEST NUMBER ALONE
2446 ;
2447 014246 ; TAST:
2448 014246 026727 163750 001606* ; CMP MTPNT,#MTPNT ;REGISTER OR MEMORY?
2449 014254 103016 ; BHS 2$ ;MEMORY
2450 ;
2451 ;
2452 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
2453 ; REGISTER TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
2454 ; TABLE
2455 ;
2456 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
2457 ; BY THE NUMBER OF REGISTERS IN THE TABLE, THE QUOTIENT
2458 ; IS THE TEST NUMBER (ZERO-RELATIVE)
2459 ;
2460 ; BACK OUT THE NUMBER OF MEMORY TEST ENTRIES FROM THE
2461 ; CURRENT COUNT IF NECESSARY, FOR A COMPLETE DISCUSSION
2462 ; OF HOW THIS IS DONE, SEE ABOVE AT 'JMPMT'
2463 ;
2464 014256 012700 000010 ; MOV #<RT*NREGS>,R0 ;TOTAL NUMBER OF ROUTINES
2465 014262 016701 163736 ; MOV MTCNT,R1 ;LOAD CURRENT COUNT
2466 014266 032767 000040 163534 ; BIT #MEMORY,BASE ;MEMORY TESTS SELECTED
2467 014274 001402 ; BEQ 1$ ;NO SKIP ADJUSTMENT
2468 014276 162701 000234 ; SUB #<MT*NMEMS>,R1 ;SUBTRACT OUT NUMBER OF MEM TESTS
2469 014302 160100 ; SUB R1,R0 ;GET CURRENT POSITION
2470 014304 012701 000002 1$: ; MOV #NREGS,R1 ;NUMBER OF REGISTERS
2471 014310 000415 ; BR 4$ ;NOW DO THE REST
2472 ;
2473 ;
2474 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
2475 ; MEMORY TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
2476 ; TABLE
2477 ;
2478 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
2479 ; BY THE NUMBER OF MEMORIES IN THE TABLE, THE QUOTIENT

```

```

2480 ; BACK OUT THE NUMBER OF REGISTER TEST ENTRIES FROM THE
2481 ; CURRENT COUNT IF NECESSARY, FOR A COMPLETE DISCUSSION
2482 ; OF HOW THIS IS DONE, SEE ABOVE AT 'MEMOFF'.
2483 ;
2484 014312 012700 000234 2$: MOV #<MT*NMEMS>,R0 ;TOTAL NUMBER OF ROUTINES
2485 014316 016701 163702 MOV MTCNT,R1 ;LOAD CURRENT COUNT
2486 014322 032767 000020 163500 BIT #REGSTR,BASE ;REGISTER TESTS SELECTED
2487 014330 001402 BEO 3$ ;NO SKIP ADJUSTMENT
2488 014332 162701 000010 SUB #<RT*NREGS>,R1 ;SUBTRACT OUT NUMBER OF REG TESTS
2489 014336 160100 SUB R1,R0 ;GET CURRENT POSITION
2490 014340 012701 000014 MOV #NMEMS,R1 ;NUMBER OF MEMORIES
2491 ;
2492 014344 4$: CALL $DIV ;DIVIDE FOR TEST NUMBER
2493 014350 010001 MOV R0,R1 ;PREPARE FOR CONVERSION
2494 014352 005201 INC R1 ;ADJUST ZERO RELATIVE NUMBER
2495 014354 012700 002672* MOV #ASWRK,R0 ;POINT TO CONVERSION WORK AREA
2496 014360 012702 000001 MOV #1,R2 ;KEEP ZEROS
2497 014364 CALL $CDSG ;CONVERT TEST NUMBER TO ASCII
2498 ;
2499 014370 012705 003337* MOV #PRINT,R5 ;POINT TO PRINT LINE
2500 014374 012700 002677* MOV #TMSG,R0 ;POINT TO TEST #
2501 014400 012701 000005 MOV #S,R1 ;NUMBER OF CHARS IN STRING
2502 014404 112025 5$: MOVB (R0)+(R5)+ ;MOVE STRING TO PRINT LINE
2503 014406 005301 DEC R1
2504 014410 001375 BNE 5$
2505 ;
2506 014412 116725 166257 MOVB ASWRK+3,(R5)+ ;MOVE A DIGIT OF TEST NUMBER
2507 014416 116725 166254 MOVB ASWRK+4,(R5)+
2508 014422 CALL CONSOL ;PRINT TEST NUMBER
2509 014426 000433 BR REST ;EXIT AST
2510 ;
2511 ;
2512 ;
2513 ;
2514 ; EXIT PROGRAM
2515 ;
2516 ;
2517 ; IF ALL MEMORIES ARE BEING TESTED, THE PASS NUMBER FROM
2518 ; THE PREVIOUS PASS HAS ALREADY BEEN PRINTED OUT (SEE 'MTMAIN').
2519 014430 EAST:
2520 014430 CALL HLTST ;WRITE EXIT MESSAGE
2521 014434 032767 000100 163366 BIT #ALLTST,BASE ;ALL TESTS
2522 014442 001025 BNE REST ;YES, PASS NUMBER ALREADY PRINTED
2523 ;
2524 ;
2525 ; PRINT NUMBER OF PASSES ALONE
2526 ;
2527 ;
2528 014444 PAST:
2529 014444 012700 003337* MOV #PRINT,R0 ;POINT TO PRINT LINE
2530 014450 012701 000116 MOV #78,R1 ;NUMBER OF CHAR POSITIONS
2531 014454 112720 000040 1$: MOVB #40,(R0)+ ;CLEAR THE LINE
2532 014460 005301 DEC R1
2533 014462 001374 BNE 1$
2534 ;
2535 014464 012705 003337* MOV #PRINT,R5 ;POINT TO PRINT LINE
2536 014470 012700 005631* MOV #PMSG2,R0 ;POINT TO NUMBER OF PASSES

```



```

2537 014474 012701 000014
2538 014500 112025
2539 014502 005301
2540 014504 001375
2541
2542 014506
2543 014512
2544
2545
2546 014516
2547
2548 014532 122767 000120 163250
2549 014540 001414
2550 014542 122767 000103 163240
2551 014550 001410
2552 014552 122767 000124 163230
2553 014560 001404
2554 014562 122767 000127 163220
2555 014570 001003
2556
2557 014572
2558
2559 014600 012701 003337
2560 014604 012700 000116
2561 014610 112721 000040
2562 014614 005300
2563 014616 001374
2564 014620 112767 000015 166511
2565 014626 112767 000012 166504
2566 014634
2567
2568 014640 016737 163142 000274
2569 014646

```

```

2$: MOV #PM2LN,R1 ;LENGTH OF STRING
MOV# (R0)+,(R5)+ ;MOVE STRING TO PRINT LINE
DEC R1
BNE 2$

;
CALL PASSC ;CONVERT NUMBER OF PASSES
CALL CONSOL ;WRITE MESSAGE

;
REST: RESTOR R0,R1,R2,R3,R4,R5

;
CMPB #'P,ASTWRD ;PRINT # PASSES AND CONTINUE
BEQ 10$
CMPB #'C,ASTWRD ;PRINT TEST PATTERN AND CONTINUE
BEQ 10$
CMPB #'T,ASTWRD ;PRINT TEST NUMBER
BEQ 10$
CMPB #'W,ASTWRD ;PRINT WHERE AND CONTINUE
BNE 1$

;
10$: ASTX$S

;
1$: MOV #PRINT,R1 ;POINT TO PRINT LINE
MOV #78,R0 ;NUMBER OF CHARS
2$: MOV# #40,(R1)+ ;CLEAR PRINT LINE
DEC R0
BNE 2$
MOV# #15,PRINT ;WRITE OUT ONE CR+LF
MOV# #12,PRINT+1
CALL CONSOL

;
MOV OLDVEC,@#274 ;RESTORE ORIGINAL VECTOR CONTENTS
EXIT$S

```

```

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 ;
2601 ;
2602 ;
2603 ;
2604 ;
2605 ;
2606 ;
2607 ;
2608 ;
2609 ;
2610 ;
2611 ;
2612 014654 ;
2613 014654 ;
2614 014660 ;
2615 014664 103472 ;
2616 014666 ;
2617 014672 103003 ;
2618 014674 ;
2619 014700 000765 ;
2620 ;
2621 ;
2622 ;
2623 014702 026766 163114 000002 1$:
2624 014710 103003 ;
2625 014712 ;
2626 014716 000756 ;
2627 ;

```

```

SUBRTN FOR MEMORY LIMITS.

INPUT:
2(SP) ABSOLUTE LOWER LIMITS.
4(SP) ABSOLUTE UPPER LIMITS.
6(SP) ADDRESS OF PROMPT ROUTINE.

OUTPUT:
(SP) CURRENT WORKING LOWER LIMITS.
2(SP) CURRENT WORKING UPPER LIMITS.

WORK FIELDS USED:
LOWER.
UPPER.

PROMPT FOR LIMITS. A <CR> RESPONSE MEANS TO TAKE
THE ABSOLUTE LOWER AND UPPER LIMITS AND RETURN
THEM ON THE STACK. OTHERWISE IF THE RESPONSE IS IN
THE FORM:
>0000 000A.

THIS ROUTINE CONVERTS THE FIRST VALUE AND COMPARES IT
AGAINST THE MEMORY'S ABSOLUTE LOWER LIMITS AT 2(SP). IF
THE NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A
TEMPORARY WORK FIELD. THE ROUTINE THEN CHECKS THE COMMAND
LINE FOR THE UPPER LIMITS, CONVERTS THEM, AND COMPARES
THEM AGAINST THE MEMORY'S ABSOLUTE UPPER LIMITS. IF THE
NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A TEMPORARY
WORK AREA. THE ROUTINE THEN COMPARES THE NEW UPPER LIMITS
WITH THE NEW LOWER LIMITS. IF THE NEW UPPER LIMITS ARE
EQUAL TO OR GREATER THAN THE NEW LOWER LIMITS, BOTH NEW
VALUES ARE PLACED ON THE STACK. THE ROUTINE THAT CALLED
LIMITS WILL TAKE THESE VALUES OFF THE STACK AND PLACE
THEM IN THE 'CURRENT LIMITS TABLE'. DURING THE MEMORY
TEST CYCLE, THE LIMITS FROM THIS TABLE ARE MADE AVAILABLE
TO THE MEMORY TEST ROUTINES.

LIMITS:
CALL @6(SP) ;PROMPT FOR LIMITS.
CALL FIND ;FIRST FIND A NUMBER
BCS LIMX2. ;NO OVERRIDES, LEAVE LIMITS ALONE.
CALL PACK ;CONVERT LOWER LIMIT TO BINARY
BCC 1$ ;VALUE OK, CONTINUE
CALL ERR4
BR LIMITS. ;TRY AGAIN

CHECK LOWER LIMITS.
CMP BINWD,2(SP) ;COMPARE LOWER LIMITS.
BHIS 2$ ;OK, CONTINUE
CALL ERR4
BR LIMITS. ;TRY AGAIN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

2628.          ;      ALSO CHECK NEW LOWER LIMITS AGAINST ABSOLUTE UPPER
2629          ;      LIMITS ON STACK. AN ERROR HERE WOULD SHOW UP BELOW
2630          ;      BUT IT IS MORE CORRECT TO REPORT AN ERROR IN LOWER
2631          ;      LIMITS IF THE NEW LOWER LIMITS ARE HIGHER THAN THE
2632          ;      ABSOLUTE UPPER LIMITS.
2633          ;
2634 014720 026766 163076 000004 2$:  CMP.   BINWD,4(SP)      ; IS NEW LOW LIMIT GT UPPER LIMIT
2635 014726 101403          BLOS.  20$          ; NO, NEW LOWER LIMIT IS OK
2636 014730          CALL.  ERR4
2637 014734 000747          BR     LIMITS
2638          ;
2639          ;      SAVE NEW LOWER LIMITS. FIND NEW UPPER LIMITS IN COMMAND
2640          ;      LINE.
2641          ;
2642 014736 016767 163060 163060 20$:  MOV.   BINWD,LOWER    ; MOVE IN NEW LOW LIMITS
2643 014744          CALL.  FIND
2644 014750 103003          BCC.   3$          ; FIND UPPER LIMITS IN COMMAND LINE
2645 014752          CALL.  ERR5
2646 014756 000736          BR     LIMITS
2647          ;
2648 014760          ; 3$:  CALL.  PACK
2649 014764 103003          BCC.   4$          ; CONVERT UPPER LIMITS
2650 014766          CALL.  ERR5
2651 014772 000730          BR     LIMITS
2652          ;
2653          ;      CHECK NEW UPPER LIMITS.
2654          ;
2655 014774 026766 163022 000004 4$:  CMP.   BINWD,4(SP)      ; COMPARE UPPER LIMITS
2656 015002 101403          BLOS.  5$          ; OK, CONTINUE
2657 015004          CALL.  ERR5
2658 015010 000721          BR     LIMITS
2659          ;
2660          ;      COMPARE NEW LOWER LIMITS WITH NEW UPPER LIMITS.
2661          ;
2662 015012 016767 163004 163006 5$:  MOV.   BINWD,UPPER    ; PLACE NEW UPPER LIMIT ON STACK
2663 015020 026767 163000 163000          CMP.   LOWER,UPPER    ; IS UPPER LIMIT GT LOWER LIMIT
2664 015026 101403          BLOS.  LIMX
2665 015030          CALL.  ERR5
2666 015034 000707          BR     LIMITS
2667          ;
2668 015036 016766 162762 000002 LIMX: MOV.   LOWER,2(SP)      ; PUT NEW LOWER LIMITS ON STACK
2669 015044 016766 162756 000004          MOV.   UPPER,4(SP)     ; PUT NEW UPPER LIMITS ON STACK
2670 015052 000207          LIMX2: RTS.   PC
    
```

```

2672. ;
2673. ;
2674. ; SCAN: A TABLE FOR A VALID COMMAND/MNEMONIC.
2675. ;
2676. ; INPUT:
2677. ; R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
2678. ; R1 -> CHAR STRING IN GCML COMMAND LINE.
2679. ; R2 -> TOP OF COMMAND TABLE.
2680. ;
2681. ; OUTPUT:
2682. ; R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
2683. ; R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
2684. ; R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
2685. ;
2686. ;
2687. 015054 SCAN:
2688. 015054 010346 MOV R3, -(SP) ;SAVE R3
2689. 015056 010046 MOV R0, -(SP) ;SAVE # OF ENTRIES
2690. 015060 010146 MOV R1, -(SP) ;SAVE POINTER TO BEGINNING OF STRING
2691. ;
2692. 015062 011601 FNOUT1: MOV (SP), R1 ;POINT TO NON-BLANK IN COMMAND LINE
2693. 015064 012703 000002 MOV #2, R3 ;NUMBER OF CHARS IN NON-BLANK FIELD
2694. 015070 122122 FNIN1: CMPB (R1), (R2)+ ;DOES COMMAND LINE MATCH TABLE ENTRY
2695. 015072 001003 BNE FNOUT2 ;NO, TRY NEXT TABLE ENTRY
2696. 015074 005303 DEC R3 ;SUB FROM LOOP COUNT
2697. 015076 001374 BNE FNIN1
2698. 015100 000410 BR FNMTCH ;COMMAND FOUND IN TABLE
2699. 015102 060302 FNOUT2: ADD R3, R2 ;ADD # UNCOMPARED CHARS TO POINTER
2700. 015104 005202 INC R2 ;THEN ADJUST TO NEXT TABLE ENTRY
2701. 015106 005300 DEC R0 ;SUB FROM OUTER LOOP COUNT
2702. 015110 001364 BNE FNOUT1 ;TRY AGAIN
2703. 015112 012601 MOV (SP)+, R1 ;RESTORE POINTER TO COMMAND LINE
2704. ; MOV (SP)+, R0 ;RESTORE R0
2705. 015114 012603 MOV (SP)+, R3 ;RESTORE R3
2706. 015116 000261 SEC ;COMMAND NOT IN TABLE
2707. 015120 RETURN
2708. ;
2709. 015122 010201 FNMTCH: MOV R2, R1 ;POINT R1 AT RTH ADDR IN TABLE
2710. 015124 062706 000002 ADD #2, SP ;POINT TO INCOMING R0 ON STACK
2711. 015130 012602 MOV (SP)+, R2 ;GET TOTAL # TABLE ENTRIES
2712. 015132 160002 SUB R0, R2 ;GET POSITION OF MATCHED ENTRY
2713. 015134 010200 MOV R2, R0 ;PUT IN R0 FOR RETURN
2714. 015136 012603 MOV (SP)+, R3 ;RESTORE R3
2715. 015140 000241 CLC
2716. 015142 RETURN

```

```

2718 ;
2719 ;
2720 ;
2721 ; FIND THE NEXT NON-BLANK IN THE COMMAND BUFFER,
2722 ; THEN FIND THE LENGTH OF THE STRING THAT STARTS WITH THAT CHARACTER.
2723 ;
2724 ; INPUT:
2725 ; GCMLN - NUMBER OF UNPROCESSED BYTES IN COMMAND LINE.
2726 ; GCMPT - ADDR OF NEXT UNPROCESSED POSITION IN COMMAND LINE.
2727 ;
2728 ; OUTPUT:
2729 ; R1 -> STRING, R0 = LENGTH OF STRING.
2730 ; GCMLN, GCMPT UPDATED FOR NEXT ENTRY INTO THIS ROUTINE.
2731 ;
2732 ; THIS ROUTINE IS DESIGNED TO BE ENTERED A NUMBER OF TIMES
2733 ; IN THE PARSING OF A COMMAND LINE. THE FIELDS GCMLN AND
2734 ; GCMPT ARE REFRESHED WHEN A NEW COMMAND LINE IS READ
2735 ; (SEE THE MESSAGE PRINTING/PROMPTING CODE).
2736 ;
2737 ;
2738 ; FIND:
2739 ;
2740 ; MOV R2, -(SP) ; SAVE R2.
2741 ; MOV GCMLN, R1 ; # BYTES REMAINING IN COMMAND BUFFER.
2742 ; BEQ FSECX ; THERE ARE NONE.
2743 ; MOV GCMPT, R2 ; LOAD CURRENT POINTER.
2744 ; CMPB #40, (R2) ; LOOK FOR A BLANK.
2745 ; BEQ 10$ ; OK, BUMP TO NEXT CHAR.
2746 ; CMPB #' ', (R2) ; COMMA IN COMMAND LINE.
2747 ; BNE 2$ ; TREAT COMMA AS BLANK.
2748 ; INC R2 ; BUMP POINTER.
2749 ; DEC R1 ; SUB FROM REMAINING LENGTH.
2750 ; BNE 1$ ;
2751 ; BR FSECX ; NO NON-BLANK FOUND.
2752 ;
2753 ; 2$: MOV R2, -(SP) ; TEMP SAVE POINTER TO BEGINNING OF STRING.
2754 ; CLR R0 ; CLEAR CHAR COUNT.
2755 ; CMPB #40, (R2) ; LOOK FOR A BLANK.
2756 ; BEQ 4$ ; FOUND END OF STRING.
2757 ; CMPB #' ', (R2) ; TREAT COMMAS AS BLANKS.
2758 ; BEQ 4$ ;
2759 ; INC R2 ; BUMP POINTER.
2760 ; INC R0 ; BUMP CHAR COUNT.
2761 ; DEC R1 ; SUB FROM BYTES REMAINING.
2762 ; BNE 3$ ;
2763 ;
2764 ; 3$: MOV R2, GCMPT ; SAVE POINTER FOR NEXT TIME.
2765 ; MOV R1, GCMLN ; SAVE BYTES REMAINING FOR NEXT TIME.
2766 ; MOV (SP)+, R1 ; POINTER TO BEGINNING OF STRING.
2767 ; MOV (SP)+, R2 ; RESTORE R2.
2768 ; CLC ;
2769 ; RETURN ;
2770 ;
2771 ; 4$: MOV (SP)+, R2 ; RESTORE R2.
2772 ; SEC ;
2773 ; RETURN ;

```

```

2773 ; CONVERT AN ASCII HEX VALUE FROM THE COMMAND LINE INTO BINARY.
2774 ; LEGAL STRINGS CONTAIN FROM 1 TO 4 CHARACTERS.
2775 ;
2776 ; INPUT:
2777 ; R0 = NUMBER OF CHARACTERS IN ASCII STRING.
2778 ; R1 -> STRING
2779 ;
2780 ; OUTPUT:
2781 ; THE FIELD 'BINWD' CONTAINS THE CONVERTED VALUE.
2782 ;
2783 ;
2784 015262 ; PACK:
2785 015262 005067 162534 CLR BINWD ; CLEAR DESTINATION.
2786 015266 005046 CLR -(SP) ; CLEAR FOR COND CODE INDICATOR
2787 015270 SAVE R0,R1,R2,R3,R4,R5
2788 ;
2789 ; DETERMINE THE CONVERSION FACTOR (POWER OF 16) FOR
2790 ; THE LEFTMOST ASCII CHARACTER.
2791 ;
2792 015304 022700 000004 CMP #4,R0 ; UPPER LIMIT FOR HEX DIGITS.
2793 015310 002455 BLT PSECX ; ERROR EXIT.
2794 015312 010002 MOV R0,R2 ; NUMBER OF CHARS CONTROLS LOOP
2795 015314 022700 000004 CMP #4,R0 ; 4 CHARS.?
2796 015320 001003 BNE 1$
2797 015322 012700 010000 MOV #4096,,R0 ; HEX CONVERSION FACTOR FOR HIGH ORDER CHAR.
2798 015326 000416 BR 4$ ; ENTER LOOP.
2799 015330 022700 000003 1$: CMP #3,R0 ; 3 CHARS.?
2800 015334 001003 BNE 2$
2801 015336 012700 000400 MOV #256,,R0 ; CONVERSION FACTOR FOR HIGH ORDER CHAR.
2802 015342 000410 BR 4$ ; ENTER LOOP.
2803 015344 022700 000002 2$: CMP #2,R0 ; 2 CHARS.?
2804 015350 001003 BNE 3$
2805 015352 012700 000020 MOV #16,,R0 ; CONVERSION FACTOR FOR HIGH ORDER CHAR.
2806 015356 000402 BR 4$
2807 015360 012700 000001 3$: MOV #1,R0 ; 1 CHAR.
2808 ;
2809 ; MULTIPLY EACH CHARACTER'S VALUE BY ITS CONVERSION
2810 ; FACTOR. THE CONVERSION FACTOR IS REDUCED BY A POWER
2811 ; OF 16 AS THE ASCII STRING IS SCANNED FROM LEFT TO RIGHT.
2812 ;
2813 015364 010105 4$: MOV R1,R5 ; MOVE INPUT POINTER TO R5
2814 015366 112503 HLOOP: MOVB (R5)+,R3 ; GET ASCII VALUE INTO A REG.
2815 015370 012704 000432' MOV #TRTBL,R4 ; POINT TO TRANSLATE TABLE.
2816 015374 060304 ADD R3,R4 ; ADD VALUE OF CHARACTER.
2817 015376 111401 MOVB (R4),R1 ; MOVE BINARY VALUE TO A REG.
2818 015400 022704 000512' CMP #TRTBL+60,R4 ; WAS INPUT CHAR ZERO
2819 015404 001402 BEQ 1$ ; YES, THIS IS OK.
2820 015406 105701 TSTB R1 ; WAS TABLE POSITION EMPTY.
2821 015410 001415 BECX ; YES, TRANSLATION UNSUCCESSFUL.
2822 015412 010046 1$: MOV R0, -(SP) ; SAVE FOR DIVISION LATER.
2823 015414 CALL $MUL ; MULT BY 16 TO SOME POWER.
2824 015420 060167 162376 ADD R1,BINWD ; ACCUM CONVERTED VALUE.
2825 015424 012600 MOV (SP)+,R0 ; RELOAD FACTOR.
2826 015426 012701 000020 MOV #16,,R1 ; LOAD DIVISOR.
2827 015432 CALL $DIV ; REDUCE FACTOR.
2828 015436 005302 DEC R2 ; SUB FROM LOOP COUNT
2829 015440 001352 BNE HLOOP

```

```

2830 015442 000403          BR      PCLCX          :AND-EXIT-
2831                      ;
2832 015444 012766 177777 000014 PSECX: MOV      #-1,12,(SP)
2833 015452          PCLCX: RESTOR  R0,R1,R2,R3,R4,R5  :INDICATE-ERROR-
2834                      ;
2835 015466 005726          TST      (SP)+      :TEST-CC-INDICATOR-
2836 015470 002402          BLT      1$         :ERROR-
2837 015472 000241          CLC
2838 015474 000401          BR      PACKX
2839 015476 000261          1$: SEC
2840 015500          PACKX: RETURN

```

```

2842. ;
2843. ;
2844. ; READ-A-RECORD.(BLOCK)
2845. ;
2846. ; FILE-NAME-BLOCK-PRE-INITIALIZED.
2847. ;
2848. ; OUTPUT:
2849. ; C-BIT-CLEAR. - GOOD READ.
2850. ; C-BIT-SET. - ERROR-ON-READ.
2851. ;
2852. ;
2853. 015502. GET:
2854. 015502. READ$ #INFDB...#VIRT.#EFN.1,#STAT.
2855. 015550 103005 BCC. 1$.
2856. 015552. CALL. ERR5
2857. 015556. CALL. ERNAME. ;TELL-WHICH-FILE-WAS-IN-ERROR.
2858. 015562. 000421 BR GETSX.
2859. ;
2860. 015564. 1$: WTSE$$ #EFN.1
2861. ;
2862. 015576. GLEF$$ #EFN.1
2863. 015610 105767 162176 TSTB. STAT
2864. 015614 003006 BGT. GETCX. ;GOOD-COMPLETION.
2865. 015616. CALL. ERR5
2866. 015622. CALL. ERNAME. ;TELL-WHICH-FILE-WAS-IN-ERROR.
2867. ;
2868. 015626 000261 GETSX: SEC.
2869. 015630 000403 BR GETX
2870. 015632. 005267 162202 GETCX: INC. VIRT+2. ;INC-BLOCK-COUNTER.
2871. 015636 000241 CLC.
2872. 015640 GETX: RETURN.

```



```
2874 ; CONVERT NAME OF FILE IN ERROR FROM RAD50 TO ASCII AND PRINT.
2875 ;
2876 015642 ERNAME:
2877 015642 SAVE R0,R1
2878 ;
2879 015646 012700 003337 MOV #PRINT,R0 ;R0 -> PRINT LINE
2880 015652 016701 170460 MOV INDNB+N.FNAM,R1 ;R1 = RAD50 WORD
2881 015656 CALL #CSTA ; CONVERT FIRST WORD
2882 015662 016701 170452 MOV INDNB+N.FNAM+2,R1 ; SECOND WORD
2883 015666 CALL #CSTA
2884 ;
2885 015672 CALL CONSOL
2886 015676 RESTOR R0,R1
2887 015702 RETURN
```

```

2889 ;
2890 ;
2891 ;
2892 ; CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
2893 ; R1 = WORD TO BE CONVERTED.
2894 ; R5 -> PRINT LINE.
2895 ;
2896 UNPK:
2897 015704 SAVE R0,R1,R2.
2898 ;
2899 015712 062705 000004 ADD #4,R5 ;DO LAST CHAR FIRST.
2900 015716 012702 000004 MOV #4,R2. ;NUMBER OF HEX DIGITS FOR A WORD.
2901 015722 010100 MOV R1,R0 ;SUBRTN EXPECTS DIVIDEND IN R0
2902 015724 012701 000020 1$: MOV #16,R1 ;LOAD DIVIDOR.
2903 015730 CALL $DIV
2904 015734 012703 000632 MOV #TRTBL2,R3 ;POINT TO TRANSLATE TABLE.
2905 015740 060103 ADD R1,R3 ;ADD 4 BIT VALUE.
2906 015742 111345 MOVB (R3),-(R5) ;MOVE CHAR TO PRINT LINE.
2907 015744 005302 DEC R2 ;DEC INNER LOOP COUNT.
2908 015746 001366 BNE 1$
2909 015750 062705 000005 ADD #5,R5 ;BUMP PRINT LINE POINTER.
2910 ;
2911 UNPKX:
2912 015754 RESTOR R0,R1,R2.
2913 015762 RETURN

```

```

2915 ;
2916 ;
2917 ;
2918 ; CONVERT NUMBER OF PASSES TO ASCII DECIMAL
2919 ; INCLUDE DOUBLE WORD CONVERSION (BINARY TO ASCII DECIMAL)
2920 ;
2921 ; INPUT: R5 -> PRINT LINE
2922 ; FIELD 'PASS' = LOW ORDER PASS COUNT
2923 ; FIELD 'PASSH' = HIGH ORDER PASS COUNT
2924 ;
2925 ; OUTPUT: R5 -> NEXT PRINT LINE POSITION AFTER CONVERTED VALUE
2926 ;
2927 015764 ; PASSC:
2928 015764 005767 162220 ; TST PASSH ; IS PASS NUMBER A DOUBLE WORD
2929 015770 001011 ; BNE DOUBLE ; YES, CONVERT DOUBLE WORD
2930 015772 010500 ; MOV R5,R0 ; PREPARE TO CALL CONVERSION RTN
2931 015774 016701 162212 ; MOV PASS,R1 ; NUMBER TO BE CONVERTED
2932 016000 012702 000001 ; MOV #1,R2 ; SUPPLY LEADING ZEROS
2933 016004 ; CALL #CDSG ;
2934 016010 010005 ; MOV R0,R5 ; RESTORE PRINT LINE POINTER
2935 016012 000453 ; BR PASSX ; EXIT
2936 ;
2937 016014 016701 162170 ; DOUBLE: MOV PASSH,R1 ; LOAD HIGH WORD
2938 016020 016702 162166 ; MOV PASS,R2 ; LOAD LOW WORD
2939 016024 010346 ; MOV R3,-(SP)
2940 016026 010446 ; MOV R4,-(SP)
2941 ;
2942 016030 012703 000012 ; MOV #10,R3 ; R3 = LOOP COUNTER
2943 016034 012704 016144 ; MOV #ASCNST,R4 ; R4->CONVERSION STORAGE AREA
2944 016040 112724 000060 1#: MOVVB #60,(R4)+ ; CLEAR STORAGE AREA TO 0'S
2945 016044 005303 ; DEC R3
2946 016046 001374 ; BNE 1#
2947 ;
2948 016050 012704 016156 ; MOV #ASCNST+12,R4 ; R4->LAST BYTE OF STORAGE
2949 016054 012703 000012 ; MOV #12,R3
2950 016060 012700 000012 2#: MOV #10,R0
2951 016064 ; CALL #DDIV ;
2952 016070 062700 000060 ; ADD #60,R0 ; ASCII NO. (REMAINDER+60 OCTAL)
2953 016074 110044 ; MOVVB R0,-(R4)
2954 016076 005303 ; DEC R3
2955 016100 001367 ; BNE 2#
2956 ;
2957 016102 012703 000012 ; MOV #12,R3
2958 016106 122724 000060 3#: CMPB #60,(R4)+ ; REMOVE LEADING ZEROS
2959 016112 001003 ; BNE 4#
2960 016114 005303 ; DEC R3
2961 016116 001373 ; BNE 3#
2962 016120 005203 ; INC R3 ; IF ALL 0'S, THE LAST ONE IS OK
2963 ;
2964 016122 005304 4#: DEC R4
2965 016124 010401 ; MOV R4,R1 ; R1-> RESULT
2966 016126 010302 ; MOV R3,R2 ; R2 = LENGTH OF RESULT
2967 ;
2968 016130 112125 5#: MOVVB (R1)+(R5)+ ; MOVE CONVERTED VALUE TO PRINT LINE
2969 016132 005302 ; DEC R2
2970 016134 001375 ; BNE 5#
2971 ;

```

2972 016136 012604
2973 016140 012603
2974 016142
2975
2976
2977 016144

MOV: (SP)+,R4
MOV: (SP)+,R3
PASSX: RETURN: ; RETURN.
; ;
ASCNST: .BLKW 5 ; CONVERSION STORAGE AREA.

```

2979 ;
2980 ;
2981 ; WRITE A PRINT LINE TO TT0
2982 ;
2983 ;
2984 016156 CONSOL:
2985 016156 012700 000120 MOV #00,R0 ;PRINT BUFFER BYTE COUNT
2986 016162 012701 003455 MOV #PRINT+70,R1 ;POINT PAST END OF BUFFER
2987 016166 122741 000040 1$: CMPB #40,-(R1) ;LOOK FOR A NON-BLANK
2988 016172 001003 BNE 2$ ;OK, WRITE LINE
2989 016174 005300 DEC R0 ;DEC CHAR COUNT
2990 016176 001373 BNE 1$
2991 016200 000436 BR ABEND2 ;NO NON-BLANKS?
2992 ;
2993 016202 2$: QIOW$ #ID,WVB,#LUN,TT,#EFN,1, #STAT,<#PRINT-2,R0>,ABEND2
2994 ;
2995 ;
2996 016260 012701 003337 MOV #PRINT,R1 ;POINT TO STRING
2997 016264 112721 000040 4$: MOVB #40,(R1)+ ;CLEAR LINE TO BLANKS
2998 016270 005300 DEC R0 ;DEC LOOP COUNT
2999 016272 001374 BNE 4$
3000 016274 RETURN
3001 ;
3002 016276 ABEND2: ABRT$ #MYSELF

```



```

3061 016572 012701 005630*
3062 016576 105741
3063 016600 001376
3064 016602 005302
3065 016604 001374
3066 016606 010100
3067 016610 105741
3068 016612 001376
3069 016614 005201
3070 016616 100100
3071 016620 012602
3072
3073 016622
3074
3075 016676
3076 016710 105767 161076
3077 016714 003433
3078
3079
3080
3081 016716 022767 000025 161072
3082 016724 002424
3083
3084 016726 012700 000062*
3085 016732 012701 000051
3086 016736 005020
3087 016740 005301
3088 016742 001375
3089
3090 016744
3091 016760 103411
3092 016762 016067 000146 161214
3093 016770 012767 000062* 161210
3094 016776 005067 161014
3095 017002
3096
3097 017004
3098 006366*

1$: MOV #ASCIZ,R1 ;POINT TO END OF MESSAGE TABLE
TSTB -(R1) ;LOOK FOR END OF MESSAGE
BNE 1$
DEC R2
BNE 1$ ;LOOP COUNT
MOV R1,R0 ;BACK UP ANOTHER MESSAGE
2$: TSTB -(R1) ;SAVE POINTER TO END OF MESSAGE
BNE 2$ ;BACK UP TO BEGINNING OF MESSAGE
INC R1 ;BUMP TO FIRST CHAR OF MESSAGE
SUB R1,R0 ;R0 NOW = MESSAGE LENGTH
MOV (SP)+,R2 ;RESTORE R2

;
; QIOW$S #IO,WVB,#LUN,TT,#EFN,1,,#STAT,<R1,R0>,ABEND
;
; CLEF$S #EFN,1
; TSTB STAT ;GOOD RETURN
; BLE ABEND ;NO
;
; ISSUE GCML
;
; CMP #NEST,ERWORD ;PROMPT WITH MESSAGE
; BLT TTX ;NO, JUST EXIT
;
; MOV #GCMBUF,R0 ;POINT TO GCML BUFFER
; MOV #41,,R1 ;NUMBER OF WORDS
3$: CLR (R0)+ ;CLEAR BUFFER
DEC R1
BNE 3$
;
; GCML$ #GCMBLK
; BCS ABEND
; MOV G,CMLD(R0),GCMLEN ;SAVE LENGTH
; MOV #GCMBUF,GCMPNT ;INITIALIZE COMMAND BUFFER POINTER
TTX: CLR ERWORD ;CLEAR ERROR NUMBER INDICATOR
RETURN ;AND RETURN
;
; ABEND: ABRT$S #MYSELF
; .END .START

```

ABEND 017004R	BYTE26 = 000032	BYTE78 = 000116	DOUBLE 016014R	F.BKEF = 000050
ABEND2 016276R	BYTE27 = 000033	BYTE79 = 000117	EAST 014430R	F.BKP1 = 000051
ALL 006550R	BYTE28 = 000034	BYTE80 = 000120	EFN.1 = 000001	F.BKST = 000024
ALLMEM 000657R	BYTE29 = 000035	BYTE81 = 000121	EFN.3 = 000003 G	F.BKVB = 000064
ALLREG 000652R	BYTE3 = 000003	BYTE82 = 000122	EMSG 002722R	F.CHR = 000075
ALLSEL 016560R	BYTE30 = 000036	BYTE83 = 000123	ENBR = 010000	F.CNTG = 000034
ALLTST = 000100	BYTE31 = 000037	BYTE84 = 000124	ENDLIM 007250R	F.DFNB = 000046
ALUCKE = 040000	BYTE32 = 000040	BYTE85 = 000125	ENDLN = 000014	F.DSPT = 000044
ALUOE = 004000	BYTE33 = 000041	BYTE86 = 000126	ENDOF 005645R	F.DVNM = 000134
AMSG 002711R	BYTE34 = 000042	BYTE87 = 000127	ENDTST 016330R	F.EFBK = 000010
APLACE 000032R	BYTE35 = 000043	BYTE88 = 000130	ERLIM 000020R	F.EFN = 000050
ASCIZ 005630R	BYTE36 = 000044	BYTE89 = 000131	ERNAME 015642R	F.EQBB = 000032
ASCHST 016144R	BYTE37 = 000045	BYTE90 = 000132	EROPT 016440R	F.ERR = 000052
AST 013536R	BYTE38 = 000046	BYTE91 = 000133	ERPRMT 010474R	F.FACC = 000043
ASTWRD 000010R	BYTE39 = 000047	BYTE92 = 000134	ERRADD 002646RG	F.FBY = 000014
ASWRK 002672R	BYTE4 = 000004	BYTE93 = 000135	ERRCT 002650RG	F.FNAM = 000110
A01 = 010000	BYTE40 = 000050	BYTE94 = 000136	ERROR = 000400	F.FNB = 000102
BASE 000030RG	BYTE41 = 000051	BYTE95 = 000137	ERR10 016354R	F.FTYP = 000116
BEGTST 016344R	BYTE42 = 000052	BYTE96 = 000140	ERR2 016434R	F.FVER = 000120
BINWD 000022RG	BYTE43 = 000053	BYTE97 = 000141	ERR20 016430R	F.HIBK = 000004
BITVAL = 000000	BYTE44 = 000054	BYTE98 = 000142	ERR3 016424R	F.LUN = 000042
BIT0 = 000001	BYTE45 = 000055	BYTE99 = 000143	ERR4 016420R	F.MBC = 000054
BIT1 = 000002	BYTE46 = 000056	BYTVAL = 000144	ERR5 016414R	F.MBC1 = 000055
BIT10 = 002000	BYTE47 = 000057	CA = 000002	ERR50 016410R	F.MBFG = 000056
BIT11 = 004000	BYTE48 = 000060	CAST 014224R	ERR6 016404R	F.NRBD = 000024
BIT12 = 010000	BYTE49 = 000061	CBKALL = 001000	ERR60 016400R	F.NREC = 000030
BIT13 = 020000	BYTE5 = 000005	CBKCLL = 000400	ERR7 016374R	F.OVBS = 000030
BIT14 = 040000	BYTE50 = 000062	CD = 000040	ERR8 016370R	F.RACC = 000016
BIT15 = 100000	BYTE51 = 000063	CHECK 010176R	ERR80 016364R	F.RATT = 000001
BIT2 = 000004	BYTE52 = 000064	CHECK0 010314R	ERR9 016360R	F.RCNM = 000034
BIT3 = 000010	BYTE53 = 000065	CHECK0 010314R	ERWORD 000016R	F.RCTL = 000017
BIT4 = 000020	BYTE54 = 000066	CKDATA 002636RG	ERW1 002652RG	F.RSIZ = 000002
BIT5 = 000040	BYTE55 = 000067	CK2 002640RG	ERW2 002654RG	F.RTYP = 000000
BIT6 = 000100	BYTE56 = 000070	CK3 002642RG	ERW3 002656RG	F.SEQN = 000100
BIT7 = 000200	BYTE57 = 000071	CMILUN = 000002	ERW4 002660RG	F.SPDV = 000072
BIT8 = 000400	BYTE58 = 000072	CHOBRE = 100000	FACODE 000052RG	F.SPUN = 000074
BIT9 = 001000	BYTE59 = 000073	CONSOL 016156R	FC = 000200	F.STBK = 000036
BPTISR 012344R	BYTE6 = 000006	CPCCEN = 010000	FD.CCL = ***** GX	F.UHIT = 000136
BYTE0 = 000000	BYTE60 = 000074	CPREAD = 040000	FD.REC = ***** GX	F.URBD = 000020
BYTE1 = 000001	BYTE61 = 000075	CPURTE = 020000	FD.RUM = ***** GX	F.VBN = 000064
BYTE10 = 000012	BYTE62 = 000076	CS = 000020	FD.TTY = ***** GX	F.VBSZ = 000060
BYTE11 = 000013	BYTE63 = 000077	CSADRD = 000004	FIND 015144R	GCMBLK 005662R
BYTE12 = 000014	BYTE64 = 000100	CSEGC1 = 100000	FIRST = 001000 G	GCMBUF 000062R
BYTE13 = 000015	BYTE65 = 000101	CSOE = 000040	FNIN1 015070R	GCMLEN 000204R
BYTE14 = 000016	BYTE66 = 000102	CSR1 012420RG	FNMTCH 015122R	GCMFNT 000206R
BYTE15 = 000017	BYTE67 = 000103	CSWRTE = 000100	FNOUT1 015062R	GET 015502RG
BYTE16 = 000020	BYTE68 = 000104	CURLIM 002276R	FNOUT2 015102R	GETCX 015632R
BYTE17 = 000021	BYTE69 = 000105	DATA1 000034RG	FP = 000100	GETSX 015626R
BYTE18 = 000022	BYTE7 = 000007	DBR.RD = 000001	FSECX 015254R	GETX 015640R
BYTE19 = 000023	BYTE70 = 000106	DB#CPP = 001457	FVER = 000042RG	GE.BIF = 177775
BYTE2 = 000002	BYTE71 = 000107	DB#SPT = 000026	F.ACTL = 000076	GE.CLD = 000004
BYTE20 = 000024	BYTE72 = 000110	DB#TPC = 000023	F.ALDC = 000040	GE.COM = 000001
BYTE21 = 000025	BYTE73 = 000111	DISPGS = 100000	F.BBFS = 000062	GE.CDN = 000020
BYTE22 = 000026	BYTE74 = 000112	DMAAWR = 000005	F.BDB = 000070	GE.EOF = 177766
BYTE23 = 000027	BYTE75 = 000113	DMARRD = 000003	F.BGBC = 000057	GE.IND = 000002
BYTE24 = 000030	BYTE76 = 000114	DMARWR = 000004	F.BKDN = 000026	GE.LOR = 177777
BYTE25 = 000031	BYTE77 = 000115		F.BKDS = 000020	GE.LC = 000010

GE.MDE = 177774
GE.OPR = 177775
GE.RBG = 177730
GE.SIZ = 000040
G.CMLD = 000146
G.DPRM = 000160
G.ERR = 000140
G.ISIZ = 000020
G.LPDL = 000060
G.MODE = 000141
G.PSDS = 000142
G.SIZE = 000224
HALT = 000200
HLOOP = 015366R
HLTST = 016334R
HRL0 = ***** GX
ICA = ***** GX
IMA = ***** GX
INDNB = 000330RG
INFDB = 000170RG
INLUN = 000003
IO.ATA = ***** GX
IO.WVB = ***** GX
JMPHT = 011226R
LASTJ = 011514R
LBPP = ***** GX
LCOUNT = 000044RG
LCS = 002666RG
LIMITS = 014654R
LIMM1 = 007142R
LIMNUM = 000030
LIMREF = 001506R
LIMX = 015036R
LIMX2 = 015052R
LMM = 002662RG
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
LOOP = 000001
LOOPCT = 000214R
LOWER = 000024R
LPRMPT = 010346R
LPTST = 016444R
LSTACK = 011476R
LUN.TT = 000001
MA = 000001
MAREN1 = 000001
MAREN2 = 004000
MARLOD = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.DU = 000040
MAST = 014032R
MBKALL = 001000
MBKCLK = 000400
MD = 000002
MEMERR = 013070RG
MEMOFF = 011302R
MEMORY = 000040
MEMS = 007572R
MEMSEL = 016550R
MEMTOP = 007070R
MEMX = 013520R
MFTBL = 002771R
MM = 000001
MMADDR = 000100
MMLEFT = 000002
MMOE = 000004
MMSEL = 007022R
MMWRTE = 000010
MNOBRE = 100000
MOVE = 006626R
MREN1 = 000001
MREN2 = 020000
MSEL = 000216R
MSYN = 000040
MT = 000015
MTBL = 000716R
MTBLN = 000030
MTCNT = 000224R
MTERR = 007732R
MTJUMP = 012334R
MTMAIN = 011004R
MTPMT = 010102R
MTPNT = 000222R
MTREF = 001016R
MTRT = 007720R
MTRTN = 006704R
MTSET = 010620R
MTSL = 007670R
MTSUB = 001606R
MT10 = 000060R
MYSELF = 000000R
M2REL = 007756R
N = 000144
NEST = 000025
NESTOP = 016440R
NMEMS = 000014
NREGS = 000002
NXTCNT = 000230R
NXTPNT = 000226R
N.DID = 000024
N.DVNM = 000032
N.FID = 000000
N.FNAM = 000006
N.FTYP = 000014
N.FVER = 000016
N.NEXT = 000022
N.STAT = 000020
N.UNIT = 000034
OLDVEC = 000006RG
OUT1 = 016350R
PACK = 015262R
PACKX = 015500R
PAR\$\$\$ = 000027
PASS = 000212R
PASSC = 015764R
PASSH = 000210R
PASSX = 016142R
PAST = 014444R
PCLCX = 015452R
PLB = 000010
PLC = 000020
PLD = 000030
PLRWR = 000200
PLR.EN = 000200
PMPT10 = 010112R
PMPT3 = 007530R
PMSG = 002704R
PMSG2 = 005631R
PMTCD = 016520R
PMTCS = 016524R
PMTFC = 016510R
PMTFP = 016514R
PMTMD = 016540R
PMTMM = 016544R
PMTQL = 016530R
PMTQR = 016504R
PMTQW = 016534R
PMTQ0 = 016500R
PMTQ1 = 016474R
PMTQ2 = 016470R
PMT10 = 016450R
PMT3 = 016454R
PM2LN = 000014
PPCR = ***** GX
PREADD = 002644RG
PRINT = 003337RG
PSECX = 015444R
QBPAGE = 000054RG
QL = 000010
QR = 000400
QR\$CR1 = 176420
QR\$CR2 = 176422
QR\$LBR = 176424
QW = 000004
QXCODE = 000050RG
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FC = 000045
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IHB = 000003
Q\$IHRL = 000002
Q\$IMRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$MNC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$MSC = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 000000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
Q0 = 001000
Q1 = 002000
Q2 = 004000
REGERR = 012460RG
REGSEL = 016554R
REGSTR = 000020
REGTOP = 006750R
REGX = 013052R
REST = 014516R
RGJUMP = 011604R
RG.EN = 000200
RGQ.VA = 020000
RGREF = 000776R
RGSEL = 006704R
RGSUB = 001566R
RMSG = 002734R
RSEL = 000220R
RT = 000004
RTBL = 000706R
RTBLN = 000004
RTERR = 007370R
RTSL = 007340R
RT3 = 000056R
RT4ADD = 002362R
RZREL = 007414R
SCAN = 015054R
SELMT = 016460R
SELRT = 016464R
SEQ.CI = 000010
STADDR = 002366R
START = 006366R
STAT = 000012RG
STCA = ***** GX
STMA = ***** GX
STOP = 016340R
STRADD = 002356R
STRMEM = 000027
STRREG = 000005
STUFCD = ***** GX
STUFCS = ***** GX
STUFFA = ***** GX
STUFMD = ***** GX
STUFMM = ***** GX
STUFQB = ***** GX
STUFQR = ***** GX
STUFQX = ***** GX
S\$CLR = 000000
S\$LA = 000001
S\$OB = 000005
S\$OR = 000006
S\$OX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
S.BFHD = 000020
S.FATT = 000016
S.FDB = 000140
S.FNAM = 000006
S.FNB = 000036
S.FNBW = 000017
S.FNTY = 000004
S.FTYP = 000002
S.NFEN = 000020
TA = 012054R
TAP = 014246R
TB = 012072R
TC = 012216R
TCCD = ***** GX

TCCDU = ***** GX
TCCSD = ***** GX
TCCSU = ***** GX
TCDADD = 002526R
TCFAD = ***** GX
TCFAU = ***** GX
TCMDD = ***** GX
TCMDU = ***** GX
TCMMD = ***** GX
TCMMU = ***** GX
TCQBD = ***** GX
TCQBU = ***** GX
TCQRD = ***** GX
TCQRU = ***** GX
TCQXD = ***** GX
TCQXU = ***** GX
TCUADD = 002556R
TD = 012322R
TDADDR = 002606R
TDCS = ***** GX
TDM1 = ***** GX
TDNUL = 012332R
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$OAR = 176344
TD\$OTR = 176346
TD\$ORD = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDW = 176364
TEST10 = 000010
TEST3 = 000002
TEST6 = 000004
TMSG = 002677R
TROCT = 000232R
TRTBL = 000432R
TRTBL2 = 000632R
TSKTCB = 000004RG
TTX = 016776R
T\$AD = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTD = 000000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DRD = 000004
T\$EMEM = 010000
T\$FSAA = 000000
T\$FSAB = 000004
T\$FSAC = 000014
T\$FSB2 = 000010
T\$IB = 000026
T\$IBAR = 000024
T\$IBE = 020000
T\$IBF = 040000
T\$ICD = 000040
T\$MODE = 004000
T\$OB = 000036
T\$OBE = 004000
T\$OBF = 010000
T\$OBRA = 000034
T\$OBWA = 000032
T\$OUTA = 100000
T\$RBD0 = 000200
T\$RNB = 000040
T\$RSET = 040000
T\$SC = 000022
T\$SCLK = 020000
T\$SEG1 = 000000
T\$SEG2 = 000001
T\$SEG3 = 000002
T\$SO = 000001
T\$UBUS = 100000
T\$1CLK = 000400
T\$8BEN = 000020
T1 = 011610R
T1ADDR = 002416R
T1CD = ***** GX
T1CS = ***** GX
T1FA = ***** GX
T1MD = ***** GX
T1MM = ***** GX
T1OB = ***** GX
T1OR = ***** GX
T1OX = ***** GX
T1R = 011524R
T2 = 011620R
T2R = 011540R
T3 = 011634R
T3R = 011556R
T4 = 011652R
T4R = 011574R
T5 = 011670R
T6 = 011720R
T6ADDR = 002446R
T6CD = ***** GX
T6CHK = 010154R
T6CS = ***** GX
T6FA = ***** GX
T6MD = ***** GX
T6MM1 = ***** GX
T6QB = ***** GX
T6QR = ***** GX
T6QX = ***** GX
T7 = 011760R
T7ADDR = 002476R
T7CD = ***** GX
T7CS = ***** GX
T7FA = ***** GX
T7MD = ***** GX
T7MM = ***** GX
T7QB = ***** GX
T7QR = ***** GX
T7QX = ***** GX
T8 = 011770R
T9 = 012020R
UBD.IN = 000020
UHMSG = 002746R
UNPK = 015704R
UNPKX = 015754R
UPPER = 000026R
VIRT = 000036RG
WAST = 013636R
WCOUNT = 000046RG
WORD0 = 000000
WORD1 = 000002
WORD10 = 000024
WORD11 = 000026
WORD12 = 000030
WORD13 = 000032
WORD14 = 000034
WORD15 = 000036
WORD16 = 000040
WORD17 = 000042
WORD18 = 000044
WORD19 = 000046
WORD2 = 000004
WORD20 = 000050
WORD21 = 000052
WORD22 = 000054
WORD23 = 000056
WORD24 = 000060
WORD25 = 000062
WORD26 = 000064
WORD27 = 000066
WORD28 = 000070
WORD29 = 000072
WORD3 = 000006
WORD30 = 000074
WORD31 = 000076
WORD32 = 000100
WORD33 = 000102
WORD34 = 000104
WORD35 = 000106
WORD36 = 000110
WORD37 = 000112
WORD38 = 000114
WORD39 = 000116
WORD4 = 000010
WORD40 = 000120
WORD41 = 000122
WORD42 = 000124
WORD43 = 000126
WORD44 = 000130
WORD45 = 000132
WORD46 = 000134
WORD47 = 000136
WORD48 = 000140
WORD49 = 000142
WORD5 = 000012
WORD50 = 000144
WORD51 = 000146
WORD52 = 000150
WORD53 = 000152
WORD54 = 000154
WORD55 = 000156
WORD56 = 000160
WORD57 = 000162
WORD58 = 000164
WORD59 = 000166
WORD6 = 000014
WORD60 = 000170
WORD61 = 000172
WORD62 = 000174
WORD63 = 000176
WORD64 = 000200
WORD65 = 000202
WORD66 = 000204
WORD67 = 000206
WORD68 = 000210
WORD69 = 000212
WORD7 = 000016
WORD70 = 000214
WORD71 = 000216
WORD72 = 000220
WORD73 = 000222
WORD74 = 000224
WORD75 = 000226
WORD76 = 000230
WORD77 = 000232
WORD78 = 000234
WORD79 = 000236
WORD8 = 000020
WORD80 = 000240
WORD81 = 000242
WORD82 = 000244
WORD83 = 000246
WORD84 = 000250
WORD85 = 000252
WORD86 = 000254
WORD87 = 000256
WORD88 = 000260
WORD89 = 000262
WORD9 = 000022
WORD90 = 000264
WORD91 = 000266
WORD92 = 000270
WORD93 = 000272
WORD94 = 000274
WORD95 = 000276
WORD96 = 000300
WORD97 = 000302
WORD98 = 000304
WORD99 = 000306
WRDVAL = 000310
XTREAD = 001000
XTWRITE = 000400
\$CBDSG = ***** GX
\$CDTB = ***** GX
\$CEFI = ***** GX
\$CSTA = ***** GX
\$DDIV = ***** GX
\$DIV = ***** GX
\$DRDSE = ***** GX
\$MUL = ***** GX
\$TKTCB = ***** GX
\$\$\$ = 000042R
\$\$\$ARG = 000002
\$\$\$T1 = 000067
\$\$\$T2 = 000027
\$.FINIT = ***** G
\$.FSRCB = ***** G
\$.GCM1 = ***** G
\$.READ = ***** G
...PC1 = 006170R
...PC2 = 006344R
...PC3 = 006170R
...TFF = 000020

. ABS. 000000 000
017036 001
\$\$FSR1 001020 002
ERRORS DETECTED: 0

QMT.....MACRO M1110 27-MAR-80 15:19 PAGE 35-5
SYMBOL TABLE:

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

VIRTUAL MEMORY USED: 10016 WORDS (40 PAGES)
DYNAMIC MEMORY: 11252 WORDS (43 PAGES)
ELAPSED TIME: 00:02:31
QMT, QMT / - SP = [20, 1] IM [20, 1] QMT

```

1
2 000000 .TITLE-MMTSUB-...
3 .PSECT-MMTSUB
4 ;
5 ; MEMORY TEST DIAGNOSTICS
6 ; MATCH REPORT PROCESSOR MICROPROGRAM MEMORY
7 ;
8 ;
9 ; WRITE LEFT HALF OF MRP MICROPGM MEMORY
10 ;
11 ;
12 000000 WRITEL::
13 000000 012746 000012 MOV #MMWRTEN+MMLEFT>,-(SP)
14 000004 CALL MRPCRA ;DIRECT CONTROL WORD TO MRP
15 000010 016746 000000G MOV CKDATA,-(SP) ;GET READY TO MOVE DATA TO MRP
16 000014 CALL LBMSC ;DO IT
17 000020 005046 CLR -(SP) ;CLEAR MRP CR NS
18 000022 CALL MRPCR
19 000025 RETURN
20 ;
21 ;
22 ; WRITE RIGHT HALF OF MRP MICROPGM MEMORY
23 ;
24 ;
25 000030 WRITER::
26 000030 012746 000010 MOV #MMWRTEN,-(SP)
27 000034 CALL MRPCRA ;DIRECT CONTROL WORD TO MRP
28 000040 016746 000000G MOV CKDATA,-(SP) ;GET READY TO MOVE DATA TO MRP
29 000044 CALL LBMSC ;DO IT
30 000050 005046 CLR -(SP) ;CLEAR MRP CR NS
31 000052 CALL MRPCR
32 000055 RETURN
33 ;
34 ;
35 ; COMPARE LEFT HALF OF MRP MICROPGM MEMORY
36 ;
37 ;
38 000060 CMPL::
39 000060 CALL UNLL ;READ A LEFT HALF WORD
40 000064 026767 000000G 000000G CMP CKDATA,ERN1 ;SAME AS WORD WRITTEN
41 000072 001421 BEQ 1$ ;YES
42 000074 112767 000114 000052G MOVB #1,PRINT+42 ;INDICATE FAILURE ON LEFT HALF WORD
43 000102 016746 000000G MOV PREADD,-(SP) ;SET ADDRESS FOR SEQUENCER
44 000106 CALL SEQMM
45 000112 CALL UNLR ;READ RIGHT HALF WORD
46 000116 016767 000000G 000000G MOV PREADD,ERRADD ;SUPPLY ERROR ADDRESS
47 000124 012767 000002 000000G MOV #2,ERRCT ;PRINT 2 WORDS
48 000132 CALL MEMERR ;PRINT ERROR MESSAGE
49 000135 1$: RETURN
50 ;
51 ;
52 ; COMPARE RIGHT HALF OF MRP MICROPGM MEMORY
53 ;
54 ;
55 000140 CMPR::
56 000140 CALL UNLR ;READ A RIGHT HALF WORD
57 000144 026767 000000G 000000G CMP CKDATA,ERN2 ;SAME AS WORD WRITTEN

```

```

58 000152 001421          BEQ      1$                ;YES
59 000154 112767 000122 000052G MOVB    #'R,PRINT+42.    ;INDICATE FAILURE ON RIGHT-HALF WORD
60 000162 016746 000000G MOV     PREADD,-(SP)     ;SET ADDRESS FOR SEQUENCER
61 000166          CALL    SEQMM           ;
62 000172          CALL    UNLL           ;READ LEFT HALF WORD
63 000176 016767 000000G 000000G MOV     PREADD,ERRADD    ;SUPPLY ERROR ADDRESS
64 000204 012767 000002 000000G MOV     #2,ERRCT        ;PRINT 2 WORDS
65 000212          CALL    MEMERR        ;PRINT ERROR MESSAGE
66 000216          ;
67          ;
68          ;
69          ;
70          ;
71          ;
72          ;
73 000220          UNLL::
74 000220 012746 000006 MOV     #<MMOE+MMLEFT>,-(SP)
75 000224          CALL    MRPCR           ;DIRECT CNTL WORD TO MRP
76 000230 005046          CLR     -(SP)           ;RESET BR INHIBIT
77 000232          CALL    LBMRP           ;
78 000236          CALL    MRPLB           ;REQUEST MRP TO LOD BUS
79 000242 012667 000000G 000000G MOV     (SP)+,ERU1      ;GET MRP WORD FROM STACK
80 000246 005046          CLR     -(SP)           ;
81 000250          CALL    MRPCR           ;CLEAR THE MRP CR BITS
82 000254          RETURN          ;
83          ;
84          ;
85          ;
86          ;
87          ;
88 000256          UNLR::
89 000256 012746 000004 MOV     #<MMOE>,-(SP)
90 000262          CALL    MRPCR           ;DIRECT CNTL WORD TO MRP
91 000266 005046          CLR     -(SP)           ;RESET BR INHIBIT
92 000270          CALL    LBMRP           ;
93 000274          CALL    MRPLB           ;REQUEST MRP TO LOD BUS
94 000300 012667 000000G 000000G MOV     (SP)+,ERU2      ;GET MRP WORD FROM STACK
95 000304 005046          CLR     -(SP)           ;
96 000306          CALL    MRPCR           ;CLEAR THE MRP CR BITS
97 000312          RETURN          ;
98          ;
99          ;
100          ;
101          ;
102          ;
103 000314          SINGLE::
104 000314 012746 140000 MOV     #0$MNC,-(SP)    ;CLEAR MRP NO-CLOCKS
105 000320 012746 040000 MOV     #0$MSC,-(SP)    ;SINGLE CLOCK SEQ
106 000324          CALL    CSR1           ;
107 000330 012746 040000 MOV     #0$MNC,-(SP)    ;CLEAR SINGLE CLOCK
108 000334 012746 140000 MOV     #0$MNC,-(SP)    ;SET NO-CLOCKS
109 000340          CALL    CSR1           ;
110 000344          RETURN          ;
111          ;
112          .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MRPLB = ***** GX	Q\$QHLT = 000013
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	MSYN = 000040	Q\$QL = 000043
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	N = 000144	Q\$QLA = 000053
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLB = 000010	Q\$QLB = 000054
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLC = 000020	Q\$QLR = 000001
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLD = 000030	Q\$QW = 000042
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PLRWR = 000200	Q\$RDCD = 000005
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PLR:EN = 000200	Q\$RDMD = 000006
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	PREADD = ***** GX	Q\$REBK = 001000
BIT13 = 020000	BYTE51 = 000063	CKDATA = ***** GX	PRINT = ***** GX	Q\$RNC = 006000
BIT14 = 040000	BYTE52 = 000064	CNPL = 000060RG	002-OR\$CR1 = 176420	Q\$RSC = 004000
BIT15 = 100000	BYTE53 = 000065	CNPR = 000140RG	002-OR\$CR2 = 176422	Q\$RSET = 000010
BIT2 = 000004	BYTE54 = 000066	CNOBRE = 100000	OR\$LBR = 176424	Q\$SM = 100000
BIT3 = 000010	BYTE55 = 000067	CPCCEN = 010000	Q\$ATTN = 000100	Q\$SP = 000120
BIT4 = 000020	BYTE56 = 000070	CPREAD = 040000	Q\$BCL = 000001	Q\$SP2 = 000340
BIT5 = 000040	BYTE57 = 000071	CPWRTE = 020000	Q\$CCCP = 000040	RGD,EN = 000200
BIT6 = 000100	BYTE58 = 000072	CSADRD = 000004	Q\$CHB = 000400	RGD,VA = 020000
BIT7 = 000200	BYTE59 = 000073	CSEQCI = 100000	Q\$CHRL = 000200	SEGNM = ***** GX
BIT8 = 000400	BYTE60 = 000074	CSOE = 000040	Q\$CLR = 000040	SEQ,CT = 000010
BIT9 = 001000	BYTE61 = 000075	CSR1 = ***** GX	Q\$CNC = 030000	SINGLE = 000314RG
BYTE0 = 000000	BYTE62 = 000076	CSWRTE = 000100	Q\$CP = 000060	S\$CLR = 000000
BYTE1 = 000001	BYTE63 = 000077	DBR,RD = 000001	Q\$CPCC = 000010	S\$LA = 000001
BYTE10 = 000012	BYTE64 = 000100	DB\$CPP = 001457	Q\$CP2 = 000260	S\$OB = 000005
BYTE11 = 000013	BYTE65 = 000101	DB\$SPT = 000026	Q\$CSC = 010000	S\$OR = 000006
BYTE12 = 000014	BYTE66 = 000102	DB\$TPC = 000023	Q\$CSEL = 000360	S\$QX = 000004
BYTE13 = 000015	BYTE67 = 000103	DISPGS = 100000	Q\$CSET = 000002	S\$SR = 000007
BYTE14 = 000016	BYTE68 = 000104	DMARUR = 000005	Q\$CSP = 020000	S\$S1 = 000010
BYTE15 = 000017	BYTE69 = 000105	DMARRD = 000003	Q\$DMA = 000001	S\$S2 = 000014
BYTE16 = 000020	BYTE70 = 000106	DMARUR = 000004	Q\$ENBK = 040000	TD\$CTR = 176370
BYTE17 = 000021	BYTE71 = 000107	ENBR = 010000	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE18 = 000022	BYTE72 = 000110	ERRADD = ***** GX	Q\$FAL = 004000	TD\$INL = 004000
BYTE19 = 000023	BYTE73 = 000111	ERRCT = ***** GX	Q\$FC = 000045	TD\$MEM = 000270
BYTE2 = 000002	BYTE74 = 000112	ERW1 = ***** GX	Q\$FD = 000044	TD\$OAR = 176344
BYTE20 = 000024	BYTE75 = 000113	ERW2 = ***** GX	Q\$FP = 000046	TD\$OTR = 176346
BYTE21 = 000025	BYTE76 = 000114	LBMRP = ***** GX	Q\$HBF = 000002	TD\$ORD = 000274
BYTE22 = 000026	BYTE77 = 000115	LBMSC = ***** GX	Q\$ICP = 000006	TD\$SW = 176376
BYTE23 = 000027	BYTE78 = 000116	LOC,EN = 000100	Q\$IHB = 000003	TD\$TAR = 176372
BYTE24 = 000030	BYTE79 = 000117	LOC,WA = 040000	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE25 = 000031	BYTE80 = 000120	LOC,WB = 100000	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE26 = 000032	BYTE81 = 000121	MAREN1 = 000001	Q\$LBD = 001000	TD\$TDW = 176364
BYTE27 = 000033	BYTE82 = 000122	MAREN2 = 004000	Q\$LBDP = 001001	T\$AD = 000020
BYTE28 = 000034	BYTE83 = 000123	MARLOD = 010000	Q\$LBP = 000001	T\$AB = 000002
BYTE29 = 000035	BYTE84 = 000124	MAROUT = 000002	Q\$LCD = 000003	T\$BD = 000010
BYTE3 = 000003	BYTE85 = 000125	MAR,LO = 002000	Q\$LDPP = 000004	T\$BSO = 100000
BYTE30 = 000036	BYTE86 = 000126	MAR,OU = 000040	Q\$LHP = 010000	T\$BT = 000020
BYTE31 = 000037	BYTE87 = 000127	MBKALL = 001000	Q\$INC = 140000	T\$BTAR = 000030
BYTE32 = 000040	BYTE88 = 000130	MBKCLK = 000400	Q\$INR = 000052	T\$BTD = 002000
BYTE33 = 000041	BYTE89 = 000131	MEMERR = ***** GX	Q\$MR = 000040	T\$CD = 000100
BYTE34 = 000042	BYTE90 = 000132	MMADR = 000100	Q\$MRP = 000040	T\$CLK = 002000
BYTE35 = 000043	BYTE91 = 000133	MMLEFT = 000002	Q\$MRP2 = 000240	T\$D ISK = 000200
BYTE36 = 000044	BYTE92 = 000134	MMOE = 000004	Q\$MSC = 040000	T\$DRD = 000004
BYTE37 = 000045	BYTE93 = 000135	MMWRTE = 000010	Q\$MSET = 000004	T\$EMEM = 010000
BYTE38 = 000046		MNOBRE = 100000	Q\$MSP = 100000	T\$PSAA = 000000
BYTE39 = 000047		MREN1 = 000001	Q\$NCLK = 176000	T\$SAB = 000004
BYTE4 = 000004		MREN2 = 020000	Q\$PP = 000100	T\$SAC = 000014
BYTE40 = 000050		MRPCR = ***** GX	Q\$PPSW = 000320	T\$SB2 = 000010
BYTE41 = 000051		MRPCRA = ***** GX	Q\$PP2 = 000300	T\$IB = 000026

T\$IBAR = 000024	WORD1 = 000002	WORD33 = 000102	WORD57 = 000162	WORD80 = 000240
T\$IBE = 020000	WORD10 = 000024	WORD34 = 000104	WORD58 = 000164	WORD81 = 000242
T\$IBF = 040000	WORD11 = 000026	WORD35 = 000106	WORD59 = 000166	WORD82 = 000244
T\$ICD = 000040	WORD12 = 000030	WORD36 = 000110	WORD6 = 000014	WORD83 = 000246
T\$MODE = 004000	WORD13 = 000032	WORD37 = 000112	WORD60 = 000170	WORD84 = 000250
T\$OB = 000036	WORD14 = 000034	WORD38 = 000114	WORD61 = 000172	WORD85 = 000252
T\$OBE = 004000	WORD15 = 000036	WORD39 = 000116	WORD62 = 000174	WORD86 = 000254
T\$OBF = 010000	WORD16 = 000040	WORD4 = 000010	WORD63 = 000176	WORD87 = 000256
T\$OBRA = 000034	WORD17 = 000042	WORD40 = 000120	WORD64 = 000200	WORD88 = 000260
T\$OBWA = 000032	WORD18 = 000044	WORD41 = 000122	WORD65 = 000202	WORD89 = 000262
T\$OUTA = 100000	WORD19 = 000046	WORD42 = 000124	WORD66 = 000204	WORD9 = 000022
T\$RBD0 = 000200	WORD2 = 000004	WORD43 = 000126	WORD67 = 000206	WORD90 = 000264
T\$RNB = 000040	WORD20 = 000050	WORD44 = 000130	WORD68 = 000210	WORD91 = 000266
T\$RSET = 040000	WORD21 = 000052	WORD45 = 000132	WORD69 = 000212	WORD92 = 000270
T\$SC = 000022	WORD22 = 000054	WORD46 = 000134	WORD7 = 000016	WORD93 = 000272
T\$SCLK = 020000	WORD23 = 000056	WORD47 = 000136	WORD70 = 000214	WORD94 = 000274
T\$SEG1 = 000000	WORD24 = 000060	WORD48 = 000140	WORD71 = 000216	WORD95 = 000276
T\$SEG2 = 000001	WORD25 = 000062	WORD49 = 000142	WORD72 = 000220	WORD96 = 000300
T\$SEG3 = 000002	WORD26 = 000064	WORD5 = 000012	WORD73 = 000222	WORD97 = 000302
T\$SO = 000001	WORD27 = 000066	WORD50 = 000144	WORD74 = 000224	WORD98 = 000304
T\$UBUS = 100000	WORD28 = 000070	WORD51 = 000146	WORD75 = 000226	WORD99 = 000306
T\$ICLK = 000400	WORD29 = 000072	WORD52 = 000150	WORD76 = 000230	WRDVAL = 000310
T\$BEN = 000020	WORD3 = 000006	WORD53 = 000152	WORD77 = 000232	WRITEL = 000000RG 002
UBD.IN = 000020	WORD30 = 000074	WORD54 = 000154	WORD78 = 000234	WRITER = 000030RG 002
UNLL = 000220RG 002	WORD31 = 000076	WORD55 = 000156	WORD79 = 000236	XTREAD = 001000
UNLR = 000256RG 002	WORD32 = 000100	WORD56 = 000160	WORD8 = 000020	XTWRTE = 000400
WORD0 = 000000				

. ABS: 000000 000
000000 001
MMTSUB: 000346 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3096 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
MMTSUB,MMTSUB /-SP=[20, 1]IM,[20, 1]MMTSUB

```

1          .TITLE--MMTST0...
2 000000   .PSECT: MMTST0
3          ;
4          ; MEMORY TEST DIAGNOSTICS
5          ; MATCH REPORT PROCESSOR MICROPROGRAM MEMORY
6          ;
7          ;
8          ;
9          ; ALL-PURPOSE WRITE AND READ OF SEQUENTIAL MEMORY LOCATIONS
10         ;
11         STUFMM:
12 000000   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
13 000006   016746 000000G 1$: MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
14 000012   005267 000000G CALL SEQMM ;DO IT
15 000016   026667 000000G CALL WRITER ;WRITE LEFT HALF OF MEMORY
16 000022   026667 000004 000000G INC PREADD ;BUMP ADDRESS
17 000026   103364 000000G CMP 4(SP),PREADD ;FINISHED?
18 000034   103364 000000G BHIS 1$ ;NO
19         ;
20 000036   016667 000002 000000G 2$: MOV 2(SP),PREADD ;WORKING ADDRESS
21 000044   016746 000000G 2$: MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
22 000044   005267 000000G CALL SEQMM ;DO IT
23 000050   026667 000000G CALL WRITER ;WRITE RIGHT HALF OF MEMORY
24 000054   026667 000004 000000G INC PREADD ;BUMP ADDRESS
25 000060   103364 000000G CMP 4(SP),PREADD ;FINISHED?
26 000064   103364 000000G BHIS 2$ ;NO
27 000072   016667 000002 000000G 3$: MOV 2(SP),PREADD ;WORKING ADDRESS
28 000074   016746 000000G 3$: MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
29 000080   005267 000000G CALL SEQMM ;DO IT
30 000082   026667 000000G CALL WRITER ;WRITE LEFT HALF OF MEMORY
31 000086   026667 000004 000000G INC PREADD ;BUMP ADDRESS
32 000090   103364 000000G CMP 4(SP),PREADD ;FINISHED?
33 000094   103364 000000G BHIS 3$ ;NO
34 000102   016667 000002 000000G 4$: MOV 2(SP),PREADD ;WORKING ADDRESS
35 000106   016746 000000G 4$: MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
36 000110   005267 000000G CALL SEQMM ;DO IT
37 000112   026667 000000G CALL WRITER ;WRITE RIGHT HALF OF MEMORY
38 000116   026667 000004 000000G INC PREADD ;BUMP ADDRESS
39 000120   103364 000000G CMP 4(SP),PREADD ;FINISHED?
40 000124   103364 000000G BHIS 4$ ;NO
41 000132   016667 000002 000000G 5$: MOV 2(SP),PREADD ;WORKING ADDRESS
42 000136   016746 000000G 5$: MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
43 000140   005267 000000G CALL SEQMM ;DO IT
44 000144   026667 000000G CALL WRITER ;WRITE LEFT HALF OF MEMORY
45 000148   026667 000004 000000G INC PREADD ;BUMP ADDRESS
46 000152   103364 000000G CMP 4(SP),PREADD ;FINISHED?
47 000156   103364 000000G BHIS 5$ ;NO
48 000164   005046 000000G CLR -(SP) ;CLEAR CONTROL REG
49 000170   005046 000000G CALL MRPCR
50 000176   000001 000000G RETURN
51 000182   000001 000000G .END

```


ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR\$LBR = 176424	Q\$SM = 100000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100	Q\$SP = 000120
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001	Q\$SP2 = 000340
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040	RGQ:EN = 000200
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400	RGQ:VA = 020000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200	SEQM1 = ***** GX
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040	SEQ:CI = 000010
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000	STUFMM = 000000RG 002
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$CP = 000060	S\$CLR = 000000
BIT13 = 020000	BYTE51 = 000063	CHPL = ***** GX	Q\$CPCC = 000010	S\$LA = 000001
BIT14 = 040000	BYTE52 = 000064	CHPR = ***** GX	Q\$CP2 = 000260	S\$OB = 000005
BIT15 = 100000	BYTE53 = 000065	CHOBRE = 100000	Q\$CSC = 010000	S\$OR = 000006
BIT2 = 000004	BYTE54 = 000066	CPCCEN = 010000	Q\$CSEL = 000360	S\$QX = 000004
BIT3 = 000010	BYTE55 = 000067	CPREAD = 040000	Q\$CSET = 000002	S\$SR = 000007
BIT4 = 000020	BYTE56 = 000070	CPWRTE = 020000	Q\$CSP = 020000	S\$S1 = 000010
BIT5 = 000040	BYTE57 = 000071	CSADRD = 000004	Q\$DMA = 000001	S\$S2 = 000014
BIT6 = 000100	BYTE58 = 000072	CSEQCI = 100000	Q\$ENBK = 040000	TD\$CTR = 176370
BIT7 = 000200	BYTE59 = 000073	CSOE = 000040	Q\$ENOP = 020000	TD\$CTW = 176360
BIT8 = 000400	BYTE60 = 000074	CSURTE = 000100	Q\$FAL = 004000	TD\$INL = 004000
BIT9 = 001000	BYTE61 = 000075	DB:RD = 000001	Q\$FC = 000045	TD\$MEM = 000270
BYTE0 = 000000	BYTE62 = 000076	DB\$CPP = 001457	Q\$FO = 000044	TD\$OAR = 176344
BYTE1 = 000001	BYTE63 = 000077	DB\$SPT = 000026	Q\$FP = 000046	TD\$OTR = 176346
BYTE10 = 000012	BYTE64 = 000100	DB\$TPC = 000023	Q\$HBF = 000002	TD\$ORD = 000274
BYTE11 = 000013	BYTE65 = 000101	DISPGS = 100000	Q\$ICP = 000006	TD\$SW = 176376
BYTE12 = 000014	BYTE66 = 000102	DMAWR = 000005	Q\$IHB = 000003	TD\$TAR = 176372
BYTE13 = 000015	BYTE67 = 000103	DMARRD = 000003	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE14 = 000016	BYTE68 = 000104	DMARWR = 000004	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE15 = 000017	BYTE69 = 000105	ENBR = 010000	Q\$LBD = 001000	TD\$TDW = 176364
BYTE16 = 000020	BYTE70 = 000106	LOC:EN = 000100	Q\$LBDP = 001001	T\$AD = 000020
BYTE17 = 000021	BYTE71 = 000107	LOC:WA = 040000	Q\$LBP = 000001	T\$BA = 000002
BYTE18 = 000022	BYTE72 = 000110	LOC:WB = 100000	Q\$LDCD = 000003	T\$BD = 000010
BYTE19 = 000023	BYTE73 = 000111	MAREN1 = 000001	Q\$LDMO = 000004	T\$BSO = 100000
BYTE2 = 000002	BYTE74 = 000112	MAREN2 = 004000	Q\$LDPP = 002000	T\$BT = 000020
BYTE20 = 000024	BYTE75 = 000113	MARLOD = 010000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE21 = 000025	BYTE76 = 000114	MAROUT = 000002	Q\$INC = 140000	T\$BTD = 002000
BYTE22 = 000026	BYTE77 = 000115	MAR:LO = 002000	Q\$MR = 000052	T\$CD = 000100
BYTE23 = 000027	BYTE78 = 000116	MAR:OU = 000040	Q\$MRP = 000040	T\$CLK = 002000
BYTE24 = 000030	BYTE79 = 000117	MBKALL = 001000	Q\$MRP2 = 000240	T\$DISK = 000200
BYTE25 = 000031	BYTE80 = 000120	MBKCLK = 000400	Q\$MSC = 040000	T\$D = 000004
BYTE26 = 000032	BYTE81 = 000121	MNADRD = 000100	Q\$MSET = 000004	T\$ENEM = 010000
BYTE27 = 000033	BYTE82 = 000122	MNLEFT = 000002	Q\$MSP = 100000	T\$FSAA = 000000
BYTE28 = 000034	BYTE83 = 000123	MNDE = 000004	Q\$SNCLK = 176000	T\$FSAB = 000004
BYTE29 = 000035	BYTE84 = 000124	MNURTE = 000010	Q\$PP = 000100	T\$FSAC = 000014
BYTE3 = 000003	BYTE85 = 000125	MNOBRE = 100000	Q\$PPSW = 000320	T\$FSB2 = 000010
BYTE30 = 000036	BYTE86 = 000126	MREN1 = 000001	Q\$PP2 = 000300	T\$IB = 000026
BYTE31 = 000037	BYTE87 = 000127	MREN2 = 020000	Q\$QHLT = 000013	T\$IBAR = 000024
BYTE32 = 000040	BYTE88 = 000130	MRPCR = ***** GX	Q\$QL = 000043	T\$IBE = 020000
BYTE33 = 000041	BYTE89 = 000131	MSYN = 000040	Q\$QLA = 000053	T\$IBF = 040000
BYTE34 = 000042	BYTE90 = 000132	N = 000144	Q\$QLB = 000054	T\$ICD = 000040
BYTE35 = 000043	BYTE91 = 000133	PLB = 000010	Q\$QLR = 000001	T\$MODE = 004000
BYTE36 = 000044	BYTE92 = 000134	PLC = 000020	Q\$QW = 000042	T\$OB = 000036
BYTE37 = 000045	BYTE93 = 000135	PLD = 000030	Q\$RDGD = 000005	T\$OBE = 004000
BYTE38 = 000046		PLRWR = 000200	Q\$RMD = 000006	T\$OBF = 010000
BYTE39 = 000047		PLR:EN = 000200	Q\$REBK = 001000	T\$OBRA = 000034
BYTE4 = 000004		PREADD = ***** GX	Q\$RNC = 005000	T\$OBTA = 000032
BYTE40 = 000050		QR\$CR1 = 176420	Q\$RSC = 004000	T\$OUTA = 100000
BYTE41 = 000051		QR\$CR2 = 176422	Q\$RSET = 000010	T\$RBD0 = 000200

T#RNB = .000040	WORD2 = .000004	WORD41 = .000122	WORD62 = .000174	WORD83 = .000246
T#RSET = .040000	WORD20 = .000050	WORD42 = .000124	WORD63 = .000176	WORD84 = .000250
T#SC = .000022	WORD21 = .000052	WORD43 = .000126	WORD64 = .000200	WORD85 = .000252
T#SCLK = .020000	WORD22 = .000054	WORD44 = .000130	WORD65 = .000202	WORD86 = .000254
T#SEG1 = .000000	WORD23 = .000056	WORD45 = .000132	WORD66 = .000204	WORD87 = .000256
T#SEG2 = .000001	WORD24 = .000060	WORD46 = .000134	WORD67 = .000206	WORD88 = .000260
T#SEG3 = .000002	WORD25 = .000062	WORD47 = .000136	WORD68 = .000210	WORD89 = .000262
T#SO = .000001	WORD26 = .000064	WORD48 = .000140	WORD69 = .000212	WORD9 = .000022
T#UBUS = .100000	WORD27 = .000066	WORD49 = .000142	WORD7 = .000016	WORD90 = .000264
T#1CLK = .000400	WORD28 = .000070	WORD5 = .000012	WORD70 = .000214	WORD91 = .000266
T#BEN = .000020	WORD29 = .000072	WORD50 = .000144	WORD71 = .000216	WORD92 = .000270
UBD.IN = .000020	WORD3 = .000006	WORD51 = .000146	WORD72 = .000220	WORD93 = .000272
WORD0 = .000000	WORD30 = .000074	WORD52 = .000150	WORD73 = .000222	WORD94 = .000274
WORD1 = .000002	WORD31 = .000076	WORD53 = .000152	WORD74 = .000224	WORD95 = .000276
WORD10 = .000024	WORD32 = .000100	WORD54 = .000154	WORD75 = .000226	WORD96 = .000300
WORD11 = .000026	WORD33 = .000102	WORD55 = .000156	WORD76 = .000230	WORD97 = .000302
WORD12 = .000030	WORD34 = .000104	WORD56 = .000160	WORD77 = .000232	WORD98 = .000304
WORD13 = .000032	WORD35 = .000106	WORD57 = .000162	WORD78 = .000234	WORD99 = .000306
WORD14 = .000034	WORD36 = .000110	WORD58 = .000164	WORD79 = .000236	WRDVAL = .000310
WORD15 = .000036	WORD37 = .000112	WORD59 = .000166	WORD8 = .000020	WRITEL = ***** GX
WORD16 = .000040	WORD38 = .000114	WORD6 = .000014	WORD80 = .000240	WRITER = ***** GX
WORD17 = .000042	WORD39 = .000116	WORD60 = .000170	WORD81 = .000242	XTREAD = .001000
WORD18 = .000044	WORD4 = .000010	WORD61 = .000172	WORD82 = .000244	XTWRITE = .000400
WORD19 = .000046	WORD40 = .000120			

. ABS: 000000 000
000000 001
MMTST0 000200 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3036 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
MMTST0,MMTST0/-SP=C20,1]IM,C20,1]MMTST0

```

1          .TITLE: MMTST1
2 000000   .PSECT: MMTST1
3          ;
4          ;
5          ; MEMORY TEST DIAGNOSTICS
6          ;
7          ;
8          ;
9          ;
10         ;
11        ; TEST-01
12        ; WRITE MEMORY ADDRESS INTO MEMORY LOCATION
13        ;
14 000000   TIMM:
15 000000   016667 000002 000000G MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER
16 000006   016667 000002 000000G MOV. 2(SP),PREADD. ;WORKING ADDRESS
17 000014   1$:
18 000014   016746 000000G MOV. PREADD, -(SP) ;SEQUENCE UP TO START ADDRESS
19 000020   CALL. SEQMM. ;DO IT
20 000024   CALL. WRITEL. ;WRITE LEFT HALF OF MEMORY
21 000030   005267 000000G INC. CKDATA. ;BUMP TEST COUNTER
22 000034   005267 000000G INC. PREADD. ;BUMP ADDRESS
23 000040   026667 000004 000000G CMP. 4(SP),PREADD. ;FINISHED?
24 000046   103362. BHIS. 1$ ;NO
25          ;
26 000050   016667 000002 000000G MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER
27 000056   016667 000002 000000G MOV. 2(SP),PREADD. ;WORKING ADDRESS
28 000064   2$:
29 000064   016746 000000G MOV. PREADD, -(SP) ;SEQ UP TO START ADDRESS
30 000070   CALL. SEQMM. ;DO IT
31 000074   CALL. WRITER. ;WRITE RIGHT HALF OF MEMORY
32 000100   005267 000000G INC. CKDATA. ;BUMP TEST COUNTER
33 000104   005267 000000G INC. PREADD. ;BUMP ADDRESS
34 000110   026667 000004 000000G CMP. 4(SP),PREADD. ;FINISHED?
35 000116   103362. BHIS. 2$ ;NO
36          ;
37 000120   016667 000002 000000G MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER
38 000126   016667 000002 000000G MOV. 2(SP),PREADD. ;WORKING ADDRESS
39 000134   3$:
40 000134   016746 000000G MOV. PREADD, -(SP) ;SEQ UP TO START ADDRESS
41 000140   CALL. SEQMM. ;DO IT
42 000144   CALL. CMPL. ;COMPARE LEFT HALF OF MEMORY
43 000150   005267 000000G INC. CKDATA. ;BUMP TEST COUNTER
44 000154   005267 000000G INC. PREADD. ;BUMP ADDRESS
45 000160   026667 000004 000000G CMP. 4(SP),PREADD. ;FINISHED?
46 000166   103362. BHIS. 3$ ;NO
47          ;
48 000170   016667 000002 000000G MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER
49 000176   016667 000002 000000G MOV. 2(SP),PREADD. ;WORKING ADDRESS
50 000204   4$:
51 000204   016746 000000G MOV. PREADD, -(SP) ;SEQ UP TO START ADDRESS
52 000210   CALL. SEQMM. ;DO IT
53 000214   CALL. CMR. ;COMPARE RIGHT HALF OF MEMORY
54 000220   005267 000000G INC. CKDATA. ;BUMP TEST COUNTER
55 000224   005267 000000G INC. PREADD. ;BUMP ADDRESS
56 000230   026667 000004 000000G CMP. 4(SP),PREADD. ;FINISHED?
57 000236   103362. BHIS. 4$ ;NO

```

MMTST1 MACRO M1110 27-MAR-80 15:04 PAGE 5-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

58
59
60 000240
61 000001

;

RETURN
.END.

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	OR\$LBR = 176424	Q\$SM = 100000
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100	Q\$SP = 000120
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001	Q\$SPZ = 000340
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040	RGD.EN = 000200
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400	RGD.VA = 020000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200	SEOMT = ***** GX
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040	SEQ.CI = 000010
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000	S\$CLA = 000000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CP = 000060	S\$LA = 000001
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q\$CPC = 000010	S\$OB = 000005
BIT14 = 040000	BYTE51 = 000063	CMPL = ***** GX	Q\$CP2 = 000260	S\$QR = 000006
BIT15 = 100000	BYTE52 = 000064	CMPR = ***** GX	Q\$CSC = 010000	S\$QX = 000004
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	Q\$CSEL = 000360	S\$SR = 000007
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	Q\$CSET = 000002	S\$S1 = 000010
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q\$CSP = 020000	S\$S2 = 000014
BIT5 = 000040	BYTE56 = 000070	CPURTE = 020000	Q\$DMA = 000001	TD\$CTR = 176370
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTW = 176360
BIT7 = 000200	BYTE58 = 000072	CSEDC1 = 100000	Q\$ENOP = 020000	TD\$INL = 004000
BIT8 = 000400	BYTE59 = 000073	CSOE = 000040	Q\$FAL = 004000	TD\$MEM = 000270
BIT9 = 001000	BYTE6 = 000006	CSURTE = 000100	Q\$FC = 000045	TD\$OAR = 176344
BYTE0 = 000000	BYTE60 = 000074	DBR.RD = 000001	Q\$FO = 000044	TD\$OTR = 176346
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$FP = 000046	TD\$ORD = 000274
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$HBF = 000002	TD\$SW = 176376
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$ICP = 000006	TD\$TAR = 176372
BYTE12 = 000014	BYTE64 = 000100	DISPGS = 100000	Q\$IHB = 000003	TD\$TAW = 176362
BYTE13 = 000015	BYTE65 = 000101	DMARWR = 000005	Q\$IHRL = 000002	TD\$TDR = 176374
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$IMRP = 000007	TD\$TDW = 176364
BYTE15 = 000017	BYTE67 = 000103	DMARWR = 000004	Q\$LBD = 001000	T\$AD = 000020
BYTE16 = 000020	BYTE68 = 000104	ENBR = 010000	Q\$LBDP = 001001	T\$BA = 000002
BYTE17 = 000021	BYTE69 = 000105	LOC.EN = 000100	Q\$LBP = 000001	T\$BD = 000010
BYTE18 = 000022	BYTE7 = 000007	LOC.WA = 040000	Q\$LDCD = 000003	T\$BSO = 100000
BYTE19 = 000023	BYTE70 = 000106	LOC.WB = 100000	Q\$LDMD = 000004	T\$BT = 000020
BYTE2 = 000002	BYTE71 = 000107	MAREN1 = 000001	Q\$LDPP = 002000	T\$BTAR = 000030
BYTE20 = 000024	BYTE72 = 000110	MAREN2 = 004000	Q\$LHP = 010000	T\$BTDR = 002000
BYTE21 = 000025	BYTE73 = 000111	MARLOD = 010000	Q\$MNC = 140000	T\$CD = 000100
BYTE22 = 000026	BYTE74 = 000112	MAROUT = 000002	Q\$MR = 000052	T\$CLK = 000200
BYTE23 = 000027	BYTE75 = 000113	MAR.LO = 002000	Q\$MRP = 000040	T\$DTSK = 000200
BYTE24 = 000030	BYTE76 = 000114	MAR.OU = 000040	Q\$MRP2 = 000240	T\$DTR = 000004
BYTE25 = 000031	BYTE77 = 000115	MBKALL = 001000	Q\$MSC = 040000	T\$ELEM = 010000
BYTE26 = 000032	BYTE78 = 000116	MBKCLK = 000400	Q\$MSET = 000004	T\$FSA = 000000
BYTE27 = 000033	BYTE79 = 000117	M\$ADRD = 000100	Q\$MSP = 100000	T\$FSAB = 000004
BYTE28 = 000034	BYTE8 = 000010	M\$LEFT = 000002	Q\$NCLK = 176000	T\$FSAC = 000014
BYTE29 = 000035	BYTE80 = 000120	M\$OE = 000004	Q\$PP = 000100	T\$FSB2 = 000010
BYTE3 = 000003	BYTE81 = 000121	M\$URTE = 000010	Q\$PPSW = 000320	T\$IB = 000026
BYTE30 = 000036	BYTE82 = 000122	M\$NOBRE = 100000	Q\$PP2 = 000300	T\$IBAR = 000024
BYTE31 = 000037	BYTE83 = 000123	MREN1 = 000001	Q\$QHLT = 000013	T\$IBE = 020000
BYTE32 = 000040	BYTE84 = 000124	MREN2 = 020000	Q\$QL = 000043	T\$IBF = 040000
BYTE33 = 000041	BYTE85 = 000125	MSYN = 000040	Q\$QLA = 000053	T\$ICD = 000040
BYTE34 = 000042	BYTE86 = 000126	N = 000144	Q\$QLB = 000054	T\$MODE = 004000
BYTE35 = 000043	BYTE87 = 000127	PLB = 000010	Q\$QLR = 000011	T\$OB = 000036
BYTE36 = 000044	BYTE88 = 000130	PLC = 000020	Q\$QW = 000042	T\$OBE = 004000
BYTE37 = 000045	BYTE89 = 000131	PLD = 000030	Q\$RDCD = 000005	T\$OBF = 010000
BYTE38 = 000046	BYTE9 = 000011	PLRWR = 000200	Q\$RDMD = 000005	T\$OBRA = 000034
BYTE39 = 000047	BYTE90 = 000132	PLR.EN = 000200	Q\$REBK = 001000	T\$OBWA = 000032
BYTE4 = 000004	BYTE91 = 000133	PREADD = ***** GX	Q\$RNC = 006000	T\$OUTA = 100000
BYTE40 = 000050	BYTE92 = 000134	Q\$RCR1 = 176420	Q\$RSC = 004000	T\$RBD0 = 000200
BYTE41 = 000051	BYTE93 = 000135	Q\$RCR2 = 176422	Q\$RSET = 000010	T\$RNB = 000040

T#RSET = 040000	WORD2 = 000004	WORD41 = 000122	WORD62 = 000174	WORD83 = 000246
T#SC = 000022	WORD20 = 000050	WORD42 = 000124	WORD63 = 000176	WORD84 = 000250
T#SCLK = 020000	WORD21 = 000052	WORD43 = 000126	WORD64 = 000200	WORD85 = 000252
T#SEG1 = 000000	WORD22 = 000054	WORD44 = 000130	WORD65 = 000202	WORD86 = 000254
T#SEG2 = 000001	WORD23 = 000056	WORD45 = 000132	WORD66 = 000204	WORD87 = 000256
T#SEG3 = 000002	WORD24 = 000060	WORD46 = 000134	WORD67 = 000206	WORD88 = 000260
T#SO = 000001	WORD25 = 000062	WORD47 = 000136	WORD68 = 000210	WORD89 = 000262
T#UBUS = 100000	WORD26 = 000064	WORD48 = 000140	WORD69 = 000212	WORD90 = 000264
T#ICLK = 000400	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD91 = 000266
T#BBEN = 000020	WORD28 = 000070	WORD50 = 000144	WORD71 = 000216	WORD92 = 000270
TIMM = 000000RG	002 WORD29 = 000072	WORD51 = 000146	WORD72 = 000220	WORD93 = 000272
UBD.IN = 000020	WORD3 = 000006	WORD52 = 000150	WORD73 = 000222	WORD94 = 000274
WORD0 = 000000	WORD30 = 000074	WORD53 = 000152	WORD74 = 000224	WORD95 = 000276
WORD1 = 000002	WORD31 = 000076	WORD54 = 000154	WORD75 = 000226	WORD96 = 000300
WORD10 = 000024	WORD32 = 000100	WORD55 = 000156	WORD76 = 000230	WORD97 = 000302
WORD11 = 000026	WORD33 = 000102	WORD56 = 000160	WORD77 = 000232	WORD98 = 000304
WORD12 = 000030	WORD34 = 000104	WORD57 = 000162	WORD78 = 000234	WORD99 = 000306
WORD13 = 000032	WORD35 = 000106	WORD58 = 000164	WORD79 = 000236	WRDVAL = 000310
WORD14 = 000034	WORD36 = 000110	WORD59 = 000166	WORD80 = 000240	WRITEL = ***** GX
WORD15 = 000036	WORD37 = 000112	WORD60 = 000170	WORD81 = 000242	WRITER = ***** GX
WORD16 = 000040	WORD38 = 000114	WORD61 = 000172	WORD82 = 000244	XTREAD = 001000
WORD17 = 000042	WORD39 = 000116			XTWRITE = 000400
WORD18 = 000044	WORD40 = 000120			
WORD19 = 000046				

. ABS. 000000 000
000000 001
MMTST1 000242. 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3036 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
MMTST1, MMTST1 / - SP = C20, IJM, C20, IJMMTST1

```

1
2 000000 .TITLE MMTST2
3 .PSECT MMTST2
4 ;
5 ; MEMORY TEST DIAGNOSTICS
6 ; MATCH REPORT PROCESSOR MICROPROGRAM MEMORY
7 ;
8 ;
9 ;
10 ;
11 ; CROSS-TALK TEST
12 ;
13 ;
14 ; WRITE ONES IN EVERY OTHER MEMORY LOCATION 10 TIMES
15 ;
16 000000 T6MM:
17 000000 012767 177777 000000 MOV #1,CKDATA ;SET TEST PATTERN = X'FFFF'
18 000012 016667 000002 000000G 10$ MOV #10,R2 ;SET LOOP COUNT
19 000020 016746 000000G 1$ MOV 2(SP),PREADD ;WORKING ADDRESS
20 000020 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
21 000024 CALL SEQMM ;DO IT
22 000030 CALL WRITEL ;WRITE LEFT HALF OF MEMORY
23 000034 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
24 000042 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
25 000050 103363 BHS 1$ ;NO
26 000052 005302 DEC R2 ;SUB FROM LOOP COUNT
27 000054 001356 BNE 10$
28 ;
29 000056 012702 000012 MOV #10,R2 ;SET LOOP COUNT
30 000062 016667 000002 000000G 20$ MOV 2(SP),PREADD ;WORKING ADDRESS
31 000070 016746 000000G 2$
32 000070 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
33 000074 CALL SEQMM ;DO IT
34 000100 CALL WRITER ;WRITE RIGHT HALF OF MEMORY
35 000104 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
36 000112 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
37 000120 103363 BHS 2$ ;NO
38 000122 005302 DEC R2 ;SUB FROM LOOP COUNT
39 000124 001356 BNE 20$
40 ;
41 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
42 ; WERE NOT WRITTEN
43 ;
44 000126 R6Z:
45 000126 005067 000000G CLR CKDATA ;SET TEST PATTERN = 0
46 000132 016667 000002 000000G MOV 2(SP),PREADD ;GET START ADDRESS
47 000140 005267 000000G INC PREADD ;BUMP START ADDRESS
48 000144 016746 000000G 1$
49 000144 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
50 000150 CALL SEQMM ;DO IT
51 000154 CALL CMPL ;COMPARE LEFT HALF OF MEMORY
52 000160 062767 000002 000000G ADD #2,PREADD ;SKIP HERE TOO
53 000166 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
54 000174 103363 BHS 1$
55 ;
56 000176 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
57 000204 005267 000000G INC PREADD ;BUMP START ADDRESS

```

58	000210				2\$:				
59	000210	016746	000000G			MOV	PREADD, -(SP)		:SET SEQUENCER TO START ADDRESS
60	000214					CALL	SEQMM		:DO IT
61	000220					CALL	CMR		:COMPARE RIGHT HALF OF MEMORY
62	000224	062767	000002	000000G		ADD	#2, PREADD		:SKIP HERE TOO
63	000232	026667	000004	000000G		CMP	4(SP), PREADD		:FINISHED ?
64	000240	103363				BHIS	2\$		
65									
66	000242					RETURN			
67		000001				.END			

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR\$LBR = 176424	Q\$SM = 100000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100	Q\$SP = 000120
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001	Q\$SP2 = 000340
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040	RGQ.EN = 000200
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400	RGQ.VA = 020000
SIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200	R6Z = 000126R 002
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040	SEOMM = *****GX
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000	SEQ.C1 = 000010
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$CP = 000060	S\$CLR = 000000
BIT13 = 020000	BYTE51 = 000063	CKDATA = *****GX	Q\$CPCC = 000010	S\$LA = 000001
BIT14 = 040000	BYTE52 = 000064	CMPL = *****GX	Q\$CP2 = 000260	S\$QB = 000005
BIT15 = 100000	BYTE53 = 000065	CMPR = *****GX	Q\$CSC = 010000	S\$QR = 000006
BIT2 = 000004	BYTE54 = 000066	CHOBRE = 100000	Q\$CSEL = 000360	S\$QX = 000004
BIT3 = 000010	BYTE55 = 000067	CPCCEN = 010000	Q\$CSET = 000002	S\$SR = 000007
BIT4 = 000020	BYTE56 = 000070	CPREAD = 040000	Q\$CSP = 020000	S\$S1 = 000010
BIT5 = 000040	BYTE57 = 000071	CPWRTE = 020000	Q\$DMA = 000001	S\$S2 = 000014
BIT6 = 000100	BYTE58 = 000072	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTR = 176370
BIT7 = 000200	BYTE59 = 000073	CSEDCI = 100000	Q\$ENOP = 020000	TD\$CTLW = 176360
BIT8 = 000400	BYTE60 = 000074	CSOE = 000040	Q\$FAL = 004000	TD\$INL = 004000
BIT9 = 001000	BYTE61 = 000075	CSWRTE = 000100	Q\$FC = 000045	TD\$MEM = 000270
BYTE0 = 000000	BYTE62 = 000076	DBR.RD = 000001	Q\$FO = 000044	TD\$OAR = 176344
BYTE1 = 000001	BYTE63 = 000077	DB\$CPP = 001457	Q\$FP = 000046	TD\$OTR = 176346
BYTE10 = 000012	BYTE64 = 000100	DB\$SPT = 000026	Q\$HBF = 000002	TD\$QRD = 000274
BYTE11 = 000013	BYTE65 = 000101	DB\$TPC = 000023	Q\$ICP = 000006	TD\$SW = 176376
BYTE12 = 000014	BYTE66 = 000102	DISPGRS = 100000	Q\$IHB = 000003	TD\$TAR = 176372
BYTE13 = 000015	BYTE67 = 000103	DMANWR = 000005	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE14 = 000016	BYTE68 = 000104	DMARRD = 000003	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE15 = 000017	BYTE69 = 000105	DMARWR = 000004	Q\$LBD = 001000	TD\$TDW = 176364
BYTE16 = 000020	BYTE70 = 000106	ENBR = 010000	Q\$LBDP = 001001	T\$AD = 000020
BYTE17 = 000021	BYTE71 = 000107	LOC.EN = 000100	Q\$LBP = 000001	T\$BA = 000002
BYTE18 = 000022	BYTE72 = 000110	LOC.WA = 040000	Q\$LCD = 000003	T\$BD = 000010
BYTE19 = 000023	BYTE73 = 000111	LOC.WB = 100000	Q\$LMD = 000004	T\$BSO = 100000
BYTE2 = 000002	BYTE74 = 000112	MAREN1 = 000001	Q\$LDPP = 002000	T\$BT = 000020
BYTE20 = 000024	BYTE75 = 000113	MAREN2 = 004000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE21 = 000025	BYTE76 = 000114	MARLDD = 010000	Q\$MNC = 140000	T\$BTD = 002000
BYTE22 = 000026	BYTE77 = 000115	MAROUT = 000002	Q\$MR = 000052	T\$CD = 000100
BYTE23 = 000027	BYTE78 = 000116	MAR.LD = 002000	Q\$MRP = 000040	T\$CLK = 000200
BYTE24 = 000030	BYTE79 = 000117	MAR.OU = 000040	Q\$MRP2 = 000240	T\$DLSK = 000200
BYTE25 = 000031	BYTE80 = 000120	MBKALL = 001000	Q\$MSC = 040000	T\$DRD = 000004
BYTE26 = 000032	BYTE81 = 000121	MBKCLK = 000400	Q\$MSET = 000004	T\$ENEM = 010000
BYTE27 = 000033	BYTE82 = 000122	MADRD = 000100	Q\$MSP = 100000	T\$FSAB = 000000
BYTE28 = 000034	BYTE83 = 000123	MHLEFT = 000002	Q\$NCLK = 176000	T\$FSAB = 000004
BYTE29 = 000035	BYTE84 = 000124	MHDE = 000004	Q\$PP = 000100	T\$FSAC = 000014
BYTE3 = 000003	BYTE85 = 000125	MHWRT = 000010	Q\$PPSW = 000320	T\$FSB2 = 000010
BYTE30 = 000036	BYTE86 = 000126	MNOBRE = 100000	Q\$PP2 = 000300	T\$IB = 000026
BYTE31 = 000037	BYTE87 = 000127	MREN1 = 000001	Q\$QHLT = 000013	T\$IBAR = 000024
BYTE32 = 000040	BYTE88 = 000130	MREN2 = 020000	Q\$QL = 000043	T\$IBE = 020000
BYTE33 = 000041	BYTE89 = 000131	MSYN = 000040	Q\$QLA = 000053	T\$IBF = 040000
BYTE34 = 000042	BYTE90 = 000132	N = 000144	Q\$QLB = 000054	T\$ICD = 000040
BYTE35 = 000043	BYTE91 = 000133	PLB = 000010	Q\$QLR = 000001	T\$MODE = 004000
BYTE36 = 000044	BYTE92 = 000134	PLC = 000020	Q\$QW = 000042	T\$OB = 000036
BYTE37 = 000045	BYTE93 = 000135	PLD = 000030	Q\$RDCD = 000005	T\$OB = 004000
BYTE38 = 000046		PLRWR = 000200	Q\$RDMD = 000006	T\$OBF = 010000
BYTE39 = 000047		PLREN = 000200	Q\$REBK = 001000	T\$OBRA = 000034
BYTE4 = 000004		PREAD = *****GX	Q\$RNC = 000000	T\$OBWA = 000032
BYTE40 = 000050		QR\$CR1 = 176420	Q\$RSC = 004000	T\$OUTA = 100000
BYTE41 = 000051		QR\$CR2 = 176422	Q\$RSET = 000010	T\$RBD0 = 000200

T#RNB = 000040	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD83 = 000246
T#RSET = 040000	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD84 = 000250
T#SC = 000022	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD85 = 000254
T#SCLK = 020000	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD86 = 000254
T#SEG1 = 000000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD87 = 000256
T#SEG2 = 000001	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD88 = 000260
T#SEG3 = 000002	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD89 = 000262
T#SD = 000001	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD9 = 000022
T#UBUS = 100000	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD90 = 000264
T#1CLK = 000400	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD91 = 000266
T#BBEN = 000020	WORD28 = 000070	WORDS = 000012	WORD71 = 000216	WORD92 = 000270
T#MM = 000000RG	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD93 = 000272
UBD.IN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD94 = 000274
WORD0 = 000000	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD95 = 000276
WORD1 = 000002	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD96 = 000300
WORD10 = 000024	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD97 = 000302
WORD11 = 000026	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD98 = 000304
WORD12 = 000030	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORD99 = 000306
WORD13 = 000032	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WRDVAL = 000310
WORD14 = 000034	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRITER = ***** GX
WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD00 = 000240	WRITER = ***** GX
WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242	XTREAD = 001000
WORD17 = 000042	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244	XTWRITE = 000400
WORD18 = 000044	WORD4 = 000010	WORD61 = 000172		

. ABS. 000000 000
000000 001
MMTST2. 000244 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3051 WORDS. (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS. (14 PAGES)
ELAPSED TIME: 00:00:40
MMTST2,MMTST2 /-SP=C20,1JIM,C20,1JMMTST2.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

58	000202	005102	COM	R2	:GET ADDRESS COMPLEMENT
59	000204	010267	MOV	R2,CKDATA	:SET TEST PATTERN = ADDR COMPLEMENT
60	000210	000000G	CALL	CMR	:COMPARE RIGHT HALF
61	000214	005267	INC	PREADD	:BUMP ADDRESS
62	000220	016702	MOV	PREADD,R2	:SET R2 TO NEXT ADDRESS
63	000224	026602	CMP	4(SP),R2	:FINISHED ?
64	000230	103360	BHIS	6\$:NO
65					
66	000232		RETURN		
67		000001	.END		

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	OR\$LBR = 176424	Q\$SM = 100000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100	Q\$SP = 000120
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001	Q\$SP2 = 000340
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040	RGQ.EN = 000200
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400	RGQ.VA = 020000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200	SEGNM = ***** GX
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040	SEQ.CI = 000010
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000	S\$CLR = 000000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CP = 000060	S\$LA = 000001
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q\$CPCC = 000010	S\$OB = 000005
BIT14 = 040000	BYTE51 = 000063	CNPL = ***** GX	Q\$CP2 = 000260	S\$OR = 000006
BIT15 = 100000	BYTE52 = 000064	CMPR = ***** GX	Q\$CSC = 010000	S\$OX = 000004
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	Q\$CSEL = 000360	S\$SR = 000007
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	Q\$CSET = 000002	S\$S1 = 000010
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q\$CSP = 020000	S\$S2 = 000014
BIT5 = 000040	BYTE56 = 000070	CPURTE = 020000	Q\$DMA = 000001	TD\$CTR = 176370
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTW = 176360
BIT7 = 000200	BYTE58 = 000072	CSEQCI = 100000	Q\$ENOP = 020000	TD\$INL = 004000
BIT8 = 000400	BYTE59 = 000073	CSOE = 000040	Q\$FAL = 004000	TD\$MEM = 000270
BIT9 = 001000	BYTE6 = 000006	CSURTE = 000100	Q\$FC = 000045	TD\$OAR = 176344
BYTE0 = 000000	BYTE60 = 000074	DBR.RD = 000001	Q\$FO = 000044	TD\$OTR = 176346
BYTE1 = 000001	BYTE61 = 000075	DB\$CPP = 001457	Q\$FP = 000046	TD\$ORD = 000274
BYTE10 = 000012	BYTE62 = 000076	DB\$SPT = 000026	Q\$HBF = 000002	TD\$SW = 176376
BYTE11 = 000013	BYTE63 = 000077	DB\$TPC = 000023	Q\$ICP = 000006	TD\$TAR = 176372
BYTE12 = 000014	BYTE64 = 000100	DISPGS = 100000	Q\$IHB = 000003	TD\$TAW = 176362
BYTE13 = 000015	BYTE65 = 000101	DMAAUR = 000005	Q\$IHRL = 000002	TD\$TDR = 176374
BYTE14 = 000016	BYTE66 = 000102	DMARRD = 000003	Q\$IHRRP = 000007	TD\$TDW = 176364
BYTE15 = 000017	BYTE67 = 000103	DMARUR = 000004	Q\$LBD = 001000	T\$AD = 000020
BYTE16 = 000020	BYTE68 = 000104	ENBR = 010000	Q\$LBDP = 001001	T\$BA = 000002
BYTE17 = 000021	BYTE69 = 000105	LOC.EN = 000100	Q\$LBP = 000001	T\$BD = 000010
BYTE18 = 000022	BYTE7 = 000007	LOC.WA = 040000	Q\$LCD = 000003	T\$BSO = 100000
BYTE19 = 000023	BYTE70 = 000106	LOC.WB = 100000	Q\$LDMD = 000004	T\$BT = 000020
BYTE2 = 000002	BYTE71 = 000107	MAREN1 = 000001	Q\$LDPP = 002000	T\$BIAR = 000030
BYTE20 = 000024	BYTE72 = 000110	MAREN2 = 004000	Q\$LHP = 010000	T\$BDT = 002000
BYTE21 = 000025	BYTE73 = 000111	MARLDD = 010000	Q\$MNC = 140000	T\$CD = 000100
BYTE22 = 000026	BYTE74 = 000112	MAROUT = 000002	Q\$MR = 000052	T\$CLK = 002000
BYTE23 = 000027	BYTE75 = 000113	MAR.LO = 002000	Q\$MRP = 000040	T\$DISK = 000200
BYTE24 = 000030	BYTE76 = 000114	MAR.OU = 000040	Q\$MRP2 = 000240	T\$DRD = 000004
BYTE25 = 000031	BYTE77 = 000115	MBKALL = 001000	Q\$MSC = 040000	T\$EMEM = 010000
BYTE26 = 000032	BYTE78 = 000116	MBKCLK = 000400	Q\$MSET = 000004	T\$FSAB = 000000
BYTE27 = 000033	BYTE79 = 000117	MMADR = 000100	Q\$MSP = 100000	T\$FSAB2 = 000004
BYTE28 = 000034	BYTE8 = 000010	MMLEFT = 000002	Q\$NCLK = 176000	T\$FSAC = 000014
BYTE29 = 000035	BYTE80 = 000120	MMDE = 000004	Q\$PP = 000100	T\$FSB2 = 000010
BYTE3 = 000003	BYTE81 = 000121	MMWRTE = 000010	Q\$PPSW = 000320	T\$IB = 000026
BYTE30 = 000036	BYTE82 = 000122	MNOBRE = 100000	Q\$PP2 = 000300	T\$IBAR = 000024
BYTE31 = 000037	BYTE83 = 000123	MREN1 = 000001	Q\$QHLT = 000013	T\$IBE = 020000
BYTE32 = 000040	BYTE84 = 000124	MREN2 = 020000	Q\$QL = 000043	T\$IBF = 040000
BYTE33 = 000041	BYTE85 = 000125	MSYN = 000044	Q\$QLA = 000053	T\$ICD = 000040
BYTE34 = 000042	BYTE86 = 000126	N = 000144	Q\$QLB = 000054	T\$MODE = 004000
BYTE35 = 000043	BYTE87 = 000127	PLB = 000010	Q\$QLR = 000001	T\$OB = 000036
BYTE36 = 000044	BYTE88 = 000130	PLC = 000020	Q\$QW = 000042	T\$OB2 = 004000
BYTE37 = 000045	BYTE89 = 000131	PLD = 000030	Q\$RDCD = 000005	T\$OBF = 010000
BYTE38 = 000046	BYTE9 = 000011	PLRW = 000200	Q\$RDMD = 000005	T\$OBRA = 000034
BYTE39 = 000047	BYTE90 = 000132	PLR.EN = 000200	Q\$REBK = 001000	T\$OBWA = 000032
BYTE4 = 000004	BYTE91 = 000133	PREADD = ***** GX	Q\$RNC = 006000	T\$OUTA = 100000
BYTE40 = 000050	BYTE92 = 000134	Q\$RCR1 = 176420	Q\$RSC = 004000	T\$RBD0 = 000200
BYTE41 = 000051	BYTE93 = 000135	Q\$RCR2 = 176422		T\$RNB = 000040

T#RSET= 000000	WORD2= 000004	WORD41= 000122	WORD62= 000174	WORD83= 000246
T#SC= 000022	WORD20= 000050	WORD42= 000124	WORD63= 000176	WORD84= 000250
T#SCLK= 020000	WORD21= 000052	WORD43= 000126	WORD64= 000200	WORD85= 000252
T#SEG1= 000000	WORD22= 000054	WORD44= 000130	WORD65= 000202	WORD86= 000254
T#SEG2= 000001	WORD23= 000056	WORD45= 000132	WORD66= 000204	WORD87= 000256
T#SEG3= 000002	WORD24= 000060	WORD46= 000134	WORD67= 000206	WORD88= 000260
T#S0= 000001	WORD25= 000062	WORD47= 000136	WORD68= 000210	WORD89= 000262
T#UBUS= 100000	WORD26= 000064	WORD48= 000140	WORD69= 000212	WORD9= 000022
T#1CLK= 000400	WORD27= 000066	WORD49= 000142	WORD7= 000016	WORD90= 000264
T#BBEN= 000020	WORD28= 000070	WORD5= 000012	WORD70= 000214	WORD91= 000266
T7MM= 000000RG	002 WORD29= 000072	WORD50= 000144	WORD71= 000216	WORD92= 000270
USD, IN= 000020	WORD3= 000006	WORD51= 000146	WORD72= 000220	WORD93= 000272
WORD0= 000000	WORD30= 000074	WORD52= 000150	WORD73= 000222	WORD94= 000274
WORD1= 000002	WORD31= 000076	WORD53= 000152	WORD74= 000224	WORD95= 000276
WORD10= 000024	WORD32= 000100	WORD54= 000154	WORD75= 000226	WORD96= 000300
WORD11= 000026	WORD33= 000102	WORD55= 000156	WORD76= 000230	WORD97= 000302
WORD12= 000030	WORD34= 000104	WORD56= 000160	WORD77= 000232	WORD98= 000304
WORD13= 000032	WORD35= 000106	WORD57= 000162	WORD78= 000234	WORD99= 000306
WORD14= 000034	WORD36= 000110	WORD58= 000164	WORD79= 000236	WORDVAL= 000310
WORD15= 000036	WORD37= 000112	WORD59= 000166	WORD8= 000020	WRITER= ***** GX
WORD16= 000040	WORD38= 000114	WORD6= 000014	WORD80= 000240	WRITER= ***** GX
WORD17= 000042	WORD39= 000116	WORD60= 000170	WORD81= 000242	XTREAD= 001000
WORD18= 000044	WORD4= 000010	WORD61= 000172	WORD82= 000244	XTWRITE= 000400
WORD19= 000046	WORD40= 000120			

. ABS: 000000 000
000000 001
MMTST3 000234 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3036 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:40
MMTST3,MMTST3/-SP=C 20,1 JIM,C 20,1 JMMTST3

```

1
2 000000 .TITLE MMTST4
3 .PSECT MMTST4
4 ;
5 ; MEMORY TEST DIAGNOSTICS
6 ; MATCH REPORT PROCESSOR MICROPROGRAM MEMORY
7 ;
8 ;
9 ;
10 ; TEST 12
11 ; LOOK FORWARD, LOOK BEHIND ADDRESSING TEST
12 ;
13 ;
14 ;
15 ; READ FROM TOP OF MEMORY DOWN, THEN WRITE
16 000000 TCMMD:
17 000000 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
18 000006 016746 000000G 1$: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
19 000012 CALL SEQMM ;DO IT
20 000016 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
21 000024 CALL CMPL ;CHECK LEFT HALF
22 000030 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
23 000034 CALL SEQMM ;WRITE SAME ADDR THAT WE READ
24 000040 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
25 000046 CALL WRITEL ;WRITE LEFT HALF OF MEMORY
26 000052 005267 000000G INC PREADD ;BUMP ADDRESS
27 000056 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
28 000064 103350 BHIS 1$ ;NO
29 ;
30 000066 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
31 000074 016746 000000G 2$: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
32 000100 CALL SEQMM ;DO IT
33 000104 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
34 000112 CALL CMPL ;CHECK RIGHT HALF
35 000116 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
36 000122 CALL SEQMM ;WRITE SAME ADDRESS THAT WE READ
37 000126 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
38 000134 CALL WRITER ;WRITE RIGHT HALF OF MEMORY
39 000140 005267 000000G INC PREADD ;BUMP ADDRESS
40 000144 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
41 000152 103350 BHIS 2$ ;NO
42 000154 RETURN
43 ;
44 ; TEST 12
45 ; READ FROM BOTTOM OF MEMORY UP, THEN WRITE
46 ;
47 000156 TCMMU:
48 000156 016667 000004 000000G MOV 4(SP),PREADD ;WORKING ADDRESS = END ADDRESS
49 000164 016746 000000G 1$: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
50 000170 CALL SEQMM ;DO IT
51 000174 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
52 000202 CALL CMPL ;CHECK LEFT HALF
53 000206 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
54 000212 CALL SEQMM ;WRITE SAME ADDR AS WE READ
55 000216 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
56 000224 CALL WRITEL ;WRITE LEFT HALF OF MEMORY
57 000230 162767 000001 000000G SUB #1,PREADD ;BACK UP

```

58	000236	026667	000002	000000G	CMP	2(SP),PREADD	:FINISHED?
59	000244	003747			BLE	1\$	
60							
61	000246	016667	000004	000000G	MOV	4(SP),PREADD	:WORKING ADDRESS = END ADDRESS
62	000254	016746	000000G	2\$:	MOV	PREADD, -(SP)	:SEQUENCE UP TO START ADDRESS
63	000260				CALL	SEQMM	:DO IT
64	000264	016767	000000G	000000G	MOV	CK2,CKDATA	:TEST PATTERN FOR READ
65	000272				CALL	CMR	:CHECK RIGHT HALF
66	000276	016746	000000G		MOV	PREADD, -(SP)	:SET SEQ ADDRESS FOR WRITE
67	000302				CALL	SEQMM	:WRITE SAME ADDR AS WE READ
68	000306	016767	000000G	000000G	MOV	CK3,CKDATA	:TEST PATTERN FOR WRITE
69	000314				CALL	WRITER	:WRITE RIGHT HALF OF MEMORY
70	000320	162767	000001	000000G	SUB	#1,PREADD	:BACK UP
71	000326	026667	000002	000000G	CMP	2(SP),PREADD	:FINISHED?
72	000334	003747			BLE	2\$	
73							
74	000336				RETURN		
75	000001				.END		

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR#CR1 = 176420	Q#RSC = 004000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR#CR2 = 176422	Q#RSET = 000010
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR#LBR = 176424	Q#SM = 100000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q#ATTN = 000100	Q#SP = 000120
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q#BCL = 000001	Q#SP2 = 000340
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q#CCCP = 000040	RGQ.EN = 000200
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q#CHB = 000400	RGQ.VA = 020000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#CHRL = 000200	SEQMM = ***** GX
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q#CLR = 000040	Q#CCLR = 000040
BIT13 = 020000	BYTE51 = 000063	CKDATA = ***** GX	Q#CNC = 030000	Q#CPC = 000060
BIT14 = 040000	BYTE52 = 000064	CK2 = ***** GX	Q#CPC2 = 000260	Q#CPCC = 000010
BIT15 = 100000	BYTE53 = 000065	CK3 = ***** GX	Q#CPC = 010000	Q#CSC = 010000
BIT2 = 000004	BYTE54 = 000066	CMPL = ***** GX	Q#CSEL = 000360	Q#CSET = 000002
BIT3 = 000010	BYTE55 = 000067	CMPR = ***** GX	Q#CSP = 020000	Q#DMA = 000001
BIT4 = 000020	BYTE56 = 000070	CNOBRE = 100000	Q#ENBK = 040000	Q#ENOP = 020000
BIT5 = 000040	BYTE57 = 000071	CPCCEN = 010000	Q#FAL = 004000	Q#FC = 000045
BIT6 = 000100	BYTE58 = 000072	CPREAD = 040000	Q#FB = 000044	Q#FP = 000046
BIT7 = 000200	BYTE59 = 000073	CPURTE = 020000	Q#HBF = 000002	Q#HCP = 000006
BIT8 = 000400	BYTE60 = 000074	CSADRD = 000004	Q#IHB = 000003	Q#IHR = 000002
BIT9 = 001000	BYTE61 = 000075	CSEQCI = 100000	Q#IMRP = 000007	Q#LBD = 001000
BYTE0 = 000000	BYTE62 = 000076	CSOE = 000040	Q#LBP = 001001	Q#LBDP = 001001
BYTE1 = 000001	BYTE63 = 000077	CSURTE = 000100	Q#LDC = 000003	Q#LDD = 000004
BYTE10 = 000012	BYTE64 = 000078	DBR.RD = 000001	Q#LDM = 000000	Q#LDP = 002000
BYTE11 = 000013	BYTE65 = 000079	DB#CPP = 001457	Q#LHP = 010000	Q#MNC = 140000
BYTE12 = 000014	BYTE66 = 000100	DB#SPT = 000026	Q#MR = 000052	Q#MRP = 000040
BYTE13 = 000015	BYTE67 = 000101	DB#TPC = 000023	Q#MRP2 = 000240	Q#MRP2 = 000240
BYTE14 = 000016	BYTE68 = 000102	DISPGS = 100000	Q#MSC = 040000	Q#MSET = 000004
BYTE15 = 000017	BYTE69 = 000103	DMAWR = 000005	Q#MSP = 100000	Q#NCL = 176000
BYTE16 = 000020	BYTE70 = 000104	DMARRD = 000003	Q#NCLK = 000100	Q#PPSW = 000320
BYTE17 = 000021	BYTE71 = 000105	DMARWR = 000004	Q#PP2 = 000300	Q#OHLT = 000013
BYTE18 = 000022	BYTE72 = 000107	ENBR = 010000	Q#QL = 000043	Q#QLA = 000053
BYTE19 = 000023	BYTE73 = 000110	LOC.EN = 000100	Q#QLB = 000054	Q#QLR = 000001
BYTE2 = 000002	BYTE74 = 000111	LOC.WA = 040000	Q#QL = 000042	Q#RD = 000005
BYTE20 = 000024	BYTE75 = 000112	LOC.WB = 100000	Q#RDCD = 000005	Q#RDMD = 000005
BYTE21 = 000025	BYTE76 = 000114	MAREN1 = 000001	Q#REBK = 001000	Q#RNC = 006000
BYTE22 = 000026	BYTE77 = 000115	MAREN2 = 004000	Q#RSC = 004000	
BYTE23 = 000027	BYTE78 = 000116	MARLOD = 010000		
BYTE24 = 000030	BYTE79 = 000117	MAROUT = 000002		
BYTE25 = 000031	BYTE80 = 000120	MAR.LO = 002000		
BYTE26 = 000032	BYTE81 = 000121	MAR.OU = 000040		
BYTE27 = 000033	BYTE82 = 000122	MBKALL = 001000		
BYTE28 = 000034	BYTE83 = 000123	MBKCLK = 000400		
BYTE29 = 000035	BYTE84 = 000124	MMADR = 000100		
BYTE3 = 000003	BYTE85 = 000125	MMLEFT = 000002		
BYTE30 = 000036	BYTE86 = 000126	MMOE = 000004		
BYTE31 = 000037	BYTE87 = 000127	MMWRTE = 000010		
BYTE32 = 000040	BYTE88 = 000130	MNOBRE = 100000		
BYTE33 = 000041	BYTE89 = 000131	MREN1 = 000001		
BYTE34 = 000042	BYTE90 = 000132	MREN2 = 020000		
BYTE35 = 000043	BYTE91 = 000133	MSYN = 000040		
BYTE36 = 000044	BYTE92 = 000134	N = 000144		
BYTE37 = 000045	BYTE93 = 000135	PLB = 000010		
BYTE38 = 000046		PLC = 000020		
BYTE39 = 000047		PLD = 000030		
BYTE4 = 000004		PLRW = 000200		
BYTE40 = 000050		PLR.EN = 000200		
BYTE41 = 000051		PREADD = ***** GX		

T#0BWA = 000032	WORD17 = 000042	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244
T#OUTA = 100000	WORD18 = 000044	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246
T#RBD0 = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T#RNB = 000040	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T#RSET = 040000	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T#SC = 000022	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T#SCLK = 020000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T#SEG1 = 000000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T#SEG2 = 000001	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T#SEG3 = 000002	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T#SO = 000001	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T#UBUS = 100000	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T#1CLK = 000400	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T#BBEN = 000020	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
UBD.IN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
WORD0 = 000000	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD1 = 000002	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD10 = 000024	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD11 = 000026	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD12 = 000030	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WORDVAL = 000310
WORD13 = 000032	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WRITEL = ***** GX
WORD14 = 000034	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRITER = ***** GX
WORD15 = 000036	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240	XTREAD = 001000
WORD16 = 000040	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242	XTWRITE = 000400

. ABS: 000000 000
000000 001
MMTST4 000340 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3051 WORDS (12 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
MMTST4,MMTST4,SP=C20,1JIM,C20,1JMMTST4

```

1 .TITLE - MMTST5
2 000000 .PSECT - MMTST5
3
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 000000 TDMM:
17 000000 016767 000000G 000000C MOV LMM,INDNB+N.FNAM ;PLACE FILE NAME INTO INPUT DNB
18 000000 016767 000002G 000000C MOV LMM+2,INDNB+N.FNAM+2
19 000014 042767 000000G 000000G BIC #FIRST,BASE ;CLEAR FIRST TIME THROUGH FLAG
20 000022 012767 000001 000002G MOV #1,VIRT+2 ;REINIT BLOCK COUNT
21 000030 OPENR #INFDB
22 ;
23 ;
24 ;
25 ;
26 ;
27 000046 005067 000000G CLR PREADD ;INIT SEQUENCER = 0
28 000052 1$: CALL GET ;READ A RECORD
29 000056 103515 BCS TDMMX ;ERROR EXIT
30 000060 016705 000000C MOV INFDB+F.BKDS+2,R5 ;POINT TO RECORD READ
31 000064 012704 000400 MOV #256,R4 ;NUMBER OF WORDS IN RECORD (MAX)
32 000070 032767 000000G 000000G BIT #FIRST,BASE ;FIRST TIME THROUGH
33 000076 001011 BNE 2$ ;NO
34 000100 052767 000000G 000000G BIS #FIRST,BASE ;SET FLAG FOR FIRST TIME THROUGH
35 000106 012567 000000G MOV (R5)+,LCOUNT ;SAVE NUMBER OF WORDS IN COLUMN
36 000112 016767 000000G 000000G MOV LCOUNT,WCOUNT ;INITIALIZE WORKING COUNTER
37 000120 005304 DEC R4 ;SUB FROM NUMBER OF WORDS IN RECORD
38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 ;
45 ;
46 ;
47 000122 2$: READ THE LEFT COLUMN OF MRP MICROPGM MEMORY
48 000122 012567 000000G MOV (R5)+,CKDATA ;SET TEST COUNTER WITH FILE WORD
49 000126 026766 000000G 000002C CMP PREADD,2(SP) ;AT LOWER MEMORY BOUND YET?
50 000134 103412 BLO 25$ ;SKIP COMPARE OF WORD
51 000136 026766 000000G 000004C CMP PREADD,4(SP) ;UPPER MEMORY BOUND EXCEEDED?
52 000144 101006 BHI 25$ ;SKIP COMPARE OF WORD
53 000146 016746 000000G MOV PREADD,-(SP) ;INITIALIZE SEQUENCER ADDRESS
54 000152 CALL SEQMM ;SET ADDRESS
55 000156 CALL CMLP ;COMPARE LEFT HALF OF MEMORY
56 000162 25$:
57 000162 005367 000000G DEC WCOUNT ;SUB FROM # WORDS IN A COLUMN

```

```

58 000166 001405          BEQ . . . . MRPRGT .          ; DO RIGHT-HAND COLUMN
59 000170 005267 000000G . INC . PREADD .          ; ADVANCE SEQUENCER ADDRESS
60 000174 005304          DEC . R4 .              ; FINISHED WITH THIS RECORD
61 000176 001725          BEQ . 1$ .              ; YES, GET NEXT
62 000200 000750          BR . 2$ .              ; NO, FETCH NEXT WORD
63                          ;
64                          ; READ THE RIGHT COLUMN OF MRP MICROPGM MEMORY
65                          ;
66 000202 .              MRPRGT:
67 000202 016767 000000G 000000G . MOV . LCOUNT, WCOUNT . ; REINITIALIZE WORKING COUNTER
68 000210 005067 000000G . CLR . PREADD .          ; INIT SEQUENCER = 0
69 000214 005304          1$: DEC . R4 .              ; FINISHED WITH THIS RECORD
70 000216 001007          BNE . 2$ .              ; NO, CONTINUE
71 000220          CALL . GET .              ; READ NEXT
72 000224 103432 .        BCS . TDMMX .          ; ERROR, EXIT
73
74 000226 016705 000000C . MOV . INFDB+F, BKDS+2, R5 . ; POINT TO RECORD READ
75 000232 012704 000400 . MOV . #256, R4 .          ; R4 = NUMBER OF WORDS IN RECORD
76                          ;
77 000236          2$:
78 000236 012567 000000G . MOV . (R5)+, CKDATA .          ; SET TEST COUNTER WITH FILE WORD
79 000242 026766 000000G 000002 . CMP . PREADD, 2(SP) .        ; AT LOWER MEMORY BOUND YET?
80 000250 103412 .        BLO . 25$ .              ; SKIP COMPARE OF WORD
81 000252 026766 000000G 000004 . CMP . PREADD, 4(SP) .        ; UPPER MEMORT BOUND EXCEEDED?
82 000260 101006 .        BHI . 25$ .              ; SKIP COMPARE OF WORD
83 000262 016746 000000G . MOV . PREADD, -(SP) .        ; INITIALIZE SEQUENCER ADDRESS
84 000266          CALL . SEQMM .          ; SET ADDRESS
85 000272          CALL . CMPR .          ; COMPARE RIGHT HALF OF MEMORY
86 000276          25$:
87 000276 005367 000000G . DEC . WCOUNT .          ; FINISHED WITH THIS COLUMN
88 000302 001403          BEQ . TDMMX .          ; YES, DONE
89 000304 005267 000000G . INC . PREADD .          ; NO, ADVANCE SEQUENCER ADDRESS
90 000310 000741          BR . 1$ .              ; SET IT
91                          ;
92 000312 .              TDMMX:
93 000312          CLOSE$ #INFDB .          ;
94 000322 105067 000000C . CLR . INDNB+H, FVER .        ; RESET FILE VERSION NUMBER
95 000326          RETURN .          ;
96 000001          .END .

```

ALUCKE = 040000	BYTE41 = 000051	BYTE93 = 000135	MRPRGT = 000202R	002 Q\$PPSW = 000320
ALUOE = 004000	BYTE42 = 000052	BYTE94 = 000136	MSYN = 000040	Q\$PP2 = 000300
A01 = 010000	BYTE43 = 000053	BYTE95 = 000137	N = 000144	Q\$QHLT = 000013
BASE = ***** GX	BYTE44 = 000054	BYTE96 = 000140	N.FNAM = ***** GX	Q\$QL = 000043
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	N.FVER = ***** GX	Q\$QLA = 000053
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PAR\$\$\$ = 000027	Q\$QLB = 000054
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLB = 000010	Q\$QLR = 000001
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PLC = 000020	Q\$QW = 000042
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PLD = 000030	Q\$RDCD = 000005
BIT12 = 010000	BYTE5 = 000005	CKDATA = ***** GX	PLRWR = 000200	Q\$RNDM = 000006
BIT13 = 020000	BYTE50 = 000062	CMPL = ***** GX	PLR.EN = 000200	Q\$REBK = 001000
BIT14 = 040000	BYTE51 = 000063	CMPR = ***** GX	PREADD = ***** GX	Q\$RNC = 006000
BIT15 = 100000	BYTE52 = 000064	CNOBRE = 100000	QR\$CR1 = 176420	Q\$RSC = 004000
BIT2 = 000004	BYTE53 = 000065	CPCCEN = 010000	QR\$CR2 = 176422	Q\$RSET = 000010
BIT3 = 000010	BYTE54 = 000066	CPREAD = 040000	QR\$LBR = 176424	Q\$SM = 100000
BIT4 = 000020	BYTE55 = 000067	CPURTE = 020000	Q\$ATTN = 000100	Q\$SP = 000120
BIT5 = 000040	BYTE56 = 000070	CSADRD = 000004	Q\$BCL = 000001	Q\$SP2 = 000340
BIT6 = 000100	BYTE57 = 000071	CSEQCI = 100000	Q\$CCCP = 000040	RGD.EN = 000200
BIT7 = 000200	BYTE58 = 000072	CSOE = 000040	Q\$CHB = 000400	RGD.VA = 020000
BIT8 = 000400	BYTE59 = 000073	CSURTE = 000100	Q\$CHRL = 000200	SEDM = ***** GX
BIT9 = 001000	BYTE6 = 000006	DBR.RD = 000001	Q\$CLR = 000040	SEQ.C1 = 000010
BYTE0 = 000000	BYTE60 = 000074	DB\$CPP = 001457	Q\$CNC = 030000	S\$CLR = 000000
BYTE1 = 000001	BYTE61 = 000075	DB\$SPT = 000026	Q\$CP = 000060	S\$LA = 000001
BYTE10 = 000012	BYTE62 = 000076	DB\$TPC = 000023	Q\$CPC = 000010	S\$OB = 000005
BYTE11 = 000013	BYTE63 = 000077	DISPGS = 100000	Q\$CP2 = 000260	S\$OR = 000006
BYTE12 = 000014	BYTE64 = 000100	DMAAWR = 000005	Q\$CSC = 010000	S\$OX = 000004
BYTE13 = 000015	BYTE65 = 000101	DMARRD = 000003	Q\$CSEL = 000360	S\$SR = 000007
BYTE14 = 000016	BYTE66 = 000102	DMARWR = 000004	Q\$CSET = 000002	S\$S1 = 000010
BYTE15 = 000017	BYTE67 = 000103	ENBR = 010000	Q\$CSP = 020000	S\$S2 = 000014
BYTE16 = 000020	BYTE68 = 000104	FIRST = ***** GX	Q\$DMA = 000001	TDMM = 000000RG 002
BYTE17 = 000021	BYTE69 = 000105	FO.RD = ***** GX	Q\$ENBK = 040000	TDMMK = 000312R 002
BYTE18 = 000022	BYTE7 = 000007	F.BKDS = ***** GX	Q\$ENOP = 020000	TD\$CTR = 176370
BYTE19 = 000023	BYTE70 = 000106	F.FACC = ***** GX	Q\$FAL = 004000	TD\$CTLW = 176360
BYTE2 = 000002	BYTE71 = 000107	GET = ***** GX	Q\$FC = 000045	TD\$INL = 004000
BYTE20 = 000024	BYTE72 = 000110	INDNB = ***** GX	Q\$FO = 000044	TD\$MEM = 000270
BYTE21 = 000025	BYTE73 = 000111	INFDB = ***** GX	Q\$FP = 000046	TD\$OAR = 176344
BYTE22 = 000026	BYTE74 = 000112	LCOUNT = ***** GX	Q\$HBF = 000002	TD\$OTR = 176346
BYTE23 = 000027	BYTE75 = 000113	LMM = ***** GX	Q\$ICP = 000006	TD\$ORD = 000274
BYTE24 = 000030	BYTE76 = 000114	LOC.EN = 000100	Q\$IBB = 000003	TD\$SW = 176376
BYTE25 = 000031	BYTE77 = 000115	LOC.WA = 040000	Q\$IHLR = 000002	TD\$TAR = 176372
BYTE26 = 000032	BYTE78 = 000116	LOC.WB = 000000	Q\$IMRP = 000007	TD\$TAW = 176362
BYTE27 = 000033	BYTE79 = 000117	MAREN1 = 000001	Q\$LBD = 001000	TD\$TDR = 176374
BYTE28 = 000034	BYTE8 = 000010	MAREN2 = 004000	Q\$LBDP = 001001	TD\$TDW = 176364
BYTE29 = 000035	BYTE80 = 000120	MARLOD = 010000	Q\$LBP = 000001	T\$AD = 000020
BYTE3 = 000003	BYTE81 = 000121	MAROUT = 000002	Q\$LCD = 000003	T\$BA = 000002
BYTE30 = 000036	BYTE82 = 000122	MAR.LO = 002000	Q\$LDM = 000004	T\$BD = 000010
BYTE31 = 000037	BYTE83 = 000123	MAR.DU = 000040	Q\$LDPP = 002000	T\$BSO = 100000
BYTE32 = 000040	BYTE84 = 000124	MBKALL = 001000	Q\$LHP = 010000	T\$BT = 000020
BYTE33 = 000041	BYTE85 = 000125	MBKCLK = 000400	Q\$MNC = 140000	T\$BTAR = 000030
BYTE34 = 000042	BYTE86 = 000126	MADRD = 000100	Q\$MR = 000052	T\$BDT = 002000
BYTE35 = 000043	BYTE87 = 000127	MMLEFT = 000002	Q\$MRP = 000040	T\$CD = 000100
BYTE36 = 000044	BYTE88 = 000130	MNOE = 000004	Q\$MRP2 = 000240	T\$CLK = 002000
BYTE37 = 000045	BYTE89 = 000131	MNWRTE = 000004	Q\$MSC = 040000	T\$DISK = 000200
BYTE38 = 000046	BYTE9 = 000011	MNOBRE = 100000	Q\$MSET = 000004	T\$DRD = 000004
BYTE39 = 000047	BYTE90 = 000132	MREN1 = 000001	Q\$MSP = 100000	T\$HEM = 010000
BYTE4 = 000004	BYTE91 = 000133	MREN2 = 020000	Q\$NCLK = 176000	T\$SAA = 000000
BYTE40 = 000050	BYTE92 = 000134		Q\$PP = 000100	T\$SAB = 000004

T\$FSAC = 000014	VIRT = ***** GX	WORD31 = 000076	WORD56 = 000160	WORD80 = 000240
T\$FSB2 = 000018	WCOUNT = ***** GX	WORD32 = 000100	WORD57 = 000162	WORD81 = 000242
T\$IB = 000026	WORD0 = 000000	WORD33 = 000102	WORD58 = 000164	WORD82 = 000244
T\$IBAR = 000024	WORD1 = 000002	WORD34 = 000104	WORD59 = 000166	WORD83 = 000246
T\$IBE = 020000	WORD10 = 000024	WORD35 = 000106	WORD6 = 000014	WORD84 = 000250
T\$IBF = 040000	WORD11 = 000026	WORD36 = 000110	WORD60 = 000170	WORD85 = 000252
T\$ICD = 000040	WORD12 = 000030	WORD37 = 000112	WORD61 = 000172	WORD86 = 000254
T\$MODE = 004000	WORD13 = 000032	WORD38 = 000114	WORD62 = 000174	WORD87 = 000256
T\$OB = 000036	WORD14 = 000034	WORD39 = 000116	WORD63 = 000176	WORD88 = 000260
T\$OBE = 004000	WORD15 = 000036	WORD4 = 000010	WORD64 = 000200	WORD89 = 000262
T\$OBF = 010000	WORD16 = 000040	WORD40 = 000120	WORD65 = 000202	WORD9 = 000022
T\$OBRA = 000034	WORD17 = 000042	WORD41 = 000122	WORD66 = 000204	WORD90 = 000264
T\$OBWA = 000032	WORD18 = 000044	WORD42 = 000124	WORD67 = 000206	WORD91 = 000266
T\$OUTA = 100000	WORD19 = 000046	WORD43 = 000126	WORD68 = 000210	WORD92 = 000270
T\$RBD0 = 000200	WORD2 = 000004	WORD44 = 000130	WORD69 = 000212	WORD93 = 000272
T\$RNB = 000040	WORD20 = 000050	WORD45 = 000132	WORD7 = 000016	WORD94 = 000274
T\$RSET = 040000	WORD21 = 000052	WORD46 = 000134	WORD70 = 000214	WORD95 = 000276
T\$SC = 000022	WORD22 = 000054	WORD47 = 000136	WORD71 = 000216	WORD96 = 000300
T\$SCLK = 020000	WORD23 = 000056	WORD48 = 000140	WORD72 = 000220	WORD97 = 000302
T\$SEG1 = 000000	WORD24 = 000060	WORD49 = 000142	WORD73 = 000222	WORD98 = 000304
T\$SEG2 = 000001	WORD25 = 000062	WORD5 = 000012	WORD74 = 000224	WORD99 = 000306
T\$SEG3 = 000002	WORD26 = 000064	WORD50 = 000144	WORD75 = 000226	WRDVAL = 000310
T\$SO = 000001	WORD27 = 000066	WORD51 = 000146	WORD76 = 000230	XTREAD = 001000
T\$UBUS = 100000	WORD28 = 000070	WORD52 = 000150	WORD77 = 000232	XTWRITE = 000400
T\$ICLK = 000400	WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	.CLOSE = *****
T\$OBEN = 000020	WORD3 = 000006	WORD54 = 000154	WORD79 = 000236	.OPEN = *****
UBD.IN = 000020	WORD30 = 000074	WORD55 = 000156	WORD8 = 000020	

. ABS. 000000 000
000000 001
MMTST5 000330 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3885 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:45
MMTST5,MMTST5/-SP=C20,1JIM,C20,1JMMTST5

```

1          .TITLE--MDTEST---
2 000000   .PSECT MDTEST
3          .MCALL WTSE$,CLEF$
4          ;
5          ;
6          ;
7          ;
8          ;
9          ;
10         ;
11         ;
12 000000   STUFMD::
13 000000
14 000004   016667 000002 000000G CALL RESET ;RESET MRP
15 000012   1$: CALL MOV 2(SP),PREADD ;WORKING ADDRESS
16 000016   005267 000000G CALL WMD ;WRITE 1 LOCATION
17 000022   026667 000004 000000G INC PREADD ;BUMP ADDRESS
18 000030   103370 CMP 4(SP),PREADD ;FINISHED?
19          BHIS 1$ ;NO
20 000032   005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
21 000034   012746 176000 MOV #0$NCLK,-(SP) ;SET NO-CLOCKS
22 000040 CALL CSR1
23 000044   005067 176422 CLR QR$CR2 ;SET LOAD MODE
24          ;
25 000050 CALL RESET ;RESET MRP
26 000054   016667 000002 000000G MOV 2(SP),PREADD ;RESET WORKING ADDRESS
27 000062   2$: CALL CMD ;READ AND COMPARE
28 000066   005267 000000G INC PREADD ;BUMP ADDRESS
29 000072   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
30 000100   103370 BHIS 2$ ;NO
31          ;
32 000102   005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
33 000104   012746 176000 MOV #0$NCLK,-(SP) ;SET NO-CLOCKS
34 000110 CALL CSR1
35 000114   005067 176422 CLR QR$CR2 ;SET LOAD MODE
36          ;
37 000120 RETURN

```

```

39 ;
40 ;
41 ;
42 ; TEST-01
43 ; WRITE MEMORY ADDRESS INTO MEMORY LOCATION
44 ;
45 ;
46 000122. TIMD::
47 000126. 016667 000002 000000G. CALL RESET ;RESET MRP
48 000134. 016667 000002 000000G. MOV 2(SP),PREADD ;WORKING ADDRESS
49 000142. 1$: CALL 2(SP),CKDATA ;SET TEST PATTERN = WORKING ADDRESS
50 000146. 005267 000000G. CALL WMD ;WRITE 1 LOCATION
51 000152. 005267 000000G. INC PREADD ;BUMP ADDRESS
52. 000156. 026667 000004 000000G. INC CKDATA ;BUMP TEST PATTERN
53 000164. 103366. CMP 4(SP),PREADD ;FINISHED?
54 ; BHS 1$ ;NO
55 000166. 005046 ;
56 000170. 012746 176000 CLR -(SP) ;CLEAR NOTHING IN CSR1
57 000174. CALL #0$NCLK, -(SP) ;SET NO-CLOCKS
58 000200. 005067 176422 CALL CSR1
59 ; CLR QR#CR2 ;SET LOAD MODE
60 000204 ;
61 000210. 016667 000002 000000G. CALL RESET ;RESET MRP
62. 000216. 016667 000002 000000G. MOV 2(SP),PREADD ;RESET WORKING ADDRESS
63 000224. 2$: CALL 2(SP),CKDATA ;RESET TEST PATTERN
64 000230. 005267 000000G. CALL CMD ;READ AND COMPARE
65 000234. 005267 000000G. INC PREADD ;BUMP ADDRESS
66 000240. 026667 000004 000000G. INC CKDATA ;BUMP TEST PATTERN
67 000246. 103366. CMP 4(SP),PREADD ;FINISHED?
68 ; BHS 2$ ;NO
69 000250. 005046 ;
70 000252. 012746 176000 CLR -(SP) ;CLEAR NOTHING IN CSR1
71 000256. CALL #0$NCLK, -(SP) ;SET NO-CLOCKS
72. 000262. 005067 176422 CALL CSR1
73 ; CLR QR#CR2 ;SET LOAD MODE
74 000266. RETURN

```



```

76 ;
77 ;
78 ; TEST-06
79 ; CROSS-TALK TEST
80 ;
81 ;
82 000270 ; T6MD:
83 000270 ;
84 000274 012767 177777 000000G CALL RESET ; RESET MRP
85 000302 012702 000012 MOV #1,CKDATA ; SET TEST PATTERN = X'FFFF'
86 000306 016667 000002 000000G 10$ MOV #10,R2 ; SET LOOP COUNT
87 000314 062767 000002 000000G 1$ CALL WMD ; WORKING ADDRESS
88 000320 026667 000004 000000G ADD #2,PREADD ; WRITE 1 LOCATION
89 000326 026667 000004 000000G CMP 4(SP),PREADD ; SKIP ONE ADDRESS
90 000334 103367 BHS 1$ ; FINISHED?
91 000336 005302 DEC R2 ; NO
92 000340 001362 BNE 10$ ; SUB FROM LOOP COUNT
93 ;
94 000342 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
95 000344 012746 176000 MOV #0$NCLK,-(SP) ; SET NO-CLOCKS
96 000350 CALL CSR1
97 000354 005067 176422 CLR QR$CR2 ; SET LOAD MODE
98 ;
99 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH 1'S
100 ; WERE NOT WRITTEN
101 ;
102 000360 ; R6Z:
103 000360 ;
104 000364 005067 000000G CALL RESET ; RESET MRP
105 000370 016667 000002 000000G CLR CKDATA ; SET TEST PATTERN
106 000376 005267 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
107 000402 062767 000002 000000G 1$ INC PREADD ; BUMP START ADDRESS
108 000406 026667 000004 000000G CALL CMD ; READ AND COMPARE 1 LOCATION
109 000414 026667 000004 000000G ADD #2,PREADD ; SKIP 1 LOCATION
110 000422 103367 CMP 4(SP),PREADD ; FINISHED?
111 ; BHS 1$ ; NO
112 000424 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
113 000426 012746 176000 MOV #0$NCLK,-(SP) ; SET NO-CLOCKS
114 000432 CALL CSR1
115 000436 005067 176422 CLR QR$CR2 ; SET LOAD MODE
116 ;
117 000442 RETURN

```

```

119      ;
120      ;
121      ;      TEST-07
122      ;      WRITE-COMPLEMENT-OF-MEMORY-ADDRESS-INTO-MEMORY-LOCATION
123      ;
124      ;
125 000444      T7MD::
126 000444
127 000450      016667      000002      000000G      CALL      RESET      ;RESET-MRP
128 000456      016602      000002      MOV      2(SP),PREADD ;WORKING-ADDRESS
129 000462      005102      1$:      MOV      2(SP),R2      ;TEST-PATTERN=-ADDRESS
130 000464      010267      000000G      COM      R2      ;GET-ADDRESS-COMPLEMENT
131 000470      MOV      R2,CKDATA ;SET-TEST-PATTERN
132 000474      005267      000000G      CALL      WMD      ;WRITE-MRP-DATA-MEMORY
133 000500      016702      000000G      INC      PREADD      ;BUMP-ADDRESS
134 000504      026667      000004      000000G      MOV      PREADD,R2 ;SET-UP-FOR-NEXT-TIME
135 000512      103363      CMP      4(SP),PREADD ;FINISHED-?
136      BHIS      1$      ;NO
137 000514      005046      CLR      -(SP)      ;CLEAR-NOTHING-IN-CSR1
138 000516      012746      176000      MOV      #0#NCLK,-(SP) ;SET-NO-CLOCKS
139 000522      CALL      CSR1
140 000526      005067      176422      CLR      QR$CR2      ;SET-LOAD-MODE
141
142 000532      CALL      RESET      ;RESET-MRP
143 000536      016667      000002      000000G      MOV      2(SP),PREADD ;WORKING-ADDRESS
144 000544      016602      000002      MOV      2(SP),R2      ;TEST-PATTERN=-ADDRESS
145 000550      005102      2$:      COM      R2      ;GET-ADDRESS-COMPLEMENT
146 000552      010267      000000G      MOV      R2,CKDATA ;SET-TEST-PATTERN
147 000556      CALL      CMD      ;READ-AND-COMPARE-MRP-DATA-MEMORY
148 000562      005267      000000G      INC      PREADD      ;BUMP-ADDRESS
149 000566      016702      000000G      MOV      PREADD,R2 ;SET-UP-FOR-NEXT-TIME
150 000572      026667      000004      000000G      CMP      4(SP),PREADD ;FINISHED-?
151 000600      103363      BHIS      2$      ;NO
152
153 000602      005046      CLR      -(SP)      ;CLEAR-NOTHING-IN-CSR1
154 000604      012746      176000      MOV      #0#NCLK,-(SP) ;SET-NO-CLOCKS
155 000610      CALL      CSR1
156 000614      005067      176422      CLR      QR$CR2      ;SET-LOAD-MODE
157
158 000620      RETURN

```

```

160 ;
161 ;
162 ; TEST-12.
163 ; LOOK FORWARD, LOOK BEHIND ADDRESSING TEST.
164 ;
165 ;
166 ; READ FROM TOP OF MEMORY DOWN, THEN WRITE.
167 ;
168 000622. TCMDU::
169 000622.
170 000626 016667 000002 000000G. CALL RESET ; RESET MRP
171 000634 016767 000000G 000000G 1$ MOV 2(SP),PREADD ; WORKING ADDRESS.
172 000642. CALL CK2,CKDATA ; TEST PATTERN FOR READ.
173 000646 016767 000000G 000000G. CALL CMD ; CHECK MEMORY LOCATION.
174 000654. MOV CK3,CKDATA ; TEST PATTERN FOR WRITE.
175 000660 005267 000000G. CALL WMD ; WRITE MRP DATA MEMORY.
176 000664 026667 000004 000000G. INC PREADD ; BUMP ADDRESS.
177 000672. 103360 CMP 4(SP),PREADD ; FINISHED?
178 ; BHIS 1$ ; NO.
179 000674 005046 ;
180 000676 012746 176000 CLR -(SP) ; CLEAR NOTHING IN CSR1
181 000702. MOV #0$NCLK,-(SP) ; SET NO-CLOCKS.
182 000706 005067 176422 CALL CSR1
183 ; CLR QR$CR2 ; SET LOAD MODE.
184 000712.
185 ; RETURN.
186 ;
187 ; TEST-12.
188 ; READ FROM BOTTOM OF MEMORY UP, THEN WRITE.
189 000714 TCMDU::
190 000714.
191 000720 016667 000004 000000G. CALL RESET ; RESET MRP
192 000726 016767 000000G 000000G 1$ MOV 4(SP),PREADD ; WORKING ADDRESS = END ADDRESS
193 000734. CALL CK2,CKDATA ; TEST PATTERN FOR READ.
194 000740 016767 000000G 000000G. CALL CMD ; CHECK MEMORY LOCATION.
195 000746. MOV CK3,CKDATA ; TEST PATTERN FOR WRITE.
196 000752. 162767 000001 000000G. CALL WMD ; WRITE MEMORY LOCATION.
197 000760 026667 000002 000000G. SUB #1,PREADD ; BACK UP 1
198 000766 003757 000000G. CMP 2(SP),PREADD ; FINISHED?
199 ; BLE 1$ ; NO.
200 000770 005046 ;
201 000772 012746 176000 CLR -(SP) ; CLEAR NOTHING IN CSR1
202 000776. MOV #0$NCLK,-(SP) ; SET NO-CLOCKS.
203 001002. 005067 176422 CALL CSR1
204 ; CLR QR$CR2 ; SET LOAD MODE.
205 001006. RETURN.

```

```

207 ;
208 ;
209 ;
210 ;
211 ;
212 001010 ; WMD:
213 001010 012767 000004 176424 MOV #Q$LDMD,QR$LBR ;MOVE ATTN CODE TO LOD BUS REG
214 001015 012767 120100 176422 MOV #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
215 001024 016701 176422 1$: QR$CR2,R1 ;READ CSR2
216 001030 032701 000100 BIT #Q$ATTN,R1 ;ATTN CLEAR
217 001034 001373 BNE 1$ ;NO, READ AGAIN
218 ;
219 001036 016767 000000 176424 MOV PREADD,QR$LBR ;CD MEMORY START ADDRESS
220 001044 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
221 001052 016701 176422 2$: QR$CR2,R1 ;READ CSR2
222 001056 032701 000040 BIT #Q$CCCP,R1 ;IS CC TO CP CLEAR
223 001062 001373 BNE 2$ ;NO, READ AGAIN
224 ;
225 ;
226 001064 012767 000001 176424 MOV #1,QR$LBR ;TRANSFER COUNT = 1 WORD
227 001072 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
228 001100 016701 176422 3$: QR$CR2,R1 ;READ CSR2
229 001104 032701 000040 BIT #Q$CCCP,R1 ;IS CC TO CP CLEAR
230 001110 001373 BNE 3$ ;NO, READ AGAIN
231 ;
232 001112 012767 000000 176424 MOV #CKDATA,QR$LBR ;CC MEMORY DATA BUFFER
233 001120 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
234 ;
235 ;
236 ;
237 001126 ; WTSE$S #EFN.3
238 ;
239 001140 ; GLEF$S #EFN.3
240 ;
241 ;
242 ; RE-ARM INTERRUPTS
243 001152 012767 100400 176422 MOV #<Q$SM+Q$CHB>,QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
244 001160 012767 101000 176422 MOV #<Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
245 001166 012767 160000 176422 MOV #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
246 001174 RETURN
    
```

```

248 ; READ AND COMPARE MRP DATA MEMORY
249 ;
250 CMD:
251 001176 012767 000006 176424 MOV #Q$RDMD,QR$LBR ;MOVE ATTN CODE TO LOD BUS REG
252 001204 012767 120100 176422 MOV #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
253 001212 016701 176422 1$: MOV QR$CR2,R1 ;READ CSR2
254 001216 032701 000100 BIT #Q$ATTN,R1 ;ATTN CLEAR
255 001222 001373 BNE 1$ ;NO, READ AGAIN
256 ;
257 001224 016767 000000G 176424 MOV PREADD,QR$LBR ;CD MEMORY START ADDRESS
258 001232 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
259 001240 016701 176422 2$: MOV QR$CR2,R1 ;READ CSR2
260 001244 032701 000040 BIT #Q$CCCP,R1 ;IS CC TO CP CLEAR
261 001250 001373 BNE 2$ ;NO, READ AGAIN
262 ;
263 ;
264 001252 012767 000001 176424 MOV #1,QR$LBR ;TRANSFER COUNT = 1 WORD
265 001260 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
266 001266 016701 176422 3$: MOV QR$CR2,R1 ;READ CSR2
267 001272 032701 000040 BIT #Q$CCCP,R1 ;IS CC TO CP CLEAR
268 001276 001373 BNE 3$ ;NO, READ AGAIN
269 ;
270 001300 012767 000000G 176424 MOV #ERW1,QR$LBR ;CC MEMORY DATA BUFFER
271 001306 012767 120040 176422 MOV #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
272 ;
273 ; WAIT FOR INTERRUPT FROM CP
274 ;
275 001314 WTSE#$ #EFN.3
276 ;
277 001326 CLEF#$ #EFN.3
278 ;
279 ; RE-ARM INTERRUPTS
280 ;
281 001340 012767 100400 176422 MOV #<Q$SM+Q$CHB>,QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
282 001346 012767 101000 176422 MOV #<Q$SM+Q$REBK>,QR$CR2 ;RE-ARM
283 001354 012767 160000 176422 MOV #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
284 ;
285 001362 026767 000000G 000000G CMP CKDATA,ERW1 ;SAME AS PATTERN WRITTEN
286 001370 001410 BEQ 4$ ;YES, EXIT
287 001372 016767 000000G 000000G MOV PREADD,ERRADD ;ADDRESS OF ERROR
288 001400 012767 000001 000000G MOV #1,ERRCT ;NUMBER OF WORDS TO PRINT
289 001406 CALL MEMERR ;GO TO ERROR ROUTINE
290 001412 4$: RETURN
291 ;

```

```

293 ; RESET-MRP...
294 ;
295 ;
296 001414 ; RESET:
297 001414 005046 CLR - (SP) ; CLEAR NOTHING
298 001416 012746 000004 MOV #Q$MSET, - (SP) ; MRP RESET
299 001422 CALL CSR1
300 001426 012746 000004 MOV #Q$MSET, - (SP) ; CLEAR RESET
301 001432 005046 CLR - (SP) ; SET NOTHING
302 001434 CALL CSR1
303 ;
304 001440 005046 100$: CLR - (SP) ; START MICROCODE AT 0
305 001442 CALL SEQCS
306 ;
307 001446 005046 CLR - (SP) ; REINHIBIT BRANCH CONTROL REGISTER
308 001450 CALL CPCR
309 ;
310 001454 012746 000377 MOV #377, - (SP) ; SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
311 001460 CALL SEQMM
312 ;
313 001464 005046 CLR - (SP) ; REINHIBIT BRANCH CONTROL REGISTER
314 001466 CALL MRPCR
315 ;
316 001472 012767 001000 176422 MOV #Q$REBK, Q$CR2 ; RE-ARM INTERRUPTS
317 001500 012767 120000 176422 MOV #Q$SM+Q$ENOP, Q$CR2 ; SET SEARCH MODE + ENABLE INTERRUPTS
318 001506 012746 000360 MOV #Q$CSEL, - (SP) ; CLEAR ALL SELECTIONS
319 001512 052716 001001 BIS #Q$LBD+Q$LBP, (SP) ; CLEAR DRIVE AND PULSE
320 001516 052716 170000 BIS #Q$CNC+Q$MNC, (SP) ; CLEAR MRP NO-CLOCK
321 001522 005046 CLR - (SP) ; SET NOTHING
322 001524 CALL CSR1
323 ;
324 001530 RETURN
325 ;
326 000001 .END
    
```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	N = 000144	Q\$QLB = 000054
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLB = 000010	Q\$QLR = 000001
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLC = 000020	Q\$QW = 000042
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLD = 000030	Q\$RDCC = 000005
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLRW = 000200	Q\$RDMD = 000006
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLREN = 000200	Q\$REBK = 001000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PREADD = ***** GX	Q\$RNC = 006000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$RCR1 = 176420	Q\$RSC = 004000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$RCR2 = 176422	Q\$RSET = 000010
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q\$ATLB = 176424	Q\$SM = 100000
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX	Q\$BLBR = 176424	Q\$SP = 000120
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT2 = 000004	BYTE53 = 000065	CHD = 001176R	002 Q\$CCCP = 000040	RESET = 001414R 002
BIT3 = 000010	BYTE54 = 000066	CNOBRE = 100000	Q\$CHB = 000400	RGD.EN = 000200
BIT4 = 000020	BYTE55 = 000067	CPCCEN = 010000	Q\$CHRL = 000200	RGD.VA = 020000
BIT5 = 000040	BYTE56 = 000070	CPCCR = ***** GX	Q\$CLR = 000040	R6Z = 000360R 002
BIT6 = 000100	BYTE57 = 000071	CPREAD = 040000	Q\$CNC = 030000	SEDCS = ***** GX
BIT7 = 000200	BYTE58 = 000072	CPURTE = 020000	Q\$CP = 000060	SEQM = ***** GX
BIT8 = 000400	BYTE59 = 000073	CSADRD = 000004	Q\$CPC = 000010	SEQ.CI = 000010
BIT9 = 001000	BYTE6 = 000006	CSEQCT = 100000	Q\$CP2 = 000260	STUFMD = 000000RG 002
BYTE0 = 000000	BYTE60 = 000074	CSDOE = 000040	Q\$CSC = 010000	S\$CLR = 000000
BYTE1 = 000001	BYTE61 = 000075	CSR1 = ***** GX	Q\$CSEL = 000360	S\$LA = 000001
BYTE10 = 000012	BYTE62 = 000076	CSURTE = 000100	Q\$CSET = 000002	S\$OB = 000005
BYTE11 = 000013	BYTE63 = 000077	DBR.RD = 000001	Q\$CSP = 020000	S\$OR = 000006
BYTE12 = 000014	BYTE64 = 000100	DB\$CPP = 001457	Q\$DMA = 000001	S\$OX = 000004
BYTE13 = 000015	BYTE65 = 000101	DB\$SPT = 000026	Q\$ENBK = 040000	S\$SR = 000007
BYTE14 = 000016	BYTE66 = 000102	DB\$TPC = 000023	Q\$ENOP = 020000	S\$S1 = 000010
BYTE15 = 000017	BYTE67 = 000103	DISPSS = 100000	Q\$FAL = 004000	S\$S2 = 000014
BYTE16 = 000020	BYTE68 = 000104	DMAUR = 000005	Q\$FC = 000045	TCMD = 000622RG 002
BYTE17 = 000021	BYTE69 = 000105	DMARRD = 000003	Q\$FO = 000044	TCMDU = 000714RG 002
BYTE18 = 000022	BYTE7 = 000007	DMARUR = 000004	Q\$FP = 000046	TD\$CTR = 176370
BYTE19 = 000023	BYTE70 = 000106	EFN.3 = ***** GX	Q\$HBF = 000002	TD\$TDW = 176360
BYTE2 = 000002	BYTE71 = 000107	ENBR = 010000	Q\$ICP = 000006	TD\$INL = 004000
BYTE20 = 000024	BYTE72 = 000110	ERRADD = ***** GX	Q\$IH0 = 000003	TD\$MEM = 000270
BYTE21 = 000025	BYTE73 = 000111	ERRCT = ***** GX	Q\$IHRL = 000002	TD\$OAR = 176344
BYTE22 = 000026	BYTE74 = 000112	ERW1 = ***** GX	Q\$IRP = 000007	TD\$OTR = 176346
BYTE23 = 000027	BYTE75 = 000113	LOC.EN = 000100	Q\$LBD = 001000	TD\$ORD = 000274
BYTE24 = 000030	BYTE76 = 000114	LOC.WA = 040000	Q\$LBDP = 001001	TD\$SW = 176376
BYTE25 = 000031	BYTE77 = 000115	LOC.WB = 100000	Q\$LBP = 000001	TD\$TAR = 176372
BYTE26 = 000032	BYTE78 = 000116	MAREN1 = 000001	Q\$LDCC = 000003	TD\$TAU = 176362
BYTE27 = 000033	BYTE79 = 000117	MAREN2 = 004000	Q\$LDMD = 000004	TD\$TDR = 176374
BYTE28 = 000034	BYTE8 = 000010	MARL0D = 010000	Q\$LDPP = 002000	TD\$TDW = 176364
BYTE29 = 000035	BYTE80 = 000120	MAROUT = 000002	Q\$LHP = 010000	T\$BD = 000020
BYTE3 = 000003	BYTE81 = 000121	MAR.LO = 002000	Q\$MNC = 140000	T\$BA = 000002
BYTE30 = 000036	BYTE82 = 000122	MAR.OU = 000040	Q\$MR = 000052	T\$BD = 000010
BYTE31 = 000037	BYTE83 = 000123	MBKALL = 001000	Q\$MRP = 000040	T\$BSO = 100000
BYTE32 = 000040	BYTE84 = 000124	MBKCLK = 000400	Q\$MRP2 = 000240	T\$BT = 000020
BYTE33 = 000041	BYTE85 = 000125	MEMERR = ***** GX	Q\$MSC = 040000	T\$BTAR = 000030
BYTE34 = 000042	BYTE86 = 000126	MNADR0 = 000100	Q\$MSET = 000004	T\$BTID = 002000
BYTE35 = 000043	BYTE87 = 000127	MNLEFT = 000002	Q\$MSP = 100000	T\$CD = 000100
BYTE36 = 000044	BYTE88 = 000130	MNDE = 000004	Q\$NCLK = 176000	T\$CLK = 002000
BYTE37 = 000045	BYTE89 = 000131	MNURTE = 000010	Q\$PP = 000100	T\$DLSK = 000200
BYTE38 = 000046	BYTE9 = 000011	MNOBRE = 100000	Q\$PP2 = 000320	T\$DRD = 000004
BYTE39 = 000047	BYTE90 = 000132	MREN1 = 000001	Q\$PP2 = 000300	T\$MEM = 010000
BYTE4 = 000004	BYTE91 = 000133	MREN2 = 020000	Q\$Q = 000043	T\$PSAA = 000000
BYTE40 = 000050	BYTE92 = 000134	MRPCR = ***** GX	Q\$QL = 000043	T\$PSAB = 000043
BYTE41 = 000051	BYTE93 = 000135	MSYN = 000040	Q\$QLA = 000053	T\$PSAC = 000014

T#FSB2 = 000010	T7MD = 000444RG	002.WORD30 = 000074	WORD55 = 000156	WORD79 = 000236
T#IB = 000026	UBD.IN = 000020	WORD31 = 000076	WORD56 = 000160	WORD8 = 000020
T#IBAR = 000024	WMD = 001010R	002.WORD32 = 000100	WORD57 = 000162	WORD80 = 000240
T#IBE = 020000	WORD0 = 000000	WORD33 = 000102	WORD58 = 000164	WORD81 = 000242
T#IBF = 040000	WORD1 = 000002	WORD34 = 000104	WORD59 = 000166	WORD82 = 000244
T#ICD = 000040	WORD10 = 000024	WORD35 = 000106	WORD6 = 000014	WORD83 = 000246
T#MODE = 004000	WORD11 = 000026	WORD36 = 000110	WORD60 = 000170	WORD84 = 000250
T#OB = 000036	WORD12 = 000030	WORD37 = 000112	WORD61 = 000172	WORD85 = 000252
T#OBE = 004000	WORD13 = 000032	WORD38 = 000114	WORD62 = 000174	WORD86 = 000254
T#OBF = 010000	WORD14 = 000034	WORD39 = 000116	WORD63 = 000176	WORD87 = 000256
T#OBRA = 000034	WORD15 = 000036	WORD4 = 000010	WORD64 = 000200	WORD88 = 000260
T#OBWA = 000032	WORD16 = 000040	WORD40 = 000120	WORD65 = 000202	WORD89 = 000262
T#OUTA = 100000	WORD17 = 000042	WORD41 = 000122	WORD66 = 000204	WORD9 = 000022
T#RBD0 = 000200	WORD18 = 000044	WORD42 = 000124	WORD67 = 000206	WORD90 = 000264
T#RNB = 000040	WORD19 = 000046	WORD43 = 000126	WORD68 = 000210	WORD91 = 000266
T#RSET = 040000	WORD2 = 000004	WORD44 = 000130	WORD69 = 000212	WORD92 = 000270
T#SC = 000022	WORD20 = 000050	WORD45 = 000132	WORD7 = 000016	WORD93 = 000272
T#SCLK = 020000	WORD21 = 000052	WORD46 = 000134	WORD70 = 000214	WORD94 = 000274
T#SEG1 = 000000	WORD22 = 000054	WORD47 = 000136	WORD71 = 000216	WORD95 = 000276
T#SEG2 = 000001	WORD23 = 000056	WORD48 = 000140	WORD72 = 000220	WORD96 = 000300
T#SEG3 = 000002	WORD24 = 000060	WORD49 = 000142	WORD73 = 000222	WORD97 = 000302
T#SO = 000001	WORD25 = 000062	WORDS = 000012	WORD74 = 000224	WORD98 = 000304
T#UBUS = 100000	WORD26 = 000064	WORD50 = 000144	WORD75 = 000226	WORD99 = 000306
T#iCLK = 000400	WORD27 = 000066	WORD51 = 000146	WORD76 = 000230	WRDVAL = 000310
T#BBEN = 000020	WORD28 = 000070	WORD52 = 000150	WORD77 = 000232	XTREAD = 001000
T1MD = 000122RG	002.WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	XTWRITE = 000400
T6MD = 000270RG	002.WORD3 = 000006	WORD54 = 000154		

. ABS: 000000 000
000000 001
MDTEST: 001532 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3425 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:49
MDTEST,MDTEST/-SP=C20,1JIM,C20,1JMDTEST


```

1
2 000000 .TITLE - CSTSUB
3 .PSECT - CSTSUB
4
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 000000 WRITE SECT A OF CP CONTROL STORE
13 000000 012746 000100 WRITEA:
14 000004 MOV #CSWRTE, -(SP) ; SECT A + WRITE ENABLE
15 000010 016746 000000G CALL CPCRA ; DIRECT CONTROL WORD TO CP
16 000014 MOV CKDATA, -(SP) ; TEST PATTERN
17 000020 005046 CALL LBCSC ; WRITE SECT A
18 000022 CLR -(SP) ; CLEAR CONTROL REG
19 000026 CALL CPCR
20 RETURN
21 ;
22 ;
23 ;
24 ;
25 000030 WRITE SECT B OF CP CONTROL STORE
26 000030 012746 000110 WRITEB:
27 000034 MOV #<CSWRTE+PLB>, -(SP) ; SECT B + WRITE ENABLE
28 000040 016746 000000G CALL CPCRA ; DIRECT CONTROL WORD TO CP
29 000044 MOV CKDATA, -(SP) ; TEST PATTERN
30 000050 005046 CALL LBCSC ; WRITE SECT B
31 000052 CLR -(SP) ; CLEAR CONTROL REG
32 000056 CALL CPCR
33 RETURN
34 ;
35 ;
36 ;
37 ;
38 000060 WRITE SECTION C OF CP CONTROL STORE
39 000060 012746 000120 WRITEC:
40 000064 MOV #<CSWRTE+PLC>, -(SP) ; SECT C + WRITE ENABLE
41 000070 016746 000000G CALL CPCRA ; DIRECT CONTROL WORD TO CP
42 000074 MOV CKDATA, -(SP) ; TEST PATTERN
43 000100 005046 CALL LBCSC ; WRITE SECT C
44 000102 CLR -(SP) ; CLEAR CONTROL REG
45 000106 CALL CPCR
46 RETURN
47 ;
48 ;
49 ;
50 ;
51 000110 WRITE SECTION D OF CP CONTROL STORE
52 000110 012746 000130 WRITED:
53 000114 MOV #<CSWRTE+PLD>, -(SP) ; SECT D + WRITE ENABLE
54 000120 016746 000000G CALL CPCRA ; DIRECT CONTROL WORD TO CP
55 000124 MOV CKDATA, -(SP) ; TEST PATTERN
56 000130 005046 CALL LBCSC ; WRITE SECT D
57 000132 CLR -(SP) ; CLEAR CONTROL REG
58 000136 CALL CPCR
59 RETURN

```

CSTSUB - MACRO - M1110 27-MAR-80 14:51 PAGE 5-1

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

58 000136

RETURN.

```

60 ;
61 ;
62 ; COMPARE SECT A OF CP CONTROL STORE
63 ;
64 ;
65 000140 CMPA::
66 000140 CALL UNLA ; READ SECT A WORD
67 000144 026767 000000G 000000G CMP CKDATA, ERW1 ; SAME AS WORD WRITTEN
68 000152 001435 BEQ 1$ ; YES
69 000154 112767 000101 000052G MOVB #'A, PRINT+42 ; INDICATE FAILURE ON SECT A
70 000162 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
71 000166 CALL SEQCS
72 000172 CALL UNLB ; READ SECT B WORD
73 000176 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
74 000202 CALL SEQCS
75 000206 CALL UNLC ; READ SECT C WORD
76 000212 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
77 000216 CALL SEQCS
78 000222 CALL UNLD ; READ SECT D WORD
79 000226 016767 000000G 000000G MOV PREADD, ERRADD ; SUPPLY ERROR ADDRESS
80 000234 012767 000004 000000G MOV #4, ERCT ; PRINT 4 WORDS
81 000242 CALL MEMERR ; PRINT ERROR MESSAGE
82 000246 1$: RETURN
83 ;
84 ;
85 ; COMPARE SECT B OF CP CONTROL STORE
86 ;
87 ;
88 000250 CMPBB::
89 000250 CALL UNLB ; READ SECT B WORD
90 000254 026767 000000G 000000G CMP CKDATA, ERW2 ; SAME AS WORD WRITTEN
91 000262 001435 BEQ 1$ ; YES
92 000264 112767 000102 000052G MOVB #'B, PRINT+42 ; INDICATE FAILURE ON SECT B
93 000272 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
94 000276 CALL SEQCS
95 000302 CALL UNLA ; READ SECT A WORD
96 000306 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
97 000312 CALL SEQCS
98 000316 CALL UNLC ; READ SECT C WORD
99 000322 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS
100 000326 CALL SEQCS
101 000332 CALL UNLD ; READ SECT D WORD
102 000336 016767 000000G 000000G MOV PREADD, ERRADD ; SUPPLY ERROR ADDRESS
103 000344 012767 000004 000000G MOV #4, ERCT ; PRINT 4 WORDS
104 000352 CALL MEMERR ; PRINT ERROR MESSAGE
105 000356 1$: RETURN
106 ;
107 ;
108 ; COMPARE SECT C OF CP CONTROL STORE
109 ;
110 ;
111 000360 CMPCC::
112 000360 CALL UNLC ; READ SECT C WORD
113 000364 026767 000000G 000000G CMP CKDATA, ERW3 ; SAME AS WORD WRITTEN
114 000372 001435 BEQ 1$ ; YES
115 000374 112767 000103 000052G MOVB #'C, PRINT+42 ; INDICATE FAILURE ON SECT C
116 000402 016746 000000G MOV PREADD, -(SP) ; SET SEQUENCER ADDRESS

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

117 000406          CALL·   SEQCS·
118 000412          CALL·   UNLA·
119 000416 016746 000000G·      MOV·   PREADD,-(SP)·      ;READ·SECT·A·WORD·
120 000422          CALL·   SEQCS·      ;SET·SEQUENCER·ADDRESS·
121 000426          CALL·   UNLB·
122 000432 016746 000000G·      MOV·   PREADD,-(SP)·      ;READ·SECT·B·WORD·
123 000436          CALL·   SEQCS·      ;SET·SEQUENCER·ADDRESS·
124 000442          CALL·   UNLD·
125 000446 016767 000000G·000000G·      MOV·   PREADD,ERRADD·      ;READ·SECT·D·WORD·
126 000454 012767 000004 000000G·      MOV·   #4,ERRCT·      ;SUPPLY·ERROR·ADDRESS·
127 000462          CALL·   MEMERR·      ;PRINT·4·WORDS·
128 000466          ;          ;PRINT·ERROR·MESSAGE
129          1$:
130          ;
131          ;
132          ;      COMPARE·SECT·D·OF·CP·CONTROL·STORE·
133          ;
134 000470          CMPDD::
135 000470          CALL·   UNLD·      ;READ·SECT·D·WORD·
136 000474 026767 000000G·000000G·      CMP·   CKDATA,ERW4·      ;SAME·AS·WORD·WRITTEN·
137 000502 001435          BEQ·   1$·      ;YES·
138 000504 112767 000104 000052G·      MOV·   #D,PRINT+42·      ;INDICATE·FAILURE·ON·SECT·D·
139 000512 016746 000000G·      MOV·   PREADD,-(SP)·      ;SET·SEQUENCER·ADDRESS·
140 000516          CALL·   SEQCS·
141 000522          CALL·   UNLA·
142 000526 016746 000000G·      MOV·   PREADD,-(SP)·      ;READ·SECT·A·WORD·
143 000532          CALL·   SEQCS·      ;SET·SEQUENCER·ADDRESS·
144 000536          CALL·   UNLB·
145 000542 016746 000000G·      MOV·   PREADD,-(SP)·      ;READ·SECT·B·WORD·
146 000546          CALL·   SEQCS·      ;SET·SEQUENCER·ADDRESS·
147 000552          CALL·   UNLC·
148 000556 016767 000000G·000000G·      MOV·   PREADD,ERRADD·      ;READ·SECT·C·WORD·
149 000564 012767 000004 000000G·      MOV·   #4,ERRCT·      ;SUPPLY·ERROR·ADDRESS·
150 000572          CALL·   MEMERR·      ;PRINT·4·WORDS·
151 000576          ;          ;PRINT·ERROR·MESSAGE
151          1$:
151          RETURN·

```

```

153 ;
154 ;
155 ; READ SECT A WORD FROM CP CONTROL STORE
156 ;
157 ;
158 UNLA::
159 000600 012746 000040 MOV #<CSOE>,-(SP) ;SET BR INHIBIT AND OUTPUT ENABLE
160 000604 CALL CPCR ;DIRECT CONTROL WORD TO CP
161 000610 005046 CLR -(SP)
162 000612 CALL LBCP ;CLOCK BR INHIBIT
163 000616 CALL CPLB ;REQUEST CP TO LOD BUS
164 000622 012667 000000G MOV (SP)+,ERU1 ;GET CP WORD
165 000626 005046 CLR -(SP) ;CLEAR CP CR
166 000630 CALL CPCR
167 000634 RETURN
168 ;
169 ;
170 ; READ SECT B WORD FROM CP CONTROL STORE
171 ;
172 ;
173 UNLB::
174 000636 012746 000050 MOV #<CSOE+PLB>,-(SP) ;SET BR INHIBIT AND OUTPUT ENABLE
175 000642 CALL CPCR ;DIRECT CONTROL WORD TO CP
176 000646 005046 CLR -(SP)
177 000650 CALL LBCP ;CLOCK BR INHIBIT
178 000654 CALL CPLB ;REQUEST CP TO LOD BUS
179 000660 012667 000000G MOV (SP)+,ERU2 ;GET CP WORD
180 000664 005046 CLR -(SP) ;CLEAR CP CR
181 000666 CALL CPCR
182 000672 RETURN
183 ;
184 ;
185 ; READ SECT C WORD FROM CP CONTROL STORE
186 ;
187 ;
188 UNLC::
189 000674 012746 000060 MOV #<CSOE+PLC>,-(SP) ;SET BR INHIBIT AND OUTPUT ENABLE
190 000700 CALL CPCR ;DIRECT CONTROL WORD TO CP
191 000704 005046 CLR -(SP)
192 000706 CALL LBCP ;CLOCK BR INHIBIT
193 000712 CALL CPLB ;REQUEST CP TO LOD BUS
194 000716 012667 000000G MOV (SP)+,ERU3 ;GET CP WORD
195 000722 005046 CLR -(SP) ;CLEAR CP CR
196 000724 CALL CPCR
197 000730 RETURN
198 ;
199 ;
200 ; READ SECT D WORD FROM CP CONTROL STORE
201 ;
202 ;
203 UNLD::
204 000732 012746 000070 MOV #<CSOE+PLD>,-(SP) ;SET BR INHIBIT AND OUTPUT ENABLE
205 000736 CALL CPCR ;DIRECT CONTROL WORD TO CP
206 000742 005046 CLR -(SP)
207 000744 CALL LBCP ;CLOCK BR INHIBIT
208 000750 CALL CPLB ;REQUEST CP TO LOD BUS
209 000754 012667 000000G MOV (SP)+,ERU4 ;GET CP WORD

```

```
210 000760 005046          CLR·      -(SP)          ;CLEAR·CP·CR·
211 000762          CALL·    CPCR
212 000766          RETURN·
213          ;
214          ;
215          ; SINGLE·CLOCK·SEQUENCER
216          ;
217          ;
218 000770          SINGLE:·
219 000770 012746 030000      MOV·      #0#CNC,-(SP)    ;CLEAR·CP·NO-CLOCKS·
220 000774 012746 010000      MOV·      #0#CSC,-(SP)    ;SINGLE·CLOCK·SEQ·
221 001000          CALL·    CSR1
222 001004 012746 010000      MOV·      #0#CSC,-(SP)    ;CLEAR·SINGLE·CLOCK·
223 001010 012746 030000      MOV·      #0#CNC,-(SP)    ;SET·NO-CLOCKS·
224 001014          CALL·    CSR1
225 001020          RETURN·
226          ;
227          000001          .END·
```

ALUCKE = 040000
ALUOE = 004000
A01 = 010000
BITVAL = 000000
BIT0 = 000001
BIT1 = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2 = 000004
BIT3 = 000010
BIT4 = 000020
BIT5 = 000040
BIT6 = 000100
BIT7 = 000200
BIT8 = 000400
BIT9 = 001000
BYTE0 = 000000
BYTE1 = 000001
BYTE10 = 000012
BYTE11 = 000013
BYTE12 = 000014
BYTE13 = 000015
BYTE14 = 000016
BYTE15 = 000017
BYTE16 = 000020
BYTE17 = 000021
BYTE18 = 000022
BYTE19 = 000023
BYTE2 = 000002
BYTE20 = 000024
BYTE21 = 000025
BYTE22 = 000026
BYTE23 = 000027
BYTE24 = 000030
BYTE25 = 000031
BYTE26 = 000032
BYTE27 = 000033
BYTE28 = 000034
BYTE29 = 000035
BYTE3 = 000003
BYTE30 = 000036
BYTE31 = 000037
BYTE32 = 000040
BYTE33 = 000041
BYTE34 = 000042
BYTE35 = 000043
BYTE36 = 000044
BYTE37 = 000045
BYTE38 = 000046
BYTE39 = 000047
BYTE4 = 000004
BYTE40 = 000050
BYTE41 = 000051
BYTE42 = 000052
BYTE43 = 000053
BYTE44 = 000054
BYTE45 = 000055
BYTE46 = 000056
BYTE47 = 000057
BYTE48 = 000060
BYTE49 = 000061
BYTE5 = 000005
BYTE50 = 000062
BYTE51 = 000063
BYTE52 = 000064
BYTE53 = 000065
BYTE54 = 000066
BYTE55 = 000067
BYTE56 = 000070
BYTE57 = 000071
BYTE58 = 000072
BYTE59 = 000073
BYTE6 = 000006
BYTE60 = 000074
BYTE61 = 000075
BYTE62 = 000076
BYTE63 = 000077
BYTE64 = 000100
BYTE65 = 000101
BYTE66 = 000102
BYTE67 = 000103
BYTE68 = 000104
BYTE69 = 000105
BYTE7 = 000007
BYTE70 = 000106
BYTE71 = 000107
BYTE72 = 000110
BYTE73 = 000111
BYTE74 = 000112
BYTE75 = 000113
BYTE76 = 000114
BYTE77 = 000115
BYTE78 = 000116
BYTE79 = 000117
BYTE8 = 000008
BYTE80 = 000120
BYTE81 = 000121
BYTE82 = 000122
BYTE83 = 000123
BYTE84 = 000124
BYTE85 = 000125
BYTE86 = 000126
BYTE87 = 000127
BYTE88 = 000130
BYTE89 = 000131
BYTE9 = 000009
BYTE90 = 000132
BYTE91 = 000133
BYTE92 = 000134
BYTE93 = 000135
BYTE94 = 000136
BYTE95 = 000137
BYTE96 = 000140
BYTE97 = 000141
BYTE98 = 000142
BYTE99 = 000143
BYTVAL = 000144
CBKALL = 001000
CBKCLK = 000400
CKDATA = ***** GX
CMPA = 000140RG
CMPBB = 000250RG
CMPCC = 000360RG
CMPDD = 000470RG
CNOBRE = 100000
CPCCEN = 010000
CPCR = ***** GX
CPCRA = ***** GX
CPLB = ***** GX
CPREAD = 040000
CPURTE = 020000
CSADRD = 000004
CSEQCI = 100000
CSOE = 000040
CSR1 = ***** GX
CSURTE = 000100
DBR RD = 000001
DBCPP = 001457
DBSPPT = 000026
DBTPC = 000023
DISPGS = 100000
DMAWR = 000005
DMARRD = 000003
DMARWR = 000004
ENBR = 010000
ERRADD = ***** GX
ERRCT = ***** GX
ERW1 = ***** GX
ERW2 = ***** GX
ERW3 = ***** GX
ERW4 = ***** GX
LBCP = ***** GX
LBCSC = ***** GX
LOC.EN = 000100
LOC.WA = 040000
LOC.WB = 100000
MAREN1 = 000001
MAREN2 = 004000
MARL0D = 010000
MAROUT = 000002
MAR.LO = 002000
MAR.OU = 000040
MAY9 = 000011
MBKALL = 001000
MBKCLK = 000400
MEMERR = ***** GX
MMADR0 = 000100
MMLEFT = 000002
MMOE = 000004
MMWRTE = 000010
MNOBRE = 100000
MREN1 = 000001
MREN2 = 020000
MSYN = 000040
N = 000144
PLB = 000010
PLC = 000020
PLD = 000030
002.PLRWR = 000200
002.PLR.EN = 000200
002.PREADD = ***** GX
002.PRINT = ***** GX
OR\$CR1 = 176420
OR\$CR2 = 176422
OR\$LBR = 176424
Q\$ATTN = 000100
Q\$BCL = 000001
Q\$CCCP = 000040
Q\$CHB = 000400
Q\$CHRL = 000200
Q\$CLR = 000040
Q\$CNC = 030000
Q\$CP = 000060
Q\$CPC = 000010
Q\$CP2 = 000260
Q\$CSC = 010000
Q\$CSEL = 000360
Q\$CSET = 000002
Q\$CSP = 020000
Q\$DMA = 000001
Q\$ENBK = 040000
Q\$ENOP = 020000
Q\$FAL = 004000
Q\$FO = 000044
Q\$FP = 000046
Q\$HBF = 000002
Q\$ICP = 000006
Q\$IH8 = 000003
Q\$IHRL = 000002
Q\$IHRP = 000007
Q\$LBD = 001000
Q\$LBDP = 001001
Q\$LBP = 000001
Q\$LDCD = 000003
Q\$LDMD = 000004
Q\$LDPP = 002000
Q\$LHP = 010000
Q\$INC = 140000
Q\$MR = 000052
Q\$MRP = 000040
Q\$MRP2 = 000240
Q\$M\$C = 040000
Q\$MSET = 000004
Q\$MSP = 100000
Q\$NCLK = 176000
Q\$PP = 000100
Q\$PPSW = 000320
Q\$PP2 = 000300
Q\$QHLT = 000013
Q\$QL = 000043
Q\$QLA = 000053
Q\$QLB = 000054
Q\$QLR = 000001
Q\$QW = 000042
Q\$RDCD = 000005
Q\$RDMD = 000006
Q\$REBK = 001000
Q\$RNC = 006000
Q\$RSC = 004000
Q\$RSET = 000010
Q\$SM = 100000
Q\$SP = 000120
Q\$SP2 = 000340
RGQ.EN = 000200
RGQ.VA = 020000
SECCS = ***** GX
SEQ.CI = 000010
SINGLE = 00070RG 002
S\$CLR = 000000
S\$LA = 000001
S\$QB = 000005
S\$QR = 000006
S\$QX = 000004
S\$SR = 000007
S\$S1 = 000010
S\$S2 = 000014
TD\$CTR = 176370
TD\$CTW = 176360
TD\$INL = 004000
TD\$MEM = 000270
TD\$QAR = 176344
TD\$QTR = 176346
TD\$QAR = 000274
TD\$SW = 176376
TD\$TAR = 176372
TD\$TAW = 176362
TD\$TDR = 176374
TD\$TDW = 176364
T\$D = 000020
T\$BA = 000002
T\$BD = 000010
T\$BSO = 100000
T\$BT = 000020
T\$BTAR = 000030
T\$BTID = 002000
T\$CD = 000100
T\$CLK = 002000
T\$DISK = 000200
T\$DRD = 000004
T\$EMEM = 010000
T\$FSAA = 000000

T#FSAB = 000004	UNLA = 000600RG	002	WORD30 = 000074	WORD56 = 000160	WORD81 = 000242
T#FSAC = 000014	UNLB = 000636RG	002	WORD31 = 000076	WORD57 = 000162	WORD82 = 000244
T#FSB2 = 000010	UNLC = 000674RG	002	WORD32 = 000100	WORD58 = 000164	WORD83 = 000246
T#IB = 000026	UNLD = 000732RG	002	WORD33 = 000102	WORD59 = 000166	WORD84 = 000250
T#IBAR = 000024	WORD0 = 000000		WORD34 = 000104	WORD6 = 000014	WORD85 = 000252
T#IBE = 020000	WORD1 = 000002		WORD35 = 000106	WORD60 = 000170	WORD86 = 000254
T#IBF = 040000	WORD10 = 000024		WORD36 = 000110	WORD61 = 000172	WORD87 = 000256
T#ICD = 000040	WORD11 = 000026		WORD37 = 000112	WORD62 = 000174	WORD88 = 000260
T#MODE = 004000	WORD12 = 000030		WORD38 = 000114	WORD63 = 000176	WORD89 = 000262
T#OB = 000036	WORD13 = 000032		WORD39 = 000116	WORD64 = 000200	WORD9 = 000022
T#OBE = 004000	WORD14 = 000034		WORD4 = 000010	WORD65 = 000202	WORD90 = 000264
T#OBF = 010000	WORD15 = 000036		WORD40 = 000120	WORD66 = 000204	WORD91 = 000266
T#OBRA = 000034	WORD16 = 000040		WORD41 = 000122	WORD67 = 000206	WORD92 = 000270
T#OBWA = 000032	WORD17 = 000042		WORD42 = 000124	WORD68 = 000210	WORD93 = 000272
T#OUTA = 100000	WORD18 = 000044		WORD43 = 000126	WORD69 = 000212	WORD94 = 000274
T#RBD0 = 000200	WORD19 = 000046		WORD44 = 000130	WORD7 = 000016	WORD95 = 000276
T#RNB = 000040	WORD2 = 000004		WORD45 = 000132	WORD70 = 000214	WORD96 = 000300
T#RSET = 040000	WORD20 = 000050		WORD46 = 000134	WORD71 = 000216	WORD97 = 000302
T#SC = 000022	WORD21 = 000052		WORD47 = 000136	WORD72 = 000220	WORD98 = 000304
T#SCLK = 020000	WORD22 = 000054		WORD48 = 000140	WORD73 = 000222	WORD99 = 000306
T#SEG1 = 000000	WORD23 = 000056		WORD49 = 000142	WORD74 = 000224	WRDVAL = 000310
T#SEG2 = 000001	WORD24 = 000060		WORD5 = 000012	WORD75 = 000226	WRITEA = 000000RG 002
T#SEG3 = 000002	WORD25 = 000062		WORD50 = 000144	WORD76 = 000230	WRITEB = 000030RG 002
T#S0 = 000001	WORD26 = 000064		WORD51 = 000146	WORD77 = 000232	WRITEC = 000060RG 002
T#UBUS = 100000	WORD27 = 000066		WORD52 = 000150	WORD78 = 000234	WRITED = 000100RG 002
T#1CLK = 000400	WORD28 = 000070		WORD53 = 000152	WORD79 = 000236	XTREAD = 001000
T#BBEN = 000020	WORD29 = 000072		WORD54 = 000154	WORD8 = 000020	XTWRITE = 000400
UBD.IN = 000020	WORD3 = 000006		WORD55 = 000156	WORD80 = 000240	

. ABS. 000000 000
000000 001
CSTSUB: 001022 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3150 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:46
CSTSUB: CSTSUB / -SP=C20.1 JIM, [20.1] CSTSUB.


```

1          .TITLE .CSTST0
2 000000   .PSECT .CSTST0
3
4          ;
5          ;   HARDWARE QUERY RESOLVER MEMORY DIAGNOSTICS
6          ;   CONTROL PROCESSOR CONTROL STORE
7          ;
8 000000   STUFCS:
9 000000   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
10 000006   1$:
11 000006   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
12 000012   CALL SEQCS ;DO IT
13 000016   CALL WRITEB ;WRITE SECT A OF MEMORY
14 000022   005267 000000G INC PREADD ;BUMP ADDRESS
15 000026   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
16 000034   103364 BHIS 1$ ;NO
17          ;
18 000036   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
19 000044   2$:
20 000044   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
21 000050   CALL SEQCS
22 000054   CALL WRITEB ;WRITE SECT B OF MEMORY
23 000060   005267 000000G INC PREADD ;BUMP ADDRESS
24 000064   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
25 000072   103364 BHIS 2$ ;NO
26          ;
27 000074   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
28 000102   3$:
29 000102   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
30 000106   CALL SEQCS
31 000112   CALL WRITEB ;WRITE SECT C OF MEMORY
32 000116   005267 000000G INC PREADD ;BUMP ADDRESS
33 000122   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
34 000130   103364 BHIS 3$ ;NO
35          ;
36 000132   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
37 000140   4$:
38 000140   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
39 000144   CALL SEQCS
40 000150   CALL WRITEB ;WRITE SECT D OF MEMORY
41 000154   005267 000000G INC PREADD ;BUMP ADDRESS
42 000160   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
43 000166   103364 BHIS 4$ ;NO
44          ;
45 000170   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
46 000176   5$:
47 000176   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
48 000202   CALL SEQCS ;DO IT
49 000206   CALL CMPA ;COMPARE SECT A OF MEMORY
50 000212   005267 000000G INC PREADD ;BUMP ADDRESS
51 000216   026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
52 000224   103364 BHIS 5$
53          ;
54 000226   016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
55 000234   6$:
56 000234   016746 000000G MOV PREADD, -(SP) ;SEQ UP TO START ADDRESS
57 000240   CALL SEQCS ;DO IT

```

58	000244			CALL	CMPBB		; COMPARE SECT B OF MEMORY
59	000250	005267	000000G	INC	PREADD		; BUMP ADDRESS
60	000254	026667	000004 000000G	CMP	4(SP),PREADD		; FINISHED ?
61	000262	103364		BHIS	6\$		
62							
63	000264	016667	000002 000000G	MOV	2(SP),PREADD		; WORKING ADDRESS
64	000272					7\$:	
65	000272	016746	000000G	MOV	PREADD, -(SP)		; SEQ UP TO START ADDRESS
66	000276			CALL	SEQCS		; DO IT
67	000302			CALL	CMPC		; COMPARE SECT C OF MEMORY
68	000306	005267	000000G	INC	PREADD		; BUMP ADDRESS
69	000312	026667	000004 000000G	CMP	4(SP),PREADD		; FINISHED ?
70	000320	103364		BHIS	7\$		
71							
72	000322	016667	000002 000000G	MOV	2(SP),PREADD		; WORKING ADDRESS
73	000330					8\$:	
74	000330	016746	000000G	MOV	PREADD, -(SP)		; SEQ UP TO START ADDRESS
75	000334			CALL	SEQCS		; DO IT
76	000340			CALL	CMPD		; COMPARE SECT D OF MEMORY
77	000344	005267	000000G	INC	PREADD		; BUMP ADDRESS
78	000350	026667	000004 000000G	CMP	4(SP),PREADD		; FINISHED ?
79	000356	103364		BHIS	8\$		
80							
81	000360	005046		CLR	-(SP)		; CLEAR CONTROL REG
82	000362			CALL	CPCR		
83	000366			RETURN			
84		000001		.END			

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR\$CR1 = 176420	Q\$RSC = 004000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR\$CR2 = 176422	Q\$RSET = 000010
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR\$LBR = 176424	Q\$SM = 100000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$ATTN = 000100	Q\$SP = 000120
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$BCL = 000001	Q\$SP2 = 000340
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CCCP = 000040	RGQ,EN = 000200
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CHB = 000400	RGQ,VA = 020000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CHRL = 000200	SEQCS = ***** GX
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CLR = 000040	SEQ,CI = 000010
BIT13 = 020000	BYTE50 = 000062	CMPA = ***** GX	Q\$CNC = 030000	STUFCS = 000000RG 002
BIT14 = 040000	BYTE51 = 000063	CMPBB = ***** GX	Q\$CP = 000060	S\$CLR = 000000
BIT15 = 100000	BYTE52 = 000064	CMPCC = ***** GX	Q\$CPC = 000010	S\$LA = 000001
BIT2 = 000004	BYTE53 = 000065	CMPDD = ***** GX	Q\$CP2 = 000260	S\$QB = 000005
BIT3 = 000010	BYTE54 = 000066	CHOBRE = 100000	Q\$CSC = 010000	S\$QR = 000006
BIT4 = 000020	BYTE55 = 000067	CPCCEN = 010000	Q\$CSEL = 000360	S\$QX = 000004
BIT5 = 000040	BYTE56 = 000070	CPCR = ***** GX	Q\$CSET = 000002	S\$SR = 000007
BIT6 = 000100	BYTE57 = 000071	CPREAD = 040000	Q\$CSP = 020000	S\$S1 = 000010
BIT7 = 000200	BYTE58 = 000072	CPURTE = 020000	Q\$DMA = 000001	S\$S2 = 000014
BIT8 = 000400	BYTE59 = 000073	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTR = 176370
BIT9 = 001000	BYTE6 = 000006	CSEQCI = 100000	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE0 = 000000	BYTE60 = 000074	C\$OE = 000040	Q\$FAL = 004000	TD\$INL = 004000
BYTE1 = 000001	BYTE61 = 000075	CSURTE = 000100	Q\$FC = 000045	TD\$MEM = 000270
BYTE10 = 000012	BYTE62 = 000076	DBR, RD = 000001	Q\$FO = 000044	TD\$OAR = 176344
BYTE11 = 000013	BYTE63 = 000077	DB\$CPP = 001457	Q\$FP = 000046	TD\$QTR = 176346
BYTE12 = 000014	BYTE64 = 000100	DB\$SPT = 000026	Q\$HBF = 000002	TD\$GRD = 000274
BYTE13 = 000015	BYTE65 = 000101	DB\$TPC = 000023	Q\$ICP = 000006	TD\$SW = 176376
BYTE14 = 000016	BYTE66 = 000102	DISPCS = 100000	Q\$IHB = 000003	TD\$TAR = 176372
BYTE15 = 000017	BYTE67 = 000103	DMARWR = 000005	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q\$LBD = 001000	TD\$TDW = 176364
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q\$LBDP = 001001	T\$AD = 000020
BYTE19 = 000023	BYTE70 = 000106	LOC,EN = 000100	Q\$LBP = 000001	T\$BA = 000002
BYTE2 = 000002	BYTE71 = 000107	LOC,WA = 040000	Q\$LCD = 000003	T\$BD = 000010
BYTE20 = 000024	BYTE72 = 000110	LOC,WB = 100000	Q\$LDMD = 000004	T\$BSO = 100000
BYTE21 = 000025	BYTE73 = 000111	MAREN1 = 000001	Q\$LDPP = 002000	T\$BT = 000020
BYTE22 = 000026	BYTE74 = 000112	MAREN2 = 004000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE23 = 000027	BYTE75 = 000113	MARLOD = 010000	Q\$MNC = 140000	T\$BTD = 002000
BYTE24 = 000030	BYTE76 = 000114	MAROUT = 000002	Q\$MR = 000052	T\$CD = 000100
BYTE25 = 000031	BYTE77 = 000115	MAR,LO = 002000	Q\$MRP = 000040	T\$CLK = 002000
BYTE26 = 000032	BYTE78 = 000116	MAR,OU = 000040	Q\$MRP2 = 000240	T\$DISK = 000200
BYTE27 = 000033	BYTE79 = 000117	MBKALL = 001000	Q\$MSC = 040000	T\$DRD = 000004
BYTE28 = 000034	BYTE8 = 000010	MBKCLK = 000400	Q\$MSET = 000004	T\$EMEN = 010000
BYTE29 = 000035	BYTE80 = 000120	MNARRD = 000100	Q\$MSP = 100000	T\$FSAA = 000000
BYTE3 = 000003	BYTE81 = 000121	MNLEFT = 000002	Q\$MNCLK = 176000	T\$FSAC = 000004
BYTE30 = 000036	BYTE82 = 000122	MNDE = 000004	Q\$PP = 000100	T\$FSAC = 000014
BYTE31 = 000037	BYTE83 = 000123	MNURTE = 000010	Q\$PPSW = 000320	T\$FSB2 = 000010
BYTE32 = 000040	BYTE84 = 000124	MNOBRE = 100000	Q\$PP2 = 000300	T\$IB = 000026
BYTE33 = 000041	BYTE85 = 000125	MREN1 = 000001	Q\$QHLT = 000013	T\$IBAR = 000024
BYTE34 = 000042	BYTE86 = 000126	MREN2 = 020000	Q\$QL = 000043	T\$IBE = 020000
BYTE35 = 000043	BYTE87 = 000127	MSYN = 000040	Q\$QLA = 000053	T\$IBF = 040000
BYTE36 = 000044	BYTE88 = 000130	N = 000144	Q\$QLB = 000054	T\$ICD = 000040
BYTE37 = 000045	BYTE89 = 000131	PLB = 000020	Q\$QLR = 000001	T\$MODE = 004000
BYTE38 = 000046	BYTE9 = 000011	PLC = 000010	Q\$QW = 000042	T\$OB = 000035
BYTE39 = 000047	BYTE90 = 000132	PLD = 000030	Q\$RCD = 000005	T\$OBF = 004000
BYTE4 = 000004	BYTE91 = 000133	PLRWR = 000200	Q\$REBK = 001000	T\$OBRA = 000034
BYTE40 = 000050	BYTE92 = 000134	PLR,EN = 000200	Q\$RNC = 006000	T\$OBWA = 000032
BYTE41 = 000051	BYTE93 = 000135	PREADD = ***** GX		

T#OUTA = 100000	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T#RBDO = 000200	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T#RNB = 000040	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T#RSET = 040000	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T#SC = 000022	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T#SCLK = 020000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T#SEG1 = 000000	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T#SEG2 = 000001	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T#SEG3 = 000002	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T#SO = 000001	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T#UBUS = 100000	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T#1CLK = 000400	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T#BBEN = 000020	WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
UBD.IN = 000020	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD0 = 000000	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD1 = 000002	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD10 = 000024	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD11 = 000026	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WRDVAL = 000310
WORD12 = 000030	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WRITEA = ***** GX
WORD13 = 000032	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRITEB = ***** GX
WORD14 = 000034	WORD37 = 000112	WORD59 = 000166	WORD80 = 000240	WRITEC = ***** GX
WORD15 = 000036	WORD38 = 000114	WORD6 = 000014	WORD81 = 000242	WRITED = ***** GX
WORD16 = 000040	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244	XTREAD = 001000
WORD17 = 000042	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246	XTWRITE = 000400
WORD18 = 000044				

. ABS. 000000 000
000000 001
CSTST0 000370 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3080 WORDS (.13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (.14 PAGES)
ELAPSED TIME: 00:00:41
CSTST0,CSTST0/-SP=C20.1]IM,C20.1]CSTST0

```

1
2 000000 .TITLE: CSTST1
3 .PSECT: CSTST1
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 000000 TIGS::
12 000000 016667 000002 000000G MOV 2(SP),CKDATA ; START ADDRESS = TEST COUNTER
13 000000 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
14 000014 1$:
15 000014 016746 000000G MOV PREADD, -(SP) ; SEQ UP TO START ADDRESS
16 000020 CALL SEQCS ; DO IT
17 000024 CALL WRITEA ; WRITE SECT A OF MEMORY
18 000030 005267 000000G INC CKDATA ; BUMP TEST COUNTER
19 000034 005267 000000G INC PREADD ; BUMP ADDRESS
20 000040 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED?
21 000046 103362 BHIS 1$ ; NO
22 ;
23 000050 016667 000002 000000G MOV 2(SP),CKDATA ; START ADDRESS = TEST COUNTER
24 000056 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
25 000064 2$:
26 000064 016746 000000G MOV PREADD, -(SP) ; SEQ UP TO START ADDRESS
27 000070 CALL SEQCS
28 000074 CALL WRITEB ; WRITE SECT B OF MEMORY
29 000100 005267 000000G INC CKDATA ; BUMP TEST COUNTER
30 000104 005267 000000G INC PREADD ; BUMP ADDRESS
31 000110 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED?
32 000116 103362 BHIS 2$ ; NO
33 ;
34 000120 016667 000002 000000G MOV 2(SP),CKDATA ; START ADDRESS = TEST COUNTER
35 000126 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
36 000134 3$:
37 000134 016746 000000G MOV PREADD, -(SP) ; SEQ UP TO START ADDRESS
38 000140 CALL SEQCS
39 000144 CALL WRITEC ; WRITE SECT C OF MEMORY
40 000150 005267 000000G INC CKDATA ; BUMP TEST COUNTER
41 000154 005267 000000G INC PREADD ; BUMP ADDRESS
42 000160 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED?
43 000166 103362 BHIS 3$ ; NO
44 ;
45 000170 016667 000002 000000G MOV 2(SP),CKDATA ; START ADDRESS = TEST COUNTER
46 000176 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
47 000204 4$:
48 000204 016746 000000G MOV PREADD, -(SP) ; SEQ UP TO START ADDRESS
49 000210 CALL SEQCS
50 000214 CALL WRITED ; WRITE SECT D OF MEMORY
51 000220 005267 000000G INC CKDATA ; BUMP TEST COUNTER
52 000224 005267 000000G INC PREADD ; BUMP ADDRESS
53 000230 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED?
54 000236 103362 BHIS 4$ ; NO
55 ;
56 000240 016667 000002 000000G MOV 2(SP),CKDATA ; START ADDRESS = TEST COUNTER
57 000246 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS

```

```

58 000254
59 000254 016746 000000G. 5$: MOV. PREADD,-(SP) ;SEQ UP TO START ADDRESS.
60 000260 CALL. SEQCS. ;DO IT.
61 000264 CALL. CMPA ;COMPARE SECT A OF MEMORY.
62 000270 005267 000000G. INC. CKDATA. ;BUMP TEST COUNTER.
63 000274 005267 000000G. INC. PREADD. ;BUMP ADDRESS.
64 000300 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED?.
65 000306 103362. BHIS. 5$
66 ;
67 000310 016667 000002 000000G. MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER.
68 000316 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS.
69 000324 6$:
70 000324 016746 000000G. MOV. PREADD,-(SP) ;SEQ UP TO START ADDRESS.
71 000330 CALL. SEQCS. ;DO IT.
72 000334 CALL. CMPBB. ;COMPARE SECT B OF MEMORY.
73 000340 005267 000000G. INC. CKDATA. ;BUMP TEST COUNTER.
74 000344 005267 000000G. INC. PREADD. ;BUMP ADDRESS.
75 000350 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED?.
76 000356 103362. BHIS. 6$
77 ;
78 000360 016667 000002 000000G. MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER.
79 000366 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS.
80 000374 7$:
81 000374 016746 000000G. MOV. PREADD,-(SP) ;SEQ UP TO START ADDRESS.
82 000400 CALL. SEQCS. ;DO IT.
83 000404 CALL. CMPC. ;COMPARE SECT C OF MEMORY.
84 000410 005267 000000G. INC. CKDATA. ;BUMP TEST COUNTER.
85 000414 005267 000000G. INC. PREADD. ;BUMP ADDRESS.
86 000420 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED?.
87 000426 103362. BHIS. 7$
88 ;
89 000430 016667 000002 000000G. MOV. 2(SP),CKDATA. ;START ADDRESS = TEST COUNTER.
90 000436 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS.
91 000444 8$:
92 000444 016746 000000G. MOV. PREADD,-(SP) ;SEQ UP TO START ADDRESS.
93 000450 CALL. SEQCS. ;DO IT.
94 000454 CALL. CMPDD. ;COMPARE SECT D OF MEMORY.
95 000460 005267 000000G. INC. CKDATA. ;BUMP TEST COUNTER.
96 000464 005267 000000G. INC. PREADD. ;BUMP ADDRESS.
97 000470 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED?.
98 000476 103362. BHIS. 8$
99 ;
100 000500 RETURN.
101 000001 .END.

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR\$CR1 = 176420	Q\$RSC = 004000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR\$CR2 = 176422	Q\$RSET = 000010
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR\$LBR = 176424	Q\$SM = 100000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$ATTN = 000100	Q\$SP = 000120
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$BCL = 000001	Q\$SP2 = 000340
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CCCP = 000040	RGD.EN = 000200
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CHB = 000400	RGD.VA = 020000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CHRL = 000200	SEQCS = ***** GX
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$CLR = 000040	SEQ.CI = 000010
BIT13 = 020000	BYTE51 = 000063	CKDATA = ***** GX	Q\$CNC = 030000	\$SCLR = 000000
BIT14 = 040000	BYTE52 = 000064	CMFA = ***** GX	Q\$CP = 000060	\$S\$A = 000001
BIT15 = 100000	BYTE53 = 000065	CMFBB = ***** GX	Q\$CPCC = 000010	\$S\$B = 000005
BIT2 = 000004	BYTE54 = 000066	CMPC = ***** GX	Q\$CP2 = 000260	\$S\$R = 000006
BIT3 = 000010	BYTE55 = 000067	CMPPD = ***** GX	Q\$CSC = 010000	\$S\$X = 000004
BIT4 = 000020	BYTE56 = 000070	CNOBRE = 100000	Q\$CSEL = 000360	\$S\$SR = 000007
BIT5 = 000040	BYTE57 = 000071	CPCCEN = 010000	Q\$CSET = 000002	\$S\$1 = 000010
BIT6 = 000100	BYTE58 = 000072	CPREAD = 040000	Q\$CSP = 020000	\$S\$2 = 000014
BIT7 = 000200	BYTE59 = 000073	CPWRTE = 020000	Q\$DMB = 000001	TD\$CTR = 176370
BIT8 = 000400	BYTE60 = 000074	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTW = 176360
BIT9 = 001000	BYTE61 = 000075	CSEQCI = 100000	Q\$ENOP = 020000	TD\$INL = 004000
BYTE0 = 000000	BYTE62 = 000076	CSOE = 000040	Q\$FAL = 004000	TD\$MEM = 000270
BYTE1 = 000001	BYTE63 = 000077	CSURTE = 000100	Q\$FC = 000045	TD\$OAR = 176344
BYTE10 = 000012	BYTE64 = 000100	DBR.RD = 000001	Q\$FO = 000044	TD\$OAR = 176344
BYTE11 = 000013	BYTE65 = 000101	DB\$CPP = 001457	Q\$FP = 000046	TD\$OTR = 176346
BYTE12 = 000014	BYTE66 = 000102	DB\$SPT = 000026	Q\$HBF = 000002	TD\$ORD = 000274
BYTE13 = 000015	BYTE67 = 000103	DB\$TPC = 000023	Q\$ICP = 000006	TD\$SW = 176376
BYTE14 = 000016	BYTE68 = 000104	DISPGS = 100000	Q\$IHB = 000003	TD\$TAR = 176372
BYTE15 = 000017	BYTE69 = 000105	DMAAWR = 000005	Q\$IHR = 000002	TD\$TAW = 176362
BYTE16 = 000020	BYTE70 = 000106	DMAARR = 000003	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE17 = 000021	BYTE71 = 000107	DMAWR = 000004	Q\$LBD = 001000	TD\$TDW = 176364
BYTE18 = 000022	BYTE72 = 000110	ENBR = 010000	Q\$LBDP = 001001	T\$AD = 000020
BYTE19 = 000023	BYTE73 = 000111	LOC.EN = 000100	Q\$LB = 000001	T\$BA = 000002
BYTE2 = 000002	BYTE74 = 000112	LOC.WA = 040000	Q\$LD = 000003	T\$BD = 000010
BYTE20 = 000024	BYTE75 = 000113	LOC.WB = 100000	Q\$LDL = 000004	T\$BSA = 100000
BYTE21 = 000025	BYTE76 = 000114	MAREN1 = 000001	Q\$LDMD = 000004	T\$BI = 000020
BYTE22 = 000026	BYTE77 = 000115	MAREN2 = 004000	Q\$LDPP = 002000	T\$BIAR = 000030
BYTE23 = 000027	BYTE78 = 000116	MARLOD = 010000	Q\$LHP = 010000	T\$BIID = 002000
BYTE24 = 000030	BYTE79 = 000117	MAROUT = 000002	Q\$MNC = 140000	T\$BI = 000100
BYTE25 = 000031	BYTE80 = 000120	MAR.LO = 002000	Q\$MR = 000052	T\$BCK = 002000
BYTE26 = 000032	BYTE81 = 000121	MAR.OU = 000040	Q\$MRP = 000040	T\$DISK = 000200
BYTE27 = 000033	BYTE82 = 000122	MBKALL = 001000	Q\$MRP2 = 000240	T\$DRD = 000004
BYTE28 = 000034	BYTE83 = 000123	MBKCLK = 000400	Q\$MSC = 040000	T\$MEM = 010000
BYTE29 = 000035	BYTE84 = 000124	MMADRD = 000100	Q\$MSET = 000004	T\$FSAA = 000000
BYTE3 = 000003	BYTE85 = 000125	MMLEFT = 000002	Q\$MSP = 100000	T\$FSAB = 000004
BYTE30 = 000036	BYTE86 = 000126	MMOE = 000004	Q\$SNCLK = 176000	T\$FSAC = 000014
BYTE31 = 000037	BYTE87 = 000127	MMURTE = 000010	Q\$PP = 000100	T\$FSB2 = 000010
BYTE32 = 000040	BYTE88 = 000130	MNOBRE = 100000	Q\$PPSW = 000320	T\$IB = 000026
BYTE33 = 000041	BYTE89 = 000131	MREN1 = 000001	Q\$PP2 = 000300	T\$IBAR = 000024
BYTE34 = 000042	BYTE90 = 000132	MREN2 = 020000	Q\$QHLT = 000013	T\$IBE = 020000
BYTE35 = 000043	BYTE91 = 000133	MSYN = 000040	Q\$QL = 000043	T\$IBF = 040000
BYTE36 = 000044	BYTE92 = 000134	N = 000144	Q\$QLA = 000053	T\$ICD = 000040
BYTE37 = 000045	BYTE93 = 000135	PLB = 000010	Q\$QLB = 000054	T\$MODE = 004000
BYTE38 = 000046		PLC = 000020	Q\$QLR = 000001	T\$OB = 000036
BYTE39 = 000047		PLD = 000030	Q\$QW = 000042	T\$OBF = 004000
BYTE4 = 000004		PLRWR = 000200	Q\$RDCD = 000005	T\$OBF = 010000
BYTE40 = 000050		PLR.EN = 000200	Q\$RDMB = 000006	T\$OBRA = 000034
BYTE41 = 000051		PREADD = ***** GX	Q\$REBK = 001000	T\$OBWA = 000032
			Q\$RNC = 000000	T\$OUTA = 100000

T\$RBD0 = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T\$RNB = 000040	WORD2 = 000004	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T\$RSET = 040000	WORD20 = 000050	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T\$SC = 000022	WORD21 = 000052	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T\$SCLK = 020000	WORD22 = 000054	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T\$SEG1 = 000000	WORD23 = 000056	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T\$SEG2 = 000001	WORD24 = 000060	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T\$SEG3 = 000002	WORD25 = 000062	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T\$SO = 000001	WORD26 = 000064	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T\$UBUS = 100000	WORD27 = 000066	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T\$1CLK = 000400	WORD28 = 000070	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T\$8BEN = 000020	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
TICS = 000000RG	002 WORD3 = 000006	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
UBD.IN = 000020	WORD30 = 000074	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
WORD0 = 000000	WORD31 = 000076	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD1 = 000002	WORD32 = 000100	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD10 = 000024	WORD33 = 000102	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD11 = 000026	WORD34 = 000104	WORD56 = 000160	WORD78 = 000234	WRDVAL = 000310
WORD12 = 000030	WORD35 = 000106	WORD57 = 000162	WORD79 = 000236	WRITEA = *****GX
WORD13 = 000032	WORD36 = 000110	WORD58 = 000164	WORD8 = 000020	WRITEB = *****GX
WORD14 = 000034	WORD37 = 000112	WORD59 = 000166	WORD00 = 000240	WRITEC = *****GX
WORD15 = 000036	WORD38 = 000114	WORD6 = 000014	WORD01 = 000242	WRITED = *****GX
WORD16 = 000040	WORD39 = 000116	WORD60 = 000170	WORD02 = 000244	XTREAD = 001000
WORD17 = 000042	WORD4 = 000010	WORD61 = 000172	WORD03 = 000246	XTWRITE = 000400
WORD18 = 000044				

. ABS: 000000 000
000000 001
CSTST1 000502 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3080 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:41
CSTST1, CSTST1 / -SP=[20, 1]IM,[20, 1]CSTST1


```

1
2 000000 .TITLE--GSTST2-
3 .PSECT--CSTST2-
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 000000 T6GS::
14 000000 012767 177777 000000G MOV #1,CKDATA ;SET TEST PATTERN = X'FFFF'
15 000000 012702 000012 MOV #10,,R2 ;SET LOOP COUNT
16 000012 016667 000002 000000G 10$ MOV 2(SP),PREADD ;WORKING ADDRESS
17 000020 1$
18 000020 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
19 000024 CALL SEQCS ;DO IT
20 000030 CALL WRITEB ;WRITE SECT A OF MEMORY
21 000034 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
22 000042 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
23 000050 103363 BHS 1$ ;NO
24 000052 005302 DEC R2 ;SUB FROM LOOP COUNT
25 000054 001356 BNE 10$
26 ;
27 000056 012702 000012 MOV #10,,R2 ;SET LOOP COUNT
28 000062 016667 000002 000000G 20$ MOV 2(SP),PREADD ;WORKING ADDRESS
29 000070 2$
30 000070 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
31 000074 CALL SEQCS ;DO IT
32 000100 CALL WRITEB ;WRITE SECT B OF MEMORY
33 000104 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
34 000112 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
35 000120 103363 BHS 2$ ;NO
36 000122 005302 DEC R2 ;SUB FROM LOOP COUNT
37 000124 001356 BNE 20$
38 ;
39 000126 012702 000012 MOV #10,,R2 ;SET LOOP COUNT
40 000132 016667 000002 000000G 30$ MOV 2(SP),PREADD ;WORKING ADDRESS
41 000140 3$
42 000140 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
43 000144 CALL SEQCS ;DO IT
44 000150 CALL WRITEB ;WRITE SECT C OF MEMORY
45 000154 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
46 000162 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
47 000170 103363 BHS 3$ ;NO
48 000172 005302 DEC R2 ;SUB FROM LOOP COUNT
49 000174 001356 BNE 30$
50 ;
51 000176 012702 000012 MOV #10,,R2 ;SET LOOP COUNT
52 000202 016667 000002 000000G 40$ MOV 2(SP),PREADD ;WORKING ADDRESS
53 000210 4$
54 000210 016746 000000G MOV PREADD, -(SP) ;SET SEQUENCER TO START ADDRESS
55 000214 CALL SEQCS ;DO IT
56 000220 CALL WRITEB ;WRITE SECT D OF MEMORY
57 000224 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS

```

```

58 000232 026667 000004 000000G.    CMP.    4(SP),PREADD.    ;FINISHED.?
59 000240 103363                    BHIS.   4$                ;NO
60 000242 005302                    DEC.    R2.                ;SUB FROM LOOP COUNT
61 000244 001356                    BNE.   40$
62.                                     ;
63.                                     ;
64.                                     ;
65.                                     ;
66 000246                                R6Z:
67 000246 005067 000000G.    CLR.    CKDATA.        ;SET TEST PATTERN = 0
68 000252 016667 000002 000000G.    MOV.    2(SP),PREADD.  ;GET START ADDRESS
69 000260 005267 000000G.    INC.    PREADD.        ;BUMP START ADDRESS
70 000264                                1$:
71 000264 016746 000000G.    MOV.    PREADD, -(SP)  ;SET SEQUENCER TO START ADDRESS
72 000270                                CALL.   SEQCS.         ;DO IT
73 000274                                CALL.   CMPA.         ;COMPARE SECT A OF MEMORY
74 000300 062767 000002 000000G.    ADD.    #2,PREADD.    ;SKIP HERE TOO
75 000306 026667 000004 000000G.    CMP.    4(SP),PREADD. ;FINISHED.?
76 000314 103363                    BHIS.   1$
77.                                     ;
78 000316 016667 000002 000000G.    MOV.    2(SP),PREADD. ;WORKING ADDRESS
79 000324 005267 000000G.    INC.    PREADD.        ;BUMP START ADDRESS
80 000330                                2$:
81 000330 016746 000000G.    MOV.    PREADD, -(SP)  ;SET SEQUENCER TO START ADDRESS
82 000334                                CALL.   SEQCS.         ;DO IT
83 000340                                CALL.   CMPBB.        ;COMPARE SECT B OF MEMORY
84 000344 062767 000002 000000G.    ADD.    #2,PREADD.    ;SKIP HERE TOO
85 000352 026667 000004 000000G.    CMP.    4(SP),PREADD. ;FINISHED.?
86 000360 103363                    BHIS.   2$
87.                                     ;
88 000362 016667 000002 000000G.    MOV.    2(SP),PREADD. ;GET START ADDRESS
89 000370 005267 000000G.    INC.    PREADD.        ;BUMP START ADDRESS
90 000374                                3$:
91 000374 016746 000000G.    MOV.    PREADD, -(SP)  ;SET SEQUENCER TO START ADDRESS
92 000400                                CALL.   SEQCS.         ;DO IT
93 000404                                CALL.   CMPC.         ;COMPARE SECT C OF MEMORY
94 000410 062767 000002 000000G.    ADD.    #2,PREADD.    ;SKIP HERE TOO
95 000416 026667 000004 000000G.    CMP.    4(SP),PREADD. ;FINISHED.?
96 000424 103363                    BHIS.   3$
97.                                     ;
98 000426 016667 000002 000000G.    MOV.    2(SP),PREADD. ;WORKING ADDRESS
99 000434 005267 000000G.    INC.    PREADD.        ;BUMP START ADDRESS
100 000440                                4$:
101 000440 016746 000000G.    MOV.    PREADD, -(SP)  ;SET SEQUENCER TO START ADDRESS
102 000444                                CALL.   SEQCS.         ;DO IT
103 000450                                CALL.   CMPDD.        ;COMPARE SECT D OF MEMORY
104 000454 062767 000002 000000G.    ADD.    #2,PREADD.    ;SKIP HERE TOO
105 000462 026667 000004 000000G.    CMP.    4(SP),PREADD. ;FINISHED.?
106 000470 103363                    BHIS.   4$
107.                                     ;
108 000472                                RETURN.
109 000001                                .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR\$CR1 = 176420	Q\$RSC = 004000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR\$CR2 = 176422	Q\$RSET = 000010
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR\$LBR = 176424	Q\$SM = 100000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$ATTN = 000100	Q\$SP = 000120
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$BCL = 000001	Q\$SP2 = 000340
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CCCP = 000040	RGQ.EN = 000200
BIT10 = 000200	BYTE48 = 000060	BYTVAL = 000144	Q\$CHB = 000400	RGQ.VA = 020000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CHL = 000200	R6Z = 000246R
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CLR = 000040	SEQCS = ***** GX
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q\$CNC = 030000	SEQ.CI = 000010
BIT14 = 040000	BYTE51 = 000063	CMFA = ***** GX	Q\$CP = 000060	S\$CLR = 000000
BIT15 = 100000	BYTE52 = 000064	CMFBB = ***** GX	Q\$CPCC = 000010	S\$LA = 000001
BIT2 = 000004	BYTE53 = 000065	CMPC = ***** GX	Q\$CP2 = 000260	S\$OB = 000005
BIT3 = 000010	BYTE54 = 000066	CMFDD = ***** GX	Q\$CSC = 010000	S\$QR = 000006
BIT4 = 000020	BYTE55 = 000067	CHOBRE = 100000	Q\$CSEL = 000360	S\$QX = 000004
BIT5 = 000040	BYTE56 = 000070	CPCCEN = 010000	Q\$CSET = 000002	S\$SR = 000007
BIT6 = 000100	BYTE57 = 000071	CPREAD = 040000	Q\$CSP = 020000	S\$S1 = 000010
BIT7 = 000200	BYTE58 = 000072	CPWRTE = 020000	Q\$DMA = 000001	S\$S2 = 000014
BIT8 = 000400	BYTE59 = 000073	CSADRD = 000004	Q\$ENBK = 040000	TD\$CTR = 176370
BIT9 = 001000	BYTE6 = 000006	CSEDCI = 100000	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE0 = 000000	BYTE60 = 000074	CSOE = 000040	Q\$FAL = 004000	TD\$INL = 004000
BYTE1 = 000001	BYTE61 = 000075	CSWRTE = 000100	Q\$FC = 000045	TD\$MEM = 000270
BYTE10 = 000012	BYTE62 = 000076	DBR.RD = 000001	Q\$FO = 000044	TD\$OAR = 176344
BYTE11 = 000013	BYTE63 = 000077	DB\$CPP = 001457	Q\$FP = 000046	TD\$OTR = 176346
BYTE12 = 000014	BYTE64 = 000100	DB\$SPT = 000026	Q\$HBF = 000002	TD\$QRB = 000274
BYTE13 = 000015	BYTE65 = 000101	DB\$TPC = 000023	Q\$ICP = 000006	TD\$SW = 176376
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q\$IHB = 000003	TD\$TAR = 176372
BYTE15 = 000017	BYTE67 = 000103	DMAWR = 000005	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q\$IMRP = 000007	TD\$TDR = 176374
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q\$LBD = 001000	TD\$TDW = 176364
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q\$LBDP = 001001	T\$AD = 000020
BYTE19 = 000023	BYTE70 = 000106	LOC.EN = 000100	Q\$LBP = 000001	T\$AB = 000002
BYTE2 = 000002	BYTE71 = 000107	LOC.WA = 040000	Q\$LCD = 000003	T\$BD = 000010
BYTE20 = 000024	BYTE72 = 000110	LOC.WB = 100000	Q\$LDMD = 000004	T\$BSO = 100000
BYTE21 = 000025	BYTE73 = 000111	MAREN1 = 000001	Q\$LDPP = 002000	T\$BT = 000020
BYTE22 = 000026	BYTE74 = 000112	MAREN2 = 004000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE23 = 000027	BYTE75 = 000113	MARLDD = 010000	Q\$MNC = 140000	T\$BTD = 002000
BYTE24 = 000030	BYTE76 = 000114	MAROUT = 000002	Q\$MR = 000052	T\$CD = 000100
BYTE25 = 000031	BYTE77 = 000115	MAR.LO = 020000	Q\$MRP = 000040	T\$CLK = 002000
BYTE26 = 000032	BYTE78 = 000116	MAR.DU = 000040	Q\$MRP2 = 000240	T\$DISK = 000200
BYTE27 = 000033	BYTE79 = 000117	MBKALL = 001000	Q\$MSE = 040000	T\$DRD = 000004
BYTE28 = 000034	BYTE8 = 000010	MBKCLK = 000400	Q\$MSET = 000004	T\$ENEM = 010000
BYTE29 = 000035	BYTE80 = 000120	MMADRD = 000100	Q\$MSP = 100000	T\$FSAA = 000000
BYTE3 = 000003	BYTE81 = 000121	MMLFT = 000002	Q\$NCLK = 176000	T\$FSAB = 000004
BYTE30 = 000036	BYTE82 = 000122	MNOE = 000004	Q\$PP = 000100	T\$FSAC = 000014
BYTE31 = 000037	BYTE83 = 000123	MNWRTE = 000010	Q\$PPSW = 000320	T\$FSB2 = 000010
BYTE32 = 000040	BYTE84 = 000124	MNOBRE = 100000	Q\$PP2 = 000300	T\$IB = 000026
BYTE33 = 000041	BYTE85 = 000125	MREN1 = 000001	Q\$QHLT = 000013	T\$IBAR = 000024
BYTE34 = 000042	BYTE86 = 000126	MREN2 = 020000	Q\$QL = 000043	T\$IBE = 020000
BYTE35 = 000043	BYTE87 = 000127	MSYN = 000040	Q\$QLA = 000053	T\$IBF = 040000
BYTE36 = 000044	BYTE88 = 000130	N = 000144	Q\$QLB = 000054	T\$ICD = 000040
BYTE37 = 000045	BYTE89 = 000131	PLB = 000010	Q\$QLR = 000001	T\$MODE = 004000
BYTE38 = 000046	BYTE9 = 000011	PLC = 000020	Q\$QW = 000042	T\$OB = 000036
BYTE39 = 000047	BYTE90 = 000132	PLD = 000030	Q\$RDCD = 000005	T\$ORF = 004000
BYTE4 = 000004	BYTE91 = 000133	PLRWR = 000200	Q\$RDND = 000006	T\$PFB = 010000
BYTE40 = 000050	BYTE92 = 000134	PLR.EN = 000200	Q\$REBK = 001000	T\$OBRA = 000034
BYTE41 = 000051	BYTE93 = 000135	PREADD = ***** GX	Q\$RNC = 006000	T\$OBWA = 000032

T\$OUTA = 100000	WORD18 = 000044	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T\$RBD0 = 000200	WORD19 = 000046	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T\$RNB = 000040	WORD2 = 000004	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T\$RSET = 040000	WORD20 = 000050	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T\$SC = 000022	WORD21 = 000052	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T\$SCLK = 020000	WORD22 = 000054	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T\$SEG1 = 000000	WORD23 = 000056	WORD46 = 000134	WORD68 = 000210	WORD9 = 000022
T\$SEG2 = 000001	WORD24 = 000060	WORD47 = 000136	WORD69 = 000212	WORD90 = 000264
T\$SEG3 = 000002	WORD25 = 000062	WORD48 = 000140	WORD7 = 000016	WORD91 = 000266
T\$SO = 000001	WORD26 = 000064	WORD49 = 000142	WORD70 = 000214	WORD92 = 000270
T\$UBUS = 100000	WORD27 = 000066	WORD5 = 000012	WORD71 = 000216	WORD93 = 000272
T\$1CLK = 000400	WORD28 = 000070	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T\$BEN = 000020	WORD29 = 000072	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
T6CS = 000000RG	WORD3 = 000006	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
UBD.IN = 000020	WORD30 = 000074	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD0 = 000000	WORD31 = 000076	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD1 = 000002	WORD32 = 000100	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD10 = 000024	WORD33 = 000102	WORD56 = 000160	WORD78 = 000234	WRDVAL = 000310
WORD11 = 000026	WORD34 = 000104	WORD57 = 000162	WORD79 = 000236	WRITE = ***** GX
WORD12 = 000030	WORD35 = 000106	WORD58 = 000164	WORD8 = 000020	WRITEB = ***** GX
WORD13 = 000032	WORD36 = 000110	WORD59 = 000166	WORD00 = 000240	WRITEC = ***** GX
WORD14 = 000034	WORD37 = 000112	WORD6 = 000014	WORD81 = 000242	WRITEE = ***** GX
WORD15 = 000036	WORD38 = 000114	WORD60 = 000170	WORD82 = 000244	XTREAD = 001000
WORD16 = 000040	WORD39 = 000116	WORD61 = 000172	WORD83 = 000246	XTWRITE = 000400
WORD17 = 000042	WORD4 = 000010			

. ABS. 000000 000
000000 001
CSTST2 000474 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3104 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
CSTST2, CSTST2 / -SP=[20, 1] JIM, [20, 1] CSTST2

```

1
2 000000 .TITLE - CSTST3
3 .PSECT - CSTST3
4
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 000000 T7GS::
12 000000 016602 000002 MOV 2(SP),R2 ;START ADDRESS = TEST COUNTER
13 000004 010267 000000G MOV R2,PREADD ;WORKING ADDRESS
14 000010 1$:
15 000010 010246 MOV R2, -(SP) ;SEQUENCE UP TO START ADDRESS
16 000012 CALL SEQCS ;DO IT
17 000016 005102 COM R2 ;GET ADDRESS COMPLEMENT
18 000020 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN = ADDR COMPLEMENT
19 000024 CALL WRITEA ;WRITE SECT A OF MEMORY
20 000030 005267 000000G INC PREADD ;BUMP ADDRESS
21 000034 016702 000000G MOV PREADD,R2 ;SET R2 TO NEXT ADDRESS
22 000040 026602 000004 CMP 4(SP),R2 ;FINISHED?
23 000044 103361 BHIS 1$ ;NO
24 ;
25 000046 016602 000002 MOV 2(SP),R2 ;START ADDRESS = TEST COUNTER
26 000052 010267 000000G MOV R2,PREADD ;WORKING ADDRESS
27 000056 2$:
28 000056 010246 MOV R2, -(SP) ;SEQ UP TO START ADDRESS
29 000060 CALL SEQCS ;DO IT
30 000064 005102 COM R2 ;GET ADDRESS COMPLEMENT
31 000066 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN
32 000072 CALL WRITEB ;WRITE SECT B OF MEMORY
33 000076 005267 000000G INC PREADD ;BUMP ADDRESS
34 000102 016702 000000G MOV PREADD,R2 ;SET UP FOR NEXT ADDRESS
35 000106 026602 000004 CMP 4(SP),R2 ;FINISHED?
36 000112 103361 BHIS 2$ ;NO
37 ;
38 000114 016602 000002 MOV 2(SP),R2 ;START ADDRESS = TEST COUNTER
39 000120 010267 000000G MOV R2,PREADD ;WORKING ADDRESS
40 000124 3$:
41 000124 010246 MOV R2, -(SP) ;SEQUENCE UP TO START ADDRESS
42 000126 CALL SEQCS ;DO IT
43 000132 005102 COM R2 ;GET ADDRESS COMPLEMENT
44 000134 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN = ADDR COMPLEMENT
45 000140 CALL WRITEC ;WRITE SECT C OF MEMORY
46 000144 005267 000000G INC PREADD ;BUMP ADDRESS
47 000150 016702 000000G MOV PREADD,R2 ;SET R2 TO NEXT ADDRESS
48 000154 026602 000004 CMP 4(SP),R2 ;FINISHED?
49 000160 103361 BHIS 3$ ;NO
50 ;
51 000162 016602 000002 MOV 2(SP),R2 ;START ADDRESS = TEST COUNTER
52 000166 010267 000000G MOV R2,PREADD ;WORKING ADDRESS
53 000172 4$:
54 000172 010246 MOV R2, -(SP) ;SEQUENCE UP TO START ADDRESS
55 000174 CALL SEQCS ;DO IT
56 000200 005102 COM R2 ;GET ADDRESS COMPLEMENT
57 000202 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN = ADDR COMPLEMENT

```

58	000206			CALL	WRITED		:WRITE SECT D OF MEMORY
59	000212	005267	000000G	INC	PREADD		:BUMP ADDRESS
60	000216	016702	000000G	MOV	PREADD,R2		:SET R2 TO NEXT ADDRESS
61	000222	026602	000004	CMF	4(SP),R2		:FINISHED ?
62	000226	103361		BHIS	4\$:NO
63				:			
64	000230	016602	000002	MOV	2(SP),R2		:START ADDRESS = TEST COUNTER
65	000234	010267	000000G	MOV	R2,PREADD		:WORKING ADDRESS
66	000240			5\$:			
67	000240	010246		MOV	R2,-(SP)		:SEQUENCE UP TO START ADDRESS
68	000242			CALL	SEQCS		:DO IT
69	000246	005102		COM	R2		:GET ADDRESS COMPLEMENT
70	000250	010267	000000G	MOV	R2,CKDATA		:SET TEST PATTERN = ADDR COMPLEMENT
71	000254			CALL	CMFA		:COMPARE SECT A
72	000260	005267	000000G	INC	PREADD		:BUMP ADDRESS
73	000264	016702	000000G	MOV	PREADD,R2		:SET R2 TO NEXT ADDRESS
74	000270	026602	000004	CMF	4(SP),R2		:FINISHED ?
75	000274	103361		BHIS	5\$:NO
76				:			
77	000276	016602	000002	MOV	2(SP),R2		:START ADDRESS = TEST COUNTER
78	000302	010267	000000G	MOV	R2,PREADD		:WORKING ADDRESS
79	000306			6\$:			
80	000306	010246		MOV	R2,-(SP)		:SEQUENCE UP TO START ADDRESS
81	000310			CALL	SEQCS		:DO IT
82	000314	005102		COM	R2		:GET ADDRESS COMPLEMENT
83	000316	010267	000000G	MOV	R2,CKDATA		:SET TEST PATTERN = ADDR COMPLEMENT
84	000322			CALL	CMFBB		:COMPARE SECT B
85	000326	005267	000000G	INC	PREADD		:BUMP ADDRESS
86	000332	016702	000000G	MOV	PREADD,R2		:SET R2 TO NEXT ADDRESS
87	000336	026602	000004	CMF	4(SP),R2		:FINISHED ?
88	000342	103361		BHIS	6\$:NO
89				:			
90	000344	016602	000002	MOV	2(SP),R2		:START ADDRESS = TEST COUNTER
91	000350	010267	000000G	MOV	R2,PREADD		:WORKING ADDRESS
92	000354			7\$:			
93	000354	010246		MOV	R2,-(SP)		:SEQUENCE UP TO START ADDRESS
94	000356			CALL	SEQCS		:DO IT
95	000362	005102		COM	R2		:GET ADDRESS COMPLEMENT
96	000364	010267	000000G	MOV	R2,CKDATA		:SET TEST PATTERN = ADDR COMPLEMENT
97	000370			CALL	CMPC		:COMPARE SECT C
98	000374	005267	000000G	INC	PREADD		:BUMP ADDRESS
99	000400	016702	000000G	MOV	PREADD,R2		:SET R2 TO NEXT ADDRESS
100	000404	026602	000004	CMF	4(SP),R2		:FINISHED ?
101	000410	103361		BHIS	7\$:NO
102				:			
103	000412	016602	000002	MOV	2(SP),R2		:START ADDRESS = TEST COUNTER
104	000416	010267	000000G	MOV	R2,PREADD		:WORKING ADDRESS
105	000422			8\$:			
106	000422	010246		MOV	R2,-(SP)		:SEQUENCE UP TO START ADDRESS
107	000424			CALL	SEQCS		:DO IT
108	000430	005102		COM	R2		:GET ADDRESS COMPLEMENT
109	000432	010267	000000G	MOV	R2,CKDATA		:SET TEST PATTERN = ADDR COMPLEMENT
110	000436			CALL	CMFDD		:COMPARE SECT D
111	000442	005267	000000G	INC	PREADD		:BUMP ADDRESS
112	000446	016702	000000G	MOV	PREADD,R2		:SET R2 TO NEXT ADDRESS
113	000452	026602	000004	CMF	4(SP),R2		:FINISHED ?
114	000456	103361		BHIS	8\$:NO

CSTSTZ..M...M1110 27-MAR-80 14:48 PAGE 5-2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

115
116 000460
117 000001

RETURN
.END.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	QR#CR1 = 176420	Q#RSC = 004000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	QR#CR2 = 176422	Q#RSET = 000010
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	QR#LBR = 176424	Q#SM = 100000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q#ATTN = 000100	Q#SP = 000120
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q#BCL = 000001	Q#SP2 = 000340
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q#CCCP = 000040	RGD.EN = 000200
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q#CHB = 000400	RGD.VA = 020000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#CHRL = 000200	SEOC.S = ***** GX
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q#CLR = 000040	SEQ.CI = 000010
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q#CNC = 030000	S#CLR = 000000
BIT14 = 040000	BYTE51 = 000063	CMPA = ***** GX	Q#CP = 000060	S#LA = 000001
BIT15 = 100000	BYTE52 = 000064	CMPBB = ***** GX	Q#CPCC = 000010	S#QB = 000005
BIT2 = 000004	BYTE53 = 000065	CMPD = ***** GX	Q#CP2 = 000260	S#QR = 000006
BIT3 = 000010	BYTE54 = 000066	CMPDD = ***** GX	Q#CSC = 010000	S#QX = 000004
BIT4 = 000020	BYTE55 = 000067	CNOBRE = 100000	Q#CSEL = 000360	S#SR = 000007
BIT5 = 000040	BYTE56 = 000070	CPCCEN = 010000	Q#CSET = 000002	S#S1 = 000010
BIT6 = 000100	BYTE57 = 000071	CPREAD = 040000	Q#CSP = 020000	S#S2 = 000014
BIT7 = 000200	BYTE58 = 000072	CPURTE = 020000	Q#DMA = 000001	TD#CTR = 176370
BIT8 = 000400	BYTE59 = 000073	CSADRD = 000004	Q#ENBK = 040000	TD#CTW = 176360
BIT9 = 001000	BYTE6 = 000006	CSEQCI = 100000	Q#ENOP = 020000	TD#INL = 004000
BYTE0 = 000000	BYTE60 = 000074	CSDOE = 000040	Q#FAL = 004000	TD#MEM = 000270
BYTE1 = 000001	BYTE61 = 000075	CSURTE = 000100	Q#FC = 000045	TD#OAR = 176344
BYTE10 = 000012	BYTE62 = 000076	DBR.RD = 000001	Q#FO = 000044	TD#OTR = 176346
BYTE11 = 000013	BYTE63 = 000077	DB#CPP = 001457	Q#FP = 000046	TD#ORD = 000274
BYTE12 = 000014	BYTE64 = 000100	DB#SPT = 000026	Q#HBF = 000002	TD#SJW = 176376
BYTE13 = 000015	BYTE65 = 000101	DB#TPC = 000023	Q#ICP = 000006	TD#TAR = 176372
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q#IHB = 000003	TD#TAW = 176362
BYTE15 = 000017	BYTE67 = 000103	DMAWR = 000005	Q#IHL = 000002	TD#TDR = 176374
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q#IMRP = 000007	TD#TDW = 176364
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q#LBD = 001000	T#AD = 000020
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q#LBDP = 001001	T#BA = 000002
BYTE19 = 000023	BYTE70 = 000106	LOC.EN = 000100	Q#LBP = 000001	T#BD = 000010
BYTE2 = 000002	BYTE71 = 000107	LOC.WA = 040000	Q#LDCD = 000003	T#BSO = 100000
BYTE20 = 000024	BYTE72 = 000110	LOC.WB = 100000	Q#LDM = 000004	T#BT = 000020
BYTE21 = 000025	BYTE73 = 000111	MAREN1 = 000001	Q#LDPP = 002000	T#BTAR = 000030
BYTE22 = 000026	BYTE74 = 000112	MAREN2 = 004000	Q#LHP = 010000	T#BTD = 002000
BYTE23 = 000027	BYTE75 = 000113	MARLOD = 010000	Q#MNC = 140000	T#CD = 000100
BYTE24 = 000030	BYTE76 = 000114	MAROUT = 000002	Q#MR = 000052	T#CLK = 002000
BYTE25 = 000031	BYTE77 = 000115	MAR.LO = 002000	Q#MRP = 000040	T#DISK = 000200
BYTE26 = 000032	BYTE78 = 000116	MAR.OU = 000040	Q#MRP2 = 000240	T#DRD = 000004
BYTE27 = 000033	BYTE79 = 000117	MBKALL = 001000	Q#MSC = 040000	T#MEM = 010000
BYTE28 = 000034	BYTE8 = 000010	MBKCLK = 000400	Q#MSET = 000004	T#FSAA = 000000
BYTE29 = 000035	BYTE80 = 000120	MMADR = 000100	Q#MSP = 100000	T#FSAB = 000004
BYTE3 = 000003	BYTE81 = 000121	MLEFT = 000002	Q#NCLK = 176000	T#FSAC = 000014
BYTE30 = 000036	BYTE82 = 000122	MMD = 000004	Q#PP = 000100	T#FSB2 = 000010
BYTE31 = 000037	BYTE83 = 000123	MNRTE = 000010	Q#PPSW = 000320	T#IB = 000026
BYTE32 = 000040	BYTE84 = 000124	MNOBRE = 100000	Q#PP2 = 000300	T#IBAR = 000024
BYTE33 = 000041	BYTE85 = 000125	MREN1 = 000001	Q#QHLT = 000013	T#IBE = 020000
BYTE34 = 000042	BYTE86 = 000126	MREN2 = 000000	Q#QL = 000043	T#IBF = 040000
BYTE35 = 000043	BYTE87 = 000127	MSYN = 000040	Q#QLA = 000053	T#ICD = 000040
BYTE36 = 000044	BYTE88 = 000130	N = 000144	Q#QLB = 000054	T#MODE = 004000
BYTE37 = 000045	BYTE89 = 000131	PLB = 000010	Q#QLR = 000001	T#DB = 000036
BYTE38 = 000046	BYTE9 = 000011	PLC = 000020	Q#QW = 000042	T#DBE = 004000
BYTE39 = 000047	BYTE90 = 000132	PLD = 000030	Q#RDCD = 000005	T#DFE = 010000
BYTE4 = 000004	BYTE91 = 000133	PLRWR = 000200	Q#RDMD = 000006	T#DRA = 000034
BYTE40 = 000050	BYTE92 = 000134	PLR.EN = 000200	Q#REBK = 001000	T#BLWA = 000032
BYTE41 = 000051	BYTE93 = 000135	PREADD = ***** GX	Q#RNC = 006000	T#OUTA = 100000

T#RBD0 = 000200	WORD19 = 000046	WORD40 = 000120	WORD62 = 000174	WORD84 = 000250
T#RNB = 000040	WORD20 = 000050	WORD41 = 000122	WORD63 = 000176	WORD85 = 000252
T#RSET = 040000	WORD21 = 000052	WORD42 = 000124	WORD64 = 000200	WORD86 = 000254
T#SC = 000022	WORD22 = 000054	WORD43 = 000126	WORD65 = 000202	WORD87 = 000256
T#SCL = 020000	WORD23 = 000056	WORD44 = 000130	WORD66 = 000204	WORD88 = 000260
T#SEG1 = 000000	WORD24 = 000060	WORD45 = 000132	WORD67 = 000206	WORD89 = 000262
T#SEG2 = 000001	WORD25 = 000062	WORD46 = 000134	WORD68 = 000210	WORD90 = 000264
T#SEG3 = 000002	WORD26 = 000064	WORD47 = 000136	WORD69 = 000212	WORD91 = 000266
T#SO = 000001	WORD27 = 000066	WORD48 = 000140	WORD70 = 000214	WORD92 = 000270
T#UBUS = 100000	WORD28 = 000070	WORD49 = 000142	WORD71 = 000216	WORD93 = 000272
T#1CLK = 000400	WORD29 = 000072	WORD50 = 000144	WORD72 = 000220	WORD94 = 000274
T#BEN = 000020	WORD30 = 000074	WORD51 = 000146	WORD73 = 000222	WORD95 = 000276
T7CS = 000000RG 002	WORD31 = 000076	WORD52 = 000150	WORD74 = 000224	WORD96 = 000300
UBD.IN = 000020	WORD32 = 000100	WORD53 = 000152	WORD75 = 000226	WORD97 = 000302
WORD0 = 000000	WORD33 = 000102	WORD54 = 000154	WORD76 = 000230	WORD98 = 000304
WORD1 = 000002	WORD34 = 000104	WORD55 = 000156	WORD77 = 000232	WORD99 = 000306
WORD10 = 000024	WORD35 = 000106	WORD56 = 000160	WORD78 = 000234	WORDVAL = 000310
WORD11 = 000026	WORD36 = 000110	WORD57 = 000162	WORD79 = 000236	WRITEA = ***** GX
WORD12 = 000030	WORD37 = 000112	WORD58 = 000164	WORD80 = 000240	WRITEB = ***** GX
WORD13 = 000032	WORD38 = 000114	WORD59 = 000166	WORD81 = 000242	WRITEC = ***** GX
WORD14 = 000034	WORD39 = 000116	WORD60 = 000170	WORD82 = 000244	WRITED = ***** GX
WORD15 = 000036	WORD4 = 000010	WORD61 = 000172	WORD83 = 000246	XTREAD = 001000
WORD16 = 000040				XTWRITE = 000400
WORD17 = 000042				
WORD18 = 000044				

. ABS. 000000 000
000000 001
CSTST3 000462 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3080 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
CSTST3,6STST3/-SP=C20,1JIM,C20,1JCSTST3

```

1
2 000000 .TITLE--CSTST4
3 .PSECT: CSTST4
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 000000 TCCSD::
15 000000 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
16 000006 016746 000000G 1#: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
17 000012 CALL SEQCS ;DO IT
18 000016 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
19 000024 CALL CMPA ;CHECK SECTION A
20 000030 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
21 000034 CALL SEQCS
22 000040 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
23 000046 CALL WRITEB ;WRITE SECT A OF MEMORY
24 000052 005267 000000G INC PREADD ;BUMP ADDRESS
25 000056 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
26 000064 103350 BHIS 1# ;NO
27 ;
28 000066 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
29 000074 016746 000000G 2#: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
30 000100 CALL SEQCS ;DO IT
31 000104 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
32 000112 CALL CMPBB ;CHECK SECTION B
33 000116 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
34 000122 CALL SEQCS
35 000126 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
36 000134 CALL WRITEB ;WRITE SECT B OF MEMORY
37 000140 005267 000000G INC PREADD ;BUMP ADDRESS
38 000144 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
39 000152 103350 BHIS 2# ;NO
40 ;
41 000154 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
42 000162 016746 000000G 3#: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
43 000166 CALL SEQCS ;DO IT
44 000172 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ
45 000200 CALL CMPC ;CHECK SECTION C
46 000204 016746 000000G MOV PREADD,-(SP) ;SET SEQ ADDRESS FOR WRITE
47 000210 CALL SEQCS
48 000214 016767 000000G 000000G MOV CK3,CKDATA ;TEST PATTERN FOR WRITE
49 000222 CALL WRITEB ;WRITE SECT C OF MEMORY
50 000226 005267 000000G INC PREADD ;BUMP ADDRESS
51 000232 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
52 000240 103350 BHIS 3# ;NO
53 ;
54 000242 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
55 000250 016746 000000G 4#: MOV PREADD,-(SP) ;SEQUENCE UP TO START ADDRESS
56 000254 CALL SEQCS ;DO IT
57 000260 016767 000000G 000000G MOV CK2,CKDATA ;TEST PATTERN FOR READ

```

```

58 000266          CALL    CMPDD          ;CHECK SECTION D
59 000272 016746 000000G  MOV    PREADD,-(SP)    ;SET SEQ ADDRESS FOR WRITE
60 000276          CALL    SEQCS          ;
61 000302 016767 000000G-000000G  MOV    CK3,CKDATA     ;TEST PATTERN FOR WRITE
62 000310          CALL    WRITED         ;WRITE SECT D OF MEMORY
63 000314 005267 000000G  INC    PREADD          ;BUMP ADDRESS
64 000320 026667 000004 000000G  CMP    4(SP),PREADD   ;FINISHED?
65 000326 103350          BHS    4$             ;NO
66 000330          RETURN          ;
67          ;
68          ;
69          ;
70          ;
71 000332          ;
72 000332 016667 000004 000000G  TCCSU: MOV    4(SP),PREADD   ;WORKING ADDRESS = END ADDRESS
73 000340 016746 000000G  1$: MOV    PREADD,-(SP)  ;SEQUENCE UP TO START ADDRESS
74 000344          CALL    SEQCS          ;DO IT
75 000350 016767 000000G-000000G  MOV    CK2,CKDATA     ;TEST PATTERN FOR READ
76 000356          CALL    CMPA          ;CHECK SECTION A
77 000362 016746 000000G  MOV    PREADD,-(SP)  ;SET SEQ ADDRESS FOR WRITE
78 000366          CALL    SEQCS          ;
79 000372 016767 000000G-000000G  MOV    CK3,CKDATA     ;TEST PATTERN FOR WRITE
80 000400          CALL    WRITEA        ;WRITE SECT A OF MEMORY
81 000404 162767 000001 000000G  SUB    #1,PREADD      ;BACK UP
82 000412 026667 000002 000000G  CMP    2(SP),PREADD   ;FINISHED?
83 000420 003747          BLE    1$             ;
84          ;
85 000422 016667 000004 000000G  MOV    4(SP),PREADD   ;WORKING ADDRESS = END ADDRESS
86 000430 016746 000000G  2$: MOV    PREADD,-(SP)  ;SEQUENCE UP TO START ADDRESS
87 000434          CALL    SEQCS          ;DO IT
88 000440 016767 000000G-000000G  MOV    CK2,CKDATA     ;TEST PATTERN FOR READ
89 000446          CALL    CMPBB         ;CHECK SECTION B
90 000452 016746 000000G  MOV    PREADD,-(SP)  ;SET SEQ ADDRESS FOR WRITE
91 000456          CALL    SEQCS          ;
92 000462 016767 000000G-000000G  MOV    CK3,CKDATA     ;TEST PATTERN FOR WRITE
93 000470          CALL    WRITEB        ;WRITE SECT B OF MEMORY
94 000474 162767 000001 000000G  SUB    #1,PREADD      ;BACK UP
95 000502 026667 000002 000000G  CMP    2(SP),PREADD   ;FINISHED?
96 000510 003747          BLE    2$             ;
97          ;
98 000512 016667 000004 000000G  MOV    4(SP),PREADD   ;WORKING ADDRESS = END ADDRESS
99 000520 016746 000000G  3$: MOV    PREADD,-(SP)  ;SEQUENCE UP TO START ADDRESS
100 000524          CALL    SEQCS          ;DO IT
101 000530 016767 000000G-000000G  MOV    CK2,CKDATA     ;TEST PATTERN FOR READ
102 000536          CALL    CMPC          ;CHECK SECTION C
103 000542 016746 000000G  MOV    PREADD,-(SP)  ;SET SEQ ADDRESS FOR WRITE
104 000546          CALL    SEQCS          ;
105 000552 016767 000000G-000000G  MOV    CK3,CKDATA     ;TEST PATTERN FOR WRITE
106 000560          CALL    WRITEC        ;WRITE SECT C OF MEMORY
107 000564 162767 000001 000000G  SUB    #1,PREADD      ;BACK UP
108 000572 026667 000002 000000G  CMP    2(SP),PREADD   ;FINISHED?
109 000600 003747          BLE    3$             ;
110          ;
111 000602 016667 000004 000000G  MOV    4(SP),PREADD   ;WORKING ADDRESS = END ADDRESS
112 000610 016746 000000G  4$: MOV    PREADD,-(SP)  ;SEQUENCE UP TO START ADDRESS
113 000614          CALL    SEQCS          ;DO IT
114 000620 016767 000000G-000000G  MOV    CK2,CKDATA     ;TEST PATTERN FOR READ

```

115 000626			CALL	CMPDD		;CHECK SECTION D
116 000632	016746	000000G	MOV	PREADD, -(SP)		;SET SEQ ADDRESS FOR WRITE
117 000636			CALL	SEQCS		
118 000642	016767	000000G 000000G	MOV	CK3, CKDATA		;TEST PATTERN FOR WRITE
119 000650			CALL	WRITED		;WRITE SECT D OF MEMORY
120 000654	162767	000001 000000G	SUB	#1, PREADD		;BACK-UP
121 000662	026667	000002 000000G	CMP	2(SP), PREADD		;FINISHED ?
122 000670	003747		BLE	4\$		
123						
124 000672			RETURN			
125	000001		.END			

SYMBOL-TABLE:

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLREN = 000200	Q\$REBK = 001000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PREADD = ***** GX.	Q\$RNC = 006000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$CR1 = 176420	Q\$RSC = 004000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CR2 = 176422	Q\$RSET = 000010
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$FLBR = 176424	Q\$SM = 100000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$ATTN = 000100	Q\$SP = 000120
BIT10 = 002000	BYTE48 = 000058	BYTVAL = 000144	Q\$BCL = 000001	Q\$SP2 = 000340
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CCCP = 000040	RGQ_VA = 000200
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$CHB = 000400	SEQC = ***** GX.
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX.	Q\$CHL = 000200	S\$CLR = 000000
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX.	Q\$CLR = 000040	S\$LA = 000001
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX.	Q\$CNC = 030000	S\$OB = 000005
BIT2 = 000004	BYTE53 = 000065	CMPA = ***** GX.	Q\$CP = 000060	S\$OR = 000006
BIT3 = 000010	BYTE54 = 000066	CMPBB = ***** GX.	Q\$CPC = 000010	S\$QX = 000004
BIT4 = 000020	BYTE55 = 000067	CMPCC = ***** GX.	Q\$CP2 = 000260	S\$SR = 000007
BIT5 = 000040	BYTE56 = 000070	CMPDD = ***** GX.	Q\$CSC = 010000	S\$S1 = 000010
BIT6 = 000100	BYTE57 = 000071	CNOBRE = 100000	Q\$CSEL = 000360	S\$S2 = 000014
BIT7 = 000200	BYTE58 = 000072	CPCCEN = 010000	Q\$CSET = 000002	TCCSD = 000000RG 002
BIT8 = 000400	BYTE59 = 000073	CPREAD = 040000	Q\$CSP = 020000	TCCSU = 000332RG 002
BIT9 = 001000	BYTE6 = 000006	CPWRT = 020000	Q\$DMA = 000001	TD\$CTR = 176370
BYTE0 = 000000	BYTE60 = 000074	CSADR = 000004	Q\$ENBK = 040000	TD\$CTW = 176360
BYTE1 = 000001	BYTE61 = 000075	CSEQCI = 100000	Q\$ENOP = 020000	TD\$INL = 004000
BYTE10 = 000012	BYTE62 = 000076	C\$OE = 000040	Q\$FAL = 004000	TD\$MEM = 000270
BYTE11 = 000013	BYTE63 = 000077	CSURTE = 000100	Q\$FC = 000045	TD\$PAR = 176344
BYTE12 = 000014	BYTE64 = 000100	DBR_RD = 000001	Q\$FO = 000044	TD\$OTR = 176346
BYTE13 = 000015	BYTE65 = 000101	DB\$CPP = 001457	Q\$FP = 000046	TD\$ORD = 000274
BYTE14 = 000016	BYTE66 = 000102	DB\$SPT = 000026	Q\$HBF = 000002	TD\$SW = 176376
BYTE15 = 000017	BYTE67 = 000103	DB\$TPC = 000023	Q\$ICP = 000006	TD\$TAR = 176372
BYTE16 = 000020	BYTE68 = 000104	DISPGS = 100000	Q\$IHB = 000003	TD\$TAR = 176362
BYTE17 = 000021	BYTE69 = 000105	DMAUR = 000005	Q\$IHRL = 000002	TD\$TOR = 176374
BYTE18 = 000022	BYTE7 = 000007	DMARRD = 000003	Q\$IMRP = 000007	TD\$TOR = 176364
BYTE19 = 000023	BYTE70 = 000106	DMARWR = 000004	Q\$LBD = 001000	T\$AD = 000020
BYTE2 = 000002	BYTE71 = 000107	ENBR = 010000	Q\$LBDP = 001001	T\$BA = 000002
BYTE20 = 000024	BYTE72 = 000110	LOC_EN = 000100	Q\$LBP = 000001	T\$BD = 000010
BYTE21 = 000025	BYTE73 = 000111	LOC_WA = 040000	Q\$LDCD = 000003	T\$BSO = 100000
BYTE22 = 000026	BYTE74 = 000112	LOC_WB = 100000	Q\$LDMO = 000004	T\$BT = 000020
BYTE23 = 000027	BYTE75 = 000113	MAREN1 = 000001	Q\$LDPP = 002000	T\$BTAR = 000030
BYTE24 = 000030	BYTE76 = 000114	MAREN2 = 004000	Q\$LHP = 010000	T\$BDT = 002000
BYTE25 = 000031	BYTE77 = 000115	MARLOD = 010000	Q\$MNC = 140000	T\$CD = 000100
BYTE26 = 000032	BYTE78 = 000116	MAROUT = 000002	Q\$MR = 000052	T\$DRD = 000004
BYTE27 = 000033	BYTE79 = 000117	MARLO = 002000	Q\$MRP = 000040	T\$MEM = 010000
BYTE28 = 000034	BYTE8 = 000010	MAROU = 000040	Q\$MRP2 = 000240	T\$FSAA = 000000
BYTE29 = 000035	BYTE80 = 000120	MBKALL = 001000	Q\$MSC = 040000	T\$FSAB = 000004
BYTE3 = 000003	BYTE81 = 000121	MBKCLK = 000400	Q\$MSET = 000004	T\$FSAC = 000014
BYTE30 = 000036	BYTE82 = 000122	MMADR = 000100	Q\$MSP = 100000	T\$FSB2 = 000010
BYTE31 = 000037	BYTE83 = 000123	MMLEFT = 000002	Q\$NCLK = 176000	T\$IB = 000026
BYTE32 = 000040	BYTE84 = 000124	MMOE = 000004	Q\$PP = 000100	T\$IBAR = 000024
BYTE33 = 000041	BYTE85 = 000125	MMURTE = 000010	Q\$PPSW = 000320	T\$IBE = 020000
BYTE34 = 000042	BYTE86 = 000126	MNOBRE = 100000	Q\$PP2 = 000300	T\$IBF = 040000
BYTE35 = 000043	BYTE87 = 000127	MREN1 = 000001	Q\$QHLT = 000013	T\$ICD = 000040
BYTE36 = 000044	BYTE88 = 000130	MREN2 = 020000	Q\$QL = 000043	T\$MODE = 004000
BYTE37 = 000045	BYTE89 = 000131	MSYN = 000040	Q\$QLB = 000054	T\$OB = 000036
BYTE38 = 000046	BYTE9 = 000011	N = 000144	Q\$QLR = 000001	T\$OBE = 004000
BYTE39 = 000047	BYTE90 = 000132	PLB = 000010	Q\$QLR = 000001	
BYTE4 = 000004	BYTE91 = 000133	PLC = 000020	Q\$RD = 000040	
BYTE40 = 000050	BYTE92 = 000134	PLD = 000030	Q\$RD = 000005	
BYTE41 = 000051	BYTE93 = 000135	PLRWR = 000200	Q\$RDMD = 000006	

T\$BBF = 010000	WORD16 = 000040	WORD39 = 000116	WORD61 = 000172	WORD84 = 000250
T\$DBRA = 000034	WORD17 = 000042	WORD4 = 000010	WORD62 = 000174	WORD85 = 000252
T\$DBWA = 000032	WORD18 = 000044	WORD40 = 000120	WORD63 = 000176	WORD86 = 000254
T\$OUTA = 100000	WORD19 = 000046	WORD41 = 000122	WORD64 = 000200	WORD87 = 000256
T\$RBD0 = 000200	WORD2 = 000004	WORD42 = 000124	WORD65 = 000202	WORD88 = 000260
T\$RNB = 000040	WORD20 = 000050	WORD43 = 000126	WORD66 = 000204	WORD89 = 000262
T\$RSET = 040000	WORD21 = 000052	WORD44 = 000130	WORD67 = 000206	WORD9 = 000022
T\$SC = 000022	WORD22 = 000054	WORD45 = 000132	WORD68 = 000210	WORD90 = 000264
T\$SCLK = 020000	WORD23 = 000056	WORD46 = 000134	WORD69 = 000212	WORD91 = 000266
T\$SEG1 = 000000	WORD24 = 000060	WORD47 = 000136	WORD7 = 000016	WORD92 = 000270
T\$SEG2 = 000001	WORD25 = 000062	WORD48 = 000140	WORD70 = 000214	WORD93 = 000272
T\$SEG3 = 000002	WORD26 = 000064	WORD49 = 000142	WORD71 = 000216	WORD94 = 000274
T\$SO = 000001	WORD27 = 000066	WORD5 = 000012	WORD72 = 000220	WORD95 = 000276
T\$UBUS = 100000	WORD28 = 000070	WORD50 = 000144	WORD73 = 000222	WORD96 = 000300
T\$1CLK = 000400	WORD29 = 000072	WORD51 = 000146	WORD74 = 000224	WORD97 = 000302
T\$0BEN = 000020	WORD3 = 000006	WORD52 = 000150	WORD75 = 000226	WORD98 = 000304
UBD, IN = 000020	WORD30 = 000074	WORD53 = 000152	WORD76 = 000230	WORD99 = 000306
WORD0 = 000000	WORD31 = 000076	WORD54 = 000154	WORD77 = 000232	WRDVAL = 000310
WORD1 = 000002	WORD32 = 000100	WORD55 = 000156	WORD78 = 000234	WRITEA = ***** GX
WORD10 = 000024	WORD33 = 000102	WORD56 = 000160	WORD79 = 000236	WRITEB = ***** GX
WORD11 = 000026	WORD34 = 000104	WORD57 = 000162	WORD8 = 000020	WRITEC = ***** GX
WORD12 = 000030	WORD35 = 000106	WORD58 = 000164	WORD90 = 000240	WRITED = ***** GX
WORD13 = 000032	WORD36 = 000110	WORD59 = 000166	WORD81 = 000242	XTREAD = 001000
WORD14 = 000034	WORD37 = 000112	WORD6 = 000014	WORD82 = 000244	XTWRITE = 000400
WORD15 = 000036	WORD38 = 000114	WORD60 = 000170	WORD83 = 000246	

. ABS: 000000 000
000000 001
CSTST4 000674 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3096 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:43
CSTST4, CSTST4/-SP=[20, 1]IM,[20, 1]CSTST4

```

1
2 000000 .TITLE - CSTST5..
3 .PSECT - CSTST5
4 ;
5 ;
6 ; .MCALL - OPEN$R,CLOSE$
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 000000 TDCS::
15 ;
16 ;
17 ;
18 000000 016767 000000G-000000C MOV LCS,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
19 000006 016767 000002G-000000C MOV LCS+2,INDNB+N,FNAM+2
20 000014 042767 000000G-000000G BIC #FIRST,BASE ;CLEAR FIRST TIME THROUGH FLAG
21 000022 012767 000001 000002G MOV #1,VIRT+2 ;REINIT BLOCK COUNT
22 000030 OPEN$R #INFDB
23 ;
24 ;
25 ; GET FIRST RECORD. THE FIRST WORD OF THE FIRST RECORD
26 ; CONTAINS THE NUMBER OF WORDS TO BE READ INTO AN
27 ; CP COLUMN. (SEE BELOW), SAVE THIS VALUE.
28 000046 005067 000000G CLR PREADD ;SET SEQUENCER TO ZERO
29 000052 15% CALL GET ;READ A RECORD
30 000056 103002 BCC 15% ;BRANCH IF OK
31 000060 000167 000452 JMP CPCSX ;ERROR, EXIT
32 000064 15%
33 000064 016705 000000C MOV INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ
34 000070 012704 000400 MOV #256,,R4 ;NUMBER OF WORDS IN RECORD (MAX)
35 000074 032767 000000G-000000G BIT #FIRST,BASE ;FIRST TIME THROUGH
36 000102 001011 BNE 2$ ;NO
37 000104 052767 000000G-000000G BIS #FIRST,BASE ;SET FLAG FOR FIRST TIME THROUGH
38 000112 012567 000000G MOV (R5)+,LCOUNT ;SAVE NUMBER OF WORDS IN COLUMN
39 000116 016767 000000G-000000G MOV LCOUNT,WCOUNT ;INITIALIZE WORKING COUNTER
40 000124 005304 DEC R4 ;SUB FROM NUMBER OF WORDS IN RECORD
41 ;
42 ;
43 ; EACH LOCATION IN CP CONTROL STORE CONSISTS OF FOUR WORDS,
44 ; SECTION 'A', SECTION 'B', SECTION 'C', SECTION 'D'. IN
45 ; WORDS ARE READ FIRST (IE, A COLUMN) THEN ALL RIGHT WORDS,
46 ; READING, ALL OF SECTION 'A' IS READ FIRST, THEN SECTION
47 ; 'B', 'C', 'D'. THE PROGRAM 'CONVRT' HAS WRITTEN THE FILE
48 ; LDCS.DAT TO CONTAIN CP CONTROL STORE IN COLUMNS.
49 ;
50 ;
51 ; READ SECTION 'A' OF CP CONTROL STORE.
52 000126 2$:
53 000126 012567 000000G MOV (R5)+,CKDATA ;SET TEST COUNTER WITH FILE WORD
54 000132 026766 000000G-000002C CMP PREADD,2(SP) ;AT LOWER MEMORY BOUND YET?
55 000140 103412 BLO 25$ ;SKIP COMPARE OF WORD
56 000142 026766 000000G-000004C CMP PREADD,4(SP) ;UPPER MEMORY BOUND EXCEEDED?
57 000150 101006 BHI 25$ ;SKIP COMPARE OF WORD
58 000152 016746 000000G MOV PREADD,-(SP) ;INITIALIZE SEQUENCER ADDRESS

```

```

58 000156          CALL  SEQCS          ;SET ADDRESS.
59 000162          CALL  CMPA           ;COMPARE SECTION A OF MEMORY.
60 000166          ;
61 000166 005367 000000G. 25$:
62 000172 001405          DEC  WCOUNT          ;SUB FROM # WORDS IN A COLUMN.
63 000174 005267 000000G.  BEQ  CPB           ;DO SECT B
64 000200 005304          INC  PREADD          ;ADVANCE SEQUENCER ADDRESS.
65 000202 001723          DEC  R4             ;FINISHED WITH THIS RECORD.
66 000204 000750          BEQ  1$            ;YES, GET NEXT.
67          ;          BR  2$            ;NO, RESET SEQUENCER ADDRESS.
68          ;
69          ;          READ SECTION 'B'
70 000206          ;          CPB:
71 000206 016767 000000G.000000G.  MOV  LCOUNT,WCOUNT ;REINIT WORKING COUNTER.
72 000210 005067 000000G.          CLR  PREADD          ;SET SEQUENCER TO ZERO.
73 000220 005304          DEC  R4             ;FINISHED WITH THIS RECORD.
74 000222 001007          BNE  2$            ;NO, CONTINUE.
75 000224          CALL  GET           ;READ NEXT
76 000230 103542          BCS  CPCSX         ;ERROR, EXIT.
77
78 000232 016705 000000C.  MOV  INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ.
79 000236 012704 000400          MOV  #256.,R4        ;R4 = NUMBER OF WORDS IN RECORD.
80          ;
81 000242          ;          2$:
82 000242 012567 000000G.  MOV  (R5)+,CKDATA     ;SET TEST COUNTER WITH FILE WORD.
83 000246 026766 000000G.000002.  CMP  PREADD,2(SP)    ;AT LOWER MEMORY BOUND YET?
84 000254 103412          BLO  25$           ;SKIP COMPARE OF WORD.
85 000256 026766 000000G.000004.  CMP  PREADD,4(SP)    ;UPPER MEMORY BOUND EXCEEDED?
86 000264 101006          BHI  25$           ;SKIP COMPARE OF WORD.
87 000266 016746 000000G.  MOV  PREADD,-(SP)    ;INITIALIZE SEQUENCER ADDRESS.
88 000272          CALL  SEQCS          ;SET ADDRESS.
89 000276          CALL  CMPBB         ;COMPARE SECT B OF MEMORY.
90 000302          ;
91 000302 005367 000000G. 25$:
92 000306 001403          DEC  WCOUNT          ;FINISHED WITH THIS COLUMN.
93 000310 005267 000000G.  BEQ  CPC           ;YES, GET NEXT.
94 000314 000741          INC  PREADD          ;NO, ADVANCE SEQUENCER ADDRESS.
95          ;          BR  1$            ;SET IT.
96          ;
97          ;          READ SECTION 'C'
98 000316          ;          CPC:
99 000316 016767 000000G.000000G.  MOV  LCOUNT,WCOUNT ;REINITIALIZE WORKING COUNTER.
100 000324 005067 000000G.          CLR  PREADD          ;INIT SEQUENCER = 0
101 000330 005304          DEC  R4             ;FINISHED WITH THIS RECORD.
102 000332 001007          BNE  2$            ;NO, CONTINUE.
103 000334          CALL  GET           ;READ NEXT
104 000340 103476          BCS  CPCSX         ;ERROR, EXIT.
105 000342 016705 000000C.  MOV  INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ.
106 000346 012704 000400          MOV  #256.,R4        ;R4 = NUMBER OF WORDS IN RECORD.
107          ;
108 000352          ;          2$:
109 000352 012567 000000G.  MOV  (R5)+,CKDATA     ;SET TEST COUNTER WITH FILE WORD.
110 000356 026766 000000G.000002.  CMP  PREADD,2(SP)    ;AT LOWER MEMORY BOUND YET?
111 000364 103412          BLO  25$           ;SKIP COMPARE OF WORD.
112 000366 026766 000000G.000004.  CMP  PREADD,4(SP)    ;UPPER MEMORY BOUND EXCEEDED?
113 000374 101006          BHI  25$           ;SKIP COMPARE OF WORD.
114 000376 016746 000000G.  MOV  PREADD,-(SP)    ;INITIALIZE SEQUENCER ADDRESS.

```



```

115 000402.          CALL  SEQCS          ;SET ADDRESS.
116 000406          CALL  CMPC           ;COMPARE SECT. D OF MEMORY.
117 000412.          25$:
118 000412. 005367 000000G.  DEC  WCOUNT.      ;FINISHED WITH THIS COLUMN.
119 000416 001403    BEQ  CPD           ;YES, GET NEXT.
120 000420 005267    INC  PREADD.     ;NO, ADVANCE SEQUENCER ADDRESS.
121 000424 000741    BR   1$         ;SET IT.
122.                ;
123.                ;
124.                ;
125 000426          CPD:
126 000426 016767 000000G.000000G.  MOV  LCOUNT,WCOUNT. ;REINITIALIZE WORKING COUNTER.
127 000434 005067 000000G.          CLR  PREADD.         ;INIT SEQUENCER = 0
128 000440 005304          1$:      DEC  R4              ;FINISHED WITH THIS RECORD.
129 000442 001007          BNE  2$         ;NO, CONTINUE.
130 000444          CALL  GET          ;READ NEXT
131 000450 103432.        BCS  CPCSX.         ;ERROR, EXIT.
132 000452 016705 000000C.        MOV  INFDB+F,BKDS+2,R5 ;POINT TO RECORD READ.
133 000456 012704 000400          MOV  #256.,R4         ;R4 = NUMBER OF WORDS IN RECORD.
134.                ;
135 000462.          2$:
136 000462 012567 000000G.          MOV  (R5)+,CKDATA.    ;SET TEST COUNTER WITH FILE WORD.
137 000466 026766 000000G.000002.  CMP  PREADD,2(SP)   ;AT LOWER MEMORY BOUND YET?
138 000474 103412.        BLO  25$         ;SKIP COMPARE OF WORD.
139 000476 026766 000000G.000004  CMP  PREADD,4(SP)   ;UPPER MEMORT BOUND EXCEEDED?
140 000504 101006          BHI  25$         ;SKIP COMPARE OF WORD.
141 000506 016746 000000G.        MOV  PREADD,-(SP)   ;INITIALIZE SEQUENCER ADDRESS.
142 000512.          CALL  SEQCS          ;SET ADDRESS.
143 000516          CALL  CMPDD.        ;COMPARE SECTION D OF MEMORY.
144 000522.          25$:
145 000522 005367 000000G.  DEC  WCOUNT.      ;FINISHED WITH THIS COLUMN.
146 000526 001403    BEQ  CPCSX.     ;YES, ALL DONE.
147 000530 005267    INC  PREADD.     ;NO, ADVANCE SEQUENCER ADDRESS.
148 000534 000741    BR   1$         ;SET IT.
149.                ;
150 000536          CPCSX:
151 000536          CLOSE$ #INFDB.
152 000546 105067 000000C.        CLR  INDNB+N,FVER.  ;RESET FILE VERSION NUMBER.
153 000552.          RETURN.
154 000001          .END.

```

ALUCKE = 040000	BYTE41 = 000051	BYTE93 = 000135	MMLEFT = 000002	Q\$MSC = 040000
ALUOE = 004000	BYTE42 = 000052	BYTE94 = 000136	MMOE = 000004	Q\$MSET = 000004
A01 = 010000	BYTE43 = 000053	BYTE95 = 000137	MMWRTE = 000010	Q\$MSP = 100000
BASE = ***** GX	BYTE44 = 000054	BYTE96 = 000140	MNOBRE = 100000	Q\$NCLK = 176000
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	MREN1 = 000001	Q\$PP = 000100
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	MREN2 = 020000	Q\$PPSW = 000320
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	MSYN = 000040	Q\$PP2 = 000300
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	N = 000144	Q\$QHLT = 000013
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	N.FNAM = ***** GX	Q\$QL = 000043
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	N.FVER = ***** GX	Q\$QLA = 000053
BIT13 = 020000	BYTE51 = 000063	CKDATA = ***** GX	PAR\$\$\$ = 000027	Q\$QLB = 000054
BIT14 = 040000	BYTE52 = 000064	CMPA = ***** GX	PLB = 000010	Q\$QLR = 000001
BIT15 = 100000	BYTE53 = 000065	CMPBB = ***** GX	PLC = 000020	Q\$QW = 000042
BIT2 = 000004	BYTE54 = 000066	CMPFC = ***** GX	PLD = 000030	Q\$RDCD = 000005
BIT3 = 000010	BYTE55 = 000067	CMPDD = ***** GX	PLRWR = 000200	Q\$RDMD = 000006
BIT4 = 000020	BYTE56 = 000070	CNOBRE = 100000	PLR.EN = 000200	Q\$REBK = 001000
BIT5 = 000040	BYTE57 = 000071	CPB = 000206R	002 PREAD = ***** GX	Q\$RNC = 006000
BIT6 = 000100	BYTE58 = 000072	CPC = 000316R	002 Q\$RCR1 = 176420	Q\$RSC = 004000
BIT7 = 000200	BYTE59 = 000073	CPCCEN = 010000	002 Q\$RCR2 = 176422	Q\$RSET = 000010
BIT8 = 000400	BYTE60 = 000074	CPCSX = 000536R	002 Q\$RLBR = 176424	Q\$SM = 100000
BIT9 = 001000	BYTE61 = 000075	CPD = 000426R	002 Q\$ATTN = 000100	Q\$SP = 000120
BYTE0 = 000000	BYTE62 = 000076	CPREAD = 040000	Q\$BCL = 000001	Q\$SP2 = 000340
BYTE1 = 000001	BYTE63 = 000077	CPWRTE = 020000	Q\$CCCP = 000040	RGQ.EN = 000200
BYTE10 = 000012	BYTE64 = 000100	CSADRD = 000004	Q\$CHB = 000400	RGQ.VA = 020000
BYTE11 = 000013	BYTE65 = 000101	CSEQCI = 100000	Q\$CHRL = 000200	SEQCS = ***** GX
BYTE12 = 000014	BYTE66 = 000102	CSOE = 000040	Q\$CLR = 000040	SEQ.CI = 000010
BYTE13 = 000015	BYTE67 = 000103	CSWRTE = 000100	Q\$CNC = 030000	S\$CLR = 000000
BYTE14 = 000016	BYTE68 = 000104	DBR.RD = 000001	Q\$CP = 000060	S\$LA = 000001
BYTE15 = 000017	BYTE69 = 000105	DB\$CPP = 001457	Q\$CPC = 000010	S\$QB = 000005
BYTE16 = 000020	BYTE70 = 000106	DB\$SPT = 000026	Q\$CP2 = 000260	S\$QR = 000006
BYTE17 = 000021	BYTE71 = 000107	DB\$TPC = 000023	Q\$CSC = 010000	S\$QX = 000004
BYTE18 = 000022	BYTE72 = 000110	DISPCS = 100000	Q\$CSEL = 000360	S\$SR = 000007
BYTE19 = 000023	BYTE73 = 000111	DMARUR = 000005	Q\$CSET = 000002	S\$S1 = 000010
BYTE2 = 000002	BYTE74 = 000112	DMARRD = 000003	Q\$C5P = 020000	S\$S2 = 000014
BYTE20 = 000024	BYTE75 = 000113	DMARWR = 000004	Q\$DMA = 000001	TDCS 000000RG 002
BYTE21 = 000025	BYTE76 = 000114	ENBR = 010000	Q\$ENBK = 040000	TD\$CTR = 176370
BYTE22 = 000026	BYTE77 = 000115	FIRST = ***** GX	Q\$ENOP = 020000	TD\$CTW = 176360
BYTE23 = 000027	BYTE78 = 000116	FO.RD = ***** GX	Q\$FAL = 004000	TD\$INL = 004000
BYTE24 = 000030	BYTE79 = 000117	F.BKDS = ***** GX	Q\$FC = 000045	TD\$NEM = 000270
BYTE25 = 000031	BYTE80 = 000120	F.FACC = ***** GX	Q\$FO = 000044	TD\$OAR = 176344
BYTE26 = 000032	BYTE81 = 000121	GET = ***** GX	Q\$FP = 000046	TD\$OTR = 176346
BYTE27 = 000033	BYTE82 = 000122	INDNB = ***** GX	Q\$HBF = 000002	TD\$ORD = 000274
BYTE28 = 000034	BYTE83 = 000123	INFDB = ***** GX	Q\$ICP = 000006	TD\$SW = 176376
BYTE29 = 000035	BYTE84 = 000124	LCOUNT = ***** GX	Q\$IHG = 000003	TD\$TAR = 176372
BYTE3 = 000003	BYTE85 = 000125	LCS = ***** GX	Q\$IHRL = 000002	TD\$TAW = 176362
BYTE30 = 000036	BYTE86 = 000126	LOC.EN = 000100	Q\$IHRP = 000007	TD\$TDR = 176374
BYTE31 = 000037	BYTE87 = 000127	LOC.WA = 040000	Q\$LBD = 001000	TD\$TDW = 176364
BYTE32 = 000040	BYTE88 = 000130	LOC.WB = 100000	Q\$LBDP = 001001	T\$AD = 000020
BYTE33 = 000041	BYTE89 = 000131	MAREN1 = 000001	Q\$LBP = 000001	T\$BA = 000002
BYTE34 = 000042	BYTE90 = 000133	MAREN2 = 004000	Q\$LDCD = 000003	T\$BD = 000010
BYTE35 = 000043	BYTE91 = 000133	MARLOD = 010000	Q\$LDMD = 000004	T\$BSO = 100000
BYTE36 = 000044	BYTE92 = 000134	MAROUT = 000002	Q\$LDFF = 002000	T\$BT = 000020
BYTE37 = 000045		MAR.LO = 002000	Q\$LHP = 010000	T\$BTAR = 000030
BYTE38 = 000046		MAR.OU = 000040	Q\$MNC = 140000	T\$BTD = 000000
BYTE39 = 000047		MBKALL = 001000	Q\$MR = 000002	T\$BTDL = 000000
BYTE4 = 000004		MBKCLK = 000400	Q\$MRP = 000040	T\$BTDL = 000100
BYTE40 = 000050		MMADR = 000100	Q\$MRP2 = 000240	T\$CLK = 002000
				T\$DISK = 000200

T\$DRD = 000004	T\$JCLK = 000400	WORD3 = 000006	WORD55 = 000156	WORD8 = 000020
T\$EMEM = 010000	T\$BBEN = 000020	WORD30 = 000074	WORD56 = 000160	WORD80 = 000240
T\$FSAA = 000000	USD.IN = 000020	WORD31 = 000076	WORD57 = 000162	WORD81 = 000242
T\$FSAB = 000004	VIRT = ***** GX	WORD32 = 000100	WORD58 = 000164	WORD82 = 000244
T\$FSAC = 000014	WCOUNT = ***** GX	WORD33 = 000102	WORD59 = 000166	WORD83 = 000246
T\$FSB2 = 000010	WORD0 = 000000	WORD34 = 000104	WORD6 = 000014	WORD84 = 000250
T\$IB = 000026	WORD1 = 000002	WORD35 = 000106	WORD60 = 000170	WORD85 = 000252
T\$IBAR = 000024	WORD10 = 000024	WORD36 = 000110	WORD61 = 000172	WORD86 = 000254
T\$IBE = 020000	WORD11 = 000026	WORD37 = 000112	WORD62 = 000174	WORD87 = 000256
T\$IBF = 040000	WORD12 = 000030	WORD38 = 000114	WORD63 = 000176	WORD88 = 000260
T\$ICD = 000040	WORD13 = 000032	WORD39 = 000116	WORD64 = 000200	WORD89 = 000262
T\$MODE = 004000	WORD14 = 000034	WORD4 = 000010	WORD65 = 000202	WORD9 = 000022
T\$OB = 000036	WORD15 = 000036	WORD40 = 000120	WORD66 = 000204	WORD90 = 000264
T\$OBE = 004000	WORD16 = 000040	WORD41 = 000122	WORD67 = 000206	WORD91 = 000266
T\$OBF = 010000	WORD17 = 000042	WORD42 = 000124	WORD68 = 000210	WORD92 = 000270
T\$OBRA = 000034	WORD18 = 000044	WORD43 = 000126	WORD69 = 000212	WORD93 = 000272
T\$OBWA = 000032	WORD19 = 000046	WORD44 = 000130	WORD7 = 000016	WORD94 = 000274
T\$OUTA = 100000	WORD2 = 000004	WORD45 = 000132	WORD70 = 000214	WORD95 = 000276
T\$RBD0 = 000200	WORD20 = 000050	WORD46 = 000134	WORD71 = 000216	WORD96 = 000300
T\$RNB = 000040	WORD21 = 000052	WORD47 = 000136	WORD72 = 000220	WORD97 = 000302
T\$RSET = 040000	WORD22 = 000054	WORD48 = 000140	WORD73 = 000222	WORD98 = 000304
T\$SC = 000022	WORD23 = 000056	WORD49 = 000142	WORD74 = 000224	WORD99 = 000306
T\$SCLK = 020000	WORD24 = 000060	WORDS = 000012	WORD75 = 000226	WRDVAL = 000310
T\$SEG1 = 000000	WORD25 = 000062	WORD50 = 000144	WORD76 = 000230	XTREAD = 001000
T\$SEG2 = 000001	WORD26 = 000064	WORD51 = 000146	WORD77 = 000232	XTWRITE = 000400
T\$SEG3 = 000002	WORD27 = 000066	WORD52 = 000150	WORD78 = 000234	.CLOSE = ***** G
T\$SD = 000001	WORD28 = 000070	WORDS3 = 000152	WORD79 = 000236	.OPEN = ***** G
T\$UBUS = 100000	WORD29 = 000072	WORDS4 = 000154		

. ABS: 000000 000
000000 001
CSTSTS 000554 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3940 WORDS (16 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:46
CSTSTS.6STSTS/-SP=C20.1]IM.[20.1]CSTSTS

```

1          ,TITLE--QBTEST...
2 000000   ,PSECT--QBTEST-
3          ;
4          ;
5          ;
6          ;
7          ;
8          ;
9          ;
10         ;
11         ;
12         ;
13 000000   ;
14 000000   016746 000000G. STUFQB: MOV    QBPAGE,-(SP)      ;SET PAGE NUMBER ARGUMENT
15 000004   016667 000002 000000G. CALL  SELPG          ;SELECT MEMORY PAGE
16 000010   016667 000002 000000G. MOV    2(SP),PREADD. ;WORKING ADDRESS
17 000016   005267 000000G. 1$: CALL  WOB          ;WRITE QLB MEMORY
18 000022   026667 000004 000000G. INC   PREADD.       ;BUMP ADDRESS
19 000026   103370 000004 000000G. CMP   4(SP),PREADD. ;FINISHED?
20 000034   103370 000004 000000G. BHIS  1$           ;NO
21         ;
22 000036   012746 000013   MOV    #0$QHLT,-(SP) ;SET HALT CODE
23 000042   012746 000040   CALL  PPCR          ;CLEAR PPS
24 000046   012746 000040   MOV    #0$CLR,-(SP) ;CLEAR PPS
25 000052   016667 000002 000000G. CALL  PPCR
26 000056   005267 000000G. 2$: MOV    2(SP),PREADD. ;WORKING ADDRESS
27 000064   005267 000000G. CALL  COB          ;READ AND COMPARE QLB MEMORY
28 000070   026667 000004 000000G. INC   PREADD.       ;BUMP ADDRESS
29 000074   103370 000004 000000G. CMP   4(SP),PREADD. ;FINISHED?
30 000102   103370 000004 000000G. BHIS  2$           ;NO
31         ;
32 000104   012746 000013   MOV    #0$QHLT,-(SP) ;SET HALT CODE
33 000110   012746 002000   CALL  PPCR
34 000114   012746 002000   MOV    #2000,-(SP)  ;=X'400' (ILLEGAL ADDRESS)
35 000120   012746 000040   CALL  LBPP
36 000124   012746 000040   MOV    #0$CLR,-(SP) ;CLEAR PPS
37 000130   012746 000040   CALL  PPCR
38 000134   012746 000040   RETURN

```

```

40 ;
41 ;
42 ; TEST·01
43 ; WRITE·MEMORY·ADDRESS·INTO·MEMORY·LOCATION·
44 ;
45 ;
46 000136 ; T10B::
47 000136 016746 000000G MOV QBPAGE,-(SP) ;SET·PAGE·NUMBER·ARGUMENT·
48 000142 CALL SELPG ;SELECT·MEMORY·PAGE·
49 000146 016667 000002 000000G MOV 2(SP),PREADD ;WORKING·ADDRESS·
50 000154 016667 000002 000000G MOV 2(SP),CKDATA ;TEST·PATTERN·=·ADDRESS·
51 000162 1$: CALL WOB ;WRITE·DLB·MEMORY·
52 000166 005267 000000G INC CKDATA ;BUMP·TEST·COUNTER·
53 000172 005267 000000G INC PREADD ;BUMP·ADDRESS·
54 000176 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED·?·
55 000204 103366 BHIS 1$ ;NO·
56 ;
57 000206 012746 000013 MOV #0$QHLT,-(SP) ;SET·HALT·CODE·
58 000212 CALL PPCR
59 000216 012746 000040 MOV #0$CLR,-(SP) ;CLEAR·PPS
60 000222 CALL PPCR
61 000226 016667 000002 000000G MOV 2(SP),PREADD ;WORKING·ADDRESS·
62 000234 016667 000002 000000G MOV 2(SP),CKDATA ;TEST·PATTERN·=·ADDRESS·
63 000242 2$: CALL CQB ;READ·AND·COMPARE·DLB·MEMORY·
64 000246 005267 000000G INC CKDATA ;BUMP·TEST·COUNTER·
65 000252 005267 000000G INC PREADD ;BUMP·ADDRESS·
66 000256 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED·?·
67 000264 103366 BHIS 2$ ;NO·
68 ;
69 000266 012746 000013 MOV #0$QHLT,-(SP) ;SET·HALT·CODE·
70 000272 CALL PPCR
71 000276 012746 002000 MOV #2000,-(SP) ;='X'400' (ILLEGAL·ADDRESS)
72 000302 CALL LBPP
73 000306 012746 000040 MOV #0$CLR,-(SP) ;CLEAR·PPS
74 000312 CALL PPCR
75 000316 RETURN

```

```

77 ;
78 ;
79 ; TEST-06
80 ; CROSS-TALK TEST.
81 ;
82 ;
83 000320 ; T6QB:
84 000320 016746 000000G MOV QBPAGE,-(SP) ;SET PAGE NUMBER ARGUMENT.
85 000324 CALL SELPG ;SELECT MEMORY PAGE.
86 000330 012767 177777 000000G MOV #-1,CKDATA ;SET TEST PATTERN =X'FFFF'
87 000336 012702 000012 MOV #10,R2 ;SET LOOP COUNT.
88 000342 016667 000002 000000G 10$: MOV 2(SP),PREADD ;WORKING ADDRESS.
89 000350 1$: CALL WQB ;WRITE OLB MEMORY.
90 000354 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS.
91 000362 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
92 000370 103367 BHIS 1$ ;NO.
93 000372 005302 DEC R2 ;SUB FROM LOOP COUNT.
94 000374 001362 BNE 10$
95 ;
96 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
97 ; WERE NOT WRITTEN.
98 ;
99 000376 ; R6Z:
100 000376 012746 000013 MOV #0,$QHLT,-(SP) ;SET HALT CODE.
101 000402 CALL PPCR
102 000406 012746 000040 MOV #0,$CLR,-(SP) ;CLEAR PPS
103 000412 CALL PPCR
104 000416 005067 000000G CLR CKDATA ;SET TEST PATTERN = 0
105 000422 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
106 000430 005267 000000G INC PREADD ;BUMP 1 ADDRESS.
107 000434 1$: CALL CQB ;READ AND COMPARE OLB MEMORY.
108 000440 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS.
109 000446 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
110 000454 103367 BHIS 1$ ;NO.
111 ;
112 000456 012746 000013 MOV #0,$QHLT,-(SP) ;SET HALT CODE.
113 000462 CALL PPCR
114 000466 012746 002000 MOV #2000,-(SP) ;=X'400' (ILLEGAL ADDRESS)
115 000472 CALL LBPP
116 000476 012746 000040 MOV #0,$CLR,-(SP) ;CLEAR PPS
117 000502 CALL PPCR
118 000506 RETURN

```

```

120 ;
121 ;
122 ; TEST-07
123 ; WRITE-COMPLEMENT-OF-MEMORY-ADDRESS-INTO-MEMORY-LOCATION.
124 ;
125 ;
126 000510 ; T7QB::
127 000510 016746 000000G MOV QBPAGE,-(SP) ;SET-PAGE-NUMBER-ARGUMENT.
128 000514 CALL SELPG ;SELECT-MEMORY-PAGE.
129 000520 016667 000002 000000G MOV 2(SP),PREADD ;WORKING-ADDRESS.
130 000526 016602 000002 MOV 2(SP),R2 ;TEST-PATTERN=-ADDRESS.
131 000532 005102 1$: COM R2 ;GET-ADDRESS-COMPLEMENT.
132 000534 010267 000000G MOV R2,CKDATA ;SET-TEST-PATTERN.
133 000540 CALL WQB ;WRITE-QLB-MEMORY.
134 000544 005267 000000G INC PREADD ;BUMP-ADDRESS.
135 000550 016702 000000G MOV PREADD,R2 ;SET-UP-FOR-NEXT-TIME.
136 000554 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED-?.
137 000562 103363 BHIS 1$ ;NO.
138 ;
139 000564 012746 000013 MOV #0$QHLT,-(SP) ;SET-HALT-CODE.
140 000570 CALL PPCR
141 000574 012746 000040 MOV #0$CLR,-(SP) ;CLEAR-PPS
142 000600 CALL PPCR
143 000604 016667 000002 000000G MOV 2(SP),PREADD ;WORKING-ADDRESS.
144 000612 016602 000002 MOV 2(SP),R2 ;TEST-PATTERN=-ADDRESS.
145 000616 005102 2$: COM R2 ;GET-ADDRESS-COMPLEMENT.
146 000620 010267 000000G MOV R2,CKDATA ;SET-TEST-PATTERN.
147 000624 CALL CQB ;READ-AND-COMPARE-QLB-MEMORY.
148 000630 005267 000000G INC PREADD ;BUMP-ADDRESS.
149 000634 016702 000000G MOV PREADD,R2 ;SET-UP-FOR-NEXT-TIME.
150 000640 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED-?.
151 000646 103363 BHIS 2$ ;NO.
152 ;
153 000650 012746 000013 MOV #0$QHLT,-(SP) ;SET-HALT-CODE.
154 000654 CALL PPCR
155 000660 012746 002000 MOV #2000,-(SP) ;='X'400' (ILLEGAL-ADDRESS)
156 000664 CALL LBPP
157 000670 012746 000040 MOV #0$CLR,-(SP) ;CLEAR-PPS
158 000674 CALL PPCR
159 000700 RETURN

```

```

161 ;
162 ;
163 ; TEST-12
164 ; LOOK FORWARD, LOOK BEHIND ADDRESSING TEST
165 ;
166 ;
167 ; READ FROM TOP OF MEMORY DOWN, THEN WRITE
168 ;
169 000702. TCQBD::
170 000702. 016746 000000G. MOV. QBPAGE, -(SP) ; SET PAGE NUMBER ARGUMENT
171 000706. CALL. SELPG. ; SELECT MEMORY PAGE
172 000712. 016667 000002 000000G. MOV. 2(SP), PREADD. ; WORKING ADDRESS
173 000720. 016767 000000G 000000G 1$. MOV. CK2, CKDATA. ; TEST PATTERN FOR READ
174 000726. CALL. CQB. ; CHECK MEMORY LOCATION
175 000732. 016767 000000G 000000G. MOV. CK3, CKDATA. ; TEST PATTERN FOR WRITE
176 000740. CALL. WQB. ; WRITE QLB MEMORY
177 000744. 005267 000000G. INC. PREADD. ; BUMP ADDRESS
178 000750. 026667 000004 000000G. CMP. 4(SP), PREADD. ; FINISHED?
179 000756. 103360. BHS. 1$. ; NO
180 ;
181 000760. 012746 000013. MOV. #0$QHLT, -(SP) ; SET HALT CODE
182 000764. CALL. PPCR. ; SEND X'400' (ILLEGAL ADDRESS)
183 000770. 012746 002000. MOV. #2000, -(SP) ; CLEAR PPS
184 000774. CALL. LBPP. ; CLEAR PPS
185 001000. 012746 000040. MOV. #0$CLR, -(SP) ; CLEAR PPS
186 001004. CALL. PPCR. ; CLEAR PPS
187 001010. RETURN.
188 ;
189 ; TEST-12
190 ; READ FROM BOTTOM OF MEMORY UP, THEN WRITE
191 ;
192 001012. TCQBU::
193 001012. 016667 000004 000000G. MOV. 4(SP), PREADD. ; WORKING ADDRESS = END ADDRESS
194 001020. 016767 000000G 000000G 1$. MOV. CK2, CKDATA. ; TEST PATTERN FOR READ
195 001026. CALL. CQB. ; CHECK MEMORY LOCATION
196 001032. 016767 000000G 000000G. MOV. CK3, CKDATA. ; TEST PATTERN FOR WRITE
197 001040. CALL. WQB. ; WRITE MEMORY LOCATION
198 001044. 162767 000001 000000G. SUB. #1, PREADD. ; BACK UP 1
199 001052. 026667 000002 000000G. CMP. 2(SP), PREADD. ; FINISHED?
200 001060. 003757. BLE. 1$. ; NO
201 ;
202 001062. 012746 000013. MOV. #0$QHLT, -(SP) ; SET HALT CODE
203 001066. CALL. PPCR. ; SEND X'400' (ILLEGAL ADDRESS)
204 001072. 012746 002000. MOV. #2000, -(SP) ; CLEAR PPS
205 001076. CALL. LBPP. ; CLEAR PPS
206 001102. 012746 000040. MOV. #0$CLR, -(SP) ; CLEAR PPS
207 001106. CALL. PPCR. ; CLEAR PPS
208 001112. RETURN.

```



```

210 ;
211 ;
212 ; WRITE-QLB-MEMORY.
213 ;
214 ;
215 ; QCB:
216 001114 012746 000053 MOV #0$QLA,-(SP) ;SELECT-QLB-MEMORY.
217 001120 CALL PPCR ;
218 001124 016746 000000G MOV PREADD,-(SP) ;MEMORY-ADDRESS.
219 001130 CALL LBPP ;
220 001134 012746 000054 MOV #0$QLB,-(SP) ;SELECT-QLB-PAGES.
221 001140 CALL PPCR ;
222 001144 016746 000000G MOV CKDATA,-(SP) ;TEST PATTERN.
223 001150 CALL LBPP ;
224 001154 RETURN ;
225 ;
226 ;
227 ; READ-AND-COMPARE-QLB-MEMORY.
228 ;
229 ;
230 ; CQB:
231 001156 012746 000053 MOV #0$QLA,-(SP) ;SELECT-QLB-MEMORY.
232 001162 CALL PPCR ;
233 001166 016746 000000G MOV PREADD,-(SP) ;MEMORY-ADDRESS.
234 001172 CALL LBPP ;
235 001176 012746 000054 MOV #0$QLB,-(SP) ;SELECT-NON-REF-PAGE
236 001202 CALL PPCR ;
237 001206 CALL PPLB ;
238 001212 012667 000000G MOV (SP)+,ERW1 ;
239 ;
240 001216 026767 000000G-000000G CMP CKDATA,ERW1 ;SAME-AS-PATTERN-WRITTEN.
241 001224 001410 BEQ 1$ ;YES,EXIT
242 001226 016767 000000G-000000G MOV PREADD,ERRADD ;ADDRESS-OF-ERROR.
243 001234 012767 000001 000000G MOV #1,ERRCT ;NUMBER-OF-WORDS-TO-PRINT.
244 001242 CALL MEMERR ;GO-TO-ERROR-ROUTINE.
245 001246 1$: RETURN ;
246 ;
247 000001 ; .END.

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLD = 000030	Q\$QLB = 000054
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLRWR = 000200	Q\$QLR = 000001
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLR:EN = 000200	Q\$QW = 000042
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PPCR = ***** GX	Q\$RDCD = 000005
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PPLB = ***** GX	Q\$RDMD = 000006
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PREAD = ***** GX	Q\$REBK = 001000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	QBPAGE = ***** GX	Q\$RNC = 006000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	QR\$CR1 = 176420	Q\$RSC = 004000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	QR\$CR2 = 176422	Q\$RSET = 000010
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	QR\$LBR = 176424	Q\$SM = 100000
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	Q\$CCCP = 000040	RG0:EN = 000200
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	Q\$CHB = 000400	RG0:VA = 020000
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q\$CHRL = 000200	R6Z = 000376R 002
BIT5 = 000040	BYTE56 = 000070	CPWRTE = 020000	Q\$CLR = 000040	SELPG = ***** GX
BIT6 = 000100	BYTE57 = 000071	C0B = 001156R 002	Q\$CNC = 030000	SEQ:CI = 000010
BIT7 = 000200	BYTE58 = 000072	CSADRD = 000004	Q\$CP = 000060	STUFQB = 000000R 002
BIT8 = 000400	BYTE59 = 000073	CSEQCI = 100000	Q\$CPC = 000010	S\$CLR = 000000
BIT9 = 001000	BYTE6 = 000006	C\$OE = 000040	Q\$CP2 = 000260	S\$LA = 000001
BYTE0 = 000000	BYTE60 = 000074	CSWRTE = 000100	Q\$CSC = 010000	S\$OB = 000005
BYTE1 = 000001	BYTE61 = 000075	DBR:RD = 000001	Q\$CSEL = 000360	S\$OR = 000006
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	Q\$CSET = 000002	S\$OX = 000004
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	Q\$CSP = 020000	S\$SR = 000007
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	Q\$DMA = 000001	S\$S1 = 000010
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q\$ENBK = 040000	S\$S2 = 000014
BYTE14 = 000016	BYTE66 = 000102	DMAUR = 000005	Q\$ENOP = 020000	TC0B = 000702RG 002
BYTE15 = 000017	BYTE67 = 000103	DHARRD = 000003	Q\$FAL = 004000	TC0BU = 001012RG 002
BYTE16 = 000020	BYTE68 = 000104	DHARWR = 000004	Q\$FC = 000045	TD\$CTR = 176370
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$FO = 000044	TD\$CTW = 176360
BYTE18 = 000022	BYTE7 = 000007	ERRADD = ***** GX	Q\$FP = 000046	TD\$INL = 004000
BYTE19 = 000023	BYTE70 = 000106	ERRCT = ***** GX	Q\$HBF = 000002	TD\$MEM = 000270
BYTE2 = 000002	BYTE71 = 000107	ERW1 = ***** GX	Q\$ICP = 000006	TD\$OAR = 176344
BYTE20 = 000024	BYTE72 = 000110	LBPP = ***** GX	Q\$IHB = 000003	TD\$OTR = 176346
BYTE21 = 000025	BYTE73 = 000111	LOC:EN = 000100	Q\$IHRL = 000002	TD\$ORD = 000274
BYTE22 = 000026	BYTE74 = 000112	LOC:WA = 040000	Q\$IRP = 000007	TD\$SW = 176376
BYTE23 = 000027	BYTE75 = 000113	LOC:WB = 100000	Q\$LBD = 001000	TD\$TAR = 176372
BYTE24 = 000030	BYTE76 = 000114	MAREN1 = 000001	Q\$LBDP = 001001	TD\$TAJW = 176362
BYTE25 = 000031	BYTE77 = 000115	MAREN2 = 004000	Q\$LBP = 000001	TD\$TDR = 176374
BYTE26 = 000032	BYTE78 = 000116	MARLOD = 010000	Q\$LDCC = 000003	TD\$TDW = 176364
BYTE27 = 000033	BYTE79 = 000117	MAROUT = 000002	Q\$LDMD = 000004	T\$AD = 000020
BYTE28 = 000034	BYTE8 = 000010	MAR:LO = 002000	Q\$LDPP = 002000	T\$BA = 000002
BYTE29 = 000035	BYTE80 = 000120	MAR:OU = 000040	Q\$LHP = 010000	T\$BD = 000010
BYTE3 = 000003	BYTE81 = 000121	MBKALL = 001000	Q\$MNC = 140000	T\$BSO = 100000
BYTE30 = 000036	BYTE82 = 000122	MBKCLK = 000400	Q\$MR = 000052	T\$BT = 000020
BYTE31 = 000037	BYTE83 = 000123	MEMERR = ***** GX	Q\$MRP = 000040	T\$BTAR = 000030
BYTE32 = 000040	BYTE84 = 000124	MHARRD = 000100	Q\$MRP2 = 000240	T\$BTD = 000200
BYTE33 = 000041	BYTE85 = 000125	MHLEFT = 000002	Q\$MSC = 040000	T\$CD = 000100
BYTE34 = 000042	BYTE86 = 000126	MHDE = 000004	Q\$MSET = 000004	T\$CLK = 002000
BYTE35 = 000043	BYTE87 = 000127	MHURTE = 000010	Q\$MSP = 100000	T\$DISK = 000200
BYTE36 = 000044	BYTE88 = 000130	MNOBRE = 100000	Q\$NCLK = 176000	T\$DRD = 000004
BYTE37 = 000045	BYTE89 = 000131	MREN1 = 000001	Q\$PP = 000100	T\$EMEM = 010000
BYTE38 = 000046	BYTE9 = 000011	MREN2 = 020000	Q\$PPSW = 000320	T\$FSAA = 000000
BYTE39 = 000047	BYTE90 = 000132	MSYN = 000040	Q\$PP2 = 000300	T\$FSAB = 000004
BYTE4 = 000004	BYTE91 = 000133	N = 000144	Q\$QHLT = 000013	T\$FSAC = 000014
BYTE40 = 000050	BYTE92 = 000134	PLB = 000010	Q\$QL = 000043	T\$FSB2 = 000010
BYTE41 = 000051	BYTE93 = 000135	PLC = 000020	Q\$QLA = 000053	T\$IB = 000026

T#IBAR = 000024	WORD0 = 000000	WORD32 = 000100	WORD56 = 000160	WORD80 = 000220
T#IBE = 020000	WORD1 = 000002	WORD33 = 000102	WORD57 = 000162	WORD81 = 000242
T#IBF = 040000	WORD10 = 000024	WORD34 = 000104	WORD58 = 000164	WORD82 = 000244
T#ICD = 000040	WORD11 = 000026	WORD35 = 000106	WORD59 = 000166	WORD83 = 000246
T#MODE = 000000	WORD12 = 000030	WORD36 = 000110	WORD60 = 000170	WORD84 = 000250
T#OB = 000036	WORD13 = 000032	WORD37 = 000112	WORD61 = 000172	WORD85 = 000252
T#OBE = 000400	WORD14 = 000034	WORD38 = 000114	WORD62 = 000174	WORD86 = 000254
T#OBF = 010000	WORD15 = 000036	WORD39 = 000116	WORD63 = 000176	WORD87 = 000256
T#OBRA = 000034	WORD16 = 000040	WORD4 = 000010	WORD64 = 000200	WORD88 = 000260
T#OBWA = 000032	WORD17 = 000042	WORD40 = 000120	WORD65 = 000202	WORD89 = 000262
T#OUTA = 100000	WORD18 = 000044	WORD41 = 000122	WORD66 = 000204	WORD90 = 000264
T#RBD0 = 000200	WORD19 = 000046	WORD42 = 000124	WORD67 = 000206	WORD91 = 000266
T#RNB = 000040	WORD2 = 000004	WORD43 = 000126	WORD68 = 000210	WORD92 = 000270
T#RSET = 040000	WORD20 = 000050	WORD44 = 000130	WORD69 = 000212	WORD93 = 000272
T#SC = 000022	WORD21 = 000052	WORD45 = 000132	WORD7 = 000016	WORD94 = 000274
T#SCLK = 020000	WORD22 = 000054	WORD46 = 000134	WORD70 = 000214	WORD95 = 000276
T#SEG1 = 000000	WORD23 = 000056	WORD47 = 000136	WORD71 = 000216	WORD96 = 000300
T#SEG2 = 000001	WORD24 = 000060	WORD48 = 000140	WORD72 = 000220	WORD97 = 000302
T#SEG3 = 000002	WORD25 = 000062	WORD49 = 000142	WORD73 = 000222	WORD98 = 000304
T#SQ = 000001	WORD26 = 000064	WORD5 = 000012	WORD74 = 000224	WORD99 = 000306
T#UBUS = 100000	WORD27 = 000066	WORD50 = 000144	WORD75 = 000226	WOB = 001114R
T#1CLK = 000400	WORD28 = 000070	WORD51 = 000146	WORD76 = 000230	WRVAL = 000310
T#BBEN = 000020	WORD29 = 000072	WORD52 = 000150	WORD77 = 000232	XTREAD = 001000
T10B = 000136RG	002 WORD3 = 000006	WORD53 = 000152	WORD78 = 000234	XTURTE = 000400
T60B = 000320RG	002 WORD30 = 000074	WORD54 = 000154	WORD79 = 000236	
T70B = 000510RG	002 WORD31 = 000076	WORD55 = 000156		
UBD IN = 000020				

. ABS. 000000 000
000000 001
QBTEST 001250 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3150 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:47
QBTEST, QBTEST / - SP = [20, 1] IM, [20, 1] QBTEST

```

1
2 000000 .TITLE- QRTEST.
3 .PSECT- QRTEST.
4 .LIST- MEB.
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 000000 STUFQR::
13 000000 016667 000002 000000G. MOV 2(SP),PREADD. ;WORKING ADDRESS.
14 000006 004767 000752 1$: JSR PC,WOR. ;WRITE QLB REF MEMORY.
15 000012 005267 000000G. INC PREADD. ;BUMP ADDRESS.
16 000016 026667 000004 000000G. CMP 4(SP),PREADD. ;FINISHED?.
17 000024 103370 BHIS 1$ ;NO.
18 ;
19 000026 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE.
20 000032 004767 000000G. JSR PC,PPCR.
21 000036 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
22 000042 004767 000000G. JSR PC,PPCR.
23 000046 016667 000002 000000G. MOV 2(SP),PREADD. ;WORKING ADDRESS.
24 000054 004767 000746 2$: JSR PC,COR. ;READ AND COMPARE QLB REF MEMORY.
25 000060 005267 000000G. INC PREADD. ;BUMP ADDRESS.
26 000064 026667 000004 000000G. CMP 4(SP),PREADD. ;FINISHED?.
27 000072 103370 BHIS 2$ ;NO.
28 ;
29 000074 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE.
30 000100 004767 000000G. JSR PC,PPCR.
31 000104 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
32 000110 004767 000000G. JSR PC,PPCR.
33 000114 000207 RTS PC.

```

```

35 ;
36 ;
37 ; TEST-01
38 ; WRITE MEMORY ADDRESS INTO MEMORY LOCATION.
39 ;
40 ;
41 000116 ; T1QR::
42 000116 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
43 000124 016667 000002 000000G MOV 2(SP),CKDATA ;TEST PATTERN = ADDRESS.
44 000132 004767 000626 1$ JSR PC,WOR ;WRITE QLB REF MEMORY.
45 000136 005267 000000G INC CKDATA ;BUMP TEST COUNTER.
46 000142 005267 000000G INC PREADD ;BUMP ADDRESS.
47 000146 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
48 000154 103366 BHS 1$ ;NO.
49 ;
50 000156 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE.
51 000162 004767 000000G JSR PC,PPCR.
52 000166 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
53 000172 004767 000000G JSR PC,PPCR.
54 000176 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
55 000204 016667 000002 000000G MOV 2(SP),CKDATA ;TEST PATTERN = ADDRESS.
56 000212 004767 000610 2$ JSR PC,COR ;READ AND COMPARE QLB REF MEMORY.
57 000216 005267 000000G INC CKDATA ;BUMP TEST COUNTER.
58 000222 005267 000000G INC PREADD ;BUMP ADDRESS.
59 000226 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
60 000234 103366 BHS 2$ ;NO.
61 ;
62 000236 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE.
63 000242 004767 000000G JSR PC,PPCR.
64 000246 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
65 000252 004767 000000G JSR PC,PPCR.
66 000256 000207 RTS PC.

```

```

68 ;
69 ;
70 ; TEST-06
71 ; CROSS-TALK TEST
72 ;
73 ;
74 000260 ; T6QR::
75 000260 012767 177777 000000G MOV #1,CKDATA ;SET TEST PATTERN =X'FFFF'
76 000266 012702 000012 MOV #10.,R2 ;SET LOOP COUNT
77 000272 016667 000002 000000G 10$: MOV 2(SP),PREADD ;WORKING ADDRESS
78 000300 004767 000460 1$: JSR PC,WQR ;WRITE QLB REF MEMORY
79 000304 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
80 000312 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
81 000320 103367 BHIS 1$ ;NO
82 000322 005302 DEC R2 ;SUB FROM LOOP COUNT
83 000324 001362 BNE 10$
84 ;
85 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
86 ; WERE NOT WRITTEN
87 ;
88 000326 ; R6Z:
89 000326 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE
90 000332 004767 000000G JSR PC,PPCR
91 000336 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS
92 000342 004767 000000G JSR PC,PPCR
93 000346 005067 000000G CLR CKDATA ;SET TEST PATTERN = 0
94 000352 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
95 000360 005267 000000G INC PREADD ;BUMP START ADDRESS
96 000364 004767 000436 1$: JSR PC,CQR ;READ AND COMPARE QLB REF MEMORY
97 000370 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
98 000376 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
99 000404 103367 BHIS 1$ ;NO
100 ;
101 000406 012746 000013 MOV #0$QHLT,-(SP) ;HALT CODE
102 000412 004767 000000G JSR PC,PPCR
103 000416 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS
104 000422 004767 000000G JSR PC,PPCR
105 000426 000207 RTS PC

```

```

107 ;
108 ;
109 ; TEST-07
110 ; WRITE COMPLEMENT OF MEMORY ADDRESS INTO MEMORY LOCATION.
111 ;
112 ;
113 000430 ; T70R::
114 000430 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
115 000436 016602 000002 MOV 2(SP),R2 ;TEST PATTERN = ADDRESS.
116 000442 005102 1$: COM R2 ;GET ADDRESS COMPLEMENT.
117 000444 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN.
118 000450 004767 000310 JSR PC,WQR ;WRITE QLB REF MEMORY.
119 000454 005267 000000G INC PREADD ;BUMP ADDRESS.
120 000460 016702 000000G MOV PREADD,R2 ;SET UP FOR NEXT TIME.
121 000464 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?.
122 000472 103363 BHIS 1$ ;NO.
123 ;
124 000474 012746 000013 MOV #0$OHLT,-(SP) ;HALT CODE.
125 000500 004767 000000G JSR PC,PPCR.
126 000504 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
127 000510 004767 000000G JSR PC,PPCR.
128 000514 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
129 000522 016602 000002 MOV 2(SP),R2 ;TEST PATTERN = ADDRESS.
130 000526 005102 2$: COM R2 ;GET ADDRESS COMPLEMENT.
131 000530 010267 000000G MOV R2,CKDATA ;SET TEST PATTERN.
132 000534 004767 000266 JSR PC,COR ;READ AND COMPARE QLB REF MEMORY.
133 000540 005267 000000G INC PREADD ;BUMP ADDRESS.
134 000544 016702 000000G MOV PREADD,R2 ;SET UP FOR NEXT TIME.
135 000550 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?.
136 000556 103363 BHIS 2$ ;NO.
137 ;
138 000560 012746 000013 MOV #0$OHLT,-(SP) ;HALT CODE.
139 000564 004767 000000G JSR PC,PPCR.
140 000570 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
141 000574 004767 000000G JSR PC,PPCR.
142 000600 000207 RTS PC.

```

```

144 ;
145 ;
146 ; TEST-12-
147 ; LOOK-FORWARD, LOOK-BEHIND-ADDRESSING-TEST-
148 ;
149 ;
150 ; READ-FROM-TOP-OF-MEMORY-DOWN, THEN-WRITE-
151 ;
152-000602. TCQRD::
153 000602. 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS-
154 000610 016767 000000G-000000G-1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ-
155 000616 004767 000204 JSR. PC,CQR. ;CHECK-MEMORY-LOCATION-
156 000622- 016767 000000G-000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE-
157 000630 004767 000130 JSR. PC,WQR. ;WRITE-QLB-REF-MEMORY-
158 000634 005267 000000G. INC. PREADD. ;BUMP-ADDRESS-
159 000640 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED-?
160 000646 103360 BHIS. 1$. ;NO-
161 ;
162-000650 012746 000013 MOV. #Q$QHLT,-(SP) ;HALT-CODE-
163 000654 004767 000000G. JSR. PC,PPCR. ;CHECK-MEMORY-LOCATION-
164 000660 012746 000040 MOV. #Q$CLR,-(SP) ;CLEAR-PPS-
165 000664 004767 000000G. JSR. PC,PPCR.
166 000670 000207 RTS. PC-
167 ;
168 ; TEST-12-
169 ; READ-FROM-BOTTOM-OF-MEMORY-UP, THEN-WRITE-
170 ;
171 000672. TCQRU::
172-000672. 016667 000004 000000G. MOV. 4(SP),PREADD. ;WORKING ADDRESS := END-ADDRESS-
173 000700 016767 000000G-000000G-1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ-
174 000706 004767 000114 JSR. PC,CQR. ;CHECK-MEMORY-LOCATION-
175 000712- 016767 000000G-000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE-
176 000720 004767 000040 JSR. PC,WQR. ;WRITE-MEMORY-LOCATION-
177 000724 162767 000001 000000G. SUB. #1,PREADD. ;BACK-UP-1-
178 000732- 026667 000002 000000G. CMP. 2(SP),PREADD. ;FINISHED-?
179 000740 003757 BLE. 1$. ;NO-
180 ;
181 000742- 012746 000013 MOV. #Q$QHLT,-(SP) ;HALT-CODE-
182-000746 004767 000000G. JSR. PC,PPCR. ;CHECK-MEMORY-LOCATION-
183 000752- 012746 000040 MOV. #Q$CLR,-(SP) ;CLEAR-PPS-
184 000756 004767 000000G. JSR. PC,PPCR.
185 000762- 000207 RTS. PC-

```



```

187 ;
188 ;
189 ; WRITE-QLB-REF-MEMORY.
190 ;
191 ;
192 WQR:
193 000764 012746 000053 MOV #0$QLA,-(SP) ;SELECT-QLB-PAGE.
194 000770 004767 000000G JSR PC,PPCR.
195 000774 016746 000000G MOV PREADD,-(SP) ;MEMORY-ADDRESS.
196 001000 004767 000000G JSR PC,LBPP.
197 001004 012746 000001 MOV #0$QLR,-(SP) ;SELECT-REF-PAGE.
198 001010 004767 000000G JSR PC,PPCR.
199 001014 016746 000000G MOV CKDATA,-(SP) ;TEST-PATTERN.
200 001020 004767 000000G JSR PC,LBPP.
201 001024 000207 RTS PC.
202 ;
203 ;
204 ; READ-AND-COMPARE-QLB-REF-MEMORY.
205 ;
206 ;
207 COR:
208 001026 012746 000053 MOV #0$QLA,-(SP) ;SELECT-QLB-PAGE.
209 001032 004767 000000G JSR PC,PPCR.
210 001036 016746 000000G MOV PREADD,-(SP) ;MEMORY-ADDRESS.
211 001042 004767 000000G JSR PC,LBPP.
212 001046 012746 000001 MOV #0$QLR,-(SP) ;SELECT-REF-PAGE.
213 001052 004767 000000G JSR PC,PPCR.
214 001056 004767 000000G JSR PC,PPLB.
215 001062 012667 000000G MOV (SP)+,ERW1 ;UNLOAD-WORD-FROM-QLB-REF.
216 ;
217 001066 026767 000000G-000000G CMP CKDATA,ERW1 ;SAME-AS-PATTERN-WRITTEN.
218 001074 001410 BEQ 1$ ;YES,EXIT.
219 001076 016767 000000G-000000G MOV PREADD,ERRADD ;ADDRESS-OF-ERROR.
220 001104 012767 000001 000000G MOV #1,ERRCT ;NUMBER-OF-WORDS-TO-PRINT.
221 001112 004767 000000G JSR PC,MEMERR ;GO-TO-ERROR-ROUTINE.
222 001116 000207 1$: RTS PC.
223 ;
224 000001 ;.END.

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLD = 000030	Q\$QLR = 000001
ALUDE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLRWR = 000200	Q\$QW = 000042
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLR,EN = 000200	Q\$RDCD = 000005
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PPCR = ***** GX	Q\$RDMD = 000006
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PPLB = ***** GX	Q\$REBK = 001000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PREADD = ***** GX	Q\$RNC = 006000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$RCR1 = 176420	Q\$RSC = 004000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$RCR2 = 176422	Q\$RSET = 000010
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q\$RLBR = 176424	Q\$SM = 100000
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX	Q\$CCCP = 000040	RGQ,EN = 000200
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	Q\$CHB = 000400	RGQ,VA = 020000
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	Q\$CHRL = 000200	RGZ = 000326R 002
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q\$CLR = 000040	SEQ,C I = 000010
BIT5 = 000040	BYTE56 = 000070	CPURTE = 020000	Q\$CNC = 030000	STUFOR = 000000RG 002
BIT6 = 000100	BYTE57 = 000071	COR = 001026R 002	Q\$CPC = 000060	S\$CLR = 000000
BIT7 = 000200	BYTE58 = 000072	CSADRD = 000004	Q\$CPC = 000010	S\$LA = 000001
BIT8 = 000400	BYTE59 = 000073	CSEQCI = 100000	Q\$CP2 = 000260	S\$OB = 000005
BIT9 = 001000	BYTE6 = 000006	CSOE = 000040	Q\$CSC = 010000	S\$OR = 000006
BYTE0 = 000000	BYTE60 = 000074	CSURTE = 000100	Q\$CSEL = 000360	S\$OX = 000004
BYTE1 = 000001	BYTE61 = 000075	DBR, RD = 000001	Q\$CSET = 000002	S\$SR = 000007
BYTE10 = 000012	BYTE62 = 000076	DB\$CPP = 001457	Q\$CSP = 020000	S\$S1 = 000010
BYTE11 = 000013	BYTE63 = 000077	DB\$SPT = 000026	Q\$DMA = 000001	S\$S2 = 000014
BYTE12 = 000014	BYTE64 = 000100	DB\$TPC = 000023	Q\$ENBK = 040000	TCORD = 000602RG 002
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q\$ENOP = 020000	TCORU = 000672RG 002
BYTE14 = 000016	BYTE66 = 000102	DMAAUR = 000005	Q\$FAL = 004000	TD\$CTR = 176370
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	Q\$FC = 000045	TD\$CTW = 176360
BYTE16 = 000020	BYTE68 = 000104	DMARWR = 000004	Q\$FO = 000044	TD\$INL = 004000
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q\$FP = 000046	TD\$MEM = 000270
BYTE18 = 000022	BYTE7 = 000007	ERRADD = ***** GX	Q\$HBF = 000002	TD\$MTR = 176344
BYTE19 = 000023	BYTE70 = 000106	ERRCT = ***** GX	Q\$ICP = 000006	TD\$QTR = 000274
BYTE2 = 000002	BYTE71 = 000107	ERW1 = ***** GX	Q\$IHB = 000003	TD\$SW = 176376
BYTE20 = 000024	BYTE72 = 000110	LBPP = ***** GX	Q\$IHRL = 000002	TD\$TAR = 176372
BYTE21 = 000025	BYTE73 = 000111	LOC,EN = 000100	Q\$IMRP = 000007	TD\$TAW = 176362
BYTE22 = 000026	BYTE74 = 000112	LOC,WA = 040000	Q\$LBD = 001000	TD\$TDR = 176374
BYTE23 = 000027	BYTE75 = 000113	LOC,WB = 100000	Q\$LBDP = 001001	TD\$TDW = 176364
BYTE24 = 000030	BYTE76 = 000114	MAREN1 = 000001	Q\$LBP = 000001	T\$AD = 000020
BYTE25 = 000031	BYTE77 = 000115	MAREN2 = 004000	Q\$LDCD = 000003	T\$BA = 000002
BYTE26 = 000032	BYTE78 = 000116	MARLOD = 010000	Q\$LDMD = 000004	T\$BD = 000010
BYTE27 = 000033	BYTE79 = 000117	MAROUT = 000002	Q\$LDPP = 002000	T\$BSO = 100000
BYTE28 = 000034	BYTE8 = 000010	MAR,LO = 002000	Q\$LHP = 010000	T\$BT = 000020
BYTE29 = 000035	BYTE80 = 000120	MAR,OU = 000040	Q\$MNC = 140000	T\$BTAR = 000030
BYTE3 = 000003	BYTE81 = 000121	MBKALL = 001000	Q\$MR = 000052	T\$BTD = 002000
BYTE30 = 000036	BYTE82 = 000122	MBKCLK = 000400	Q\$MRP = 000040	T\$CD = 000100
BYTE31 = 000037	BYTE83 = 000123	MEMERR = ***** GX	Q\$MRP2 = 000240	T\$CLK = 002000
BYTE32 = 000040	BYTE84 = 000124	MMADRD = 000100	Q\$MSC = 040000	T\$DISK = 000200
BYTE33 = 000041	BYTE85 = 000125	MMLEFT = 000002	Q\$MSET = 000004	T\$DRD = 000004
BYTE34 = 000042	BYTE86 = 000126	MMOE = 000004	Q\$MSP = 100000	T\$MEM = 010000
BYTE35 = 000043	BYTE87 = 000127	MMURTE = 000010	Q\$NCLK = 176000	T\$FSA = 000000
BYTE36 = 000044	BYTE88 = 000130	MNOBRE = 100000	Q\$PP = 000100	T\$FSAB = 000004
BYTE37 = 000045	BYTE89 = 000131	MREN1 = 000001	Q\$PPSW = 000320	T\$FSAC = 000014
BYTE38 = 000046	BYTE9 = 000011	MREN2 = 020000	Q\$PP2 = 000300	T\$FSB = 000010
BYTE39 = 000047	BYTE90 = 000132	MSYN = 000040	Q\$QMLT = 000013	T\$IBAR = 000024
BYTE4 = 000004	BYTE91 = 000133	N = 000144	Q\$QL = 000043	T\$IBE = 020000
BYTE40 = 000050	BYTE92 = 000134	PLB = 000010	Q\$QLA = 000053	
BYTE41 = 000051	BYTE93 = 000135	PLC = 000020	Q\$QLB = 000054	

T\$IBF = 040000	WORD1 = 000002	WORD33 = 000102	WORDS57 = 000162	WORD80 = 000240
T\$ICD = 000040	WORD10 = 000024	WORD34 = 000104	WORDS58 = 000164	WORD81 = 000242
T\$MODE = 004000	WORD11 = 000026	WORD35 = 000106	WORDS59 = 000166	WORD82 = 000244
T\$OB = 000036	WORD12 = 000030	WORD36 = 000110	WORD6 = 000014	WORD83 = 000246
T\$OBE = 004000	WORD13 = 000032	WORD37 = 000112	WORD60 = 000170	WORD84 = 000250
T\$OBF = 010000	WORD14 = 000034	WORD38 = 000114	WORD61 = 000172	WORD85 = 000252
T\$OBRA = 000034	WORD15 = 000036	WORD39 = 000116	WORD62 = 000174	WORD86 = 000254
T\$OBWA = 000032	WORD16 = 000040	WORD4 = 000010	WORD63 = 000176	WORD87 = 000256
T\$OUTA = 100000	WORD17 = 000042	WORD40 = 000120	WORD64 = 000200	WORD88 = 000260
T\$RBD0 = 000200	WORD18 = 000044	WORD41 = 000122	WORD65 = 000202	WORD89 = 000262
T\$RNB = 000040	WORD19 = 000046	WORD42 = 000124	WORD66 = 000204	WORD9 = 000022
T\$RSET = 040000	WORD2 = 000004	WORD43 = 000126	WORD67 = 000206	WORD90 = 000264
T\$SC = 000022	WORD20 = 000050	WORD44 = 000130	WORD68 = 000210	WORD91 = 000266
T\$SCLK = 020000	WORD21 = 000052	WORD45 = 000132	WORD69 = 000212	WORD92 = 000270
T\$SEG1 = 000000	WORD22 = 000054	WORD46 = 000134	WORD7 = 000016	WORD93 = 000272
T\$SEG2 = 000001	WORD23 = 000056	WORD47 = 000136	WORD70 = 000214	WORD94 = 000274
T\$SEG3 = 000002	WORD24 = 000060	WORD48 = 000140	WORD71 = 000216	WORD95 = 000276
T\$SO = 000001	WORD25 = 000062	WORD49 = 000142	WORD72 = 000220	WORD96 = 000300
T\$UBUS = 100000	WORD26 = 000064	WORDS = 000012	WORD73 = 000222	WORD97 = 000302
T\$1CLK = 000400	WORD27 = 000066	WORDS0 = 000144	WORD74 = 000224	WORD98 = 000304
T\$8BEN = 000020	WORD28 = 000070	WORDS1 = 000146	WORDS5 = 000226	WORD99 = 000306
T10R = 000116RG	WORD29 = 000072	WORDS2 = 000148	WORD76 = 000230	WQR = 000764R
T60R = 000260RG	WORD3 = 000006	WORDS3 = 000150	WORD77 = 000232	WRDVAL = 000310
T70R = 000430RG	WORD30 = 000074	WORDS4 = 000152	WORD78 = 000234	XTREAD = 001000
UBD, IN = 000020	WORD31 = 000076	WORDS5 = 000154	WORD79 = 000236	XTWRITE = 000400
WORD0 = 000000	WORD32 = 000100	WORDS6 = 000156	WORD8 = 000020	

. ABS: 000000 000
000000 001
QRTEST: 001120 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3136 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:44
QRTEST, QRTEST / -SP=C20, IJIM, C20, IJQRTEST

```

1 .TITLE: QXTEST.
2 000000 .PSECT: QXTEST.
3 .LIST: MEB.
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ALL-PURPOSE WRITE AND READ OF SEQUENTIAL MEMORY LOCATIONS.
14 000000 STUFQX:
15 000000 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
16 000006 004767 000622 1$ JSR PC,MOX ;WRITE QEX MEMORY.
17 000012 005267 000000G INC PREADD ;BUMP ADDRESS.
18 000016 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
19 000024 103370 BHIS 1$ ;NO.
20 ;
21 000026 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
22 000032 004767 000000G JSR PC,PPCR.
23 000036 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS.
24 000044 004767 000616 2$ JSR PC,CQX ;READ AND COMPARE QEX MEMORY
25 000050 005267 000000G INC PREADD ;BUMP ADDRESS.
26 000054 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
27 000062 103370 BHIS 2$ ;NO.
28 ;
29 000064 012746 000040 MOV #0$CLR,-(SP) ;CLEAR PPS.
30 000070 004767 000000G JSR PC,PPCR.
31 000074 000207 RTS PC.

```

```

32.      ;
33      ;
34      ;
35      ;
36      ;
37      ;
38 000076      ;
39 000076 016667 000002 000000G.  T10X:  MOV.  2(SP),PREADD.  ;WORKING ADDRESS.
40 000104 016667 000002 000000G.  MOV.  2(SP),CKDATA.  ;TEST PATTERN = ADDRESS.
41 000112 004767 000516      1$:  JSR.  PC,WQX.  ;WRITE QEX MEMORY.
42 000116 005267 000000G.  INC.  CKDATA.  ;BUMP TEST COUNTER.
43 000122 005267 000000G.  INC.  PREADD.  ;BUMP ADDRESS.
44 000126 026667 000004 000000G.  CMP.  4(SP),PREADD.  ;FINISHED?
45 000134 103366      BHIS.  1$.  ;NO.
46      ;
47 000136 012746 000040      MOV.  #0$CLR,-(SP)  ;CLEAR PPS.
48 000142 004767 000000G.  JSR.  PC,PPCR.
49 000146 016667 000002 000000G.  MOV.  2(SP),PREADD.  ;WORKING ADDRESS.
50 000154 016667 000002 000000G.  MOV.  2(SP),CKDATA.  ;TEST PATTERN = ADDRESS.
51 000162 004767 000500      2$:  JSR.  PC,CQX.  ;READ AND COMPARE QEX MEMORY.
52 000166 005267 000000G.  INC.  CKDATA.  ;BUMP TEST COUNTER.
53 000172 005267 000000G.  INC.  PREADD.  ;BUMP ADDRESS.
54 000176 026667 000004 000000G.  CMP.  4(SP),PREADD.  ;FINISHED?
55 000204 103366      BHIS.  2$.  ;NO.
56      ;
57 000206 012746 000040      MOV.  #0$CLR,-(SP)  ;CLEAR PPS.
58 000212 004767 000000G.  JSR.  PC,PPCR.
59 000216 000207      RTS.  PC.

```

```

61 ;
62 ;
63 ; TEST-06
64 ; CROSS-TALK TEST.
65 ;
66 ;
67 000220 T60X::
68 000220 012767 177777 000000G MOV #1,CKDATA ;SET TEST PATTERN =X'FFFF'
69 000226 012702 000012 MOV #10,R2 ;SET LOOP COUNT
70 000232 016667 000002 000000G 10$ MOV 2(SP),PREADD ;WORKING ADDRESS
71 000240 004767 000370 1$ JSR PC,W0X ;WRITE QEX MEMORY
72 000244 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
73 000252 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
74 000260 103367 BHS 1$ ;NO
75 000262 005302 DEC R2 ;SUB FROM LOOP COUNT
76 000264 001362 BNE 10$
77 ;
78 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
79 ; WERE NOT WRITTEN.
80 ;
81 000266 R6Z:
82 000266 012746 000040 MOV #0$CLR, -(SP) ;CLEAR PPS
83 000272 004767 000000G JSR PC,PPCR
84 000276 005067 000000G CLR CKDATA ;SET TEST PATTERN = 0
85 000302 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
86 000310 005267 000000G INC PREADD ;BUMP START ADDRESS
87 000314 004767 000346 1$ JSR PC,C0X ;READ AND COMPARE QEX MEMORY
88 000320 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
89 000326 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
90 000334 103367 BHS 1$ ;NO
91 ;
92 000336 012746 000040 MOV #0$CLR, -(SP) ;CLEAR PPS
93 000342 004767 000000G JSR PC,PPCR
94 000346 000207 RTS PC

```

```

96 ;
97 ;
98 ;
99 ;
100 ;
101 ;
102-000350 ;
103 000350 016667 000002 000000G. ;
104 000356 016682 000002 ;
105 000362 005182 ;
106 000364 010267 000000G. ;
107 000370 004767 000240 ;
108 000374 005267 000000G. ;
109 000400 016702 000000G. ;
110 000404 026667 000004 000000G. ;
111 000412 103363 ;
112 ;
113 000414 016667 000002 000000G. ;
114 000422 016682 000002 ;
115 000426 005182 ;
116 000430 010267 000000G. ;
117 000434 004767 000226 ;
118 000440 005267 000000G. ;
119 000444 016702 000000G. ;
120 000450 026667 000004 000000G. ;
121 000456 103363 ;
122 ;
123 000460 012746 000040 ;
124 000464 004767 000000G. ;
125 000470 000207 ;

TEST-07
WRITE-COMPLEMENT-OF-MEMORY-ADDRESS-INTO-MEMORY-LOCATION-

T7QX:
1$:
MOV. 2(SP),PREADD. ;WORKING ADDRESS.
MOV. 2(SP),R2. ;TEST-PATTERN.=.ADDRESS.
COM. R2. ;GET-ADDRESS-COMPLEMENT.
MOV. R2,CKDATA. ;SET-TEST-PATTERN.
JSR. PC,WQX. ;WRITE-OEX-MEMORY.
INC. PREADD. ;BUMP-ADDRESS.
MOV. PREADD,R2. ;SET-UP-FOR-NEXT-TIME.
CMP. 4(SP),PREADD. ;FINISHED-?.
BHS. 1$. ;NO.

2$:
MOV. 2(SP),PREADD. ;WORKING ADDRESS.
MOV. 2(SP),R2. ;TEST-PATTERN.=.ADDRESS.
COM. R2. ;GET-ADDRESS-COMPLEMENT.
MOV. R2,CKDATA. ;SET-TEST-PATTERN.
JSR. PC,CQX. ;READ-AND-COMPARE-OEX-MEMORY.
INC. PREADD. ;BUMP-ADDRESS.
MOV. PREADD,R2. ;SET-UP-FOR-NEXT-TIME.
CMP. 4(SP),PREADD. ;FINISHED-?.
BHS. 2$. ;NO.

MOV. #0$CLR,-(SP) ;CLEAR-PPS.
JSR. PC,PPCR.
RTS. PC.

```

```

127 ;
128 ;
129 ; TEST-12-
130 ; LOOK-FORWARD, LOOK-BEHIND-ADDRESSING-TEST-
131 ;
132 ;
133 ; READ-FROM-TOP-OF-MEMORY-DOWN, THEN-WRITE-
134 ;
135 000472. TCQXD:
136 000472. 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS-
137 000500 016767 000000G-000000G-1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ-
138 000506 004767 000154 JSR. PC,CQX. ;CHECK-MEMORY-LOCATION-
139 000512. 016767 000000G-000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE-
140 000520 004767 000110 JSR. PC,WQX. ;WRITE-OEX-MEMORY-
141 000524 005267 000000G. INC. PREADD. ;BUMP-ADDRESS-
142 000530 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED-?
143 000536 103360 BHIS. 1$. ;NO-
144 ;
145 000540 012746 000040 MOV. #0$CLR,-(SP) ;CLEAR-PPS-
146 000544 004767 000000G. JSR. PC,PPCR.
147 000550 000207 RTS. PC.
148 ;
149 ; TEST-12-
150 ; READ-FROM-BOTTOM-OF-MEMORY-UP, THEN-WRITE-
151 ;
152 000552. TCQXU:
153 000552. 016667 000004 000000G. MOV. 4(SP),PREADD. ;WORKING ADDRESS = END-ADDRESS-
154 000560 016767 000000G-000000G-1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ-
155 000566 004767 000074 JSR. PC,CQX. ;CHECK-MEMORY-LOCATION-
156 000572. 016767 000000G-000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE-
157 000600 004767 000030 JSR. PC,WQX. ;WRITE-MEMORY-LOCATION-
158 000604 162767 000001 000000G. SUB. #1,PREADD. ;BACK-UP 1
159 000612 026667 000002 000000G. CMP. 2(SP),PREADD. ;FINISHED-?
160 000620 003757 BLE. 1$. ;NO-
161 ;
162 000622. 012746 000040 MOV. #0$CLR,-(SP) ;CLEAR-PPS-
163 000626 004767 000000G. JSR. PC,PPCR.
164 000632. 000207 RTS. PC.

```



```

166 ;
167 ;
168 ; WRITE QEX MEMORY.
169 ;
170 ;
171 000634 ; WQX:
172 000634 016746 000000G MOV QXCODE,-(SP) ;PUT MEMORY SELECT ON STACK.
173 000640 004767 000000G JSR PC,PPCR.
174 000644 016746 000000G MOV PREADD,-(SP) ;MEMORY ADDRESS.
175 000650 004767 000074 JSR PC,QREG ;LOAD Q-REGISTER.
176 000654 016746 000000G MOV CKDATA,-(SP) ;TEST PATTERN.
177 000660 004767 000000G JSR PC,LBPP.
178 000664 000207 RTS PC.
179 ;
180 ;
181 ; READ AND COMPARE QEX MEMORY.
182 ;
183 ;
184 000666 ; CQX:
185 000666 016746 000000G MOV QXCODE,-(SP) ;PUT MEMORY SELECT ON STACK.
186 000672 004767 000000G JSR PC,PPCR.
187 000676 016746 000000G MOV PREADD,-(SP) ;MEMORY ADDRESS.
188 000702 004767 000042 JSR PC,QREG ;LOAD Q-REGISTER.
189 000706 004767 000000G JSR PC,PPLB.
190 000712 012667 000000G MOV (SP)+,ERW1
191 ;
192 000716 026767 000000G 000000G CMP CKDATA,ERW1 ;SAME AS PATTERN WRITTEN.
193 000724 001410 BEQ 1$ ;YES, EXIT.
194 000726 016767 000000G 000000G MOV PREADD,ERRADD ;ADDRESS OF ERROR.
195 000734 012767 000001 000000G MOV #1,ERRCT ;NUMBER OF WORDS TO PRINT.
196 000742 004767 000000G JSR PC,MEMERR ;GO TO ERROR ROUTINE.
197 000746 000207 1$: RTS PC.

```

```

199 ;
200 ;
201 ; LOAD-Q-REGISTER
202 ;
203 ;
204 000750 ; QREG:
205 000750 012746 000000C MOV: #<PLR.ERT+MD.INR>,-(SP) ;BIT-PATTERN-FOR-PLR-RIGHT
206 000754 004767 000000G JSR: PC,MRPCR ;WRITE-TO-MRP-CONTROL-REGISTER
207 000760 012746 000200 MOV: #RGQ.EN,-(SP) ;ENABLE-Q-REG-LOAD
208 000764 004767 000000G JSR: PC,LBMRP ;SEND-ENABLE-TO-MRP
209 000770 012746 000000C MOV: #<PLR.ELT+MD.INL>,-(SP) ;BIT-PATTERN-FOR-PLR-LEFT
210 000774 004767 000000G JSR: PC,MRPCR ;WRITE-TO-MRP-CONTROL-REG
211 001000 012746 020000 MOV: #RGQ.VAL,-(SP) ;SET-Q-REG-READY
212 001004 004767 000000G JSR: PC,LBMRP ;SEND-COMMAND-TO-MRP
213 001010 016646 000002 MOV: 2(SP),-(SP) ;SEND-QEX-ADDRESS
214 001014 004767 000000G JSR: PC,LBMRP
215 ;
216 ; EXTRA-CLOCK-FOR-PPS
217 ;
218 001020 012746 001001 MOV: #<Q#LBD+Q#LBP>,-(SP) ;CLEAR-DRIVE-AND-PULSE
219 001024 052716 000360 BIS: #Q#CSEL,(SP) ;CLEAR-SELECTION-BITS
220 001030 012746 176000 MOV: #Q#NCLK,-(SP) ;SET-NO-CLOCKS
221 001034 052716 000300 BIS: #Q#PP2,(SP) ;SELECT-PPS
222 001040 004767 000000G JSR: PC,CSR1
223 ;
224 001044 012746 006000 MOV: #Q#RNC,-(SP) ;CLEAR-PPS-NO-CLOCK
225 001050 005046 CLR: -(SP) ;SELECT-NOTHING
226 001052 004767 000000G JSR: PC,CSR1
227 ;
228 ; DE-SELECTION
229 ;
230 001056 012746 001001 MOV: #<Q#LBD+Q#LBP>,-(SP) ;CLEAR-DRIVE-AND-PULSE
231 001062 052716 000360 BIS: #Q#CSEL,(SP) ;CLEAR-SELECTION-BITS
232 001066 012746 176000 MOV: #Q#NCLK,-(SP) ;SET-NO-CLOCKS
233 001072 004767 000000G JSR: PC,CSR1
234 ;
235 001076 011666 000002 MOV: (SP),2(SP) ;MOVE-RETURN-ADDRESS-DOWN-STACK
236 001102 005726 TST: (SP)+ ;POINT-TO-RETURN-ADDRESS
237 001104 000207 RTS: PC
238 ;
239 000001 .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MRPCR = ***** GX	Q\$MSP = 100000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	MSYN = 000040	Q\$NCLK = 176000
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	N = 000144	Q\$PP = 000100
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLB = 000010	Q\$PPSW = 000320
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLC = 000020	Q\$PP2 = 000300
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLD = 000030	Q\$QHLT = 000013
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PLRWR = 000200	Q\$QL = 000043
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PLR.EL = ***** GX	Q\$QLA = 000053
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	PLR.EN = 000200	Q\$QLB = 000054
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	PLR.ER = ***** GX	Q\$QLR = 000001
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX	PPCR = ***** GX	Q\$QLW = 000042
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX	PPLB = ***** GX	Q\$RDCD = 000005
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	PREADD = ***** GX	Q\$RDMD = 000006
BIT3 = 000010	BYTE54 = 000066	CPCCEN = 010000	QREG = 000750R	002 Q\$REBK = 001000
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	QR\$CR1 = 176420	Q\$RNC = 006000
BIT5 = 000040	BYTE56 = 000070	CPURTE = 020000	QR\$CR2 = 176422	Q\$RSC = 004000
BIT6 = 000100	BYTE57 = 000071	CQX = 000666R	002 QR\$LBR = 176424	Q\$RSET = 000010
BIT7 = 000200	BYTE58 = 000072	CSADRD = 000004	QXCDC = ***** GX	Q\$SM = 100000
BIT8 = 000400	BYTE59 = 000073	CSEQCI = 100000	Q\$ATTN = 000100	Q\$SP = 000120
BIT9 = 001000	BYTE6 = 000006	CSOE = 000040	Q\$BCL = 000001	Q\$SE2 = 000340
BYTE0 = 000000	BYTE60 = 000074	CSR1 = ***** GX	Q\$CCCP = 000040	RG0.EN = 000200
BYTE1 = 000001	BYTE61 = 000075	CSURTE = 000100	Q\$CHB = 000400	RG1.VA = 020000
BYTE10 = 000012	BYTE62 = 000076	DBR.RD = 000001	Q\$CHRL = 000200	R62 = 000266R
BYTE11 = 000013	BYTE63 = 000077	DB\$CPP = 001457	Q\$CLR = 000040	002
BYTE12 = 000014	BYTE64 = 000100	DB\$SPT = 000026	Q\$CHC = 030000	SEQ.CI = 000010
BYTE13 = 000015	BYTE65 = 000101	DB\$TPC = 000023	Q\$CP = 000060	STURQX = 000000R
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q\$CPCC = 000010	002
BYTE15 = 000017	BYTE67 = 000103	DMAWR = 000005	Q\$CP2 = 000260	S\$CLR = 000000
BYTE16 = 000020	BYTE68 = 000104	DMARD = 000003	Q\$CSC = 010000	S\$LA = 000001
BYTE17 = 000021	BYTE69 = 000105	DMARUR = 000004	Q\$CSEL = 000360	S\$B = 000005
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q\$CSET = 000002	S\$OR = 000006
BYTE19 = 000023	BYTE70 = 000106	ERRADD = ***** GX	Q\$CSP = 020000	S\$QX = 000004
BYTE2 = 000002	BYTE71 = 000107	ERRCT = ***** GX	Q\$DMA = 000001	S\$R = 000007
BYTE20 = 000024	BYTE72 = 000110	ERW1 = ***** GX	Q\$ENBK = 040000	S\$S1 = 000010
BYTE21 = 000025	BYTE73 = 000111	LBMRP = ***** GX	Q\$ENOP = 020000	S\$S2 = 000014
BYTE22 = 000026	BYTE74 = 000112	LBPP = ***** GX	Q\$FAL = 004000	TCGX = 000472R
BYTE23 = 000027	BYTE75 = 000113	LOC.EN = 000100	Q\$FC = 000045	002
BYTE24 = 000030	BYTE76 = 000114	LOC.WA = 040000	Q\$FO = 000044	TCOXU = 000552R
BYTE25 = 000031	BYTE77 = 000115	LOC.LO = 100000	Q\$FP = 000046	002
BYTE26 = 000032	BYTE78 = 000116	MAREN1 = 000001	Q\$HBF = 000002	TD\$CTR = 176370
BYTE27 = 000033	BYTE79 = 000117	MAREN2 = 004000	Q\$ICP = 000006	TD\$CTW = 176360
BYTE28 = 000034	BYTE8 = 000010	MARLOD = 010000	Q\$IHB = 000003	TD\$INL = 004000
BYTE29 = 000035	BYTE80 = 000120	MAROUT = 000002	Q\$IHRL = 000002	TD\$MEM = 000270
BYTE3 = 000003	BYTE81 = 000121	MAR.LO = 002000	Q\$IMRP = 000007	TD\$OAR = 176344
BYTE30 = 000036	BYTE82 = 000122	MAR.OU = 000040	Q\$LBD = 001000	TD\$OTR = 176346
BYTE31 = 000037	BYTE83 = 000123	MBKALL = 001000	Q\$LBDP = 001001	TD\$ORD = 000274
BYTE32 = 000040	BYTE84 = 000124	MBKCLK = 000400	Q\$LBP = 000001	TD\$QW = 176376
BYTE33 = 000041	BYTE85 = 000125	MD.INL = ***** GX	Q\$LDCD = 000003	TD\$STAR = 176372
BYTE34 = 000042	BYTE86 = 000126	MD.INR = ***** GX	Q\$LDMD = 000004	TD\$TAU = 176362
BYTE35 = 000043	BYTE87 = 000127	MEMERR = ***** GX	Q\$LDPP = 002000	TD\$TDR = 176374
BYTE36 = 000044	BYTE88 = 000130	MMADR = 000100	Q\$LHP = 010000	TD\$TDW = 176364
BYTE37 = 000045	BYTE89 = 000131	MMLFT = 000002	Q\$MNC = 140000	T\$D = 000020
BYTE38 = 000046	BYTE9 = 000011	MNOE = 000004	Q\$MR = 000052	T\$BA = 000002
BYTE39 = 000047	BYTE90 = 000132	MNURTE = 000010	Q\$MRP = 000040	T\$B = 000010
BYTE4 = 000004	BYTE91 = 000133	MNOBRE = 100000	Q\$MRP2 = 000240	T\$SDO = 100000
BYTE40 = 000050	BYTE92 = 000134	MREN1 = 000001	Q\$MSC = 040000	T\$BT = 000020
BYTE41 = 000051	BYTE93 = 000135	MREN2 = 000000	Q\$MST = 000004	T\$BTAR = 000030
				T\$CD = 000100
				T\$CLK = 002000
				T\$DISK = 000200

T\$DRD = 000004	T\$1CLK = 000400	WORD29 = 000072	WORD54 = 000154	WORD79 = 000236
T\$EMEM = 010000	T\$8BEN = 000020	WORD3 = 000006	WORD55 = 000156	WORD8 = 000020
T\$FSAA = 000000	T10X = 000076RG	002 WORD30 = 000074	WORD56 = 000160	WORD00 = 000240
T\$FSAB = 000004	T60X = 000220RG	002 WORD31 = 000076	WORD57 = 000162	WORD01 = 000242
T\$FSAC = 000014	T70X = 000350RG	002 WORD32 = 000100	WORD58 = 000164	WORD02 = 000244
T\$FSB2 = 000010	UBD.IN = 000020	WORD33 = 000102	WORD59 = 000166	WORD03 = 000246
T\$IB = 000026	WORD0 = 000000	WORD34 = 000104	WORD6 = 000014	WORD04 = 000250
T\$IBAR = 000024	WORD1 = 000002	WORD35 = 000106	WORD60 = 000170	WORD05 = 000252
T\$IBE = 020000	WORD10 = 000024	WORD36 = 000110	WORD61 = 000172	WORD06 = 000254
T\$IBF = 040000	WORD11 = 000026	WORD37 = 000112	WORD62 = 000174	WORD07 = 000256
T\$ICD = 000040	WORD12 = 000030	WORD38 = 000114	WORD63 = 000176	WORD08 = 000260
T\$MODE = 004000	WORD13 = 000032	WORD39 = 000116	WORD64 = 000200	WORD09 = 000262
T\$OB = 000036	WORD14 = 000034	WORD4 = 000010	WORD65 = 000202	WORD9 = 000022
T\$OBE = 004000	WORD15 = 000036	WORD40 = 000120	WORD66 = 000204	WORD90 = 000264
T\$OBF = 010000	WORD16 = 000040	WORD41 = 000122	WORD67 = 000206	WORD91 = 000266
T\$OBRA = 000034	WORD17 = 000042	WORD42 = 000124	WORD68 = 000210	WORD92 = 000270
T\$OBWA = 000032	WORD18 = 000044	WORD43 = 000126	WORD69 = 000212	WORD93 = 000272
T\$OUTA = 100000	WORD19 = 000046	WORD44 = 000130	WORD7 = 000016	WORD94 = 000274
T\$RBD0 = 000200	WORD2 = 000004	WORD45 = 000132	WORD70 = 000214	WORD95 = 000276
T\$RNB = 000040	WORD20 = 000050	WORD46 = 000134	WORD71 = 000216	WORD96 = 000300
T\$RSET = 040000	WORD21 = 000052	WORD47 = 000136	WORD72 = 000220	WORD97 = 000302
T\$SC = 000022	WORD22 = 000054	WORD48 = 000140	WORD73 = 000222	WORD98 = 000304
T\$SCLK = 020000	WORD23 = 000056	WORD49 = 000142	WORD74 = 000224	WORD99 = 000306
T\$SEG1 = 000000	WORD24 = 000060	WORDS = 000012	WORD75 = 000226	WORD9 = 000634R
T\$SEG2 = 000001	WORD25 = 000062	WORDS0 = 000144	WORD76 = 000230	WORDVAL = 000310
T\$SEG3 = 000002	WORD26 = 000064	WORDS1 = 000146	WORD77 = 000232	XTREAD = 001000
T\$SO = 000001	WORD27 = 000066	WORDS2 = 000150	WORD78 = 000234	XTWRITE = 000400
T\$UBUS = 100000	WORD28 = 000070	WORDS3 = 000152		

. ABS. 000000 000
000000 001
QXTEST: 001106 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3181 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:45
QXTEST, QXTEST / - SP = C20, 1 JIM, C20, 1 JQXTEST

```

1
2 000000
3
4
5
6
7
8
9
10
11
12
13 000000
14 000000 016667 000002 000000G
15 000006 004767 000752 1$:
16 000012 005267 000000G
17 000016 026667 000004 000000G
18 000024 103370
19
20 000026 012746 077777
21 000032 004767 001062
22 000036 012746 000040
23 000042 004767 000000G
24 000046 016667 000002 000000G
25 000054 004767 000746 2$:
26 000060 005267 000000G
27 000064 026667 000004 000000G
28 000072 103370
29
30 000074 012746 077777
31 000100 004767 001014
32 000104 012746 000040
33 000110 004767 000000G
34 000114 000207

```

.TITLE: FATEST.
.PSECT: FATEST.
.LIST: MEB.

HARDWARE QUERY RESOLVER MEMORY DIAGNOSTICS
FAL POINTER MEMORY
FAL COUNTER MEMORY

ALL-PURPOSE WRITE AND READ OF SEQUENTIAL MEMORY LOCATIONS

STUFFA::

```

MOV 2(SP),PREADD ;WORKING ADDRESS
JSR PC,WFA ;WRITE FAL MEMORY
INC PREADD ;BUMP ADDRESS
CMP 4(SP),PREADD ;FINISHED?
BHIS 1$ ;NO

MOV #077777,-(SP) ;VALUE FOR QCL POINTER
JSR PC,STOP

MOV #0$CLR,-(SP) ;CLEAR PPS
JSR PC,PPCR

MOV 2(SP),PREADD ;WORKING ADDRESS
JSR PC,CFA ;READ AND COMPARE FAL MEMORY
INC PREADD ;BUMP ADDRESS
CMP 4(SP),PREADD ;FINISHED?
BHIS 2$ ;NO

MOV #077777,-(SP) ;VALUE FOR QCL POINTER
JSR PC,STOP

MOV #0$CLR,-(SP) ;CLEAR PPS
JSR PC,PPCR
RTS PC

```

```

36      ;
37      ;
38      ;
39      ;      TEST-01
40      ;      WRITE MEMORY ADDRESS INTO MEMORY LOCATION.
41      ;
42      ;
43      000116      ;      TIFA::
44      000116      016667      000002      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS.
45      000132      004767      000626      000000G      MOV      2(SP),CKDATA      ;TEST PATTERN = ADDRESS.
46      000136      005267      000000G      1$      JSR      PC,WFA      ;WRITE FAL MEMORY.
47      000142      005267      000000G      INC      CKDATA      ;BUMP TEST COUNTER
48      000146      026667      000004      000000G      INC      PREADD      ;BUMP ADDRESS.
49      000154      103366      ;      CMP      4(SP),PREADD      ;FINISHED?
50      ;      BHIS      1$      ;NO.
51      000156      012746      077777      ;      MOV      #077777,-(SP)      ;VALUE FOR QCL POINTER.
52      000162      004767      000732      ;      JSR      PC,STOP      ;
53      000166      012746      000040      ;      MOV      #0$CLR,-(SP)      ;CLEAR PPS.
54      000172      004767      000000G      ;      JSR      PC,PPCR      ;
55      000176      016667      000002      000000G      ;      MOV      2(SP),PREADD      ;WORKING ADDRESS.
56      000204      016667      000002      000000G      MOV      2(SP),CKDATA      ;TEST PATTERN = ADDRESS.
57      000212      004767      000610      2$      JSR      PC,CFA      ;READ AND COMPARE FAL MEMORY
58      000216      005267      000000G      INC      CKDATA      ;BUMP TEST COUNTER
59      000222      005267      000000G      INC      PREADD      ;BUMP ADDRESS.
60      000226      026667      000004      000000G      CMP      4(SP),PREADD      ;FINISHED?
61      000234      103366      ;      BHIS      2$      ;NO.
62      ;
63      000236      012746      077777      ;      MOV      #077777,-(SP)      ;VALUE FOR QCL POINTER.
64      000242      004767      000652      ;      JSR      PC,STOP      ;
65      000246      012746      000040      ;      MOV      #0$CLR,-(SP)      ;CLEAR PPS.
66      000252      004767      000000G      ;      JSR      PC,PPCR      ;
67      000256      000207      ;      RTS      PC      ;

```

```

69      ;
70      ;
71      ;      TEST-06
72      ;      CROSS-TALK TEST
73      ;
74      ;
75 000260      ;      T6FA::
76 000260 012767 177777 000000G  MOV.  #-1,CKDATA.  ;SET TEST PATTERN =X'FFFF'
77 000266 012702 000012      MOV.  #10.,R2.  ;SET LOOP COUNT
78 000272 016667 000002 000000G 10$  MOV.  2(SP),PREADD. ;WORKING ADDRESS
79 000300 004767 000460      JSR.  PC,WFA.  ;WRITE FAL MEMORY
80 000304 062767 000002 000000G 1$  ADD.  #2,PREADD.  ;SKIP ONE ADDRESS
81 000312 026667 000004 000000G  CMP.  4(SP),PREADD. ;FINISHED?
82 000320 103367      BHIS.  1$      ;NO
83 000322 005302      DEC.  R2.  ;SUB FROM LOOP COUNT
84 000324 001362      BNE.  10$
85      ;
86      ;      READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
87      ;      WERE NOT WRITTEN
88      ;
89 000326      ;      R6Z:
90 000326 012746 077777      MOV.  #077777,-(SP) ;VALUE FOR QCL POINTER
91 000332 004767 000562      JSR.  PC,STOP.
92 000336 012746 000040      MOV.  #0$CLR,-(SP) ;CLEAR PPS
93 000342 004767 000000G  JSR.  PC,PPCR.
94 000346 005067 000000G  CLR.  CKDATA.  ;SET TEST PATTERN = 0
95 000352 016667 000002 000000G  MOV.  2(SP),PREADD. ;WORKING ADDRESS
96 000360 005267 000000G  INC.  PREADD.  ;BUMP START ADDRESS
97 000364 004767 000436      JSR.  PC,CFA.  ;READ AND COMPARE FAL MEMORY
98 000370 062767 000002 000000G 1$  ADD.  #2,PREADD.  ;SKIP ONE ADDRESS
99 000376 026667 000004 000000G  CMP.  4(SP),PREADD. ;FINISHED?
100 000404 103367      BHIS.  1$      ;NO
101      ;
102 000406 012746 077777      MOV.  #077777,-(SP) ;VALUE FOR QCL POINTER
103 000412 004767 000502      JSR.  PC,STOP.
104 000416 012746 000040      MOV.  #0$CLR,-(SP) ;CLEAR PPS
105 000422 004767 000000G  JSR.  PC,PPCR.
106 000426 000207      RTS.  PC.

```

```

108      ;
109      ;
110      ;
111      ;
112      ;
113      ;
114      000430      ;
115      000430      016667      000002      000000G      T7FA::
116      000436      016682      000002
117      000442      005182      1$:
118      000444      010267      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS
119      000450      004767      000310      MOV      2(SP),R2      ;TEST PATTERN = ADDRESS
120      000454      005267      000000G      COM      R2      ;GET ADDRESS COMPLEMENT
121      000460      016702      000000G      MOV      R2,CKDATA      ;SET TEST PATTERN
122      000464      026667      000004      000000G      JSR      PC,WFA      ;WRITE FAL MEMORY
123      000472      103363      ;
124      ;
125      000474      012746      077777      INC      PREADD      ;BUMP ADDRESS
126      000500      004767      000414      MOV      PREADD,R2      ;SET UP FOR NEXT TIME
127      000504      012746      000040      CMP      4(SP),PREADD      ;FINISHED?
128      000510      004767      000000G      BHS      1$      ;NO
129      000514      016667      000002      000000G      ;
130      000522      016682      000002      ;
131      000526      005182      ;
132      000530      010267      000000G      2$:
133      000534      004767      000266      MOV      2(SP),PREADD      ;WORKING ADDRESS
134      000540      005267      000000G      MOV      2(SP),R2      ;TEST PATTERN = ADDRESS
135      000544      016702      000000G      COM      R2      ;GET ADDRESS COMPLEMENT
136      000550      026667      000004      000000G      MOV      R2,CKDATA      ;SET TEST PATTERN
137      000556      103363      ;
138      ;
139      000560      012746      077777      JSR      PC,CFA      ;READ AND COMPARE FAL MEMORY
140      000564      004767      000330      INC      PREADD      ;BUMP ADDRESS
141      000570      012746      000040      MOV      PREADD,R2      ;SET UP FOR NEXT TIME
142      000574      004767      000000G      CMP      4(SP),PREADD      ;FINISHED?
143      000600      000207      ;
144      ;
145      ;
146      ;
147      ;
148      ;
149      ;
150      ;
151      ;
152      ;
153      ;
154      ;
155      ;
156      ;
157      ;
158      ;
159      ;
160      ;
161      ;
162      ;
163      ;
164      ;
165      ;
166      ;
167      ;
168      ;
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      ;
224      ;
225      ;
226      ;
227      ;
228      ;
229      ;
230      ;
231      ;
232      ;
233      ;
234      ;
235      ;
236      ;
237      ;
238      ;
239      ;
240      ;
241      ;
242      ;
243      ;
244      ;
245      ;
246      ;
247      ;
248      ;
249      ;
250      ;
251      ;
252      ;
253      ;
254      ;
255      ;
256      ;
257      ;
258      ;
259      ;
260      ;
261      ;
262      ;
263      ;
264      ;
265      ;
266      ;
267      ;
268      ;
269      ;
270      ;
271      ;
272      ;
273      ;
274      ;
275      ;
276      ;
277      ;
278      ;
279      ;
280      ;
281      ;
282      ;
283      ;
284      ;
285      ;
286      ;
287      ;
288      ;
289      ;
290      ;
291      ;
292      ;
293      ;
294      ;
295      ;
296      ;
297      ;
298      ;
299      ;
300      ;
301      ;
302      ;
303      ;
304      ;
305      ;
306      ;
307      ;
308      ;
309      ;
310      ;
311      ;
312      ;
313      ;
314      ;
315      ;
316      ;
317      ;
318      ;
319      ;
320      ;
321      ;
322      ;
323      ;
324      ;
325      ;
326      ;
327      ;
328      ;
329      ;
330      ;
331      ;
332      ;
333      ;
334      ;
335      ;
336      ;
337      ;
338      ;
339      ;
340      ;
341      ;
342      ;
343      ;
344      ;
345      ;
346      ;
347      ;
348      ;
349      ;
350      ;
351      ;
352      ;
353      ;
354      ;
355      ;
356      ;
357      ;
358      ;
359      ;
360      ;
361      ;
362      ;
363      ;
364      ;
365      ;
366      ;
367      ;
368      ;
369      ;
370      ;
371      ;
372      ;
373      ;
374      ;
375      ;
376      ;
377      ;
378      ;
379      ;
380      ;
381      ;
382      ;
383      ;
384      ;
385      ;
386      ;
387      ;
388      ;
389      ;
390      ;
391      ;
392      ;
393      ;
394      ;
395      ;
396      ;
397      ;
398      ;
399      ;
400      ;
401      ;
402      ;
403      ;
404      ;
405      ;
406      ;
407      ;
408      ;
409      ;
410      ;
411      ;
412      ;
413      ;
414      ;
415      ;
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      ;
472      ;
473      ;
474      ;
475      ;
476      ;
477      ;
478      ;
479      ;
480      ;
481      ;
482      ;
483      ;
484      ;
485      ;
486      ;
487      ;
488      ;
489      ;
490      ;
491      ;
492      ;
493      ;
494      ;
495      ;
496      ;
497      ;
498      ;
499      ;
500      ;
501      ;
502      ;
503      ;
504      ;
505      ;
506      ;
507      ;
508      ;
509      ;
510      ;
511      ;
512      ;
513      ;
514      ;
515      ;
516      ;
517      ;
518      ;
519      ;
520      ;
521      ;
522      ;
523      ;
524      ;
525      ;
526      ;
527      ;
528      ;
529      ;
530      ;
531      ;
532      ;
533      ;
534      ;
535      ;
536      ;
537      ;
538      ;
539      ;
540      ;
541      ;
542      ;
543      ;
544      ;
545      ;
546      ;
547      ;
548      ;
549      ;
550      ;
551      ;
552      ;
553      ;
554      ;
555      ;
556      ;
557      ;
558      ;
559      ;
560      ;
561      ;
562      ;
563      ;
564      ;
565      ;
566      ;
567      ;
568      ;
569      ;
570      ;
571      ;
572      ;
573      ;
574      ;
575      ;
576      ;
577      ;
578      ;
579      ;
580      ;
581      ;
582      ;
583      ;
584      ;
585      ;
586      ;
587      ;
588      ;
589      ;
590      ;
591      ;
592      ;
593      ;
594      ;
595      ;
596      ;
597      ;
598      ;
599      ;
600      ;
601      ;
602      ;
603      ;
604      ;
605      ;
606      ;
607      ;
608      ;
609      ;
610      ;
611      ;
612      ;
613      ;
614      ;
615      ;
616      ;
617      ;
618      ;
619      ;
620      ;
621      ;
622      ;
623      ;
624      ;
625      ;
626      ;
627      ;
628      ;
629      ;
630      ;
631      ;
632      ;
633      ;
634      ;
635      ;
636      ;
637      ;
638      ;
639      ;
640      ;
641      ;
642      ;
643      ;
644      ;
645      ;
646      ;
647      ;
648      ;
649      ;
650      ;
651      ;
652      ;
653      ;
654      ;
655      ;
656      ;
657      ;
658      ;
659      ;
660      ;
661      ;
662      ;
663      ;
664      ;
665      ;
666      ;
667      ;
668      ;
669      ;
670      ;
671      ;
672      ;
673      ;
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      ;
730      ;
731      ;
732      ;
733      ;
734      ;
735      ;
736      ;
737      ;
738      ;
739      ;
740      ;
741      ;
742      ;
743      ;
744      ;
745      ;
746      ;
747      ;
748      ;
749      ;
750      ;
751      ;
752      ;
753      ;
754      ;
755      ;
756      ;
757      ;
758      ;
759      ;
760      ;
761      ;
762      ;
763      ;
764      ;
765      ;
766      ;
767      ;
768      ;
769      ;
770      ;
771      ;
772      ;
773      ;
774      ;
775      ;
776      ;
777      ;
778      ;
779      ;
780      ;
781      ;
782      ;
783      ;
784      ;
785      ;
786      ;
787      ;
788      ;
789      ;
790      ;
791      ;
792      ;
793      ;
794      ;
795      ;
796      ;
797      ;
798      ;
799      ;
800      ;
801      ;
802      ;
803      ;
804      ;
805      ;
806      ;
807      ;
808      ;
809      ;
810      ;
811      ;
812      ;
813      ;
814      ;
815      ;
816      ;
817      ;
818      ;
819      ;
820      ;
821      ;
822      ;
823      ;
824      ;
825      ;
826      ;
827      ;
828      ;
829      ;
830      ;
831      ;
832      ;
833      ;
834      ;
835      ;
836      ;
837      ;
838      ;
839      ;
840      ;
841      ;
842      ;
843      ;
844      ;
845      ;
846      ;
847      ;
848      ;
849      ;
850      ;
851      ;
852      ;
853      ;
854      ;
855      ;
856      ;
857      ;
858      ;
859      ;
860      ;
861      ;
862      ;
863      ;
864      ;
865      ;
866      ;
867      ;
868      ;
869      ;
870      ;
871      ;
872      ;
873      ;
874      ;
875      ;
876      ;
877      ;
878      ;
879      ;
880      ;
881      ;
882      ;
883      ;
884      ;
885      ;
886      ;
887      ;
888      ;
889      ;
890      ;
891      ;
892      ;
893      ;
894      ;
895      ;
896      ;
897      ;
898      ;
899      ;
900      ;
901      ;
902      ;
903      ;
904      ;
905      ;
906      ;
907      ;
908      ;
909      ;
910      ;
911      ;
912      ;
913      ;
914      ;
915      ;
916      ;
917      ;
918      ;
919      ;
920      ;
921      ;
922      ;
923      ;
924      ;
925      ;
926      ;
927      ;
928      ;
929      ;
930      ;
931      ;
932      ;
933      ;
934      ;
935      ;
936      ;
937      ;
938      ;
939      ;
940      ;
941      ;
942      ;
943      ;
944      ;
945      ;
946      ;
947      ;
948      ;
949      ;
950      ;
951      ;
952      ;
953      ;
954      ;
955      ;
956      ;
957      ;
958      ;
959      ;
960      ;
961      ;
962      ;
963      ;
964      ;
965      ;
966      ;
967      ;
968      ;
969      ;
970      ;
971      ;
972      ;
973      ;
974      ;
975      ;
976      ;
977      ;
978      ;
979      ;
980      ;
981      ;
982      ;
983      ;
984      ;
985      ;
986      ;
987      ;
988      ;
989      ;
990      ;
991      ;
992      ;
993      ;
994      ;
995      ;
996      ;
997      ;
998      ;
999      ;
1000     ;

```



```

145 ;
146 ;
147 ; TEST-12.
148 ; LOOK-FORWARD, LOOK-BEHIND-ADDRESSING-TEST.
149 ;
150 ;
151 ; READ-FROM-TOP-OF-MEMORY-DOWN, THEN-WRITE.
152 ;
153 000602. TCFAD::
154 000602. 016667 000002 000000G. MOV. 2(SP),PREADD. ;WORKING ADDRESS.
155 000610 016767 000000G.000000G.1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ.
156 000616 004767 000204 JSR. PC,CFA. ;CHECK-MEMORY-LOCATION.
157 000622. 016767 000000G.000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE.
158 000630 004767 000130 JSR. PC,WFA. ;WRITE-FAL-MEMORY.
159 000634 005267 000000G. INC. PREADD. ;BUMP-ADDRESS.
160 000640 026667 000004 000000G. CMP. 4(SP),PREADD. ;FINISHED-?.
161 000646 103360 BHIS. 1$. ;NO.
162. ;
163 000650 012746 077777 MOV. #077777,-(SP) ;VALUE-FOR-QCL-POINTER.
164 000654 004767 000240 JSR. PC,STQP.
165 000660 012746 000040 MOV. #0$CLR,-(SP) ;CLEAR-PPS.
166 000664 004767 000000G. JSR. PC,PPCR.
167 000670 000207 RTS. PC.
168 ;
169 ; TEST-12.
170 ; READ-FROM-BOTTOM-OF-MEMORY-UP, THEN-WRITE.
171 ;
172.000672. TCFAU::
173 000672. 016667 000004 000000G. MOV. 4(SP),PREADD. ;WORKING ADDRESS.= END-ADDRESS.
174 000700 016767 000000G.000000G.1$. MOV. CK2,CKDATA. ;TEST-PATTERN-FOR-READ.
175 000706 004767 000114 JSR. PC,CFA. ;CHECK-MEMORY-LOCATION.
176 000712. 016767 000000G.000000G. MOV. CK3,CKDATA. ;TEST-PATTERN-FOR-WRITE.
177 000720 004767 000040 JSR. PC,WFA. ;WRITE-MEMORY-LOCATION.
178 000724 162767 000001 000000G. SUB. #1,PREADD. ;BACK-UP-1.
179 000732. 026667 000002 000000G. CMP. 2(SP),PREADD. ;FINISHED-?.
180 000740 003757 BLE. 1$. ;NO.
181 ;
182.000742. 012746 077777 MOV. #077777,-(SP) ;VALUE-FOR-QCL-POINTER.
183 000746 004767 000146 JSR. PC,STQP.
184 000752. 012746 000040 MOV. #0$CLR,-(SP) ;CLEAR-PPS.
185 000756 004767 000000G. JSR. PC,PPCR.
186 000762. 000207 RTS. PC.

```

```

188 ;
189 ;
190 ; WRITE FAL MEMORY.
191 ;
192 ;
193 WFA:
194 000764 016746 000000G MOV PREADD, -(SP) ;LOAD MEMORY ADDRESS INTO QCL POINTER
195 000770 004767 000124 JSR PC, STOP ;LOAD QCL POINTER
196 000774 016746 000000G MOV FACODE, -(SP) ;PUT MEMORY SELECT ON STACK
197 001000 004767 000000G JSR PC, PPCR
198 001004 016746 000000G MOV CKDATA, -(SP) ;TEST PATTERN
199 001010 004767 000000G JSR PC, LBPP
200 001014 012746 000040 MOV #0$CLR, -(SP) ;CLEAR PPS
201 001020 004767 000000G JSR PC, PPCR
202 001024 000207 RTS PC
203 ;
204 ;
205 ; READ AND COMPARE FAL MEMORY.
206 ;
207 ;
208 CFA:
209 001026 016746 000000G MOV PREADD, -(SP) ;LOAD MEMORY ADDRESS INTO QCL POINTER
210 001032 004767 000062 JSR PC, STOP ;LOAD QCL POINTER
211 001036 016746 000000G MOV FACODE, -(SP) ;PUT MEMORY SELECT ON STACK
212 001042 004767 000000G JSR PC, PPCR
213 001046 004767 000000G JSR PC, PPLB
214 001052 012667 000000G MOV (SP)+, ERU1
215 001056 012746 000040 MOV #0$CLR, -(SP) ;CLEAR PPS
216 001062 004767 000000G JSR PC, PPCR
217 ;
218 001066 026767 000000G CMP CKDATA, ERU1 ;SAME AS PATTERN WRITTEN
219 001074 001410 BEQ I$ ;YES, EXIT
220 001076 016767 000000G MOV PREADD, ERRADD ;ADDRESS OF ERROR
221 001104 012767 000001 MOV #1, ERRCT ;NUMBER OF WORDS TO PRINT
222 001112 004767 000000G JSR PC, MEMERR ;GO TO ERROR ROUTINE
223 001116 000207 I$: RTS PC

```

```

225 ;
226 ;
227 ;
228 ; LOAD QCL POINTER
229 ;
230 ;
231 001120 ; STOP
232 001120 016667 000002 176424 MOV 2(SP),QR#LBR ;MOVE POINTER WORD TO LOD BUS REG
233 001126 012746 001001 MOV #<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
234 001132 052716 000360 BIS #Q#CSEL,(SP) ;CLEAR SELECTION BITS
235 001136 012746 176000 MOV #Q#NCLK,-(SP) ;SET NO-CLOCKS
236 001142 004767 000000G JSR PC,CSR1
237 ;
238 001146 005046 CLR -(SP) ;CLEAR NOTHING
239 001150 012746 001300 MOV #<Q#PP2+Q#LBD>,-(SP) ;SELECT PPS AND SET DRIVE
240 001154 004767 000000G JSR PC,CSR1
241 ;
242 ; SET FAL LOAD
243 ;
244 001160 012767 004000 176422 MOV #Q#FAL,QR#CR2 ;SET FAL LOAD
245 ;
246 ; EXTRA CLOCK FOR PPS
247 ;
248 001166 012746 000001 MOV #Q#LBP,-(SP) ;CLEAR PULSE
249 001172 052716 006000 BIS #Q#RNC,(SP) ;CLEAR PPS NO-CLOCK
250 001176 005046 CLR -(SP) ;SET NOTHING
251 001200 004767 000000G JSR PC,CSR1
252 ;
253 ; TURN OFF FAL LOAD
254 ;
255 001204 012767 000000 176422 MOV #0,QR#CR2
256 ;
257 ; DE-SELECTION
258 ;
259 001212 012746 001001 MOV #<Q#LBD+Q#LBP>,-(SP) ;CLEAR DRIVE AND PULSE
260 001216 052716 000360 BIS #Q#CSEL,(SP) ;CLEAR SELECTION BITS
261 001222 012746 176000 MOV #Q#NCLK,-(SP) ;SET NO-CLOCKS
262 001226 004767 000000G JSR PC,CSR1
263 ;
264 001232 011666 000002 MOV (SP),2(SP) ;MOVE RETURN ADDRESS DOWN STACK
265 001236 005726 TST (SP)+ ;POINT TO RETURN ADDRESS
266 001240 000207 RTS PC
267 ;
268 000001 .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLB = 000010	Q\$QLA = 000053
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLC = 000020	Q\$QLB = 000054
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLD = 000030	Q\$QLR = 000001
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLRWR = 000200	Q\$QW = 000042
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLREN = 000200	Q\$RDCD = 000005
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PPCR = ***** GX	Q\$RDMD = 000006
BIT10 = 002000	BYTE48 = 000050	BYTVAL = 000144	PPLB = ***** GX	Q\$REBK = 001000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PREADD = ***** GX	Q\$RNC = 006000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	QR\$CR1 = 176420	Q\$RSC = 004000
BIT13 = 020000	BYTE50 = 000062	CFA = 001026R	002 QR\$CR2 = 176424	Q\$RSET = 000010
BIT14 = 040000	BYTE51 = 000063	CKDATA = ***** GX	QR\$LBR = 176424	Q\$SM = 100000
BIT15 = 100000	BYTE52 = 000064	CK2 = ***** GX	Q\$ATTN = 000100	Q\$SP = 000120
BIT2 = 000004	BYTE53 = 000065	CK3 = ***** GX	Q\$BCL = 000001	Q\$SP2 = 000340
BIT3 = 000010	BYTE54 = 000066	CHOBRE = 100000	Q\$CCCP = 000040	RGQEN = 000200
BIT4 = 000020	BYTE55 = 000067	CPCCEN = 010000	Q\$CHB = 000400	RGQVA = 020000
BIT5 = 000040	BYTE56 = 000070	CPREAD = 040000	Q\$CHRL = 000200	R6Z = 000326R 002
BIT6 = 000100	BYTE57 = 000071	CPWRTE = 020000	Q\$CLR = 000040	SEQCI = 000010
BIT7 = 000200	BYTE58 = 000072	CSADRD = 000004	Q\$CNC = 030000	STQP 001120R 002
BIT8 = 000400	BYTE59 = 000073	CSEDCI = 100000	Q\$CPC = 000050	STUFFA 000000RG 002
BIT9 = 001000	BYTE6 = 000006	CSOE = 000040	Q\$CPC2 = 000260	S\$CLR = 000000
BYTE0 = 000000	BYTE60 = 000074	CSR1 = ***** GX	Q\$CSC = 010000	S\$LA = 000001
BYTE1 = 000001	BYTE61 = 000075	CSURTE = 000100	Q\$CSEL = 000360	S\$QB = 000005
BYTE10 = 000012	BYTE62 = 000076	DBR = 000001	Q\$CSET = 000002	S\$QR = 000006
BYTE11 = 000013	BYTE63 = 000077	DB\$CPP = 001457	Q\$CSP = 020000	S\$QX = 000004
BYTE12 = 000014	BYTE64 = 000100	DB\$SPT = 000026	Q\$DMA = 000001	S\$SR = 000007
BYTE13 = 000015	BYTE65 = 000101	DB\$TPC = 000023	Q\$ENK = 040000	S\$S1 = 000010
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q\$ENOP = 020000	S\$S2 = 000014
BYTE15 = 000017	BYTE67 = 000103	DMAAWR = 000005	Q\$FAL = 004000	TCFAD 000602RG 002
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q\$FC = 000045	TCFAU 000672RG 002
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q\$FO = 000044	TD\$CTR = 176370
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q\$FP = 000046	TD\$CTW = 176360
BYTE19 = 000023	BYTE70 = 000106	ERRADD = ***** GX	Q\$HBF = 000002	TD\$INL = 004000
BYTE2 = 000002	BYTE71 = 000107	ERRCT = ***** GX	Q\$ICP = 000006	TD\$MEM = 000270
BYTE20 = 000024	BYTE72 = 000110	ERW1 = ***** GX	Q\$IHB = 000003	TD\$OAR = 176344
BYTE21 = 000025	BYTE73 = 000111	FACODE = ***** GX	Q\$IHRL = 000002	TD\$OTR = 176346
BYTE22 = 000026	BYTE74 = 000112	LBPP = ***** GX	Q\$IRMP = 000007	TD\$ORD = 000274
BYTE23 = 000027	BYTE75 = 000113	LOCEN = 000100	Q\$LBD = 001000	TD\$SW = 176376
BYTE24 = 000030	BYTE76 = 000114	LOCWA = 040000	Q\$LBDP = 001001	TD\$TAR = 176372
BYTE25 = 000031	BYTE77 = 000115	LOCWB = 100000	Q\$LBPF = 000001	TD\$TAW = 176362
BYTE26 = 000032	BYTE78 = 000116	MAREN1 = 000001	Q\$LDCD = 000003	TD\$TDR = 176374
BYTE27 = 000033	BYTE79 = 000117	MAREN2 = 004000	Q\$LDMD = 000004	TD\$TDW = 176364
BYTE28 = 000034	BYTE8 = 000010	MARLOD = 010000	Q\$LDPP = 002000	T\$AD = 000020
BYTE29 = 000035	BYTE80 = 000120	MAROUT = 000002	Q\$LHP = 010000	T\$BA = 000002
BYTE3 = 000003	BYTE81 = 000121	MARLO = 002000	Q\$MNC = 140000	T\$BD = 000010
BYTE30 = 000036	BYTE82 = 000122	MAROU = 000040	Q\$MR = 000052	T\$BSO = 100000
BYTE31 = 000037	BYTE83 = 000123	MBKALL = 001000	Q\$MRP = 000040	T\$BT = 000020
BYTE32 = 000040	BYTE84 = 000124	MBKCLK = 000400	Q\$MRP2 = 000240	T\$BTAR = 000030
BYTE33 = 000041	BYTE85 = 000125	MEMERR = ***** GX	Q\$MSC = 040000	T\$BTD = 002000
BYTE34 = 000042	BYTE86 = 000126	MHARD = 000100	Q\$MSET = 000004	T\$CD = 000100
BYTE35 = 000043	BYTE87 = 000127	MHLEFT = 000002	Q\$MSP = 100000	T\$CLK = 002000
BYTE36 = 000044	BYTE88 = 000130	MHDE = 000004	Q\$NCLK = 176000	T\$ISK = 000200
BYTE37 = 000045	BYTE89 = 000131	MHWRT = 000010	Q\$PP = 000100	T\$DRD = 000004
BYTE38 = 000046	BYTE9 = 000011	MNOBRE = 100000	Q\$PPSW = 000320	T\$EMEM = 010000
BYTE39 = 000047	BYTE90 = 000132	MREN1 = 000001	Q\$PP2 = 000300	T\$FSA = 000000
BYTE4 = 000004	BYTE91 = 000133	MREN2 = 020000	Q\$QHLT = 000013	T\$FSA8 = 000004
BYTE40 = 000050	BYTE92 = 000134	MSYN = 000040	Q\$QL = 000043	T\$FSAC = 000014
BYTE41 = 000051	BYTE93 = 000135	N = 000144		T\$FSB2 = 000010

T#IB.. = .000026	UBD.IN = .000020	WORD31 = .000076	WORD55 = .000156	WORD79 = .000236
T#IBAR = .000024	WFA. = .000764R	002.WORD32 = .000100	WORD56 = .000160	WORD8 = .000020
T#IBE. = .020000	WORD0 = .000000	WORD33 = .000102	WORD57 = .000162	WORD80 = .000240
T#IBF. = .040000	WORD1 = .000002	WORD34 = .000104	WORD58 = .000164	WORD81 = .000242
T#ICD. = .000040	WORD10 = .000024	WORD35 = .000106	WORD59 = .000166	WORD82 = .000244
T#MODE = .004000	WORD11 = .000026	WORD36 = .000110	WORD6 = .000014	WORD83 = .000246
T#OB. = .000036	WORD12 = .000030	WORD37 = .000112	WORD60 = .000170	WORD84 = .000250
T#OBE. = .004000	WORD13 = .000032	WORD38 = .000114	WORD61 = .000172	WORD85 = .000252
T#OBF. = .010000	WORD14 = .000034	WORD39 = .000116	WORD62 = .000174	WORD86 = .000254
T#OBRA = .000034	WORD15 = .000036	WORD4 = .000010	WORD63 = .000176	WORD87 = .000256
T#OBWA = .000032	WORD16 = .000040	WORD40 = .000120	WORD64 = .000200	WORD88 = .000260
T#OUTA = .100000	WORD17 = .000042	WORD41 = .000122	WORD65 = .000202	WORD89 = .000262
T#RBD0 = .000200	WORD18 = .000044	WORD42 = .000124	WORD66 = .000204	WORD9 = .000022
T#RNB. = .000040	WORD19 = .000046	WORD43 = .000126	WORD67 = .000206	WORD90 = .000264
T#RSET = .040000	WORD2 = .000004	WORD44 = .000130	WORD68 = .000210	WORD91 = .000266
T#SC. = .000022	WORD20 = .000050	WORD45 = .000132	WORD69 = .000212	WORD92 = .000270
T#SCLK = .020000	WORD21 = .000052	WORD46 = .000134	WORD7 = .000016	WORD93 = .000272
T#SEG1 = .000000	WORD22 = .000054	WORD47 = .000136	WORD70 = .000214	WORD94 = .000274
T#SEG2 = .000001	WORD23 = .000056	WORD48 = .000140	WORD71 = .000216	WORD95 = .000276
T#SEG3 = .000002	WORD24 = .000060	WORD49 = .000142	WORD72 = .000220	WORD96 = .000300
T#SO. = .000001	WORD25 = .000062	WORDS = .000012	WORD73 = .000222	WORD97 = .000302
T#UBUS = .100000	WORD26 = .000064	WORD50 = .000144	WORD74 = .000224	WORD98 = .000304
T#1CLK = .000400	WORD27 = .000066	WORDS1 = .000146	WORD75 = .000226	WORD99 = .000306
T#0BEN = .000020	WORD28 = .000070	WORDS2 = .000150	WORD76 = .000230	WORDVAL = .000310
T1FA. = .000116RG	002.WORD29 = .000072	WORDS3 = .000152	WORD77 = .000232	XTREAD = .001000
T6FA. = .000260RG	002.WORD3 = .000006	WORDS4 = .000154	WORD78 = .000234	XTWRITE = .000400
T7FA. = .000430RG	002.WORD30 = .000074			

. ABS. 000000 000
000000 001
FATEST: 001242. 002.
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3151 WORDS. (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS. (14 PAGES)
ELAPSED TIME: 00:00:44
FATEST.FATEST/SP=C20.1JIM.C20.1JFATEST.

```

1
2 000000 .TITLE--RTEST-.
3 .PSECT: RTEST.
4
5
6
7
8
9
10 000000 STMA::
11 000000 012746 000000C MOV: #<PLR,ERT+MD,INR>,-(SP) ;SET CNTL BITS FOR MRP.
12 000004 004767 000000G JSR: PC,MRPCR: ;DIRECT CNTL WORD TO MRP.
13 000010 012746 002000 MOV: #<MAR,LOD>,-(SP) ;CNTL BITS TO LOAD MAR.
14 000014 004767 000000G JSR: PC,LBMRP: ;SEND THEM TO MRP. NS.
15 000020 012746 000001 MOV: #MAREN1,-(SP) ;CLEAR PLR-RT-ENABLE BITS. NS.
16 000024 004767 000000G JSR: PC,MRPCR: ;AND SET MAREN1 IN CR. NS.
17 000030 016746 000000G MOV: CKDATA,-(SP) ;MOVE DATA WORD TO LOD BUS REG.
18 000034 004767 000000G JSR: PC,LBMRP: ;SEND DATA TO MRP (MAR)
19
20 ;
21 ; READ MEMORY ADDRESS REG.
22 ; READ MRP MAR
23
24 000040 012746 000040 MOV: #MAR.OUT,-(SP)
25 000044 004767 000000G JSR: PC,MRPCR: ;DIRECT CNTL WORD TO MRP.
26 000050 004767 000000G JSR: PC,MRPLB: ;GET WORD FROM MRP
27 000054 005046 CLR: -(SP) ;CLEAR THE CONTROL REG. NS.
28 000056 004767 000000G JSR: PC,MRPCR: ; NS.
29 000062 012601 MOV: (SP)+,R1 ;WORD RETURNED ON STACK.
30 000070 000207 JSR: PC,RSCMP: ;DID WE READ WHAT WE WROTE.
RTS: PC.

```

```

32. ;
33. ;
34. ;
35. ;
36. ;
37 000072. ;
38 000072. 012746 000230 STCA: MOV #<PLRWR+PLD>,-(SP)
39 000076 004767 000000G JSR PC,CPCR ;DIRECT CNTL WORD TO CP
40 000102. 012746 010000 MOV #<MARLOD>,-(SP) ;SEND MAR LOD BIT NS
41 000106 004767 000000G JSR PC,LBCP ;SEND DATA TO CP
42 000112. 012746 000001 MOV #<MREN1>,-(SP) ;SET MAREN BIT AND ALSO NS
43 000116 004767 000000G JSR PC,CPCR ;CLEAR PLR-D ENABLE BITS NS
44 000122. 016746 000000G MOV CKDATA,-(SP) ;MOVE DATA WORD TO LOD BUS REG
45 000126 004767 000000G JSR PC,LBCP ;SEND DATA TO CP (MAR)
46 000132. 012746 000230 MOV #<PLRWR+PLD>,-(SP) ; NS
47 000136 004767 000000G JSR PC,CPCR ;DIRECT CNTL WORD TO CP CR NS
48 000142. 012746 000000 MOV #0,-(SP) ;CLEAR PLR-D BITS NS
49 000146 004767 000000G JSR PC,LBCP ;SEND DATA TO CP NS
50 000152. 012746 000000 MOV #0,-(SP) ;CLEAR CP CR BITS NS
51 000156 004767 000000G JSR PC,CPCR ;DIRECT CNTL WORD TO CP CR NS
52. ;
53. ;
54. ;
55 000162. 012746 000002G MOV #<MAROUT+REGEN>,-(SP)
56 000166 004767 000000G JSR PC,CPCR ;DIRECT CNTL WORD TO CP
57 000172. 004767 000000G JSR PC,CPLB ;REQUEST CP TO LOD BUS
58 000176 005046 CLR -(SP) ;CLEAR THE CONTROL REGISTER NS
59 000200 004767 000000G JSR PC,CPCR ; NS
60 000204 012601 MOV (SP)+,R1 ;CP WORD RETURNED ON STACK
61 000206 004767 000362 JSR PC,RGCMPC ;DID WE READ WHAT WE READ
62 000212. 000207 RTS PC

```

```

64 ;
65 ;
66 ; INCREMENT MEMORY ADDRESS REGISTERS UP TO LIMIT.
67 ;
68 ; MATCH REPORT PROCESSOR
69 ; FIRST SET MAR = 0
70 ;
71 IMA::
72 000214 005067 000000G CLR CKDATA ; START AT ZERO.
73 000220 005046 CLR -(SP) ; CLEAR NOTHING.
74 000222 012746 000004 MOV #0$MSET, -(SP) ; MRP RESET.
75 000226 004767 000000G JSR PC, CSR1
76 000232 012746 000004 MOV #0$MSET, -(SP) ; CLEAR RESET.
77 000236 005046 CLR -(SP) ; SET NOTHING.
78 000240 004767 000000G JSR PC, CSR1
79 ;
80 ; INCREMENT MRP MAR.
81 ;
82 000244 012746 000001 1$: MOV #MAREN1, -(SP) ; ENABLE CLOCK TO MAR.
83 000250 004767 000000G JSR PC, MRPCR
84 ;
85 000254 012746 001001 MOV #<0$LBD+0$LBP>, -(SP) ; CLEAR DRIVE AND PULSE.
86 000260 052716 000360 BIS #0$CSEL, (SP) ; CLEAR PROCESSOR SELECTION.
87 000264 012746 176000 MOV #0$NCLK, -(SP) ; SET NO-CLOCKS.
88 000270 052716 000240 BIS #0$MRP2, (SP) ; SELECT MRP.
89 000274 004767 000000G JSR PC, CSR1
90 ;
91 000300 012746 140000 MOV #0$MNC, -(SP) ; CLEAR MRP NO-CLOCK.
92 000304 012746 100000 MOV #0$MSP, -(SP) ; SINGLE CLOCK PLR.
93 000310 004767 000000G JSR PC, CSR1
94 ;
95 000314 012746 100000 MOV #0$MSP, -(SP) ; CLEAR SINGLE CLOCK.
96 000320 012746 140000 MOV #0$MNC, -(SP) ; SET MRP NO-CLOCK.
97 000324 004767 000000G JSR PC, CSR1
98 ;
99 000330 005046 CLR -(SP) ; CLEAR MRP CR.
100 000332 004767 000000G JSR PC, MRPCR
101 ;
102 ; READ MRP MAR
103 ;
104 000336 012746 000040 MOV #MAR, OUT, -(SP)
105 000342 004767 000000G JSR PC, MRPCR ; DIRECT CNTL WORD TO MRP.
106 000346 004767 000000G JSR PC, MRPLB ; GET WORD FROM MRP.
107 000352 005046 CLR -(SP) ; CLEAR THE CONTROL REG.
108 000354 004767 000000G JSR PC, MRPCR ;
109 000360 012601 MOV (SP)+, R1 ; WORD RETURNED ON STACK.
110 ;
111 ; COMPARE RESULTS.
112 ;
113 000362 005267 000000G INC CKDATA ; INCREMENT TEST COUNTER.
114 000366 004767 000202 JSR PC, RGCMP ; COMPARE AGAINST TEST PATTERN.
115 000372 022767 177777 000000G CMP #177777, CKDATA ; FINISHED?
116 000400 001321 BNE 1$ ; NO, INCREMENT MAR.
117 000402 000207 RTS PC

```



```

119 ;
120 ;
121 ;
122 ; INCREMENT CONTROL PROCESSOR MEMORY ADDRESS REGISTER
123 ; FIRST SET MAR = 0
124 ;
125 ; ICA:
126 000404 005067 000000G CLR CKDATA ; CLEAR TEST COUNTER
127 000410 005046 CLR -(SP) ; CLEAR NOTHING
128 000412 012746 000002 MOV #0%$CSET,-(SP) ; CP RESET
129 000416 004767 000000G JSR PC,CSR1
130 000422 012746 000002 MOV #0%$CSET,-(SP) ; CLEAR CP RESET
131 000426 005046 CLR -(SP) ; SET NOTHING
132 000430 004767 000000G JSR PC,CSR1
133 ;
134 ; INCREMENT MAR
135 ;
136 000434 012746 000001 1$: MOV #MREN1,-(SP) ; ENABLE CLOCK TO MAR
137 000440 004767 000000G JSR PC,CPCR
138 ;
139 000444 012746 001001 MOV #<Q$LBD+Q$LBP>,-(SP) ; CLEAR DRIVE AND PULSE
140 000450 052716 000360 BIS #Q%$CSEL,(SP) ; CLEAR PROCESSOR SELECTION
141 000454 012746 176000 MOV #Q%$NCLK,-(SP) ; SET NO CLOCKS
142 000460 052716 000260 BIS #Q%$CP2,(SP) ; SELECT CP
143 000464 004767 000000G JSR PC,CSR1
144 ;
145 000470 012746 030000 MOV #Q%$CNC,-(SP) ; CLEAR CP NO CLOCK
146 000474 012746 020000 MOV #Q%$CSP,-(SP) ; SINGLE CLOCK PLR
147 000500 004767 000000G JSR PC,CSR1
148 ;
149 000504 012746 020000 MOV #Q%$CSP,-(SP) ; CLEAR SINGLE CLOCK
150 000510 012746 030000 MOV #Q%$CNC,-(SP) ; SET MRP NO CLOCK
151 000514 004767 000000G JSR PC,CSR1
152 ;
153 000520 005046 CLR -(SP) ; CLEAR CP CR
154 000522 004767 000000G JSR PC,CPCR
155 ;
156 ; READ MAR
157 ;
158 000526 012746 000002G MOV #<MAROUT+REGEN>,-(SP)
159 000532 004767 000000G JSR PC,CPCR ; DIRECT CNTL WORD TO CP
160 000536 004767 000000G JSR PC,CPLB ; REQUEST CP TO LOD BUS
161 000542 005046 CLR -(SP) ; CLEAR THE CONTROL REGISTER
162 000544 004767 000000G JSR PC,CPCR ;
163 000550 012601 MOV (SP)+,R1 ; GET MAR VALUE NS
164 ;
165 ; COMPARE RESULTS
166 ;
167 000552 005267 000000G INC CKDATA ; BUMP TEST COUNTER
168 000556 004767 000012 JSR PC,RGCMPC ; COMPARE RESULTS
169 000562 022767 177777 000000G CMP #177777,CKDATA ; FINISHED?
170 000570 001321 BNE 1$ ; NO CHECK NEXT
171 000572 000207 RTS PC

```

```
173 ;
174 ;
175 ; COMPARE REGISTER TEST RESULTS
176 ;
177 ;
178 000574 026701 000000G RGCOMP: CMP: CKDATA,R1 ;HAVE WE READ WHAT WE HAVE WRITTEN
179 000600 001407 BEQ: 1$ ;YES, CONTINUE
180 000602 012767 000001 000000G MOV: #1,ERRCT ;PRINT 1 WORD
181 000610 010167 000000G MOV: R1,ERW1 ;MOVE DATA RECEIVED TO ERROR LIST
182 000614 004767 000000G JSR: PC,REGERR ;PRINT MESSAGE
183 000620 000207 1$: RTS: PC
184 ;
185 000001 .END
```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	MSYN = 000040	Q#QLA = 000053
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	N = 000144	Q#QLB = 000054
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLB = 000010	Q#QLR = 000001
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLC = 000020	Q#QW = 000042
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLD = 000030	Q#RDCD = 000005
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLRWR = 000200	Q#RDND = 000006
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PLR:EN = 000200	Q#REBK = 001000
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	PLR:ER = ***** GX	Q#RNC = 006000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q#RCR1 = 176420	Q#RSC = 004000
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q#RCR2 = 176422	Q#RSET = 000010
BIT14 = 040000	BYTE51 = 000063	CHOBRE = 100000	Q#RLBR = 176424	Q#SM = 100000
BIT15 = 100000	BYTE52 = 000064	CPCCEN = 010000	Q#ATTN = 000100	Q#SP = 000120
BIT2 = 000004	BYTE53 = 000065	CPCCR = ***** GX	Q#BCL = 000001	Q#SP2 = 000340
BIT3 = 000010	BYTE54 = 000066	CPLB = ***** GX	Q#CCCP = 000040	REGEN = ***** GX
BIT4 = 000020	BYTE55 = 000067	CPREAD = 040000	Q#CHB = 000400	REGERR = ***** GX
BIT5 = 000040	BYTE56 = 000070	CPWRTE = 020000	Q#CHRL = 000200	RGCMF = 000574R 002
BIT6 = 000100	BYTE57 = 000071	CSADRD = 000004	Q#CLR = 000040	RGD:EN = 000200
BIT7 = 000200	BYTE58 = 000072	CSEQCI = 100000	Q#CNC = 030000	RGD:VA = 020000
BIT8 = 000400	BYTE59 = 000073	CSDOE = 000040	Q#PCP = 000060	SEQ:CI = 000010
BIT9 = 001000	BYTE6 = 000006	CSR1 = ***** GX	Q#PCPC = 000010	STCA 000072RG 002
BYTE0 = 000000	BYTE60 = 000074	CSWRTE = 000100	Q#PCP2 = 000260	STMA 000000RG 002
BYTE1 = 000001	BYTE61 = 000075	DBR:RD = 000001	Q#CSC = 010000	S#CLR = 000000
BYTE10 = 000012	BYTE62 = 000076	DB#CPP = 001457	Q#CSEL = 000360	S#LAT = 000001
BYTE11 = 000013	BYTE63 = 000077	DB#SPT = 000026	Q#CSET = 000002	S#OP = 000005
BYTE12 = 000014	BYTE64 = 000100	DB#TPC = 000023	Q#CSP = 020000	S#OR = 000006
BYTE13 = 000015	BYTE65 = 000101	DISPGS = 100000	Q#DMA = 000001	S#OX = 000004
BYTE14 = 000016	BYTE66 = 000102	DNAWRD = 000005	Q#ENBK = 040000	S#SR = 000007
BYTE15 = 000017	BYTE67 = 000103	DMARRD = 000003	Q#ENOP = 020000	S#S1 = 000010
BYTE16 = 000020	BYTE68 = 000104	DMARWR = 000004	Q#FAL = 004000	S#S2 = 000014
BYTE17 = 000021	BYTE69 = 000105	ENBR = 010000	Q#FC = 000045	TD#CTR = 176370
BYTE18 = 000022	BYTE7 = 000007	ERRCT = ***** GX	Q#FD = 000044	TD#CTW = 176360
BYTE19 = 000023	BYTE70 = 000106	ERWI = ***** GX	Q#FF = 000046	TD#INL = 004000
BYTE2 = 000002	BYTE71 = 000107	ICA = 000404RG 002	Q#HDF = 000002	TD#MEM = 000270
BYTE20 = 000024	BYTE72 = 000110	IMA = 000214RG 002	Q#ICP = 000006	TD#OAR = 176344
BYTE21 = 000025	BYTE73 = 000111	LBCP = ***** GX	Q#IHB = 000003	TD#OTR = 176346
BYTE22 = 000026	BYTE74 = 000112	LBMRP = ***** GX	Q#IHLR = 000002	TD#ORD = 000274
BYTE23 = 000027	BYTE75 = 000113	LOC:EN = 000100	Q#IHLP = 000007	TD#SW = 176376
BYTE24 = 000030	BYTE76 = 000114	LOC:WA = 040000	Q#LBD = 001000	TD#TAR = 176372
BYTE25 = 000031	BYTE77 = 000115	LOC:WB = 100000	Q#LBDP = 001001	TD#TAJW = 176362
BYTE26 = 000032	BYTE78 = 000116	MAREN1 = 000001	Q#LBP = 000001	TD#TDR = 176374
BYTE27 = 000033	BYTE79 = 000117	MAREN2 = 004000	Q#LDCD = 000003	TD#TDW = 176364
BYTE28 = 000034	BYTE8 = 000010	MARLOD = 010000	Q#LDMD = 000004	T#AD = 000020
BYTE29 = 000035	BYTE80 = 000120	MAROUT = 000002	Q#LDPP = 002000	T#BA = 000002
BYTE3 = 000003	BYTE81 = 000121	MAR:LO = 002000	Q#LHP = 010000	T#BD = 000010
BYTE30 = 000036	BYTE82 = 000122	MAR:OU = 000040	Q#MNC = 140000	T#BSO = 100000
BYTE31 = 000037	BYTE83 = 000123	MBKALL = 001000	Q#MR = 000052	T#BL = 000020
BYTE32 = 000040	BYTE84 = 000124	MBKCLK = 000400	Q#MRP = 000040	T#BTAR = 000030
BYTE33 = 000041	BYTE85 = 000125	MD:INR = ***** GX	Q#MRP2 = 000240	T#BTD = 002000
BYTE34 = 000042	BYTE86 = 000126	MMADR = 000100	Q#MSC = 040000	T#CD = 000100
BYTE35 = 000043	BYTE87 = 000127	MMLEFT = 000002	Q#MSET = 000004	T#CLK = 002000
BYTE36 = 000044	BYTE88 = 000130	MMDE = 000004	Q#MSP = 100000	T#DISK = 000200
BYTE37 = 000045	BYTE89 = 000131	MMURTE = 000010	Q#NCLK = 176000	T#DRD = 000004
BYTE38 = 000046	BYTE9 = 000011	MNOBRE = 100000	Q#PP = 000100	T#EMEM = 010000
BYTE39 = 000047	BYTE90 = 000132	MREN1 = 000001	Q#PPSW = 000300	T#FSAB = 000000
BYTE4 = 000004	BYTE91 = 000133	MREN2 = 020000	Q#PP2 = 000300	T#FSAB = 000004
BYTE40 = 000050	BYTE92 = 000134	MRPCR = ***** GX	Q#QHLT = 000013	T#FSAC = 000014
BYTE41 = 000051	BYTE93 = 000135	MRPLB = ***** GX	Q#QL = 000043	T#FSB2 = 000010

T#IB.. = 000026	WORD1 = 000002	WORD33 = 000102	WORD57 = 000162	WORD8 = 000020
T#IBAR = 000024	WORD10 = 000024	WORD34 = 000104	WORD58 = 000164	WORD80 = 000240
T#IBE = 020000	WORD11 = 000026	WORD35 = 000106	WORD59 = 000166	WORD81 = 000242
T#IBF = 040000	WORD12 = 000030	WORD36 = 000110	WORD6 = 000014	WORD82 = 000244
T#ICD = 000040	WORD13 = 000032	WORD37 = 000112	WORD60 = 000170	WORD83 = 000246
T#MODE = 004000	WORD14 = 000034	WORD38 = 000114	WORD61 = 000172	WORD84 = 000250
T#OB = 000036	WORD15 = 000036	WORD39 = 000116	WORD62 = 000174	WORD85 = 000252
T#OBE = 004000	WORD16 = 000040	WORD4 = 000010	WORD63 = 000176	WORD86 = 000254
T#OBF = 010000	WORD17 = 000042	WORD40 = 000120	WORD64 = 000200	WORD87 = 000256
T#OBRA = 000034	WORD18 = 000044	WORD41 = 000122	WORD65 = 000202	WORD88 = 000260
T#OBWA = 000032	WORD19 = 000046	WORD42 = 000124	WORD66 = 000204	WORD89 = 000262
T#OUTA = 100000	WORD2 = 000004	WORD43 = 000126	WORD67 = 000206	WORD9 = 000022
T#RBD0 = 000200	WORD20 = 000050	WORD44 = 000130	WORD68 = 000210	WORD90 = 000264
T#RNB = 000040	WORD21 = 000052	WORD45 = 000132	WORD69 = 000212	WORD91 = 000266
T#RSET = 040000	WORD22 = 000054	WORD46 = 000134	WORD7 = 000016	WORD92 = 000270
T#SC = 000022	WORD23 = 000056	WORD47 = 000136	WORD70 = 000214	WORD93 = 000272
T#SCLK = 020000	WORD24 = 000060	WORD48 = 000140	WORD71 = 000216	WORD94 = 000274
T#SEG1 = 000000	WORD25 = 000062	WORD49 = 000142	WORD72 = 000220	WORD95 = 000276
T#SEG2 = 000001	WORD26 = 000064	WORD5 = 000012	WORD73 = 000222	WORD96 = 000300
T#SEG3 = 000002	WORD27 = 000066	WORD50 = 000144	WORD74 = 000224	WORD97 = 000302
T#SO = 000001	WORD28 = 000070	WORD51 = 000146	WORD75 = 000226	WORD98 = 000304
T#UBUS = 100000	WORD29 = 000072	WORD52 = 000150	WORD76 = 000230	WORD99 = 000306
T#1CLK = 000400	WORD3 = 000006	WORD53 = 000152	WORD77 = 000232	WORDVAL = 000310
T#BBEN = 000020	WORD30 = 000074	WORD54 = 000154	WORD78 = 000234	XTREAD = 001000
UBD, IN = 000020	WORD31 = 000076	WORD55 = 000156	WORD79 = 000236	XTWRITE = 000400
WORD0 = 000000	WORD32 = 000100	WORD56 = 000160		

. ABS. 000000 000
000000 001
RTEST: 000622 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3088 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:45
RTEST, RTEST / - SP = [20, 1] IM, [20, 1] RTEST

```

1
2 000000 .TITLE .GDTEST...
3 .PSECT CDTEST
4 .MCALL WTSE#S,CLEF#S
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 000000 STUFCD::
13 000000 CALL RESET ;RESET CP AND INITIALIZE MICROCODE
14 000004 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
15 000012 CALL WCD ;WRITE CP DATA MEMORY (VIA MICROCODE)
16 000016 005267 000000G 1$: INC PREADD ;BUMP ADDRESS
17 000022 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
18 000030 103370 BHIS 1$ ;NO
19 ;
20 000032 005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
21 000034 012746 176000 MOV #0#NCLK,-(SP) ;SET NO-CLOCKS
22 000040 CALL CSR1
23 000044 005067 176422 CLR OR#CR2 ;SET LOAD MODE
24 ;
25 000050 CALL RESET ;RESET CP AND INITIALIZE MICROCODE
26 000054 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
27 000062 CALL CCD ;READ AND COMPARE CP DATA MEMORY
28 000066 005267 000000G 2$: INC PREADD ;BUMP ADDRESS
29 000072 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
30 000100 103370 BHIS 2$ ;NO
31 ;
32 000102 005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
33 000104 012746 176000 MOV #0#NCLK,-(SP) ;SET NO-CLOCKS
34 000110 CALL CSR1
35 000114 005067 176422 CLR OR#CR2 ;SET LOAD MODE
36 000120 RETURN

```

```

38 ;
39 ;
40 ;
41 ;
42 ;
43 ;
44 000122. TICD::
45 000122. CALL RESET ; RESET CP AND INITIALIZE MICROCODE
46 000126 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
47 000134 016667 000002 000000G MOV 2(SP),CKDATA ; TEST PATTERN = ADDRESS
48 000142. 1$: CALL WCD ; WRITE CP DATA MEMORY
49 000146 005267 000000G INC CKDATA ; BUMP TEST COUNTER
50 000152 005267 000000G INC PREADD ; BUMP ADDRESS
51 000156 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED ?
52 000164 103366 BHIS 1$ ; NO
53 ;
54 000166 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
55 000170 012746 176000 MOV #0$NCLK,-(SP) ; SET NO-CLOCKS
56 000174 CALL CSR1
57 000200 005067 176422 CLR OR$CR2 ; SET LOAD MODE
58 ;
59 000204 CALL RESET ; RESET CP AND INITIALIZE MICROCODE
60 000210 016667 000002 000000G MOV 2(SP),PREADD ; WORKING ADDRESS
61 000216 016667 000002 000000G MOV 2(SP),CKDATA ; TEST PATTERN = ADDRESS
62 000224. 2$: CALL CCD ; READ AND COMPARE CP DATA MEMORY
63 000230 005267 000000G INC CKDATA ; BUMP TEST COUNTER
64 000234 005267 000000G INC PREADD ; BUMP ADDRESS
65 000240 026667 000004 000000G CMP 4(SP),PREADD ; FINISHED ?
66 000246 103366 BHIS 2$ ; NO
67 ;
68 000250 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
69 000252 012746 176000 MOV #0$NCLK,-(SP) ; SET NO-CLOCKS
70 000256 CALL CSR1
71 000262 005067 176422 CLR OR$CR2 ; SET LOAD MODE
72 ;
73 000266 RETURN

```

```

75 ;
76 ;
77 ; TEST-06
78 ; CROSS-TALK TEST.
79 ;
80 ;
81 000270 T6GD:
82 000270 CALL RESET ;RESET CP AND INITIALIZE MICROCODE
83 000274 012767 177777 000000G MOV #1,CKDATA ;SET TEST PATTERN =X'FFFF'
84 000302 012702 000012 MOV #10,R2 ;SET LOOP COUNT
85 000306 016667 000002 000000G 10$ MOV 2(SP),PREADD ;WORKING ADDRESS
86 000314 1$ CALL WCD ;WRITE CP DATA MEMORY
87 000320 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
88 000326 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
89 000334 103367 BHIS 1$ ;NO
90 000336 005302 DEC R2 ;SUB FROM LOOP COUNT
91 000340 001362 BNE 10$
92 ;
93 000342 005046 CLR -(SP) ;CLEAR NOTHING IN CSRT
94 000344 012746 176000 MOV #0$NCLK,-(SP) ;SET NO-CLOCKS
95 000350 CALL CSR1
96 000354 005067 176422 CLR QR$CR2 ;SET LOAD MODE
97 ;
98 ; READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
99 ; WERE NOT WRITTEN.
100 ;
101 000360 R6Z:
102 000360 CALL RESET ;RESET CP AND INITIALIZE MICROCODE
103 000364 005067 000000G CLR CKDATA ;SET TEST PATTERN = 0
104 000370 016667 000002 000000G MOV 2(SP),PREADD ;WORKING ADDRESS
105 000376 005267 000000G INC PREADD ;BUMP START ADDRESS
106 000402 1$ CALL CCD ;READ AND COMPARE CP DATA MEMORY
107 000406 062767 000002 000000G ADD #2,PREADD ;SKIP ONE ADDRESS
108 000414 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED?
109 000422 103367 BHIS 1$ ;NO
110 ;
111 000424 005046 CLR -(SP) ;CLEAR NOTHING IN CSR1
112 000426 012746 176000 MOV #0$NCLK,-(SP) ;SET NO-CLOCKS
113 000432 CALL CSR1
114 000436 005067 176422 CLR QR$CR2 ;SET LOAD MODE
115 ;
116 000442 RETURN

```

```

118 ;
119 ;
120 ; TEST-07
121 ; WRITE-COMPLEMENT-OF-MEMORY-ADDRESS-INTO-MEMORY-LOCATION.
122 ;
123 ;
124 000444 ; T7CD::
125 000444 ;
126 000450 016667 000002 000000G. CALL RESET ; RESET-CP-AND-INITIALIZE-MICROCODE
127 000456 016602 000002 MOV 2(SP),PREADD ; WORKING-ADDRESS
128 000462 005102 1$: COM R2 ; TEST-PATTERN=-ADDRESS
129 000464 010267 000000G. MOV R2,CKDATA ; GET-ADDRESS-COMPLEMENT
130 000470 CALL WCD ; SET-TEST-PATTERN
131 000474 005267 000000G. INC PREADD ; WRITE-CP-DATA-MEMORY
132 000500 016702 000000G. MOV PREADD,R2 ; BUMP-ADDRESS
133 000504 026667 000004 000000G. CMP 4(SP),PREADD ; SET-UP-FOR-NEXT-TIME
134 000512 103363 BHIS 1$ ; FINISHED-?
135 ; ; NO
136 000514 005046 CLR -(SP) ; CLEAR-NOTHING-IN-CSR1
137 000516 012746 176000 MOV #0$NCLK,-(SP) ; SET-NO-CLOCKS
138 000522 CALL CSR1
139 000526 005067 176422 CLR QR$CR2 ; SET-LOAD-MODE
140 ;
141 000532 CALL RESET ; RESET-CP-AND-INITIALIZE-MICROCODE
142 000536 016667 000002 000000G. MOV 2(SP),PREADD ; WORKING-ADDRESS
143 000544 016602 000002 MOV 2(SP),R2 ; TEST-PATTERN=-ADDRESS
144 000550 005102 2$: COM R2 ; GET-ADDRESS-COMPLEMENT
145 000552 010267 000000G. MOV R2,CKDATA ; SET-TEST-PATTERN
146 000556 CALL CCD ; READ-AND-COMPARE-CP-DATA-MEMORY
147 000562 005267 000000G. INC PREADD ; BUMP-ADDRESS
148 000566 016702 000000G. MOV PREADD,R2 ; SET-UP-FOR-NEXT-TIME
149 000572 026667 000004 000000G. CMP 4(SP),PREADD ; FINISHED-?
150 000600 103363 BHIS 2$ ; NO
151 ;
152 000602 005046 CLR -(SP) ; CLEAR-NOTHING-IN-CSR1
153 000604 012746 176000 MOV #0$NCLK,-(SP) ; SET-NO-CLOCKS
154 000610 CALL CSR1
155 000614 005067 176422 CLR QR$CR2 ; SET-LOAD-MODE
156 ;
157 000620 RETURN

```



```

159 ;
160 ;
161 ; TEST-12.
162 ; LOOK-FORWARD, LOOK-BEHIND-ADDRESSING-TEST.
163 ;
164 ;
165 ; READ-FROM-TOP-OF-MEMORY-DOWN, THEN-WRITE.
166 ;
167 000622. TCCDD::
168 000622. CALL RESET ;RESET-CP-AND-INITIALIZE-MICROCODE
169 000626 016667 000002 000000G MOV 2(SP),PREADD ;WORKING-ADDRESS
170 000634 016767 000000G 000000G 1$ MOV CK2,CKDATA ;TEST-PATTERN-FOR-READ
171 000642. CALL CCD ;CHECK-MEMORY-LOCATION
172 000646 016767 000000G 000000G MOV CK3,CKDATA ;TEST-PATTERN-FOR-WRITE
173 000654 CALL WCD ;WRITE-CP-DATA-MEMORY
174 000660 005267 000000G INC PREADD ;BUMP-ADDRESS
175 000664 026667 000004 000000G CMP 4(SP),PREADD ;FINISHED-?
176 000672 103360 BHIS 1$ ;NO
177 ;
178 000674 005046 CLR -(SP) ;CLEAR-NOTHING-IN-CSR1
179 000676 012746 176000 MOV #0$NCLK,-(SP) ;SET-NO-CLOCKS
180 000702. CALL CSR1
181 000706 005067 176422 CLR QR$CR2 ;SET-LOAD-MODE
182 ;
183 000712. RETURN
184 ;
185 ; TEST-12.
186 ; READ-FROM-BOTTOM-OF-MEMORY-UP, THEN-WRITE.
187 ;
188 000714 TCCDU::
189 000714 CALL RESET ;RESET-CP-AND-INITIALIZE-MICROCODE
190 000720 016667 000004 000000G MOV 4(SP),PREADD ;WORKING-ADDRESS-=-END-ADDRESS
191 000726 016767 000000G 000000G 1$ MOV CK2,CKDATA ;TEST-PATTERN-FOR-READ
192 000734. CALL CCD ;CHECK-MEMORY-LOCATION
193 000740 016767 000000G 000000G MOV CK3,CKDATA ;TEST-PATTERN-FOR-WRITE
194 000746 CALL WCD ;WRITE-MEMORY-LOCATION
195 000752 162767 000001 000000G SUB #1,PREADD ;BACK-UP-1
196 000760 026667 000002 000000G CMP 2(SP),PREADD ;FINISHED-?
197 000766 003757 BLE 1$ ;NO
198 ;
199 000770 005046 CLR -(SP) ;CLEAR-NOTHING-IN-CSR1
200 000772 012746 176000 MOV #0$NCLK,-(SP) ;SET-NO-CLOCKS
201 000776. CALL CSR1
202 001002 005067 176422 CLR QR$CR2 ;SET-LOAD-MODE
203 ;
204 001006. RETURN

```

```

206 ;
207 ;
208 ; WRITE CP DATA MEMORY.
209 ;
210 ;
211 ; WCD:
212 001010 012767 000003 176424 MOV #0$LDCD,QR$LBR ;MOVE ATTN CODE TO LOD BUS REG
213 001016 012767 120100 176422 MOV *(<Q$ATTN+Q$SM+Q$ENOP>),QR$CR2 ;SET ATTN CODE READY
214 001024 016701 176422 1$: MOV QR$CR2,R1 ;READ CSR2
215 001030 032701 000100 BIT *Q$ATTN,R1 ;ATTN CLEAR
216 001034 001373 BNE 1$ ;NO, READ AGAIN
217 ;
218 001036 016767 000006 176424 MOV PREADD,QR$LBR ;CD MEMORY START ADDRESS
219 001044 012767 120040 176422 MOV *(<Q$CCCP+Q$SM+Q$ENOP>),QR$CR2 ;SET CC TO CP
220 001052 016701 176422 2$: MOV QR$CR2,R1 ;READ CSR2
221 001056 032701 000040 BIT *Q$CCCP,R1 ;IS CC TO CP CLEAR
222 001062 001373 BNE 2$ ;NO, READ AGAIN
223 ;
224 ;
225 ;
226 001064 012767 000001 176424 MOV #1,QR$LBR ;TRANSFER COUNT = 1 WORD
227 001072 012767 120040 176422 MOV *(<Q$CCCP+Q$SM+Q$ENOP>),QR$CR2 ;SET CC TO CP
228 001100 016701 176422 3$: MOV QR$CR2,R1 ;READ CSR2
229 001104 032701 000040 BIT *Q$CCCP,R1 ;IS CC TO CP CLEAR
230 001110 001373 BNE 3$ ;NO, READ AGAIN
231 ;
232 001112 012767 000006 176424 MOV *CKDATA,QR$LBR ;CC MEMORY DATA BUFFER
233 001120 012767 120040 176422 MOV *(<Q$CCCP+Q$SM+Q$ENOP>),QR$CR2 ;SET CC TO CP
234 ;
235 ;
236 ; WAIT FOR INTERRUPT FROM CP.
237 001126 ; WTSE$: #EFN,3
238 ;
239 001140 ; CLEF$: #EFN,3
240 ;
241 ;
242 ; RE-ARM INTERRUPTS.
243 001152 012767 100400 176422 MOV *(<Q$SM+Q$CHB>),QR$CR2 ;CLEAR INTERRUPT (USE HIT BUFFER INT)
244 001160 012767 101000 176422 MOV *(<Q$SM+Q$REBK>),QR$CR2 ;RE-ARM
245 001166 012767 160000 176422 MOV *(<Q$SM+Q$ENBK+Q$ENOP>),QR$CR2 ;ENABLE
246 001174 ; RETURN
247 ;
248 ;

```

```

250          ;      READ AND COMPARE CP DATA MEMORY.
251          ;
252          ;
253 001176          ; CCD:
254 001176 012767 000005 176424      MOV.    #Q$RDCD,QR$LBR      ;MOVE ATTN CODE TO LOD BUS REG
255 001204 012767 120100 176422      MOV.    #<Q$ATTN+Q$SM+Q$ENOP>,QR$CR2 ;SET ATTN CODE READY
256 001212 016701 176422      1$:    MOV.    QR$CR2,R1      ;READ CSR2
257 001216 032701 000100          BIT.    #Q$ATTN,R1      ;ATTN CLEAR
258 001222 001373          BNE.    1$      ;NO, READ AGAIN
259          ;
260 001224 016767 000000G 176424      MOV.    PREADD,QR$LBR      ;CD MEMORY START ADDRESS
261 001232 012767 120040 176422      MOV.    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
262 001240 016701 176422      2$:    MOV.    QR$CR2,R1      ;READ CSR2
263 001244 032701 000040          BIT.    #Q$CCCP,R1      ;IS CC TO CP CLEAR
264 001250 001373          BNE.    2$      ;NO, READ AGAIN
265          ;
266 001252 012767 000001 176424      MOV.    #1,QR$LBR      ;TRANSFER COUNT = 1 WORD
267 001260 012767 120040 176422      MOV.    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
268 001266 016701 176422      3$:    MOV.    QR$CR2,R1      ;READ CSR2
269 001272 032701 000040          BIT.    #Q$CCCP,R1      ;IS CC TO CP CLEAR
270 001276 001373          BNE.    3$      ;NO, READ AGAIN
271          ;
272 001300 012767 000000G 176424      MOV.    #ERW1,QR$LBR      ;CC MEMORY DATA BUFFER
273 001306 012767 120040 176422      MOV.    #<Q$CCCP+Q$SM+Q$ENOP>,QR$CR2 ;SET CC TO CP
274          ;
275          ;      WAIT FOR INTERRUPT FROM CP.
276          ;
277 001314          ;      WTSE$: #EFN,3
278          ;
279 001326          ;      CLEF$: #EFN,3
280          ;
281          ;
282          ;      RE-ARM INTERRUPTS.
283 001340 012767 100400 176422      MOV.    #<Q$SM+Q$CHB>,QR$CR2      ;CLEAR INTERRUPT (USE HIT BUFFER INT)
284 001346 012767 101000 176422      MOV.    #<Q$SM+Q$REBK>,QR$CR2      ;RE-ARM
285 001354 012767 160000 176422      MOV.    #<Q$SM+Q$ENBK+Q$ENOP>,QR$CR2 ;ENABLE
286          ;
287 001362 026767 000000G 000000G      CMP.    CKDATA,ERW1      ;SAME AS PATTERN WRITTEN
288 001370 001410          BEQ.    8$      ;YES, EXIT
289 001372 016767 000000G 000000G      MOV.    PREADD,ERRADD      ;ADDRESS OF ERROR
290 001400 012767 000001 000000G      MOV.    #1,ERRCT      ;NUMBER OF WORDS TO PRINT
291 001406          CALL.    MEMERR      ;GO TO ERROR ROUTINE #4
292 001412      8$:    RETURN
293          ;

```

```

295 ; RESET GR .
296 ;
297 ;
298 001414 ; RESET:
299 001414 005046 CLR - (SP) ; CLEAR NOTHING
300 001416 012746 000002 MOV #0$CSET, - (SP) ; CP RESET
301 001422 CALL CSR1
302 001426 012746 000002 MOV #0$CSET, - (SP) ; CLEAR RESET
303 001432 005046 CLR - (SP) ; SET NOTHING
304 001434 CALL CSR1
305 ;
306 001440 005046 100$: CLR - (SP) ; START MICROCODE AT 0
307 001442 CALL SEQCS
308 ;
309 001446 005046 CLR - (SP) ; REINHIBIT BRANCH CONTROL REGISTER
310 001450 CALL CPCR
311 ;
312 001454 012746 000377 MOV #377, - (SP) ; SET MRP MICRO ADDRESS = X'FF' (JUMP SELF)
313 001460 CALL SEQMM
314 ;
315 001464 005046 CLR - (SP) ; REINHIBIT BRANCH CONTROL REGISTER
316 001466 CALL MRPCR
317 ;
318 001472 012767 001000 176422 MOV #0$REBK, OR$CR2 ; RE-ARM INTERRUPTS
319 001500 012767 120000 176422 MOV #<0$SM+0$ENOP>, OR$CR2 ; SET SEARCH MODE + ENABLE INTERRUPTS
320 001506 012746 000360 MOV #0$CSEL, - (SP) ; CLEAR ALL SELECTIONS
321 001512 052716 001001 BIS #<0$LBD+0$LBP>, (SP) ; CLEAR DRIVE AND PULSE
322 001516 052716 030000 BIS #0$CNC, (SP) ; CLEAR CP NO-CLOCK
323 001522 005046 CLR - (SP) ; SET NOTHING
324 001524 CALL CSR1
325 ;
326 001530 ; RETURN
327 ;
328 000001 .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	N = 000144	Q\$DLB = 000054
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLB = 000010	Q\$DLR = 000001
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLC = 000020	Q\$QJW = 000042
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PLD = 000030	Q\$RDCD = 000005
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	PLRWR = 000200	Q\$RDMD = 000006
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	PLREN = 000200	Q\$REBK = 001000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	PREADD = ***** GX	Q\$RNC = 006000
BIT11 = 004000	BYTE49 = 000061	BYTE50 = 000062	QR\$CR1 = 176420	Q\$RSC = 004000
BIT12 = 010000	BYTE51 = 000063	BYTE52 = 000064	QR\$CR2 = 176422	Q\$RSET = 000010
BIT13 = 020000	BYTE53 = 000065	BYTE54 = 000066	002 QR\$LBR = 176424	Q\$SM = 100000
BIT14 = 040000	BYTE55 = 000067	BYTE56 = 000070	CKDATA = ***** GX	Q\$SP = 000120
BIT15 = 100000	BYTE57 = 000071	BYTE58 = 000072	CK2 = ***** GX	Q\$SP2 = 000340
BIT2 = 000004	BYTE59 = 000073	BYTE60 = 000074	CK3 = ***** GX	RESET = 001414R 002
BIT3 = 000010	BYTE61 = 000075	BYTE62 = 000076	CNOBRE = 100000	RG0.EN = 000400
BIT4 = 000020	BYTE63 = 000077	BYTE64 = 000100	CPCCEN = 010000	RG0.VA = 020000
BIT5 = 000040	BYTE65 = 000101	BYTE66 = 000102	CPCR = ***** GX	R6Z = 000350R 002
BIT6 = 000100	BYTE67 = 000103	BYTE68 = 000104	CPREAD = 040000	SEDCS = ***** GX
BIT7 = 000200	BYTE69 = 000105	BYTE70 = 000106	CPWRTE = 020000	SEMM = ***** GX
BIT8 = 000400	BYTE71 = 000107	BYTE72 = 000110	CPWRITE = 000100	SEQ.CT = 000010
BIT9 = 001000	BYTE73 = 000111	BYTE74 = 000112	CSADRD = 000004	STUFCD = 000000RG 002
BYTE0 = 000000	BYTE75 = 000113	BYTE76 = 000114	CSEDCI = 100000	Q\$CLR = 000000
BYTE1 = 000001	BYTE77 = 000115	BYTE78 = 000116	CSOE = 000040	\$S\$A = 000001
BYTE10 = 000012	BYTE79 = 000117	BYTE80 = 000120	CSR1 = ***** GX	\$S\$B = 000005
BYTE11 = 000013	BYTE81 = 000121	BYTE82 = 000122	CSWRTE = 000100	\$S\$C = 000006
BYTE12 = 000014	BYTE83 = 000123	BYTE84 = 000124	DBR.RD = 000001	\$S\$X = 000004
BYTE13 = 000015	BYTE85 = 000125	BYTE86 = 000126	DB\$CPP = 001457	\$S\$R = 000007
BYTE14 = 000016	BYTE87 = 000127	BYTE88 = 000130	DB\$SPT = 000026	\$S\$1 = 000010
BYTE15 = 000017	BYTE89 = 000131	BYTE90 = 000132	DB\$TPC = 000023	\$S\$2 = 000014
BYTE16 = 000020	BYTE91 = 000133	BYTE92 = 000134	DISPGS = 100000	TCDD = 000622RG 002
BYTE17 = 000021	BYTE93 = 000135	MSYN = 000040	DMARUR = 000005	TCDDU = 000714RG 002
BYTE18 = 000022			DMARRD = 000003	TD\$CTR = 176370
BYTE19 = 000023			DMARUR = 000004	TD\$CTW = 176360
BYTE2 = 000002			EFN.3 = ***** GX	TD\$INL = 004000
BYTE20 = 000024			ENBR = 010000	TD\$MEM = 000270
BYTE21 = 000025			ERRADD = ***** GX	TD\$OAR = 176344
BYTE22 = 000026			ERRCT = ***** GX	TD\$OTR = 176346
BYTE23 = 000027			ERW1 = ***** GX	TD\$QRD = 000274
BYTE24 = 000030			LOC.EN = 000100	TD\$SW = 176376
BYTE25 = 000031			LOC.WA = 040000	TD\$TAR = 176372
BYTE26 = 000032			LOC.WB = 100000	TD\$TAW = 176362
BYTE27 = 000033			MAREN1 = 000001	TD\$TDR = 176374
BYTE28 = 000034			MAREN2 = 004000	TD\$TDW = 176364
BYTE29 = 000035			MARLOD = 010000	T\$AD = 000020
BYTE3 = 000003			MAROUT = 000002	T\$BA = 000002
BYTE30 = 000036			MAR.LO = 002000	T\$BD = 000010
BYTE31 = 000037			MAR.OU = 000040	T\$BSO = 100000
BYTE32 = 000040			MBKALL = 001000	T\$BT = 000020
BYTE33 = 000041			MBKCLK = 000400	T\$BTAR = 000030
BYTE34 = 000042			MEMERR = ***** GX	T\$BTB = 002000
BYTE35 = 000043			MMADR = 000100	T\$CLC = 000100
BYTE36 = 000044			MMLEFT = 000002	T\$CLR = 002000
BYTE37 = 000045			MMOE = 000004	T\$D = 000200
BYTE38 = 000046			MMWRTE = 000010	T\$DISK = 000200
BYTE39 = 000047			CNOBRE = 100000	T\$DRD = 000004
BYTE4 = 000004			MREN1 = 000001	T\$EMEM = 010000
BYTE40 = 000050			MREN2 = 020000	T\$FSAN = 000000
BYTE41 = 000051			MRPCR = ***** GX	T\$FSAB = 000004
				T\$FSAC = 000014

CDTEST: MACRO-M1110 27-MAR-80 14:37 PAGE:12-2
SYMBOL TABLE

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

T#FSB2 = 000010	T7CD = 000444RG	002	WORD30 = 000074	WORD55 = 000156	WORD79 = 000236
T#IB = 000026	UBD:IN = 000020		WORD31 = 000076	WORD56 = 000160	WORD80 = 000220
T#IBAR = 000024	WCD = 001010R	002	WORD32 = 000100	WORD57 = 000162	WORD81 = 000242
T#IBE = 020000	WORD0 = 000000		WORD33 = 000102	WORD58 = 000164	WORD82 = 000244
T#IBF = 040000	WORD1 = 000002		WORD34 = 000104	WORD59 = 000166	WORD83 = 000246
T#ICD = 000040	WORD10 = 000024		WORD35 = 000106	WORD6 = 000014	WORD84 = 000250
T#MODE = 004000	WORD11 = 000026		WORD36 = 000110	WORD60 = 000170	WORD85 = 000252
T#OB = 000036	WORD12 = 000030		WORD37 = 000112	WORD61 = 000172	WORD86 = 000254
T#OBE = 004000	WORD13 = 000032		WORD38 = 000114	WORD62 = 000174	WORD87 = 000256
T#OBF = 010000	WORD14 = 000034		WORD39 = 000116	WORD63 = 000176	WORD88 = 000260
T#OBRA = 000034	WORD15 = 000036		WORD4 = 000010	WORD64 = 000200	WORD89 = 000262
T#OBWA = 000032	WORD16 = 000040		WORD40 = 000120	WORD65 = 000202	WORD9 = 000022
T#OUTA = 100000	WORD17 = 000042		WORD41 = 000122	WORD66 = 000204	WORD90 = 000264
T#RBDO = 000200	WORD18 = 000044		WORD42 = 000124	WORD67 = 000206	WORD91 = 000266
T#RNB = 000040	WORD19 = 000046		WORD43 = 000126	WORD68 = 000210	WORD92 = 000270
T#RSET = 040000	WORD2 = 000004		WORD44 = 000130	WORD69 = 000212	WORD93 = 000272
T#SC = 000022	WORD20 = 000050		WORD45 = 000132	WORD7 = 000016	WORD94 = 000274
T#SCLK = 020000	WORD21 = 000052		WORD46 = 000134	WORD70 = 000214	WORD95 = 000276
T#SEG1 = 000000	WORD22 = 000054		WORD47 = 000136	WORD71 = 000216	WORD96 = 000300
T#SEG2 = 000001	WORD23 = 000056		WORD48 = 000140	WORD72 = 000220	WORD97 = 000302
T#SEG3 = 000002	WORD24 = 000060		WORD49 = 000142	WORD73 = 000222	WORD98 = 000304
T#SO = 000001	WORD25 = 000062		WORD5 = 000012	WORD74 = 000224	WORD99 = 000306
T#UBUS = 100000	WORD26 = 000064		WORD50 = 000144	WORD75 = 000226	WORDVAL = 000310
T#1CLK = 000400	WORD27 = 000066		WORD51 = 000146	WORD76 = 000230	XTREAD = 001000
T#BEN = 000020	WORD28 = 000070		WORD52 = 000150	WORD77 = 000232	XTURTE = 000400
T1CD = 000122RG	002	WORD29 = 000072	WORD53 = 000152	WORD78 = 000234	
T6CD = 000270RG	002	WORD3 = 000006	WORD54 = 000154		

. ABS: 000000 000
000000 001
CDTEST: 001532 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3425 WORDS (14 PAGES)
DYNAMIC MEMORY: 4916 WORDS (18 PAGES)
ELAPSED TIME: 00:00:49
CDTEST, CDTEST, /-SP=C 20, 1 JIM, C 20, 1 JCDTEST

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

TASK NAME : ...QMT.
PARTITION NAME : GEN
IDENTIFICATION : 08
TASK UIC : [7,5]
TASK PRIORITY : 100.
STACK LIMITS : 040236 040535 000300 00192.
PRG XFR ADDRESS : 047124
TASK ATTRIBUTES : AL,CP
TOTAL ADDRESS WINDOWS : 3.
TASK IMAGE SIZE : 8160. WORDS
TASK ADDRESS LIMITS : 040000 077667
R-W DISK BLK LIMITS : 000042 000142 000101 00065.

QMT.TSK:16 OVERLAY DESCRIPTION:

BASE	TOP	LENGTH	
040000	074037	034040	14368. QMT
074040	074547	000510	00328. MRPSUB
074550	075117	000350	00232. MMTSUB
075120	075317	000200	00120. MMTST0
075120	075363	000244	00164. MMTST1
075120	075363	000244	00164. MMTST2
075120	075353	000234	00156. MMTST3
075120	075457	000340	00224. MMTST4
075120	077477	002360	01264. MMTST5
074550	076303	001534	00860. MDTEST
074040	075063	001024	00532. CSTSUB
075064	075453	000370	00248. CSTST0
075064	075567	000504	00324. CSTST1
075064	075557	000474	00316. CSTST2
075064	075547	000464	00308. CSTST3
075064	075757	000674	00444. CSTST4
075064	077667	002604	01412. CSTST5
074040	075533	001474	00828. MRPSUB
075534	077003	001250	00680. QBTEST
075534	076653	001120	00592. QRTEST
075534	076643	001110	00584. QXTEST
075534	076777	001244	00676. FATEST
075534	076357	000624	00404. RTEST
074040	076277	002240	01184. MRPSUB

*** ROOT-SEGMENT: QMT.

R/W-MEM-LIMITS: 040000 074037 034040 14368.
 DISK-BLK-LIMITS: 000042-000076 000035 00029.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE..IDENT.	FILE...
. BLK: (RW, I, LCL, REL, CON)	040536 024112-10314.	
	040536 017036 07710.	QMT
CPSUB: (RW, I, LCL, REL, CON)	064650 000502-00322.	QMT.OBJ:1
	064650 000502-00322.	CPSUB.
		CPSUB.OBJ:1
\$\$ALER: (RW, I, LCL, REL, CON)	065352-000024 00020.	
\$\$ALVC: (RW, D, LCL, REL, CON)	065376 000700 00448.	
\$\$AUTO: (RW, I, LCL, REL, CON)	066276 000130 00088.	
\$\$FSR1: (RW, D, GBL, REL, OVR)	066426 001020 00528.	
	066426 001020 00528.	QMT
		QMT.OBJ:1
\$\$FSR2: (RW, D, GBL, REL, CON)	067446 000104 00068.	
\$\$MRKS: (RW, I, LCL, REL, OVR)	073570 000076 00062.	
\$\$OVDT: (RW, D, LCL, REL, OVR)	067552-000020 00016.	
\$\$OVRS: (RW, I, LCL, ABS, CON)	000000 000000 00000.	
\$\$RDSG: (RW, I, LCL, REL, OVR)	073666 000150 00104.	
\$\$RESL: (RW, I, LCL, REL, CON)	067572-003332 01754.	
\$\$RESM: (RW, I, LCL, REL, CON)	132000 007656 04014.	
\$\$RGDS: (RW, D, LCL, REL, CON)	073124 000000 00000.	
\$\$RTS: (RW, I, GBL, REL, OVR)	073124 000002-00002.	
\$\$SGD0: (RW, D, LCL, REL, OVR)	073126 000000 00000.	
\$\$SGD1: (RW, D, LCL, REL, CON)	073126 000440 00288.	
\$\$SGD2: (RW, D, LCL, REL, OVR)	073566 000002-00002.	
\$\$UNDS: (RW, D, LCL, REL, CON)	073570 000000 00000.	

GLOBAL-SYMBOLS:

BASE	040566-R	ERW3	043414-R	LCS	043424-R	STUFCS	065546-R	TCMMD	065436-R	T1MM	065406-R	T7ND	065536-R
BINWD	040560-R	ERW4	043416-R	LMM	043420-R	STUFFA	066076-R	TCMMU	065446-R	T1QB	065706-R	T7NM	065426-R
CKDATA	043374-R	FACODE	040610-R	MEMERR	053626-R	STUFMD	065466-R	TCQBD	065666-R	T1QR	065766-R	T7OB	065726-R
CK2	043376-R	FIRST	001000	OLDVEC	040544-R	STUFMM	065376-R	TCQBU	065676-R	T1QX	066046-R	T7OR	066006-R
CK3	043400-R	FVER	040600-R	PPCR	065646-R	STUFQB	065656-R	TCORD	065746-R	T6CD	066256-R	T7OX	066066-R
CPCR	065262-R	GET	056240-R	PREADD	043402-R	STUFQR	065736-R	TCQRU	065756-R	T6CS	065566-R	VIRT	040574-R
CPCRA	065270-R	ICA	066156-R	PRINT	044075-R	STUFQX	066016-R	TCQXD	066026-R	T6FA	066136-R	WCOUNT	040604-R
CPLB	065212-R	IMA	066166-R	QBPAGE	040612-R	TCCDD	066226-R	TCQXU	066036-R	T6MD	065526-R	WRTCS	064736-R
CSR1	053156-R	INDNB	047066-R	QXCDE	040606-R	TCCDU	066236-R	TDCS	065626-R	T6MM	065416-R	\$CEFI	005174
DATA1	040572-R	INFDB	046726-R	REGERR	053216-R	TCCSD	065606-R	TDMM	065456-R	T6OB	065716-R	\$CSTA	004226
EFN.3	000003	IO:WVB	011000	SEOCS	064650-R	TCCSU	065616-R	TSKTCB	040542-R	T6OR	065776-R	\$DIV	007146
ERRADD	043404-R	LBCP	065016-R	STAT	040550-R	TCFAD	066106-R	T1CD	066246-R	T6OX	066056-R	\$DRDSE	017134
ERRCT	043406-R	LBCSC	065114-R	STCA	066176-R	TCFAU	066116-R	T1CS	065556-R	T7CD	066266-R	\$MUL	007116
ERW1	043410-R	LBPP	065636-R	STMA	066206-R	TCMDD	065476-R	T1FA	066126-R	T7CS	065576-R	\$TKTCB	004026
ERW2	043412-R	LCCOUNT	040602-R	STUFCD	066216-R	TCMDU	065506-R	T1MD	065516-R	T7FA	066146-R		

QMT:TSK:16 MEMORY-ALLOCATION MAP: TKB.
MRPSUB: 27-MAR-80 1

PAGE 3

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MRPSUB.

R/W MEM LIMITS: 074040 074547 000510 00320.
DISK BLK LIMITS: 000077 000077 000001 00001.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	074040	000000	00000.
MRPSUB: (RW, I, LCL, REL, CON)	074040	000506	00326.
	074040	000506	00326. MRPSUB: MRPSUB.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	074546	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

LBMRP- 074212-R LBMSC- 074310-R MRPCR- 074456-R MRPCRA- 074464-R MRPLB- 074406-R SEOM1 074040-R URTMM- 074126-R

*** SEGMENT: MMTSUB

R/W MEM. LIMITS: 074550 075117 000350 00232.
DISK BLK. LIMITS: 000100 000100 000001 00001.

MEMORY-ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	074550	000000	00000.
MMTSUB: (RW, I, LCL, REL, CON)	074550	000346	00230.
	074550	000346	00230. MMTSUB
\$\$ALVC: (RW, D, LCL, REL, CON)	075116	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

CMPL: 074630-R CMPR: 074710-R SINGLE 075064-R UNLL: 074770-R UNLR: 075026-R WRITEL: 074550-R WRITER: 074600-R

QMT,TSK:16 MEMORY-ALLOCATION MAP: TKB:
MMTST0 27-MAR-80

PAGE 5

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST0.

R/W-MEM-LIMITS: 075120 075317 000200 00120.
DISK-BLK-LIMITS: 000101 000101 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST0: (RW, I, LCL, REL, CON)	075120	000200	00120.
	075120	000200	00120.
MMTST0			MMTST0.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	075320	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

STUFMM-075120-R.

QMT.TSK: MEMORY-ALLOCATION MAP. TKB.
MMTST1 27-MAR-80

PAGE 6

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST1.

R/W MEM LIMITS: 075120 075363 000244 00164.
DISK BLK LIMITS: 000102 000102 000001 00001.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST1: (RW, I, LCL, REL, CON)	075120	000242	00162.
	075120	000242	00162.
MMTST1			MMTST1.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	075362	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

TMM 075120-R.

QMT,TSK:16 MEMORY-ALLOCATION MAP: TKB:
MMTST2: 27-MAR-80

PAGE 2

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST2

R/W MEM. LIMITS: 075120 075363 000244 00164.
DISK BLK. LIMITS: 000103 000103 000001 00001.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST2: (RW, I, LCL, REL, CON)	075120	000244	00164.
	075120	000244	00164.
MMTST2:			MMTST2.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	075364	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

T6MM: 075120-R

QMT,TSK: MEMORY-ALLOCATION MAP: TKB.
MMTST3 27-MAR-80

PAGE 8

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST3.

R/W-MEM-LIMITS: 075120 075353 000234 00156.
DISK-BLK-LIMITS: 000104 000104 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE...	IDENT...	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST3: (RW, I, LCL, REL, CON)	075120	000234	00156.
	075120	000234	00156.
\$\$ALVC: (RW, D, LCL, REL, CON)	075354	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

TZHM: 075120-R.

DMT:TSK:16 MEMORY-ALLOCATION MAP. TKB.
MMTST4 27-MAR-80

PAGE 9

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST4

R/W MEM. LIMITS: 075120 075457 000340 00224.
DISK BLK. LIMITS: 000105 000105 000001 00001.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE..	IDENT.	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST4: (RW, I, LCL, REL, CON)	075120	000340	00224.
	075120	000340	00224. MMTST4
\$\$ALVC: (RW, D, LCL, REL, CON)	075460	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

TCHMD: 075120-R TCHMU: 075276-R

QMT,TSK: MEMORY-ALLOCATION MAP, TKB.
MMTST5 27-MAR-80

PAGE 18

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MMTST5.

R/W-MEM. LIMITS: 075120 077477 002360 01264.
DISK-BLK-LIMITS: 000106 000110 000003 00003.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE	IDENT.	FILE...
. BLK: (RW, I, LCL, REL, CON)	075120	000000	00000.
MMTST5: (RW, I, LCL, REL, CON)	075120	000330	00216.
	075120	000330	00216.
MMTST5			MMTST5.OBJ: 1
\$\$ALVC: (RW, D, LCL, REL, CON)	075450	000000	00000.
\$\$RESL: (RW, I, LCL, REL, CON)	075450	002026	01046.
\$\$RTS: (RW, I, GBL, REL, DVR)	073124	000002	00002.

GLOBAL SYMBOLS:

TDMH: 075120-R.

QMT.TSK:16 MEMORY-ALLOCATION MAP: TKB.
MDTEST 27-MAR-80

PAGE 11

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MDTEST

R/W MEM. LIMITS: 074550 076303 001534 00860.
DISK BLK LIMITS: 000111 000112 000002 00002.

MEMORY-ALLOCATION SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	074550	000000	00000.
MDTEST: (RW, I, LCL, REL, CON)	074550	001532	00058.
	074550	001532	00058.
MDTEST:			MDTEST.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	076302	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

STUFMD 074550-R TCMD0 075372-R TCMDU 075464-R T1MD 074672-R T6MD 075040-R T7MD 075214-R

QMT,TSK: MEMORY-ALLOCATION MAP,TKB:
CSTSUB: 27-MAR-80

PAGE 12

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTSUB:

R/W-MEM-LIMITS: 074040 075063 001024 00532.
DISK-BLK-LIMITS: 000113 000114 000002-00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
.BLK: (RW,I,LCL,REL,CON)	074040	000000	00000.
CSTSUB: (RW,I,LCL,REL,CON)	074040	001022-00530.	
\$\$\$ALVC: (RW,D,LCL,REL,CON)	074040	001022-00530.	CSTSUB: CSTSUB.OBJ:1
\$\$\$RTS: (RW,I,GBL,REL,OVR)	073124	000002-00002.	

GLOBAL-SYMBOLS:

CMPA: 074200-R CMPC: 074420-R SINGLE 075030-R UNLB: 074676-R UNLD: 074772-R WRITB: 074070-R WRITD: 074150-R
CMPBB: 074310-R CMPDD: 074530-R UNLA: 074640-R UNLC: 074734-R WRITEA: 074040-R WRITEC: 074120-R

QMT,TSK:16 MEMORY-ALLOCATION MAP,TKB
CSTST0 27-MAR-80

PAGE 17

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTST0

R/W-MEM-LIMITS: 075064 075453 000370 00248.
DISK-BLK-LIMITS: 000115 000115 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE..	IDENT.	FILE...
.BLK.: (RW,I,LCL,REL,CON)	075064	000000	00000.
CSTST0: (RW,I,LCL,REL,CON)	075064	000370	00248.
	075064	000370	00248.
\$\$\$ALVC: (RW,D,LCL,REL,CON)	075454	000000	00000.
\$\$\$RTS: (RW,I,GBL,REL,OVR)	073124	000002	00002.
			CSTST0
			CSTST0.OBJ:1

GLOBAL-SYMBOLS:

STUFCS-075064-R.

QMT:TSK: MEMORY-ALLOCATION MAP, TKB
CSTST1 27-MAR-80

PAGE 14

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTST1

R/W MEM LIMITS: 075064 075567 000504 00324.
DISK-BLK LIMITS: 000116 000116 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE	IDENT	FILE...
. BLK: (RW, I, LCL, REL, CON)	075064	000000	00000.
CSTST1: (RW, I, LCL, REL, CON)	075064	000502-00322.	
	075064	000502-00322.	CSTST1
			CSTST1.OBJ:1
\$\$ALVC: (RW, D, LCL, REL, CON)	075566	000000	00000.
\$\$RTS: (RW, I, GBL, REL, DVR)	073124	000002-00002.	

GLOBAL SYMBOLS:

T40S-- 075064-R.

DMT.TSK:16 MEMORY-ALLOCATION MAP. TKB.
CSTST2. 27-MAR-80

PAGE 15

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTST2.

R/W-MEM. LIMITS: 075064 075557 000474 00316.
DISK-BLK-LIMITS: 000117 000117 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION...	TITLE..	IDENT.	FILE...
. BLK.: (RW, I, LCL, REL, CON)	075064	000000	00000.
CSTST2: (RW, I, LCL, REL, CON)	075064	000474	00316.
	075064	000474	00316.
\$\$ALVC: (RW, D, LCL, REL, CON)	075560	000000	00000.
\$\$RTS.: (RW, I, GBL, REL, OVR)	073124	000002	00002.
			CSTST2. CSTST2.OBJ:1

GLOBAL-SYMBOLS:

T00S.. 075064-R.

DMT.TSK:16 MEMORY-ALLOCATION MAP. TKB.
CSTST3. 27-MAR-80 16:14

PAGE 16

*** SEGMENT: CSTST3.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTST3.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

R/W-MEM-LIMITS: 075064 075547 000464 00308.
DISK-BLK-LIMITS: 000120 000120 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION:	TITLE	IDENT	FILE
. BLK.: (RW, I, LCL, REL, CON)	075064	000000	00000.
CSTST3: (RW, I, LCL, REL, CON)	075064	000462	00306.
	075064	000462	00306.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	075546	000000	00000.
\$\$\$RTS.: (RW, I, GBL, REL, OVR)	073124	000002	00002.
		CSTST3	CSTST3.OBJ:1

GLOBAL-SYMBOLS:

T7CS: 075064-R.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

GMT,FSK:1 MEMORY-ALLOCATION MAP:TKR
CSTST4 27-MAR-80 16:14

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTST4.

R/W-MEM. LIMITS: 075064 075757 000674 00444.
DISK-BLK-LIMITS: 000121 000121 000001 00001.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION:	TITLE	IDENT	FILE
. BLK.: (RW, I, LCL, REL, CON)	075064	000000	00000.
CSTST4: (RW, I, LCL, REL, CON)	075064	000674	00444.
	075064	000674	00444. CSTST4
\$\$ALVC: (RW, D, LCL, REL, CON)	075760	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

TCCSD: 075064-R. TCCSU: 075416-R.

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: CSTSTS

R/W-MEM-LIMITS: 075064 077667 002604 01412.
DISK-BLK-LIMITS: 000122-000124 000003 00003.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	075064	000000	00000.
CSTST5: (RW, I, LCL, REL, CON)	075064	000554	00364.
	075064	000554	00364.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	075640	000000	00000.
\$\$\$RESL: (RW, I, LCL, REL, CON)	075640	002026	01046.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.
	CSTST5		CSTST5.OBJ:1

GLOBAL-SYMBOLS:

TDCS: 075064-R:

QNT: TSK: 1 MEMORY-ALLOCATION MAP: TKB: 18.14
MRPSUB: 27-MAR-80

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: MRPSUB

R/W-MEM-LIMITS: 074040 075533 001474 00828.
DISK-BLK-LIMITS: 000125 000126 000002 00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	074040	000000	00000.
MRPSUB: (RW, I, LCL, REL, CON)	074040	000506	00326.
	074040	000506	00326. MRPSUB.
PPSUB: (RW, I, LCL, REL, CON)	074546	000766	00502.
	074546	000766	00502. PPSUB.
\$\$ALVC: (RW, D, LCL, REL, CON)	075534	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

LBMRP: 074212-R LBPP: 074546-R MRPCR: 074456-R MRPLB: 074406-R PPLB: 074742-R SEQMM: 074040-R
LBMSC: 074310-R LBPSC: 074644-R MRPCRA: 074464-R PPCR: 075012-R SELPG: 075074-R WRTMM: 074126-R

QMT.TSK:16 MEMORY-ALLOCATION MAP, TKB, BASE 20
QBTEST 27-MAR-80 16,14 Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

*** SEGMENT: QBTEST

R/W-MEM. LIMITS: 075534 077003 001250 00600.
DISK-BLK. LIMITS: 000127 000130 000002.00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK.: (RW, I, LCL, REL, CON)	075534	000000	00000.
QBTEST: (RW, I, LCL, REL, CON)	075534	001250	00600.
	075534	001250	00600. QBTEST
\$\$\$ALVC: (RW, D, LCL, REL, CON)	077004	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

STUFQB-075534-R TCQBD-076436-R TCQBU-076546-R T1QB-075672-R T6QB-076054-R T7QB-076244-R

*** SEGMENT: QRTEST.

R/W MEM. LIMITS: 075534 076653 001120 00592.
DISK-BLK LIMITS: 000131 000132 000002-00002.

MEMORY-ALLOCATION SYNOPSIS:

SECTION:	TITLE	IDENT.	FILE
. BLK: (RW, I, LCL, REL, CON)	075534	000000	00000.
QRTEST: (RW, I, LCL, REL, CON)	075534	001120	00592.
	075534	001120	00592. QRTEST.
\$\$ALVC: (RW, D, LCL, REL, CON)	076654	000000	00000.
\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002.

GLOBAL SYMBOLS:

STUFQR: 075534-R TCQRD: 076336-R TCQRU: 076426-R T1QR: 075652-R T6QR: 076014-R T7QR: 076164-R

*** SEGMENT: QXTEST

R/W-MEM-LIMITS: 075534 076643 001110 00584.
DISK-BLK-LIMITS: 000133 000134 000002 00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION	TITLE	IDENT	FILE
.BLK: (RW,I,LCL,REL,CON)	075534	000000	00000.
QXTEST: (RW,I,LCL,REL,CON)	075534	001106	00582.
	075534	001106	00582.
QXTEST			QXTEST.OBJ:1
\$\$ALVC: (RW,D,LCL,REL,CON)	076642	000000	00000.
\$\$RTS: (RW,I,GBL,REL,DVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

STUFQX: 075534-R TCQXD: 076226-R TCQXU: 076306-R T10X: 075632-R T60X: 075754-R T70X: 076104-R

UNDEFINED-REFERENCES:

MD:INL
MD:INR
PLR:EL
PLR:ER

*** SEGMENT: FATEST.

R/W MEM. LIMITS: 075534 076777 001244 00676.
DISK-BLK-LIMITS: 000135 000136 000002 00002.

MEMORY-ALLOCATION-SYNOPSIS:

SECTION.	TITLE	IDENT.	FILE.
. BLK.: (RW, I, LCL, REL, CON)	075534	000000	00000.
FATEST: (RW, I, LCL, REL, CON)	075534	001242	00674.
	075534	001242	00674. FATEST.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	076776	000000	00000.
\$\$\$RTS.: (RW, I, GBL, REL, DVR)	073124	000002	00002.

GLOBAL-SYMBOLS:

STUFFA: 075534-R. TCFAD: 076336-R. TCFAD: 076426-R. T1FA: 075652-R. T6FA: 076014-R. T7FA: 076164-R

GMT,FSK:16 MEMORY ALLOCATION MAP Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2
RTEST 27-MAR-80 16:14

*** SEGMENT: RTEST.

R/W MEM LIMITS: 075534 076357 000624 00404.
DISK BLK LIMITS: 000137 000137 000001 00001.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	075534	000000	00000.
RTEST: (RW, I, LCL, REL, CON)	075534	000622	00402.
\$\$\$ALVC: (RW, D, LCL, REL, CON)	076356	000000	00000.
\$\$\$RTS: (RW, I, GBL, REL, DVR)	073124	000002	00002.

RTEST: RTEST.OBJ:1

GLOBAL SYMBOLS:

ICA 076140-R IMA 075750-R STCA 075626-R STMA 075534-R

UNDEFINED REFERENCES:

MD: INR
PLR: ER
REGEN:

*** SEGMENT: MRPSUB

R/W MEM LIMITS: 074040 076277 002240 01184.
DISK BLK LIMITS: 000140 000142 000003 00003.

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK: (RW, I, LCL, REL, CON)	074040	000000	00000
CDTEST: (RW, I, LCL, REL, CON)	074040	001532	00058
	074040	001532	00058
MRPSUB: (RW, I, LCL, REL, CON)	075572	000506	00326
	075572	000506	00326
\$\$\$ALVC: (RW, D, LCL, REL, CON)	076300	000000	00000
\$\$\$RTS: (RW, I, GBL, REL, OVR)	073124	000002	00002
			CDTEST.OBJ:1
			MRPSUB.OBJ:1

GLOBAL SYMBOLS:

LBMRP: 075744-R MRPCR: 076210-R MRPLB: 076140-R STUFCD: 074040-R TCCDU: 074754-R T6CD: 074330-R WRTMM: 075660-R
LBMSC: 076042-R MRPCRA: 076216-R SEQMM: 075572-R TCCDD: 074662-R TICD: 074162-R T7CD: 074504-R

*** TASK BUILDER STATISTICS:

TOTAL WORK FILE REFERENCES: 63015.
WORK FILE READS: 0.
WORK FILE WRITES: 0.
SIZE OF CORE POOL: 12970. WORDS (50. PAGES)
SIZE OF WORK FILE: 11776. WORDS (46. PAGES)

ELAPSED TIME: 00:00:55

```

1          .TITLE - SMT.
2          ;
3          ;
4          ;
5          SUBDOCUMENT PROCESSOR DIAGNOSTICS.
6          MAIN MODULE.
7          ;
8          ;
9          ;
10         THIS MODULE EXECUTES TEST CYCLES. ALL CONTROL
11         INFORMATION FOR A TEST CYCLE IS OBTAINED FROM THE
12         COMMAND LINE. SMT BUILDS TABLES AND SETS FLAGS IN
13         ORDER THAT THERE NEED BE NO USER INTERVENTION DURING
14         THE EXECUTION OF THE CYCLE. THE PROMPTS FOR COMMAND
15         LINE INPUT AND GENERAL SMT ACTION ARE:
16
17         1. PROMPT FOR ALL TESTS, ALL MEMORIES, FULL RANGE.
18         IF THE ANSWER IS 'Y', SMT BUILDS A COMPLETE MEMORY
19         TEST CYCLING TABLE, THEN PROCEEDS TO 7. (FOR THE
20         STRUCTURE OF THE TABLE, SEE BELOW REF TABLE - CURRENT JUMP
21         TABLE).
22         IF THE ANSWER IS 'N', PROCEED TO 2 FOR MORE SELECTIVE PROMPTS.
23         2. [DELETED]
24         3. PROMPT FOR MEMORIES TO BE TESTED.
25         SMT SCANS THE COMMAND LINE RESPONSE AND SETS A FLAG FOR
26         EACH MEMORY MNEMONIC IT FINDS.
27         4. PROMPT FOR MEMORY LIMITS IF ANY MEMORIES WERE SELECTED.
28         IF A MEMORY WAS SELECTED IN 3 (IE, ITS FLAG WAS SET),
29         SMT PROMPTS FOR NUMERICAL VALUES WHICH DEFINE THAT
30         PORTION OF A MEMORY ON WHICH THE TESTS ARE TO BE RUN. SMT
31         PUTS THE MEMORY LIMITS FROM THE COMMAND LINE RESPONSE
32         INTO A TABLE.
33         5. [DELETED]
34         6. PROMPT FOR MEMORY TESTS IF ANY MEMORIES WERE SELECTED.
35         SMT SCANS THE COMMAND LINE RESPONSE FOR TEST NUMBERS,
36         THEN, FOR EACH WHOSE FLAG IS UP, SMT BUILDS AN ENTRY
37         IN THE MEMORY CURRENT JUMP TABLE (SEE BELOW).
38         7. PROMPT FOR LOOP ON TEST.
39         SMT SETS A FLAG THAT DETERMINES WHETHER A TEST CYCLE
40         WILL BE EXECUTED ONCE OR EXECUTED REPEATEDLY.
41         7. PROMPT FOR ERROR OPTIONS.
42         SMT SETS FLAGS WHICH DETERMINE WHAT ACTION WILL BE
43         TAKEN IN THE EVENT OF AN ERROR.
44         8. PUT OUT DIRECTIONS FOR STOPPING TEST.
45         SMT ALLOWS THE TEST CYCLE TO BE INTERRUPTED BY
46         AN UNSOLICITED CHARACTER INTERRUPT FROM THE TERMINAL.
47         9. TEST CYCLE BEGINS.
48         NO MORE PROMPTING.
49
50         REFERENCE TABLE - CURRENT JUMP TABLE.
51
52         THE ACTUAL EXECUTION OF A TEST CYCLE DEPENDS UPON THE
53         CONTENTS OF THE CURRENT JUMP TABLE.
54
55         EACH MEMORY TEST 1 - 13 HAS ITS OWN CONTROL ROUTINE IN SMT.
56         THE REFERENCE TABLE ENTRIES ARE THE ADDRESSES OF THESE
57         ROUTINES. IN CONTIGUOUS POSITIONS THE CONTROL ROUTINE

```



```

58 ; ADDRESS REPLICATED A NUMBER OF TIMES; THE NUMBER OF
59 ; REPLICATIONS IS EQUAL TO THE NUMBER OF MEMORIES THAT
60 ; CAN BE TESTED. (IE. 12, THE VALUE OF THE EQUATE 'NMEMS').
61 ; SO, FOR EXAMPLE, SINCE THERE ARE 6 MEMORIES, TEST 1'S
62 ; CONTROL ROUTINE ADDRESS WILL BE REPEATED IN THE REFERENCE
63 ; TABLE 6 TIMES:
64 ;
65 ; .WORD T1.T1.T1.T1.T1.T1.T1.T1.T1.T1.T1.T1
66 ;
67 ; EACH ADDRESS HERE IS A PLACE-HOLDER FOR A MEMORY. THE
68 ; FOLLOWING IS A TABLE OF PLACE-HOLDER VALUES (POSITIONS OF
69 ; ADDRESSES IN THE ABOVE LINE) AND THE MEMORIES TO WHICH THEY
70 ; CORRESPOND:
71 ;
72 ; 0 DEX MEMORY
73 ; 1 QLB REFERENCE PAGE
74 ; 2 QLB PAGE
75 ; 3 SUBREAD MEMORY
76 ; 4 SIDMEM PAGE 1
77 ; 5 SIDMEM PAGE 2
78 ;
79 ; THE IDEA HERE IS THAT RATHER THAN HAVING SEPARATE CONTROL
80 ; ROUTINES FOR EACH MEMORY FOR EACH TEST, QMT CAN MAKE
81 ; USE OF THE POSITIONS OF ADDRESSES IN THE TABLE.
82 ;
83 ; THE FILLING OF THE CURRENT JUMP TABLE TAKES PLACE AS
84 ; FOLLOWS: A UNIQUE FLAG IS SET IN A FLAG WORD FOR EACH
85 ; MEMORY WHOSE MNEMONIC QMT ENCOUNTERS IN THE COMMAND
86 ; LINE RESPONSE. THE THE PROMPT 'SELECT MEMORIES'; EACH
87 ; POSITION IN THE FLAG WORD (0 - 11) CORRESPONDS TO A
88 ; MEMORY PLACE HOLDER POSITION IN THE REF TABLE AND
89 ; CURRENT JUMP TABLE, EG. THE FLAG FOR QLB PAGE 0 IS IN
90 ; POSITION 9 IN THE FLAG WORD AND THE PLACE-HOLDER
91 ; POSITION FOR QLB PAGE 0 IS 9 (SEE ABOVE). THEN FOR EACH
92 ; TEST NUMBER QMT ENCOUNTERS IN THE COMMAND LINE RESPONSE TO
93 ; THE PROMPT 'SELECT MEMORY TEST(S)', QMT MOVES THE ADDRESS
94 ; OF THAT TEST'S CONTROL ROUTINE FROM THE REF TABLE TO THE
95 ; CURRENT JUMP TABLE DEPENDING UPON THE MEMORY FLAG SETTINGS.
96 ; IE. QMT SCANS THE MEMORY FLAG WORD AND FOR EVERY BIT SET MOVES
97 ; AN ADDRESS FROM THE REF TABLE TO THE CURRENT JUMP TABLE.
98 ;
99 ; EXAMPLE:
100 ; IF QMT ENCOUNTERS A '1' IN THE COMMAND LINE RESPONSE TO
101 ; 'SELECT MEMORY TEST(S)' AND THE FLAGS FOR MRP DATA MEMORY
102 ; AND QLB PAGE 1 HAVE BEEN PREVIOUSLY SET, QMT WILL MOVE
103 ; TEST 1'S CONTROL ROUTINE ADDRESS FROM THE REF TABLE TO THE
104 ; CURRENT JUMP TABLE IN POSITIONS 1 AND 10 FOR TEST 1.
105 ;
106 ; REF TABLE:
107 ; .WORD T1.T1.T1.T1.T1.T1.T1.T1.T1.T1.T1.T1
108 ; .WORD T2.T2.T2.T2.T2.T2.T2.T2.T2.T2.T2.T2
109 ;
110 ; CURRENT JUMP TABLE:
111 ; .WORD 0.T1.0.0.0.0.0.0.0.0.T1.0
112 ; .WORD 0.0.0.0.0.0.0.0.0.0.0.0
113 ;
114 ;

```

```

115      ; TEST CYCLING
116      ; CYCLING IS DESCRIBED HERE AS IF ONLY MEMORIES ARE
117      ; BEING TESTED. THE ADJUSTMENTS NECESSARY IF EITHER
118      ; ONLY REGISTERS OR BOTH REGISTERS AND MEMORIES ARE
119      ; BEING TESTED ARE DESCRIBED BELOW AT 'MTSET' AND
120      ; 'JMPMT'.
121      ;
122      ; THE CYCLE CONTROL ROUTINE MAINTAINS A POINTER TO THE
123      ; CURRENT JUMP TABLE. THE CONTROL ROUTINE SCANS THE TABLE
124      ; UNTIL IT FINDS A NON-ZERO ENTRY. IT DERIVES THE MEMORY
125      ; TO BE TESTED FROM THE PLACE-HOLDING CHARACTERISTICS OF
126      ; THE TABLE:
127      ;
128      ; 1. GETS THE POSITION OF THE CURRENT NON-ZERO ENTRY
129      ;    RELATIVE TO THE BEGINNING OF THE TABLE.
130      ; 2. DIVIDES THIS NUMBER BY THE NUMBER OF MEMORIES. THE
131      ;    REMAINDER GIVES THE PLACE-HOLDER VALUE.
132      ;
133      ; FOR EXAMPLE, IF 'QEX WINDOW' AND 'TEST 2' HAVE BEEN
134      ; SELECTED, THE CURRENT JUMP TABLE WILL CONTAIN THE
135      ; FOLLOWING INFORMATION:
136      ;
137      ; .WORD 0.0.0.0.0.0.0.0.0.0
138      ; .WORD 0.0.0.T2.0.0.0.0.0.0.0
139      ;
140      ; THE FIRST NON-ZERO ENTRY IN THE TABLE IS AT OFFSET 15.
141      ; THE REMAINDER FROM THE DIVISION OF 15 BY 12 (12 = NUMBER
142      ; OF MEMORIES) IS 3. THE VALUE 3 IS THE PLACE-HOLDER
143      ; VALUE FOR THE QEX WINDOW MEMORY.
144      ;
145      ; THE CONTROL ROUTINE PASSES CONTROL TO THE TEST CONTROL
146      ; ROUTINE WHOSE ADDRESS IS THE CURRENT NON-ZERO ENTRY IN
147      ; THE CURRENT JUMP TABLE. THE CYCLE CONTROL ROUTINE PASSES
148      ; THE REMAINDER FROM THE ABOVE DIVISION IN R0. THE TEST
149      ; CONTROL ROUTINE USES THE CONTENTS OF R0 AS AN INDEX
150      ; INTO A TABLE OF QMT SUB-MODULE MEMORY TEST ADDRESSES.
151      ; THE TEST CONTROL ROUTINE IN TURN PASSES CONTROL TO THE
152      ; ROUTINE IN THE SUB-MODULE THAT WILL EXECUTE THE TEST ON
153      ; THE CORRECT MEMORY.
154      ;
155      ; THE SUB-MODULES OF QMT ARE:
156      ; MMTST. TEST MRP MICROPGM MEMORY.
157      ; MDTEST. TEST MRP DATA MEMORY.
158      ; QXTEST. TEST QEX MEMORIES.
159      ; CSTEST. TEST CP CONTROL STORE.
160      ; CDTEST. TEST CP DATA MEMORY.
161      ; FATST. TEST FAL MEMORIES.
162      ; QRTEST. TEST QLB REFERENCE PAGE.
163      ; QBTEST. TEST QLB MEMORIES.
164      ;
165      ; THE MODULE RTEST CONTAINS THE REGISTER TESTS.
166      ; THERE ARE MODULES WHICH CONTAIN SUBROUTINES FOR THE
167      ; QMT SUB-MODULES. THESE SUBROUTINE MODULES ARE:
168      ; MRPSUB, CPSUB, AND PPSUB. ROUTINES IN THESE MODULES
169      ; ARE GLOBAL, ALLOWING CROSS-USAGE.
170      ;
171      ; ALL ROUTINES IN THE QMT SUB-MODULES RETURN TO THE TEST

```

```

172. ; CONTROL ROUTINE THAT CALLED THEM. THE TEST CONTROL ROUTINES
173. ; RETURN TO THE CYCLING ROUTINE WHICH SCANS THE CURRENT JUMP
174. ; TABLE FOR THE NEXT NON-ZERO ENTRY.
175. ;
176. ;
177. ; EXIT FROM THE PROGRAM DEPENDS UPON THE STATUS OF THE CURRENT
178. ; JUMP TABLE, LOOP OPTIONS, HALT OPTIONS, OR TERMINAL INPUT.
179. ;
180. ; JUMP TABLE EMPTY. - EXIT.
181. ; LOOP OPTION OFF. - EXECUTE ONE TEST CYCLE.
182. ; LOOP COUNT. - EXECUTE A NUMBER OF CYCLES EQUAL TO
183. ; THE LOOP COUNT.
184. ; HALT OPTION ON. - HALT AFTER ONE ERROR.
185. ; COUNT + 'H'. - PRINT A NUMBER OF MESSAGES EQUAL TO
186. ; THE COUNT AND HALT.
187. ;
188. ; WHILE THE TESTS ARE RUNNING, THE ENTERING FROM THE TERMINAL
189. ; OF ANY CHARACTER OTHER THAN W, C, P, OR T (THESE HAVE SPECIAL
190. ; MEANINGS - SEE THE ROUTINE 'AST?') STOPS THE TESTS IMMEDIATELY.
191. ;
192. ;
193. ; REGISTER TESTS.
194. ;
195. ; REGISTERS:
196. ; MRP MEMORY ADDRESS REGISTER.
197. ; CP MEMORY ADDRESS REGISTER.
198. ;
199. ;
200. ; TESTS:
201. ; TEST 01 WRITE ZEROS.
202. ; TEST 02 WRITE ONES.
203. ; TEST 03 WRITE USER SUPPLIED TEST PATTERN.
204. ; TEST 04 INCREMENT MAR.
205. ;
206. ; MEMORY TESTS
207. ;
208. ; MEMORIES:
209. ; MRP MICROPROGRAM MEMORY.
210. ; MRP DATA MEMORY.
211. ; QEX WINDOW MEMORY.
212. ; QEX LOCATION MEMORY.
213. ; CP CONTROL STORE.
214. ; CP DATA MEMORY.
215. ; FAL POINTER MEMORY.
216. ; FAL COUNTER MEMORY.
217. ; QLB REFERENCE PAGE.
218. ; QLB PAGE 0.
219. ; QLB PAGE 1.
220. ; QLB PAGE 2.
221. ;
222. ; TESTS:
223. ; TEST 01 WRITE ADDRESS INTO ITSELF.
224. ; TEST 02 WRITE ZEROS.
225. ; TEST 03 WRITE ONES.
226. ; TEST 04 WRITE AAAA.
227. ; TEST 05 WRITE CCCC AND 3333.
228. ; TEST 06 WRITE TALK TEST.

```

```
229      ;      TEST-07  WRITE ADDRESS COMPLEMENT INTO ADDRESS.  
230      ;      TEST-08  WRITE 00FF AND FF00  
231      ;      TEST-09  SHIFT-BIT TEST.  
232      ;      TEST-10  WRITE USER SUPPLIED TEST PATTERN.  
233      ;      TEST-11  BIT MARCH TEST.  
234      ;      TEST-12  ADDRESSING TEST.  
235      ;      TEST-13  FILE COMPARE TEST.  
236      ;  
237      ;  
238      ;      ASSEMBLY:      FROM [5,3].  
239      ;      MCR>MAC SMT,LP=IM04,SMT.  
240      ;  
241      ;      TASK BUILD:      ON HQR PACK.  
242      ;      SMT/DA,SMT=SMT,RTEST,MMTEST,MDTEST,OXTEST,CSTEST,CDTEST.  
243      ;      FATEST,QRTEST,QBTEST,MRPSUB,CPSUB,PPSUB.  
244      ;      /.  
245      ;      PAR=PAR14K.  
246      ;      TASK=...SMT.  
247      ;      ASG=TT0:1  
248      ;      ASG=TT0:2.  
249      ;      /.  
250      ;      //
```

```

252.      ;
253.      ;
254.      ; LOCAL DATA AREAS
255.      ;
256.      ;
257.      ;
258.      ; .MCALL . Q10W$, Q10$, EXIT$, ABRT$, GCML$, GCMLB$, FRSZ$, CLEF$, ASTX$
259.      ; .MCALL . FDBDF$, FDRC$, FDBK$, FDOP$, NMBLK$, OPEN$, FINIT$, CLOSE$, READ$
260.      ; .MCALL . WTSE$
261.      ;
262.      ;
263.      ; LUN: TT = 1 ; LUN FOR TERMINAL
264.      ; EFN: 1 = 1 ; EVENT FLAG FOR TERMINAL
265.      ; EFN: 3 = 3 ; EVENT FLAG FOR HOR INTERRUPTS
266.      ; CMILUN = 2 ; LUN FOR GCML
267.      ; INLUN = 3 ; LUN FOR LDXX DAT FILES
268.      ;
269.      ; SETTINGS FOR FLAG 'BASE'
270.      ;
271.      ; LOOP = 1 ; LOOP ON TEST
272.      ; TEST3 = 2 ; USER PATTERN REGISTER TEST SELECTED
273.      ; TEST6 = 4 ; CROSS-TALK TEST SELECTED
274.      ; TEST10 = 10 ; USER PATTERN MEMORY TEST SELECTED
275.      ; REGISTR = 20 ; REGISTERS SELECTED
276.      ; MEMORY = 40 ; MEMORIES SELECTED
277.      ; ALLTST = 100 ; ALL MEMORY TESTS, ALL MEMORIES
278.      ; HALT = 200 ; HALT FLAG
279.      ; ERROR = 400 ; ERROR ENCOUNTERED
280.      ; FIRST == 1000 ; FIRST RECORD READ IN LDXX FILE
281.      ;
282.      ; SELECTION FLAGS
283.      ; TO BE USED WITH FLAG WORD 'MSEL' (MEMORIES)
284.      ;
285.      ; QX = 1 ; QEX MEMORY
286.      ; QR = 2 ; QLB REFERENCE PAGE
287.      ; QB = 4 ; QLB PAGE
288.      ; SR = 10 ; SUBREAD MEMORY
289.      ; S1 = 20 ; SIDMEM PAGE 1
290.      ; S2 = 40 ; SIDMEM PAGE 2
291.      ;
292.      ; .NLIST BEX
293.      ; MYSELF: .RAD50 /...SMT/
294.      ; TSKTCB: .WORD 0 ; TCB OF MY TASK
295.      ; OLDVEC: .WORD 0 ; OLD VECTOR AT 274
296.      ; ASTWRD: .WORD 0 ; UNSOLICITED CHAR FROM TERMINAL
297.      ; STAT: .BLKW 2 ; IO STATUS
298.      ; ERWORD: .WORD 0 ; MESSAGE INDEX
299.      ; ERLIM: .WORD 0 ; ERROR MESSAGE COUNT
300.      ; BINWD: .WORD 0 ; VALUE CONVERTED FROM COMMAND LINE
301.      ; LOWER: .WORD 0 ; LOWER MEMORY LIMITS HOLD AREA
302.      ; UPPER: .WORD 0 ; UPPER MEMORY LIMITS HOLD AREA
303.      ; BASE: .WORD 0 ; ALL-PURPOSE FLAG WORD
304.      ; CODE: .WORD 0 ; MEMORY SELECT CODE
305.      ; APLACE: .WORD 0 ; WORK AREA FOR CSRI ROUTINE
306.      ; DATA1: .WORD 0
307.      ; VIRT: .WORD 0,1 ; RELATIVE BLOCK IN FILE
308.      ; FVER: .WORD 0 ; FILE VERSION NO
309.      ; LCOUNT: .WORD 0 ; WORD COUNT IN RECORD

```

```

309 000050 000000          WCOUNT: .WORD 0          ;WORKING COUNTER
310 000052 000000          RT3: .WORD 0          ;USER PATTERN REGISTER TEST
311 000054 000000          MT10: .WORD 0          ;USER PATTERN MEMORY TEST
312 000056 000000          GCMBUF: .BLKW 41      ; TERMINAL INPUT
313 000200 000000          GCMLN: .WORD 0          ;COMMAND LINE LENGTH
314 000202 000000          GCMPT: .WORD 0          ; POINTER TO COMMAND LINE
315 000204 000000          PASSH: .WORD 0          ;PASS COUNT UPPER WORD
316 000206 000001          PASS: .WORD 1          ;PASS COUNT LOWER WORD
317 000210 000000          LOOPCT: .WORD 0          ; LOOP COUNT
318 000212 000000          MSEL: .WORD 0          ;MEMORY SELECT FLAG WORD
319 000214 000000          MTPNT: .WORD 0          ;CURRENT JUMP TABLE POINTER
320 000216 000000          MTCNT: .WORD 0          ;CURRENT JUMP TABLE COUNT
321 000220 000000          NXTPNT: .WORD 0          ;CURRENT JUMP TABLE POINTER - REFRESH *MTPNT
322 000222 000000          NXTCNT: .WORD 0          ;CURRENT JUMP TABLE COUNT - REFRESH *MTCNT
323
324
325          ;
326 000224          ; TABLE USED IN TRANSLATING TEST NUMBERS FROM ASCII DECIMAL TO BINARY
327          ;
328 000305' 000305' 002 003          TROCT:
329          ; = . +61
330          ; .BYTE 1,2,3,4,5,6,7,8,9,10,11,12,13
331          ; = TROCT+200
332          ;
333 000424          ; TABLE USED IN TRANSLATING FROM ASCII HEX TO BINARY
334          ;
335 000504' 000504' 001 002          TRTBL:
336          ; = . +60
337 000525' 000525' 013 014          ; .BYTE 0,1,2,3,4,5,6,7,8,9
338          ; = TRTBL+101
339          ; .BYTE 10,11,12,13,14,15
340          ; = TRTBL+200
341          ;
342 000624 060 061 062          ; TABLE USED IN CONVERTING FROM BINARY TO ASCII HEX
343          ;
344          ; TRTBL2: .ASCII /0123456789ABCDEF/
345          ;
346          ; STRING OF ALL REGISTER TESTS (USE AS PSEUDO COMMAND LINE)
347 000644 061 040 062          ; STRING OF ALL MEMORY TESTS (SAME USE)
348          ;
349          ; ALLMEM: .ASCII /1 2 3 4 5 6 7 8 9 11 12/
350          ; STRMEM: = .-ALLMEM
351          ; .EVEN
352          ;
353          ; MEMORY TABLE
354          ; VALID MEMORY MNEMONICS AND MEMORY SELECT FLAG SETTINGS
355          ; USED IN SETTING FLAG WORD *MSEL FOR MEMORIES TO BE TESTED
356 000674          ;
357 000674 121 130          MTBL:
358 000676 000001          ; .ASCII /OX/          ; OEX MEMORY
359 000700 123 106          ; .WORD OX
360 000702 000002          ; .ASCII /SF/          ; OLB REFERENCE PAGE
361 000704 123 060          ; .WORD OR
362 000706 000004          ; .ASCII /S0/          ; OLB PAGE
363 000710 123 122          ; .WORD 0B
364 000712 000010          ; .ASCII /SR/          ; SUBREAD
365 000714 123 061          ; .WORD SR

```

```

366 000716 000020          .WORD  S1
367 000720      123      062  .ASCII  /S2/          :SIDMEM PAGE 2.
368 000722 000040          .WORD  S2
369 000724 000014          MTBLN  =  <.-MTBL/2>
370 000726 000006          NMEMS  =  MTBLN/2.          :NUMBER OF MEMORIES.
371
372
373
374 000724          :
375 000724 000004          : MEMORY SELECT CODE TABLES.
376 000726 000006          :
377 000730 000005          CDLOW:  .WORD  S$0X
378 000732 000007          .WORD  S$0R
379 000734 000010          .WORD  S$0B
380 000736 000014          .WORD  S$SR
381
382 000740          .WORD  S$S1
383 000740 000004          .WORD  S$S2
384 000742 000006          :
385 000744 000005          CDHIGH: .WORD  S$0X
386 000746 000007          .WORD  S$0R
387 000750 000012          .WORD  S$0B
388 000752 000016          .WORD  S$SR
389
390          :
391          : MEMORY TEST REFERENCE TABLE.
392 000754          :
393 000754 006742' 006742' 006742' 006742' MTREF: .WORD  T1,T1,T1,T1,T1,T1
394 000770 006752' 006752' 006752' 006752' .WORD  T2,T2,T2,T2,T2,T2
395 001004 006766' 006766' 006766' 006766' .WORD  T3,T3,T3,T3,T3,T3
396 001020 007004' 007004' 007004' 007004' .WORD  T4,T4,T4,T4,T4,T4
397 001034 007022' 007022' 007022' 007022' .WORD  T5,T5,T5,T5,T5,T5
398 001050 007052' 007052' 007052' 007052' .WORD  T6,T6,T6,T6,T6,T6
399 001064 007112' 007112' 007112' 007112' .WORD  T7,T7,T7,T7,T7,T7
400 001100 007122' 007122' 007122' 007122' .WORD  T8,T8,T8,T8,T8,T8
401 001114 007152' 007152' 007152' 007152' .WORD  T9,T9,T9,T9,T9,T9
402 001130 007206' 007206' 007206' 007206' .WORD  TA,TA,TA,TA,TA,TA
403 001144 007224' 007224' 007224' 007224' .WORD  TB,TB,TB,TB,TB,TB
404 001160 007350' 007350' 007350' 007350' .WORD  TC,TC,TC,TC,TC,TC
405 001174 007454' 007454' 007454' 007454' .WORD  TD,TD,TD,TD,TD,TD
406 000015          MT  =  <.-MTREF>/MTBLN
407
408          :
409          : LOWER AND UPPER MEMORY LIMITS (REFERENCE)
410
411 001210          :
412 001210 000377          LIMREF: .WORD  255.          :OEX MEMORY UPPER LIMIT
413 001214 000000          .WORD  0          :LOWER LIMIT
414 001216 000377          .WORD  255.          :OLB REFERENCE PAGE UPPER LIMIT
415 001220 000000          .WORD  0          :OLB PAGE UPPER LIMIT
416 001222 000377          .WORD  0          :SUBREAD UPPER LIMIT
417 001224 000377          .WORD  255.          :SIDMEM PAGE 1 UPPER LIMIT
418 001226 000000          .WORD  0          :SIDMEM PAGE 2 UPPER LIMIT
419 001230 007777          .WORD  4095.
420 001232 000000          .WORD  0
421 001234 007777          .WORD  4095.
422 001236 000000          .WORD  0
    
```

```

423          000014          LIMNUM: =.      <.-LIMREF>/2.
424          ;
425          ;
426          ;
427          000000          ;
428          000000          MEMORY TEST ROUTINE ADDRESSES (CURRENT JUMP TABLE)
429          ;
430          001240          RT: =.      0          :NUMBER OF REGISTER TESTS
431          001240          NREGS: =.     0          :NUMBER OF REGISTERS
432          ;
433          ;
434          ;
435          001474          MTSUB:
436          001474          ;
437          ;
438          ;
439          ;
440          ;
441          ;
442          001524          ;
443          001540          ;
444          001554          ;
445          001570          ;
446          001604          ;
447          001620          ;
448          001634          ;
449          ;
450          ;
451          ;
452          001650          ;
453          001652          ;
454          001654          ;
455          001656          ;
456          001660          ;
457          001662          ;
458          001664          ;
459          001666          ;
460          001670          ;
461          001672          ;
462          ;
463          ;
464          001674          ;
465          001700          ;
466          001704          ;
467          001711          ;
468          001716          ;
469          001723          ;
470          001734          ;
471          001746          ;
472          001760          ;
473          ;
474          ;
475          ;
476          001765          ;
477          001765          ;
478          002010          ;
479          002033          ;

```

```

STADDR: .WORD STUFSP,STUFSP,STUFSP,STUFSP,STUFSP,STUFSP.
T1ADDR: .WORD T1SP,T1SP,T1SP,T1SP,T1SP,T1SP.
T6ADDR: .WORD T6SP,T6SP,T6SP,T6SP,T6SP,T6SP.
T7ADDR: .WORD T7SP,T7SP,T7SP,T7SP,T7SP,T7SP.
TCADDR: .WORD TCSPD,TCSPD,TCSPD,TCSPD,TCSPD,TCSPD.
TCUADD: .WORD TCSPU,TCSPU,TCSPU,TCSPU,TCSPU,TCSPU.
TDADDR: .WORD TDNUL,TDNUL,TDNUL,TDNUL,TDNUL,TDNUL.

```

```

ERROR ROUTINE WORK AREAS
CKDATA: .WORD 0          : TEST PATTERN
CK2: .WORD 0          : TEST 12 READ TEST PATTERN
CK3: .WORD 0          : TEST 12 WRITE TEST PATTERN
PREADD: .WORD 0          : CURRENT MEMORY ADDRESS
ERRADD: .WORD 0          : ADDRESS AT ERROR
ERRCT: .WORD 0          : NUMBER OF ERROR WORDS TO PRINT
ERW1: .WORD 0          : ERRONEOUS DATA FROM MEMORY - 1
ERW2: .WORD 0          : - 2
ERW3: .WORD 0          : - 3
ERW4: .WORD 0          : - 4

```

```

LMM: .RAD50 /LDMM/
LCS: .RAD50 /LDCS/
ASWRK: .ASCII // //
TMSG: .ASCII /TEST//
PMSG: .ASCII /PASS//
AMSG: .ASCII /ADDRESS//
EMSG: .ASCII /EXPECTED//
RMSG: .ASCII /RECEIVED//
UNMSG: .ASCII /* * */

```

```

MEMORY NAMES
MFTBL:

```

```

.ASCII /QEX MEMORY/
.ASCII /QLB REFERENCE PAGE/
.ASCII /QLB PAGE/

```



```

480 002056      123      125      102      .ASCII /SUBREAD MEMORY /
481 002101      123      111      104      .ASCII /SIDMEM PAGE 1 /
482 002124      123      111      104      .ASCII /SIDMEM PAGE 2 /
483
484          000023      ; MNAMSZ = <.-MFTBL>/NMEMS
485      ;
486      ; PRINT LINE
487      ;
488 002147      015      012      .BYTE 15,12 ;PRECEDE PRINT LINE WITH CRLF
489 002151      PRINT:
490          000116      .REPT 78
491      .BYTE 40
492      .ENDR
493
494      ;
495      ; TABLE OF MESSAGES
496      ;
497      ;
498 002267      000      .BYTE 0
499 002270      015      012      015      .BYTE 15,12,15,12
500 002274      124      105      123      .ASCIZ /TEST(S) ENDED/
501 002312      015      012      015      .BYTE 15,12,15,12
502 002316      124      105      123      .ASCIZ /TEST(S) HALTED/
503 002335      015      012      .BYTE 15,12
504 002337      105      116      124      .ASCIZ /ENTER ANY CHARACTER TO STOP TEST(S)/
505 002403      015      012      015      .BYTE 15,12,15,12
506 002407      124      105      123      .ASCIZ /TEST(S) STARTED/
507 002427      015      012      015      .BYTE 15,12,15,12,15,12
508 002435      123      125      102      .ASCII /SUBDOCUMENT PROCESSOR DIAGNOSTICS/
509 002476      015      012      000      .BYTE 15,12,0
510 002501      015      012      .BYTE 15,12
511 002503      124      105      123      .ASCIZ /TEST 6 INCOMPATABLE WITH MEMORY LIMITS. TEST DISCARDED./
512 002573      015      012      .BYTE 15,12
513 002575      105      122      122      .ASCII /ERROR: NO SELECTIONS. EXIT./
514 002630      015      012      000      .BYTE 15,12,0
515 002633      015      012      .BYTE 15,12
516 002635      105      122      122      .ASCIZ /ERROR ON READ/
517 002653      015      012      .BYTE 15,12
518 002655      111      116      126      .ASCIZ /INVALID ERROR OPTION/
519 002702      015      012      .BYTE 15,12
520 002704      111      116      126      .ASCIZ /INVALID LOOP OPTION/
521 002730      015      012      .BYTE 15,12
522 002732      111      116      126      .ASCIZ /INVALID TEST PATTERN/
523 002757      015      012      .BYTE 15,12
524 002761      111      116      126      .ASCIZ /INVALID TEST NUMBER/
525 003005      015      012      .BYTE 15,12
526 003007      111      114      114      .ASCIZ /ILLEGAL ODD ADDRESS/
527 003033      015      012      .BYTE 15,12
528 003035      111      116      126      .ASCIZ /INVALID UPPER MEMORY LIMITS/
529 003071      015      012      .BYTE 15,12
530 003073      111      116      126      .ASCIZ /INVALID LOWER MEMORY LIMITS/
531 003127      015      012      .BYTE 15,12
532 003131      111      116      126      .ASCIZ /INVALID MEMORY MNEMONIC/
533 003161      015      012      .BYTE 15,12
534 003163      111      116      126      .ASCIZ /INVALID RESPONSE/
535 003204      015      012      .BYTE 15,12
536 003206      105      116      124      .ASCIZ /ENTER ERROR CONTROL/

```

```

537 003232 015 012 .BYTE 15,12
538 003234 114 117 117 .ASCIZ /LOOP ON TEST(S)?/
539 003255 015 012 .BYTE 15,12
540 003257 105 116 124 .ASCIZ /ENTER PATTERN FOR TEST 10/
541 003311 015 012 .BYTE 15,12
542 003313 105 116 124 .ASCIZ /ENTER PATTERN FOR TEST 3/
543 003344 015 012 .BYTE 15,12
544 003346 123 105 114 .ASCIZ /SELECT MEMORY TEST(S)/
545 003374 015 012 .BYTE 15,12
546 003376 105 116 124 .ASCIZ /ENTER LIMITS FOR QEX MEMORY/
547 003432 015 012 .BYTE 15,12
548 003434 105 116 124 .ASCIZ /ENTER LIMITS FOR QLB REFERENCE PAGE/
549 003500 015 012 .BYTE 15,12
550 003502 105 116 124 .ASCIZ /ENTER LIMITS FOR QLB PAGE/
551 003535 015 012 .BYTE 15,12
552 003537 105 116 124 .ASCIZ /ENTER LIMITS FOR SUBREAD MEMORY/
553 003577 015 012 .BYTE 15,12
554 003601 105 116 124 .ASCIZ /ENTER LIMITS FOR SIDMEM PAGE 1/
555 003640 015 012 .BYTE 15,12
556 003642 105 116 124 .ASCIZ /ENTER LIMITS FOR SIDMEM PAGE 2/
557 003701 015 012 .BYTE 15,12
558 003703 123 105 114 .ASCIZ /SELECT MEMORIES/
559 003723 015 012 .BYTE 15,12
560 003725 101 114 114 .ASCIZ /ALL TESTS, ALL MEMORIES, FULL RANGE?/
561 003773 377 .ASCIZ .BYTE 377
562 ;
563 003774 ;
564 003774 120 101 123 .ASCIZ /PASS NUMBER /
565 000014 .-PMSG2
566 ;
567 004010 105 116 104 .ASCIZ /END OF PASS /
568 000014 .-ENDOF
569 .LIST BEX
570 .NLIST CND
571 .EVEN
572 ;
573 ;
574 ; COMMAND LINE MACRO
575 ;
576 ;
577 004024 GCMBLK: GCMLB$ 2, GCMBUF, CMILUN
578 ;
579 ; INPUT FILE FDB
580 ;
581 004332 INFDB:
582 004332 FDBDF$
583 004472 FDRCA$ FD,RWM
584 004472 FDBCA$ HRL0,512,....STAT
585 004472 FDOPA$ INLUN, INDNB
586 ;
587 004472 INDNB:
588 004472 NMBLK$ ,DAT
589 004530 FSRSZ$ 1

```

```

591 ;
592 ;
593 ; ENTER HERE
594 ;
595 ;
596 ; INITIALIZE HQR
597 ;
598 ; START:
599 004530 CALL OUT1 ; ISSUE INFORMATION MESSAGE
600 004534 016767 000000G 173242 MOV $TKTCB,TSKTCB ; SAVE MY TCB
601 004542 013767 000274 173236 MOV @#274,OLDVEC ; SAVE VECTOR AT 274
602 004550 012737 007516 000274 MOV #BPTISR,@#274 ; MOVE IN MY INTERRUPT HANDLER ADDR
603 004556 FINIT$
604 ;
605 ; PERFORM MASTER RESET AND NO CLOCKS
606 ;
607 004562 012746 177777 MOV #177777,-(SP) ; CLEAR CSR1
608 004566 012746 000010 MOV #0$RSET,-(SP) ; SET MASTER RESET
609 004572 CALL CSR1 ; DO IT
610 004576 012746 000010 MOV #0$RSET,-(SP) ; CLEAR RESET
611 004602 012746 176000 MOV #<0$NCLK>,-(SP) ; SET NO-CLOCKS
612 004606 CALL CSR1
613 ;
614 ; INITIALIZE PPS
615 ;
616 004612 012746 000053 MOV #0$QLA,-(SP) ; ADDRESS SELECT FOR QLB PAGES
617 004616 CALL PPCR
618 004622 012746 002000 MOV #2000,-(SP) ; SEND ADDRESS X'400' (ILLEGAL)
619 004626 CALL LBPP
620 ;
621 ; RESET MRP AND CP
622 ;
623 004632 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
624 004634 012746 000004 MOV #0$MSET,-(SP) ; SET RESET
625 004640 CALL CSR1
626 004644 012746 000004 MOV #0$MSET,-(SP) ; CLEAR RESET
627 004650 005046 CLR -(SP) ; SET NOTHING
628 004652 CALL CSR1
629 ;
630 004656 005046 CLR -(SP) ; CLEAR NOTHING IN CSR1
631 004660 012746 000002 MOV #0$CSET,-(SP) ; SET RESET
632 004664 CALL CSR1
633 004670 012746 000002 MOV #0$CSET,-(SP) ; CLEAR RESET
634 004674 005046 CLR -(SP) ; SET NOTHING
635 004676 CALL CSR1
636 ;
637 004702 012746 000100 MOV #100,-(SP) ; SET QLB ERASE
638 004706 CALL PPCR

```

```

640 ;
641 ;
642 ; PROMPT FOR ALL TESTS, ALL MEMORIES, FULL RANGE.
643 ;
644 ;
645 004712 ALL:
646 004712 CALL ALLSEL ; ISSUE PROMPT
647 004716 103003 BCC 1$ ; NEED A RESPONSE
648 004720 CALL ERR2 ; 'INVALID RESPONSE'
649 004724 000772 BR ALL ; PROMPT AGAIN
650 ;
651 004726 1$: CALL FIND ; LOCATE RESPONSE IN COMMAND LINE
652 004732 103003 BCC 2$ ; OK, VALIDATE RESPONSE
653 004734 122711 CALL ERR2
654 004740 000764 BR ALL
655 ;
656 ; PARSE RESPONSE.
657 ;
658 004742 122711 000116 2$: CMPB #'N,(R1) ; N = NO
659 004746 001002 BNE 3$ ; TRY YES
660 004750 000167 000072 JMP MMSL ; PROMPT FOR REGISTER TESTS
661 004754 122711 000131 3$: CMPB #'Y,(R1) ; Y = YES
662 004760 001403 BEQ MOVE ; OK, SET UP FOR ALL
663 004762 CALL ERR2 ; MUST BE Y OR N
664 004766 000751 BR ALL ; PROMPT AGAIN
665 ;
666 ; SET UP MEMORY LIMITS TABLE FOR CURRENT TEST.
667 ; COPY REFERENCE TABLE TO CURRENT TABLE (IE. TEST
668 ; MEMORIES OVER THE IR FULL RANGE).
669 ;
670 004770 012700 001210 MOVE: MOV #LIMREF,R0 ; POINT TO REF TABLE
671 004774 012701 001474 MOV #CURLIM,R1 ; POINT TO CURRENT TABLE
672 005000 012702 000014 MOV #LIMNUM,R2 ; NUMBER OF WORDS TO MOVE
673 005004 012021 1$: MOV (R0)+,(R1)+
674 005006 005302 DEC R2
675 005010 001375 BNE 1$
676 ;
677 ; MOVE ALL OF MEMORY ROUTINE REFERENCE TABLE TO CURRENT JUMP
678 ; TABLE.
679 ;
680 005012 012700 000754 MOV #MTRF,R0 ; POINT TO REF TABLE
681 005016 012701 001240 MOV #MTSUB,R1 ; POINT TO CURRENT TABLE
682 005022 012702 000116 MOV #<MT*MEMS>,R2 ; NUMBER OF WORDS
683 005026 012021 2$: MOV (R0)+,(R1)+
684 005030 005302 DEC R2
685 005032 001375 BNE 2$
686 ;
687 ; JUMP TO LOOP SELECTION
688 ;
689 005034 052767 000140 172766 BIS #<MEMORY+ALLTST>,BASE ; SET FLAG FOR MEMORIES AND ALL TESTS
690 005042 000167 001006 JMP LPRMPT

```

```

692. ;
693. ;
694. ;
695. ; PROMPT FOR MEMORY SELECTIONS.
696. ; IF RESPONSE IS <CR>, SELECT ALL MEMORIES.
697. ; IF RESPONSE IS 'N', SKIP MEMORY TESTS.
698. ;
699. ;
700. 005046 ; MMSEL:
701. 005046 005067 173140 CLR MSEL ; CLEAR MEMORY SELECT FLAG.
702. 005052 CALL MEMSEL ; PROMPT FOR SELECTION.
703. 005056 CALL FIND ; FIND MEM MNEMONIC IN COMMAND LINE.
704. 005062 103004 BCC 1$ ; NON-BLANK WAS FOUND.
705. 005064 052767 000077 173120 BIS #<QX+SR+QR+OB+S1+S2>,MSEL.
706. 005072 000437 BR LIMM1 ; PROMPT FOR LIMITS.
707. ;
708. 005074 122711 000116 1$: CMPB #'N,(R1) ; NO MEMORY TESTS.
709. 005100 001004 BNE 2$ ; THERE ARE MEMORY TESTS.
710. 005102 005067 172722 CLR BASE
711. 005106 000167 177600 JMP ALL ; RESTART.
712. 005112 052767 000040 172710 2$: BIS #MEMORY,BASE ; FLAG MEMORY TESTS SELECTED.
713. ;
714. ;
715. ; LOOP TO PROCESS MEMORY MNEMONICS. THERE MAY BE MORE THAN
716. ; ONE IN THE COMMAND LINE. EG:
717. ; >MM QL FC Q2
718. ;
719. 005120 ; MEMTOP:
720. 005120 022700 000002 CMP #2,R0 ; CORRECT NUMBER OF CHARS.
721. 005124 001403 BEQ 2$ ; YES, CONTINUE.
722. 005126 CALL ERR3 ; WRITE ERROR MESSAGE.
723. 005132 000745 BR MMSEL ; AND START OVER.
724. ;
725. ; MATCH MEM MNEMONIC FROM THE COMMAND LINE AGAINST
726. ; A TABLE OF VALID MNEMONICS AND THEIR ASSOCIATED
727. ; MEMORY SELECT FLAG SETTINGS.
728. ;
729. 005134 012700 000014 2$: MOV #MTBLN,R0 ; LENGTH OF SCAN TABLE.
730. 005140 012702 000674 MOV #MTBL,R2 ; POINT TO SCAN TABLE.
731. 005144 CALL SCAN ; MATCH COMMAND LINE AGAINST TABLE.
732. 005150 103003 BCC 3$ ; MATCH WAS MADE.
733. 005152 CALL ERR3
734. 005156 000733 BR MMSEL ; START OVER.
735. ;
736. ; SUBROUTINE SCAN SETS R1 -> FLAG.
737. ; LOOK FOR THE NEXT MNEMONIC IN THE COMMAND LINE. IF
738. ; THERE IS ONE, PROCESS IT. ELSE FALL THROUGH TO PROMPT.
739. ; FOR MEMORY LIMITS.
740. ;
741. 005160 051167 173026 3$: BIS (R1),MSEL ; SET FLAG FOR MEMORY SELECTED.
742. 005164 CALL FIND ; ANYTHING ELSE IN COMMAND LINE
743. 005170 103353 BCC MEMTOP ; YES.

```



```

781 ;
782 ;
783 ; SELECT MEMORY TESTS.
784 ;
785 ;
786 ; PROMPT FOR TEST NUMBERS. IF THE RESPONSE IS <CR>
787 ; (CARRIAGE RETURN ONLY), MOVE A PSEUDO COMMAND LINE
788 ; INTO THE COMMAND LINE BUFFER. THIS PSEUDO LINE
789 ; CONSISTS OF THE TEST NUMBERS FOR ALL TESTS EXCEPT
790 ; TEST 10 (USER PATTERN). PROCEED TO PROCESS THIS
791 ; LINE AS THOUGH IT WAS ENTERED FROM THE TERMINAL.
792 ;
793 005300 MEMS:
794 005300 032767 000040 172522 BIT #MEMORY, BASE ; MEMORIES SELECTED.
795 005306 001002 BNE 10$ ; YES, PROMPT.
796 005310 000167 000506 JMP CHECK0 ; CHECK CURRENT JUMP TABLE.
797 ;
798 005314 10$: CALL SELMT ; PROMPT FOR MEMORY SELECTION.
799 005320 CALL FIND ; LOOK FOR TEST NUMBER IN COMMAND LINE.
800 005324 103024 BCC MTSL ; SOMETHING WAS THERE.
801 005326 052767 000100 172474 BIS #ALLTST, BASE ; SET FLAG FOR ALL TESTS (REPORT PASSES)
802 005334 012700 000644 MOV #ALLMEM, R0 ; POINT TO STRING OF ALL MEM TESTS.
803 005340 012701 000056 MOV #GCMBUF, R1 ; POINT TO COMMAND LINE BUFFER.
804 005344 012702 000027 MOV #STRMEM, R2 ; LENGTH OF STRING.
805 005350 112021 1$: MOVB (R0)+, (R1)+ ; MOVE STRING TO COMMAND BUFFER
806 005352 005302 DEC R2.
807 005354 001375 BNE 1$
808 005356 012767 000027 172614 MOV #STRMEM, GCMLN ; PRETEND LINE HAS BEEN READ IN FROM CONSOLE.
809 005364 012767 000056 172610 MOV #GCMBUF, GCMPT ; INIT COMMAND LINE POINTER.
810 005372 CALL FIND ; LOCATE FIRST TEST NUMBER IN PSEUDO LINE
811 ;
812 ; PROCESS ONE TEST NUMBER AT A TIME. FIRST VALIDATE THE
813 ; NUMBER.
814 ;
815 005376 MTSL:
816 005376 022700 000002 CMP #2, R0 ; TEST NUMBERS ARE 1 OR 2 BYTES
817 005402 002416 BLT MTERR ; NO GOOD (TOO MANY)
818 005404 003007 BGT 1$ ; SINGLE DIGIT, PROCESS IT.
819 005406 122721 000061 CMPB #1, (R1)+ ; TENS DIGIT?
820 005412 001012 BNE MTERR ; NO, ERROR
821 005414 111103 MOVB (R1), R3 ; LOAD ASCII TEST NUMBER.
822 005416 062703 000012 ADD #10, R3 ; ADD VALUE OF TENS DIGIT.
823 005422 000401 BR MTRT ; AND CONTINUE.
824 ;
825 ; TRANSLATE SINGLE DIGIT FROM ASCII DECIMAL INTO BINARY.
826 ; IF A ZERO VALUE IS RETURNED FROM THE TRANSLATION,
827 ; THE ASCII CHARACTER IS INVALID.
828 ;
829 005424 111103 1$: MOVB (R1), R3 ; LOAD ASCII TEST NUMBER.
830 005426 012704 000224 MTRT: MOV #TROCT, R4 ; POINT TO TRANSLATE TABLE.
831 005432 060304 ADD R3, R4 ; INDEX TO BINARY VALUE.
832 005434 111403 MOVB (R4), R3 ; LOAD BINARY VALUE.
833 005436 001012 BNE M2REL ; NON-ZERO VALUE, CONTINUE.
834 ;
835 ; INVALID TEST NUMBER. PUT OUT MESSAGE, CLEAR THE MEMORY.
836 ; CURRENT JUMP TABLE AND GO BACK TO PROMPT.
837 ;

```

```

838 005440
839 005444 012700 001240* MTERR: CALL ERR6 ;WRITE ERROR MESSAGE
840 005450 012701 000116 MOV #MTSUB,R0 ;POINT TO JUMP TABLE
841 005454 005020 MOV #<MT*NMEMS>,R1 ;LOAD NUMBER OF WORDS IN TABLE
842 005456 005301 1$: CLR (R0)+ ;RESET TABLE
843 005460 001375 DEC R1
844 005462 000706 BNE 1$ ;TRY AGAIN
845 ;
846 ; IF TEST NUMBER = 6 OR 10, SET A FLAG FOR
847 ; LATER ACTION.
848 ;
849 005464 MZREL:
850 005464 122703 000006 CMPB #6,R3 ;TEST 6
851 005470 001003 BNE 10$ ;NO, DO NOT SET FLAG
852 005472 052767 000004 172330 BIS #TEST6,BASE ;SET FLAG FOR TEST 6 SELECTED
853 005500 122703 000012 10$: CMPB #10,,R3 ;TEST 10
854 005504 001003 BNE 1$ ;NO, NO PROMPT LATER
855 005506 052767 000010 172314 BIS #TEST10,BASE ;SET FLAG FOR PROMPT
856 ;
857 ;
858 ; MAKE TEST NUMBER ZERO-RELATIVE, MULTIPLY THE ZERO-
859 ; RELATIVE TEST NUMBER BY THE NUMBER OF MEMORIES X 2.
860 ; TO GET A BYTE OFFSET INTO THE REFERENCE TABLE AND
861 ; CURRENT JUMP TABLE, FOR EXAMPLE, IF THE ASCII TEST NUMBER
862 ; WAS 2, THE ZERO-RELATIVE NUMBER IS 1, THIS NUMBER IS
863 ; MULTIPLIED BY 24 TO GET A BYTE OFFSET = 24.
864 ;
865 ;
866 ; ADD THE PRODUCT TO THE START ADDRESS OF THE CURRENT JUMP
867 ; TABLE AND PUT THE RESULT IN R0, ADD THE SAME PRODUCT TO
868 ; THE START ADDRESS OF THE REF TABLE AND PUT THE RESULT IN
869 ; R1, THE RESULTS ARE:
870 ;
871 ; CURRENT JUMP TABLE (ASSUMING TEST 2 SELECTED)
872 ; .WORD 0,0,0,0,0,0,0,0,0,0,0
873 ; .WORD 0,0,0,0,0,0,0,0,0,0,0
874 ;
875 ; REFERENCE TABLE
876 ; .WORD T1,T1,T1,T1,T1,T1,T1,T1,T1,T1,T1
877 ; .WORD T2,T2,T2,T2,T2,T2,T2,T2,T2,T2,T2
878 ;
879 ; R0 -> FIRST 0 IN THE SECOND LINE FOLLOWING 'CURRENT JUM TABLE'
880 ; R1 -> FIRST T2 IN THE REFERENCE TABLE
881 ;
882 005514 005303 1$: DEC R3 ;MAKE TEST NUMBER ZERO REL
883 005516 010301 MOV R3,R1 ;SYSTEM EXPECTS MULTIPLICAND IN R1
884 005520 012700 000014 MOV #<NMEMS*2>,R0 ;AND MULTIPLIER IN R0
885 005524 CALL #NUL ;GET OFFSET INTO TABLE OF WORDS
886 005530 010103 MOV R1,R3 ;LOAD PRODUCT INTO R3
887 005532 012700 001240* MOV #MTSUB,R0 ;R0 -> TOP OF MEM TABLE
888 005536 060300 ADD R3,R0 ;TEST ADDRESSES GO HERE
889 005540 012701 000754* MOV #MTREF,R1 ;R1 -> MEM TEST REF TABLE
890 005544 060301 ADD R3,R1 ;TEST ADDRESSES COME FROM HERE
891 ;
892 ;
893 ; DEPENDING UPON WHICH MEMORIES HAVE BEEN SELECTED FOR TESTING
894 ; MOVE ADDRESSES OF ROUTINES THAT GOVERN THE TESTS FROM THE
      REFERENCE TABLE TO THE CURRENT JUMP TABLE (R0) START TESTING

```



```

895 ; THE MEMORY FLAG AT POSITION 0 (MRP MICROPGM MEMORY).
896 ;
897 ;
898 ; PROCEEDING WITH THE ABOVE EXAMPLE ASSUMING IN ADDITION THAT
899 ; QEX WINDOW WAS THE MEMORY SELECTED, THIS ROUTINE WOULD FILL
900 ; THE MEMORY CURRENT JUMP TABLE IN THE FOLLOWING MANNER:
901 ;
902 ; .WORD 0.0.0.0.0.0.0.0.0.0
903 ; .WORD 0.0.T2.0.0.0.0.0.0.0.0
904 ;
904 005546 012702 000001 MOV #BIT0,R2 ;START WITH FIRST MEMORY.
905 005552 012703 000006 MOV #NMEMS,R3 ;LOOP COUNT = NUMBER OF MEMORIES.
906 005556 030267 172430 2$: BIT R2,MSEL ;WAS MEMORY SELECTED
907 005562 001401 BEQ 3$ ;NO, BUMP TO NEXT
908 005564 011110 MOV (R1),(R0) ;MOVE FROM REF TO JUMP
909 005566 022120 3$: CMP (R1)+,(R0)+ ;INCR POINTERS
910 005570 006302 ASL R2 ;SHIFT TO TEST NEXT BIT
911 005572 005303 DEC R3 ;FINISHED?
912 005574 001370 BNE 2$ ;NO, CONTINUE
913 ;
914 ; TRANSFERS BETWEEN REF TABLE AND CURRENT JUMP TABLE ARE
915 ; COMPLETE FOR ONE TEST NUMBER, NOW SCAN THE COMMAND LINE
916 ; FOR THE NEXT TEST NUMBER.
917 ;
918 005576 CALL FIND ;ANYTHING ELSE IN COMMAND LINE?
919 005602 103402 BCS MTPMT ;NO
920 005604 000167 177566 JMP MTSL ;YES, PROCESS IT

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

922.
923
924 ; FINISHED WITH COMMAND LINE FOR TESTS.
925 ; PROMPT FOR TEST PATTERN IF TEST 10 WAS SELECTED.
926
927 005610 032767 000010 172212 MTPMT: BIT: #TEST10.BASE: ;USER TEST PATTERN?
928 005616 001421 BEQ: T6CHK: ;NO, SKIP ALL THIS
929 005620 PMPT10: CALL: PMT10 ;PROMPT
930 005624 CALL: FIND ;FIND A NON-BLANK IN COMMAND LINE.
931 005630 103003 BCC: 1$ ;OK, CONTINUE.
932 005632 CALL: ERR60
933 005636 000770 BR: PMPT10 ;TRY AGAIN
934 005640 1$: CALL: PACK ;CONVERT TEST PATTERN TO BINARY.
935 005644 103003 BCC: 2$ ;CONVERSION OK.
936 005646 CALL: ERR60
937 005652 000762 BR: PMPT10
938 005654 016767 172142 172172 2$: MOV: BINWD,MT10 ;PUT PATTERN IN A SAFE PLACE.
939
940 ;
941 ;
942 ; CHECK FLAG TO SEE WHETHER TEST 6 WAS SELECTED. IF
943 ; IT WAS, SET UP A POINTER TO THE MEMORY CURRENT
944 ; JUMP TABLE ENTRIES FOR TEST 6.
945 ;
946 ;
947 005662 T6CHK:
948 005670 032767 000004 172140 BIT: #TEST6.BASE: ;WAS TEST 6 SELECTED
949 005672 001454 BEQ: CHECK0 ;NO, SKIP AROUND.
950 005674 012700 001334 CLR: R1 ;START MEMORY OFFSET = 0
951 005700 012702 000006 MOV: #MTSUB+<5*<NMEMS*2>>,R0 ;POINT TO TEST 6 ADDRESSES.
952. MOV: #NMEMS,R2 ;LOOP COUNT = NUMBER OF MEMORIES.
953 ;
954 ; FOR EACH MEMORY TO COME UNDER TEST 6, CHECK WHETHER
955 ; THE MEMORY LIMITS ARE COMPATIBLE WITH THE TEST. TEST 6
956 ; REQUIRES AT LEAST THREE MEMORY LOCATIONS IN ORDER TO
957 ; WORK CORRECTLY.
958 ;
959 ; *****
960 ;
961 ; HOW TEST 6 WORKS: RATIONALE BEHIND 3 LOCATION RULE.
962 ;
963 ; TEST 6 CLEARS MEMORY FROM THE LOWER TO THE UPPER LIMIT.
964 ; IT THEN WRITES ALL 1'S IN THE FIRST LOCATION AND EVERY
965 ; OTHER LOCATION TO THE UPPER LIMIT. IT THEN READS ZEROS
966 ; FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE 1'S. IT
967 ; THEN BUMPS THE LOWER LIMIT BY ONE MEMORY INCREMENT (VALUE
968 ; VARIES DEPENDING UPON THE MEMORY). IT CLEARS MEMORY UP TO
969 ; THE UPPER LIMIT. IT WRITES 1'S INTO THE NEW LOWER LIMIT
970 ; AND EVERY OTHER LOCATION TO THE UPPER LIMIT. IT READS
971 ; ZEROS FROM THE LOCATIONS INTO WHICH IT DID NOT WRITE
972 ; 1'S. THREE MEMORY LOCATIONS ARE THE MINIMUM ON WHICH
973 ; TEST 6 CAN WORK.
974 ;
975 ; WRITE 1'S INTO LOCATIONS 0 AND 2
976 ; READ ZEROS FROM LOCATION 1
977 ; FFFF
978 ; 000000

```

```

979      ;      FFFF...
980      ;
981      ;      BUMP LOWER LIMIT TO 1
982      ;      CLEAR LOCATIONS 1 AND 2
983      ;      WRITE 1'S INTO LOCATION 1
984      ;      READ ZEROS FROM LOCATION 2
985      ;      FFFF
986      ;      FFFF
987      ;      000000
988      ;
989      ; *****
990      ;
991      ;
992      ;      CHECK EACH NON-ZERO TEST 6 ENTRY IN THE CURRENT JUMP TABLE
993      ;
994 005704 005710 CHECK: TST      (R0)          ;TEST 6 SELECTED FOR THIS MEMORY
995 005706 001440 BEQ      2$          ;NO, SKIP CHECK
996 005710 010103 MOV      R1,R3      ;SHIFT IN ANOTHER REG
997 005712 006303 ASL      R3          ;
998 005714 006303 ASL      R3          ;SHIFT FOR DOUBLE WORD OFFSET
999 005716 016304 MOV      CURLIM+2(R3),R4 ;GET LOWER LIMIT
1000 005722 062704 000002 ADD      #2,R4        ;UPPER LIMIT MUST BE AT LEAST 2 GT LOWER
1001 005726 026304 001474* CMP      CURLIM(R3),R4 ;IS UPPER LIMIT OK FOR TEST 6
1002 005732 103026 BHS      2$          ;YES, CONTINUE
1003      ;
1004      ;      FAILED CHECK, CLEAR THE ENTRY IN THE CURRENT JUMP TABLE
1005      ;      AND REPORT TO THE CONSOLE, R1 = PLACEHOLDER VALUE, THIS
1006      ;      VALUE IS USED AS AN INDEX INTO A TABLE OF MEMORY NAMES
1007      ;      (AFTER MULTIPLYING THE VALUE BY 19, THE LENGTH OF EACH
1008      ;      NAME).
1009      ;
1010 005734 005010 CLR      (R0)        ;CLEAR TEST 6 ADDRESS FROM CURRENT TABLE
1011 005736 010046 MOV      R0,-(SP)    ;SAVE POINTER
1012 005740 010146 MOV      R1,-(SP)    ;SAVE OFFSET
1013 005742 CALL     ERR10       ;PRINT GENERAL ERROR MESSAGE
1014 005746 012700 000023 MOV      #MNAMSZ,R0  ;LENGTH OF MEMORY NAMES
1015 005752 CALL     #MUL        ;GET OFFSET INTO MEMORY NAME TABLE (R0XR1)
1016 005756 012700 001765* MOV      #MFTBL,R0  ;POINT TO MEMORY NAME TABLE
1017 005762 012701 000023 MOV      #19,R1      ;NUMBER OF CHARS IN NAME
1018 005766 012705 002151* MOV      #PRINT,R5   ;POINT TO PRINT LINE
1019 005772 112025 1$:  MOVB    (R0)+(R5)+ ;MOVE NAME TO PRINT LINE
1020 005774 005301 DEC      R1          ;
1021 005776 001375 BNE     1$          ;
1022 006000 CALL     CONSOL      ;WRITE MEMORY IN ERROR
1023 006004 012601 MOV      (SP)+,R1    ;
1024 006006 012600 MOV      (SP)+,R0    ;
1025      ;
1026      ;
1027      ;      PREPARE TO CHECK NEXT TEST 6 ENTRY
1028 006010 005201 2$:  INC      R1          ;BUMP MEMORY OFFSET
1029 006012 062700 000002 ADD      #2,R0        ;POINT TO NEXT TEST 6 ADDRESS
1030 006016 005302 DEC      R2          ;SUB FROM LOOP COUNT
1031 006020 001331 BNE     CHECK       ;
1032      ;
1033      ;      MAKE SURE THAT THERE IS AT LEAST ONE NON-ZERO ENTRY
1034      ;
1035 006022 CHECK0:

```

```
1036 006022 012700 001240'      MOV:      #MTSUB,R0      :POINT TO TOP OF TABLE.
1037 006026 012701 000116      MOV:      #<MT*NMEMS>,R1 :NUMBER OF TABLE ENTRIES.
1038 006032 005720              1$: TST:      (R0)+      :IS A TABLE ENTRY PRESENT.
1039 006034 001007              BNE:      LPRMPT.    :YES, EXIT THIS ROUTINE.
1040 006036 005301              DEC:      R1         :SUB FROM ROUTINE COUNT.
1041 006040 001374              BNE:      1$        :TRY NEXT POSITION.
1042 006042              CALL:     ERR9       :EXECUTION IMPOSSIBLE.
1043                               :
1044 006046              EXIT$S.
```

```

1046 ;
1047 ;
1048 ;
1049 ;
1050 ;
1051 ;
1052 ;
1053 ;
1054 ;
1055 ;
1056 ;
1057 006054 ;
1058 006054 ;
1059 006060 ;
1060 006064 103004 ;
1061 006066 052767 000001 171734 ;
1062 006074 000442 ;
1063 ;
1064 006076 122711 000131 1$: CMPB #?Y,(R1) ;YES - LOOP ON TESTS
1065 006102 001004 ; BNE 2$ ;TRY 'N'
1066 006104 052767 000001 171716 ; BIS #LOOP,BASE ;SET FLAG FOR LOOP
1067 006112 000433 ; BR ERPRMT ;PROMPT FOR ERROR OPTIONS
1068 006114 122711 000116 2$: CMPB #?N,(R1) ;NO - DO NOT LOOP ON TESTS
1069 006120 001004 ; BNE 3$ ;NO TEST FOR LOOP COUNT
1070 006122 042767 000001 171700 ; BIC #LOOP,BASE ;CLEAR LOOP FLAG
1071 006130 000424 ; BR ERPRMT
1072 ;
1073 ;
1074 ;
1075 ;
1076 006132 060100 3$: ADD R1,R0 ;POINT 1 PAST STRING
1077 006134 005200 ; INC R0 ;BUMP FOR STUPID SYSTEM SUBRTN
1078 006136 010046 ; MOV R0,-(SP) ;SAVE FOR LATER COMPARISON
1079 006140 010100 ; MOV R1,R0 ;MOVE POINTER TO R0 FOR SYSTEM SUBRTN
1080 006142 ; CALL $CDB ;CONVERT DECIMAL TO BINARY
1081 006146 020026 ; CMP R0,(SP)+ ;WHOLE STRING CONVERTED
1082 006150 001403 ; BEQ 4$ ;YES CONTINUE
1083 006152 ; CALL ERR7
1084 006156 000736 ; BR LPRMPT ;PROMPT AGAIN
1085 006160 010167 172024 4$: MOV R1,LOOPCT ;SAVE LOOP COUNT
1086 006164 001003 ; BNE 5$
1087 006166 ; CALL ERR7
1088 006172 000730 ; BR LPRMPT
1089 006174 052767 000001 171626 5$: BIS #LOOP,BASE ;SET LOOP FLAG

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103 006202.
1104 006202.
1105 006206
1106 006212. 103445
1107
1108 006214 122711 000110
1109 006220 001007
1110 006222. 052767 000200 171600
1111 006230 012767 000002 171562.
1112 006236 000433
1113
1114
1115
1116
1117 006240 060100
1118 006242. 005200
1119 006244 010046
1120 006246 010100
1121 006250
1122 006254 020026
1123 006256 001403
1124 006260
1125 006264 000746
1126
1127 006266 010167 171526
1128 006272. 005267 171522
1129 006276
1130 006302. 103411
1131 006304 122711 000110
1132 006310 001403
1133 006312.
1134 006316 000731
1135
1136 006320 052767 000200 171502. 4$:

```

PROMPT FOR ERROR OPTIONS.

RESPONSES:

```

<CR> - CARRIAGE RETURN; DISPLAY ALL ERRORS.
H. - HALT. STOP TESTS AFTER 1ST ERROR.
NUMERIC VALUE - PRINT ONLY THIS NUMBER OF ERROR MESSAGES.
BUT CONTINUE TESTS.
N: VALUE + H - PRINT THIS NUMBER OF ERROR MESSAGES AND HALT.

```

ERPRMT:

```

CALL EROPT. ;PROMPT FOR OPTIONS.
CALL FIND ;FIND RESPONSE.
BCS MTSET ;DEFAULT TO 'DISPLAY'

```

```

CMPB #'H,(R1) ;HALT AFTER 1 ERROR.
BNE 1$ ;NO.
BIS #HALT,BASE ;SET FLAG FOR HALT.
MOV #2,ERLIM ;PRINT ONLY 1 ERROR MESSAGE.
BR MTSET.

```

ASSUME THAT THERE IS AN ASCII DECIMAL VALUE IN THE
COMMAND LINE. CONVERT IT TO BINARY AND STORE.

```

1$: ADD R1,R0 ;POINT PAST STRING IN COMMAND LINE.
INC R0 ;BUMP FOR STUPID SYSTEM SUBRTH
MOV R0, -(SP) ;SAVE FOR LATER COMPARISON.
MOV R1,R0 ;PREPARE TO CONVERT.
CALL $CDTB.
CMP R0, (SP)+ ;ENTIRE STRING CONVERTED.
BEQ 2$ ;YES.
CALL ERR8
BR ERPRMT. ;TRY AGAIN

```

```

2$: MOV R1,ERLIM ;NUMBER OF ERROR MSGS TO PRINT
INC ERLIM ;ADJUST FOR PRE-DECREMENT.
CALL FIND ;SCAN COMMAND LINE.
BCS MTSET ;NOTHING ELSE THERE.
CMPB #'H,(R1) ;HALT AFTER MESSAGE COUNT EXHAUSTED.
BEQ 4$ ;YES.
CALL ERR8 ;BAD OPTION.
BR ERPRMT. ;TRY AGAIN

```

```

4$: BIS #HALT,BASE ;SET HALT FLAG.

```

```

1138 ;
1139 ;
1140 ; PREPARE TO ENTER MAIN LOOP.
1141 ;
1142 ;
1143 ; SET UP POINTER TO CURRENT JUMP TABLE.
1144 ; SET UP COUNT OF NUMBER OF ENTRIES IN TABLE.
1145 ;
1146 ;
1147 ; IF MEMORY TESTS ARE SELECTED AND NO REGISTER
1148 ; TESTS ARE SELECTED, SET POINTER -> TOP OF THE MEMORY TEST
1149 ; CURRENT JUMP TABLE AND COUNT = NUMBER OF ENTRIES IN THE
1150 ; TABLE.
1151 ;
1152 006326 MTSET:
1153 006326 012767 001240' 171660 MOV. #MTSUB, MTPNT. ; POINT AT MEMORY TEST RTNS.
1154 006334 012767 000116 171654 MOV. #<MT*NMEMS>, MTCNT. ; NUMBER OF MEM RTNS.
1155 ;
1156 ; ADJUST FOR PRE-INCREMENT OF POINTER AND PRE-DECREMENT
1157 ; OF COUNT AT 'MTMAIN', SAVE THE POINTER AND COUNT IN
1158 ; THEIR INITIAL STATES FOR REFRESH ON REPEATED CYCLES.
1159 ;
1160 006342 005267 171650 2$: INC. MTCNT. ; ADJUST COUNT FOR PRE-DECREMENT.
1161 006346 162767 000002 171640 SUB. #2, MTPNT. ; ADJUST POINTER FOR PRE-INCREMENT.
1162 006354 016767 171636 171640 MOV. MTCNT, NXTCNT. ; SAVE FOR FUTURE PASSES.
1163 006362 016767 171626 171630 MOV. MTPNT, NXTPNT. ; SAME.
1164 ;
1165 ; THE STOP MESSAGE SAYS 'ENTER ANY CHARACTER TO STOP TEST(S)'.
1166 ; ACTUALLY THE CHARACTERS W, P, C, AND T ARE SPECIAL IN THAT
1167 ; THEY ALLOW THE RUN TO CONTINUE AFTER THE PRINTING OUT OF
1168 ; STATUS INFORMATION. SEE THE ROUTINE 'AST'.
1169 ;
1170 006370 CALL. BEGTST. ; PUT OUT START TEST MESSAGE.
1171 006374 CALL. STOP. ; GIVE DIRECTIONS FOR STOPPING TEST.
1172 006400 QIO$S. #IO,ATA, #LUN,TT, , , , , <#AST>

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

1174
1175
1176          ;      MAIN LOOP.
1177          ;
1178          ;
1179          ;
1180          ;      CHECK FOR THE HALT OPTION. IF THE OPTION IS
1181          ;      ON, CHECK TO SEE WHETHER ANY ERRORS HAVE OCCURRED.
1182          ;      IF THEY HAVE (AND THE HALT COUNT IS EXHAUSTED),
1183          ;      TERMINATE THE TESTS.
1184          ;
1185          ;      MTMAIN:
1185 006442 032767 000200 171360 BIT      #HALT,BASE      ;HALT AFTER ERROR.
1186 006450 001413          BEQ      1$           ;NO.
1187 006452 032767 000400 171350 BIT      #ERROR,BASE   ;HAS AN ERROR OCCURRED.
1188 006460 001407          BEQ      1$           ;NO.
1189 006462 026727 171332 000001 CMP      ERLIM,#1      ;IS REMAINING PRINTOUT COUNT 1 OR LESS.
1190 006470 003003          BGT      1$           ;NO, CONTINUE.
1191 006472          CALL    HLTST.       ;PUT OUT HALT MESSAGE.
1192 006476 000464          BR       8$           ;AND EXIT.
1193          ;
1194          ;
1195          ;      FIND A NON-ZERO ENTRY IN THE 'CURRENT JUMP TABLE'.
1196          ;      (A NON-ZERO ENTRY IS THE ADDRESS OF A TEST CONTROL
1197          ;      ROUTINE). IF NO NON-ZERO ENTRIES ARE FOUND BEFORE
1198          ;      THE END OF THE TABLE IS REACHED, THEN ONE MEMORY TEST
1199          ;      CYCLE OR 'PASS' IS COMPLETE.
1200 006500 016701 171510 1$:      MOV      MTPNT,R1      ;POINT TO JUMP TABLE
1201 006504 005367 171506 2$:      DEC      MTCNT.      ;FIRST SUB FROM # RTNS LEFT.
1202 006510 001406          BEQ      3$           ;ALL DONE, TEST LOOP FLAG.
1203 006512 005721          TST      (R1)+        ;ADVANCE POINTER.
1204 006514 005711          TST      (R1)         ;IS THERE AN ADDRESS IN THE TABLE.
1205 006516 001772          BEQ      2$           ;NO, BUMP TO NEXT.
1206 006520 010167 171470 MOV      R1,MTPNT.     ;SAVE JUMP TABLE POINTER.
1207 006524 000457          BR       JMPMT.      ;AND JUMP TO ROUTINE
1208          ;
1209          ;
1210          ;
1211          ;      PASS FINISHED.
1212          ;
1213          ;
1214          ;
1215          ;      IF THE LOOP FLAG IS NOT ON, EXIT QMT.
1216          ;      IF THE LOOP FLAG IS ON AND THE LOOP COUNT IS
1217          ;      EXHAUSTED, EXIT QMT.
1217 006526 032767 000001 171274 3$:    BIT      #LOOP,BASE   ;IS LOOP FLAG ON.
1218 006534 001443          BEQ      7$           ;NO, GET OUT.
1219 006536 005767 171446 TST      LOOPCT.      ;IS LOOP COUNT BEING USED.
1220 006542 001403          BEQ      4$           ;NO, JUST KEEP LOOPING.
1221 006544 005367 171440 DEC      LOOPCT.      ;SUB FROM LOOP COUNT
1222 006550 001435          BEQ      7$           ;FINISHED.
1223          ;
1224          ;
1225          ;      IF ALL TESTS, PRINT 'END OF PASS NNNN'.
1226          ;      REINITIALIZE FOR NEXT PASS.
1227 006552 032767 000100 171250 4$:    BIT      #ALLTST,BASE ;ALL TESTS
1228 006560 001415          BEQ      6$           ;NO, GET OUT.
1229 006562 012702 004010 MOV      #ENDOF,R2.   ;POINT TO MESSAGE.
1230 006566 012703 000014 MOV      #ENDLN,R3   ;LENGTH OF MESSAGE

```



```

1231 006572 012705 002151'
1232 006576 112225
1233 006600 005303          5$:  MOV.  #PRINT,R5          ;POINT TO PRINT LINE
1234 006602 001375          MOV.  (R2)+,(R5)+      ;MOVE MESSAGE TO PRINT LINE
1235 006604          DEC.  R3
1236 006610          BNE.  5$
1237          CALL.  PASSC          ;ADD # PASSES TO PRINT LINE
1238 006614 062767 000001 171364 6$:  CALL.  CONSOL          ;PRINT MESSAGE
1239 006622 005567 171356          ADD.  #1,PASS          ;COUNT NUMBER OF PASSES
1240 006626 016767 171366 171360          ADC.  PASSH          ;CARRY TO HIGH WORD
1241 006634 016767 171362 171354          MOV.  NXTPNT,MTPTNT   ;RESET POINTER
1242 006642 000677          MOV.  NXTCNT,MTCHT   ;RESET COUNT
1243          BR.  MTMAIN          ;AND ENTER LOOP
1244          ;
1245          ;
1246 006644          ;
1247 006650          7$:  CALL.  ENDTST          ;MEMORY TEST(S) ENDED
1248 006658 016737 171132 000274 8$:  MOV.  OLDVEC,@#274    ;RESTORE OLD VECTOR CONTENTS
1249 006656          EXIT$S          ;AND LEAVE
1250          ;
1251          ;
1252          ;
1253          ;
1254          ;
1255          ;
1256          ;
1257          ;
1258          ;
1259          ;
1260          ;
1261          ;
1262          ;
1263          ;
1264 006664          ;
1265          ;
1266 006664          ;
1267 006664 012700 000116          ;
1268 006670 016701 171322          ;
1269          ;
1270          ;
1271          ;
1272          ;
1273          ;
1274          ;
1275          ;
1276          ;
1277          ;
1278          ;
1279          ;
1280 006674 160100          ;
1281 006676 012701 000006          ;
1282 006702          ;
1283          ;
1284          ;
1285          ;
1286          ;
1287          ;

```

IF THE CURRENT JUMP TABLE POINTER -> REGISTER CURRENT JUMP TABLE, SET UP TO FIND OUT THE REGISTER PLACEHOLDER VALUE.
 LOAD R0 = NUMBER OF REGISTER TESTS. LOAD R1 = CURRENT JUMP TABLE COUNT. (IE. THE NUMBER OF CURRENT JUMP TABLE ENTRIES, INCLUDING THE CURRENT NON-ZERO ENTRY, REMAINING TO BE PROCESSED BY MTMAIN).
 JMPMT:
 MEMOFF: MOV. #<MT*NMEMS>,R0 ;TOTAL NUMBER OF MEMORY ROUTINES.
 MOV. MTCNT,R1 ;LOAD CURRENT COUNT.
 SUBTRACT THE CURRENT COUNT FROM THE TOTAL NUMBER OF ENTRIES IN THE CURRENT JUMP TABLE (COMBINED OR MEM ONLY) TO GET POSITION RELATIVE TO THE TOP OF THE MEMORY TABLE OF THE CURRENT NON-ZERO ENTRY.
 DIVIDE BY THE NUMBER OF MEMORIES. THE QUOTIENT WILL BE THE MEMORY TEST NUMBER (IN ZERO-RELATIVE FORM). THE REMAINDER WILL BE THE PLACEHOLDER VALUE FOR THE MEMORY TO BE TESTED.
 100\$: SUB. R1,R0 ;GET CURRENT POSITION.
 MOV. #NMEMS,R1 ;NUMBER OF MEMORIES.
 CALL. \$DIV
 USE THE REMAINDER FROM THE ABOVE DIVISION (X4) AS AN OFFSET INTO THE DOUBLE-WORD MEMORY LIMITS TABLE. MOVE THE 'CURRENT' MEMORY LIMITS TO THE STACK.
 SET THE MEMORY SELECT CODE FOR APPROPRIATE MEMORY.

```

1288 ;
1289 ; STORE THE REMAINDER (X2 FOR WORD OFFSET) INTO R0 FOR
1290 ; USE BY THE MEMORY TEST CONTROL ROUTINE.
1291 ;
1292 006706 006301 LSTACK: ASL R1 ; SHIFT FOR WORD OFFSET
1293 006710 010100 MOV R1,R0 ; SAVE WORD OFFSET FOR INDEXING
1294 006712 006301 ASL R1 ; SHIFT FOR DOUBLE WORD OFFSET
1295 006714 016067 000724' 171110 MOV CDLOW(R0),CODE ; SET INITIAL MEMORY SELECT CODE
1296 006722 016146 001474' MOV CURLIM(R1),-(SP) ; MOVE UPPER LIMIT TO STACK
1297 006726 016146 001476' MOV CURLIM+2(R1),-(SP) ; MOVE LOWER LIMIT TO STACK
1298 ;
1299 006732 016701 171256 LASTJ: MOV MTPNT,R1 ; POINT TO TEST CONTROL ROUTINE
1300 006736 000171 000000 JMP @(R1) ; GO THERE

```

```

1302. ;
1303. ;
1304. ; MEMORY TEST CONTROL ROUTINES
1305. ;
1306. ; TESTS ARE FULLY DESCRIBED IN THE QMT MEMORY TEST SUB-MODULES
1307. ;
1308. ;
1309. ;
1310. ; TEST 01
1311. 006742. T1:
1312. 006742. CALL. @T1ADDR(R0)
1313. 006746. 000167. 000514. JMP. MTJUMP
1314. ;
1315. ;
1316. ; TEST 02
1317. ;
1318. 006752. T2:
1319. 006752. 005067. 172672. CLR. CKDATA. ; TEST PATTERN = 0
1320. 006756. CALL. @STADDR(R0)
1321. 006762. 000167. 000500. JMP. MTJUMP
1322. ;
1323. ;
1324. ; TEST 03
1325. ;
1326. 006766. T3:
1327. 006766. 012767. 177777. 172654. MOV. #-1,CKDATA. ; TEST PATTERN = -1
1328. 006774. CALL. @STADDR(R0)
1329. 007000. 000167. 000462. JMP. MTJUMP
1330. ;
1331. ;
1332. ; TEST 04
1333. ;
1334. 007004. T4:
1335. 007004. 012767. 125252. 172636. MOV. #125252,CKDATA. ; TEST PATTERN = 'X'AAAA'
1336. 007012. CALL. @STADDR(R0)
1337. 007016. 000167. 000444. JMP. MTJUMP
1338. ;
1339. ;
1340. ; TEST 05
1341. ;
1342. 007022. T5:
1343. 007022. 012767. 146314. 172620. MOV. #146314,CKDATA. ; TEST PATTERN = 'X'CCCC'
1344. 007030. CALL. @STADDR(R0)
1345. 007034. 012767. 031463. 172606. MOV. #031463,CKDATA. ; TEST PATTERN = '3333'
1346. 007042. CALL. @STADDR(R0)
1347. 007046. 000167. 000414. JMP. MTJUMP
1348. ;
1349. ;
1350. ; TEST 06
1351. ;
1352. 007052. T6:
1353. 007052. 005067. 172572. CLR. CKDATA. ; SET TEST PATTERN TO ZERO
1354. 007056. CALL. @STADDR(R0)
1355. 007062. CALL. @T6ADDR(R0) ; PERFORM CROSS-TALK TEST
1356. 007066. 005216. INC. (SP) ; PERFORM AT NEXT ADDRESS
1357. 007070. 005067. 172554. CLR. CKDATA. ; RESET TEST PATTERN
1358. 007074. CALL. @STADDR(R0)

```

```

1359 007100          CALL   @T6ADDR(R0)
1360 007104 005316    DEC    (SP)                ;RESTORE ORIGINAL LOWER LIMITS
1361 007106 000167 000354 JMP    MTJUMP
1362                ;
1363                ;
1364                ;      TEST-07
1365                ;
1366 007112          T7:
1367 007112          CALL   @T7ADDR(R0)
1368 007116 000167 000344 JMP    MTJUMP
1369                ;
1370                ;
1371                ;      TEST-08
1372                ;
1373 007122          T8:
1374 007122 012767 000377 172520 MOV    #377,CKDATA        ;SET TEST PATTERN TO X'00FF'
1375 007130          CALL   @STADDR(R0)
1376 007134 012767 177400 172506 MOV    #177400,CKDATA    ;SET TEST PATTERN TO X'FF00'
1377 007142          CALL   @STADDR(R0)
1378 007146 000167 000314 JMP    MTJUMP
1379                ;
1380                ;
1381                ;      TEST-09
1382                ;
1383 007152          T9:
1384 007152 012767 000001 172470 MOV    #1,CKDATA        ;START TEST PATTERN AT 1
1385 007160          1#: CALL   @STADDR(R0)
1386 007164 016702 172460 MOV    CKDATA,R2        ;LOAD FOR SHIFT
1387 007170 006302 ASL    R2                ;SHIFT A BIT
1388 007172 010267 172452 MOV    R2,CKDATA        ;NEXT TEST PATTERN
1389 007176 005702 TST   R2                ;FINISHED (SHIFTED TO ZERO)
1390 007200 001367 BNE   1#                ;NO
1391 007202 000167 000260 JMP    MTJUMP
1392                ;
1393                ;
1394                ;      TEST-0A
1395                ;
1396 007206          TA:
1397 007206 016767 170642 172434 MOV    MT10,CKDATA      ;USER PATTERN
1398 007214          CALL   @STADDR(R0)
1399 007220 000167 000242 JMP    MTJUMP
1400                ;
1401                ;
1402                ;      TEST-0B
1403                ;
1404 007224          TB:
1405 007224 012767 100001 172416 MOV    #100001,CKDATA   ;=B'1000000000000001'
1406 007232          CALL   @STADDR(R0)
1407 007236 012767 040002 172404 MOV    #040002,CKDATA   ;=B'010000000000010'
1408 007244          CALL   @STADDR(R0)
1409 007250 012767 020004 172372 MOV    #020004,CKDATA   ;=B'001000000000100'
1410 007256          CALL   @STADDR(R0)
1411 007262 012767 010010 172360 MOV    #010010,CKDATA   ;=B'0001000000001000'
1412 007270          CALL   @STADDR(R0)
1413 007274 012767 004020 172346 MOV    #004020,CKDATA   ;=B'0000100000010000'
1414 007302          CALL   @STADDR(R0)
1415 007306 012767 002040 172334 MOV    #002040,CKDATA   ;=B'0000010000100000'
    
```

```

1416 007314          CALL @STADDR(R0)
1417 007320 012767 001100 172322 MOV #001100,CKDATA ;=B'0000001001000000'
1418 007326          CALL @STADDR(R0)
1419 007332 012767 000600 172310 MOV #000600,CKDATA ;=B'0000000110000000'
1420 007340          CALL @STADDR(R0)
1421 007344 000167 000116          JMP MTJUMP
1422          ;
1423          ;
1424          ; TEST 0C
1425          ;
1426 007350          TD:
1427 007350 005067 172274          CLR CKDATA ;SET TEST PATTERN = 0
1428 007354          CALL @STADDR(R0)
1429 007360 005067 172266          CLR CK2 ;SET READ PATTERN = 0
1430 007364 012767 177777 172262 MOV #-1,CK3 ;SET WRITE PATTERN = X'FFFF'
1431 007372          CALL @TCDADD(R0) ;READ X'0000', WRITE X'FFFF'
1432 007376 012767 177777 172246 MOV #-1,CK2 ;SET READ PATTERN = X'FFFF'
1433 007404 005067 172244          CLR CK3 ;SET WRITE PATTERN = 0
1434 007410          CALL @TCDADD(R0) ;READ X'FFFF', WRITE X'0000'
1435          ;
1436 007414 005067 172232          CLR CK2 ;SET READ PATTERN = 0
1437 007420 012767 177777 172226 MOV #-1,CK3 ;SET WRITE PATTERN = X'FFFF'
1438 007426          CALL @TCUADD(R0) ;READ/WRITE FROM BOTTOM UP
1439 007432 012767 177777 172212 MOV #-1,CK2 ;SET READ PATTERN = X'FFFF'
1440 007440 005067 172210          CLR CK3 ;CLEAR READ PATTERN
1441 007444          CALL @TCUADD(R0)
1442 007450 000167 000012          JMP MTJUMP
1443          ;
1444          ;
1445          ; TEST 0D
1446          ;
1447 007454          TD:
1448 007454          CALL @TDADDR(R0)
1449 007460 000167 000002          JMP MTJUMP
1450          ;
1451 007464          TDNUL:
1452 007464          RETURN
1453          ;
1454          ;
1455 007466          MTJUMP:
1456 007466 026760 170340 000740 CMP CODE,CDHIGH(R0) ;ALL WORDS OF MULTI-WORD MEMORY PROCESSED
1457 007474 001404          BEQ 10$ ;BRANCH IF SO
1458 007476 005267 170330          INC CODE ;NEXT ARRAY OF WORDS
1459 007502 000171 000000          JMP @ (R1) ;JUMP TO PROCESSING ROUTINE
1460 007506          10$:
1461 007506 062706 000004          ADD #4,SP
1462 007512 000167 176724          JMP MTMAIN

```

```

1464
1465
1466
1467
1468
1469
1470
1471
1472
1473 007516
1474 007516
1475
1476 007532 016705 170246
1477 007536 012700 000003
1478 007542
1479 007546 050011
1480 007550
1481
1482 007554
1483 007570 000002

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

:
: INTERRUPT SERVICE ROUTINE
: TRAP INTERRUPTS FROM HQR THROUGH VECTOR ADDRESS 274
: SET EVENT FLAG 3
: MICROCODE (MRP AND CP) DEBUGGING ROUTINES WILL READ CSR #2 AND
: DECODE THE INTERRUPT
:
:
: BPTISR:
: SAVE R0,R1,R2,R3,R4,R5
:
: MOV TSKTCB,R5 :LOAD MY TCB
: MOV #EFN,3,R0 :EVENT FLAG TO BE SET
: CALL #CEFI
: BIS R0,(R1) :SET LOCAL FLAG
: CALL $DRDSE :DECLARE SIGNIFICANT EVENT
:
: RESTOR R0,R1,R2,R3,R4,R5
: RTI

```

```

1485 ;
1486 ;
1487 ; ROUTINE TO PLACE A VALUE INTO CONTROL/STATUS REGISTER
1488 ; NUMBER 1.
1489 ;
1490 ; READ THE CURRENT VALUE OF THE CSR INTO A WORK AREA,
1491 ; CLEAR THE BITS AT 4(SP), SET THE BITS AT 2(SP),
1492 ; REWRITE CSR1 FROM THE WORK AREA.
1493 ;
1494 ; INPUT:
1495 ; 2(SP) BITS TO BE SET IN CSR1
1496 ; 4(SP) BITS TO BE CLEARED IN CSR1
1497 ;
1498 ;
1499 007572 CSR1::
1500 007572 016767 176420 170234 MOV QR$CR1, APLACE ; GET THE CURRENT VALUE
1501 007600 046667 000004 170226 BIC 4(SP), APLACE ; CLEAR FIRST
1502 007606 056667 000002 170220 BIS 2(SP), APLACE ; THEN SET
1503 007614 016767 170214 176420 MOV APLACE, QR$CR1 ; NOW RETURN IT
1504 007622 011666 000004 MOV (SP), 4(SP) ; MOVE RETURN ADDR TO TOP OF STACK
1505 007626 022626 CMP (SP)+, (SP)+ ; BUMP STACK POINTER PAST ARGS
1506 007630 RETURN ; SPLIT

```

```

1508 ;
1509 ;
1510 ; MEMORY TEST ERROR ROUTINE
1511 ;
1512 007632 MEMERR:
1513 007632 SAVE R0,R1,R2,R3,R4,R5
1514 ;
1515 ; SET FLAG FOR ERROR ENCOUNTERED, AN ERROR MESSAGE LIMIT
1516 ; COUNT OF ZERO MEANS THAT THE COUNT IS NOT BEING USED.
1517 ; A COUNT OF -1 MEANS THAT THE LIMIT HAS BEEN REACHED
1518 ; (NO MORE ERROR MESSAGES ARE TO BE PRINTED).
1519 ;
1520 007646 052767 000400 170154 BIS #ERROR,BASE ;SET FLAG FOR ERROR ENCOUNTERED
1521 007654 005767 170140 TST ERLIM ;IS ERROR COUNT BEING USED
1522 007660 001412 BEQ 2$
1523 007662 003002 BGT 1$
1524 007664 000167 000330 JMP MEMX
1525 ;
1526 ; DECREMENT ERROR LIMIT COUNT, IF IT GOES ZERO HERE, SET
1527 ; IT TO -1.
1528 ;
1529 007670 005367 170124 1$: DEC ERLIM
1530 007674 001004 BNE 2$
1531 007676 005367 170116 DEC ERLIM ;SET ERROR LIMIT FIELD TO -1
1532 007702 000167 000312 JMP MEMX
1533 ;
1534 ; PRINT TEST NUMBER
1535 ;
1536 007706 012705 002151 2$: MOV #PRINT,R5 ;POINT AT PRINT LINE
1537 007712 012700 001711 MOV #THSG,R0 ;POINT AT TEST
1538 007716 012701 000005 MOV #5,R1 ;NUMBER OF BYTES IN STRING
1539 007722 112025 3$: MOVB (R0)+,(R5)+ ;MOVE LABEL TO PRINT LINE
1540 007724 005301 DEC R1
1541 007726 001375 BNE 3$
1542 ;
1543 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
1544 ; MEMORY TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
1545 ; TABLE.
1546 ;
1547 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
1548 ; BY THE NUMBER OF MEMORIES IN THE TABLE, THE QUOTIENT
1549 ; IS THE TEST NUMBER (ZERO RELATIVE), THE REMAINDER IS THE
1550 ; MEMORY PLACEHOLDER VALUE, SAVE IT.
1551 ;
1552 007730 012700 000116 MOV #<MT*NMEMS>,R0 ;TOTAL NUMBER OF ROUTINES
1553 007734 016701 170256 MOV MTCNT,R1 ;LOAD CURRENT COUNT
1554 007740 160100 4$: SUB R1,R0 ;GET CURRENT POSITION
1555 007742 012701 000006 MOV #NMEMS,R1 ;NUMBER OF MEMORIES
1556 007746 CALL #DIV ;DIVIDE FOR TEST NUMBER
1557 007752 010146 MOV R1,-(SP) ;SAVE REMAINDER
1558 007754 010001 MOV R0,R1 ;PREPARE FOR CONVERSION
1559 007756 005201 INC R1 ;ADJUST ZERO RELATIVE NUMBER
1560 007760 012700 001704 MOV #ASWRK,R0 ;POINT TO CONVERSION WORK AREA
1561 007764 012702 000001 MOV #1,R2 ;KEEP ZEROS
1562 007770 CALL #CBDSG ;CONVERT TEST NUMBER TO ASCII
1563 ;
1564 007774 116725 171707 MOVB ASWRK+3,(R5)+ ;MOVE A DIGIT OF TEST NUMBER

```



```

1565 010000 116725 171704      MOV.  ASWRK+4, (R5)+
1566 010004 062705 000002      ADD.  #2, R5                ;BUMP PRINT LINE POINTER.
1567                               ;
1568                               ;
1569                               ;
1570 010010 012700 001716'     MOV.  #PMMSG, R0           ;POINT AT 'PASS.'
1571 010014 012701 000005     MOV.  #5, R1              ;NUMBER OF BYTES IN STRING.
1572 010020 112025 5#:      MOV.  (R0)+, (R5)+       ;MOVE LABEL TO PRINT LINE.
1573 010022 005301            DEC.  R1
1574 010024 001375            BNE.  5$
1575 010026                    CALL.  PASSC.             ;CONVERT NUMBER OF PASSES.
1576 010032 062705 000003     ADD.  #3, R5                ;ADVANCE PRINT LINE POINTER.
1577                               ;
1578 010036                    CALL.  MEMNAM.            ;MOVE MEMORY NAME TO OUTPUT BUFFER.
1579 010042                    CALL.  CONSOL.           ;WRITE TO CONSOLE.
1580                               ;
1581                               ;
1582                               ;
1583                               ;
1584 010046 012705 002151'     MOV.  #PRINT, R5          ;POINT TO PRINT LINE
1585 010052 012700 001723'     MOV.  #AMSG, R0          ;POINT TO 'ADDRESS.'
1586 010056 012701 000011     MOV.  #9, R1              ;LOAD LENGTH OF STRING.
1587 010062 112025 7#:      MOV.  (R0)+, (R5)+       ;MOVE LABEL.
1588 010064 005301            DEC.  R1
1589 010066 001375            BNE.  7$
1590                               ;
1591 010070 016701 171564     MOV.  ERRADD, R1          ;LOAD ERROR ADDRESS.
1592 010074                    CALL.  UNPK              ;CONVERT TO PRINTABLE CHARS.
1593 010100 005205            INC.  R5                    ;BUMP PRINT LINE POINTER.
1594                               ;
1595 010102 012700 001734'     MOV.  #EMSG, R0          ;POINT TO 'EXPECTED.'
1596 010106 012701 000012     MOV.  #10, R1            ;LOAD NUMBER OF CHARS.
1597 010112 112025 8#:      MOV.  (R0)+, (R5)+       ;MOVE LABEL.
1598 010114 005301            DEC.  R1
1599 010116 001375            BNE.  8$
1600                               ;
1601 010120 016701 171524     MOV.  CKDATA, R1         ;LOAD MEMORY TEST PATTERN.
1602 010124                    CALL.  UNPK              ;CONVERT.
1603 010130 005205            INC.  R5                    ;BUMP PRINT LINE POINTER.
1604                               ;
1605 010132 012700 001746'     MOV.  #RMSG, R0          ;POINT TO 'RECEIVED.'
1606 010136 012701 000012     MOV.  #10, R1            ;LOAD NUMBER OF CHARS.
1607 010142 112025 9#:      MOV.  (R0)+, (R5)+       ;MOVE LABEL.
1608 010144 005301            DEC.  R1
1609 010146 001375            BNE.  9$
1610                               ;
1611                               ;
1612                               ;
1613                               ;
1614                               ;
1615 010150 016700 171506     MOV.  ERRCT, R0          ;LOAD NUMBER OF WORDS TO CONVERT.
1616 010154 003010            BGT.  11$                ;PRINT WORDS.
1617 010156 012700 001760'     MOV.  #UNMSG, R0        ;***
1618 010162 012701 000005     MOV.  #5, R1              ;LENGTH OF MESSAGE.
1619 010166 112025 10#:     MOV.  (R0)+, (R5)+
1620 010170 005301            DEC.  R1
1621 010172 001375            BNE.  10$

```

```
1622 010174 000407          BR      13$
1623                          ;
1624 010176 012702 001664'  11$:  MOV.  #ERW1,R2.      ;POINT TO FIRST OF THEM
1625 010202 012201          12$:  MOV.  (R2)+,R1      ;LOAD THW WORD ITSELF
1626 010204          CALL.  UNPK
1627 010210 005300          DEC.  R0
1628 010212 001373          BNE.  12$
1629 010214          13$:  CALL.  CONSOL.      ;ELSE WRITE TO CONSOLE
1630                          ;
1631 010220          MEMX:
1632 010220          RESTOR. R0,R1,R2,R3,R4,R5
1633 010234          RETURN.
```

```

1635 ;
1636 ;
1637 ; AST:
1638 ;
1639 ;
1640 ;
1641 ; INTERCEPT CHARACTER FROM CONSOLE
1642 ;
1643 ; ON 'W' PRINT TEST NUMBER, PASS COUNT, MEMORY NAME, ADDRESS,
1644 ; AND TEST PATTERN (MEMORIES ONLY) OR TEST NUMBER, PASS COUNT,
1645 ; REGISTER NAME, AND TEST PATTERN (REGISTERS ONLY)
1646 ; ON 'C' PRINT TEST PATTERN ('CKDATA')
1647 ; ON 'P' PRINT NUMBER OF PASSES.
1648 ; ON 'T' PRINT TEST NUMBER.
1649 ; EXIT ON ANY OTHER CHARACTER.
1650 ;
1651 ;
1652 010236 ; AST:
1653 010236 012667 167546 MOV. (SP)+,ASTWRD. ;GET CHAR
1654 010242 SAVE. R0,R1,R2,R3,R4,R5
1655 ;
1656 010256 122767 000120 167524 CMPB. #'P,ASTWRD. ;PRINT NUMBER OF PASSES AND CONTINUE
1657 010264 001002 BNE. 1$ ;NO. NEXT OPTION
1658 010266 000167 000334 JMP. PAST
1659 010272 122767 000103 167510 1$: CMPB. #'C,ASTWRD. ;PRINT OUT TEST PATTERN
1660 010300 001002 BNE. 2$ ;NO. NEXT OPTION
1661 010302 000167 000160 JMP. CAST
1662 010306 122767 000124 167474 2$: CMPB. #'T,ASTWRD. ;PRINT OUT TEST NUMBER
1663 010314 001002 BNE. 3$ ;NO. NEXT OPTION
1664 010316 000167 000166 JMP. TAST
1665 010322 122767 000127 167460 3$: CMPB. #'W,ASTWRD. ;PRINT WHERE
1666 010330 001402 BEQ. WAST
1667 010332 000167 000254 JMP. EAST ;PRINT NUMBER OF PASSES AND EXIT
1668 ;
1669 ;
1670 ; ROUTINE FOR 'W'
1671 ;
1672 010336 ; WAST:
1673 ;
1674 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
1675 ; MEMORY TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
1676 ; TABLE.
1677 ;
1678 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
1679 ; BY THE NUMBER OF MEMORIES IN THE TABLE, THE QUOTIENT
1680 ; IS THE TEST NUMBER (ZERO-RELATIVE), THE REMAINDER IS THE
1681 ; MEMORY PLACE-HOLDER VALUE, SAVE IT.
1682 010336 012700 000116 MOV. #<MT*NMEMS>,R0 ;TOTAL NUMBER OF ROUTINES
1683 010342 016701 167650 MOV. MTCNT,R1 ;LOAD CURRENT COUNT
1684 010346 160100 SUB. R1,R0 ;GET CURRENT POSITION
1685 010350 012701 000006 MOV. #NMEMS,R1 ;NUMBER OF MEMORIES
1686 010354 CALL. $DIV ;DIVIDE FOR TEST NUMBER
1687 010360 010146 MOV. R1,-(SP) ;SAVE REMAINDER
1688 010362 010001 MOV. R0,R1 ;PREPARE FOR CONVERSION
1689 010366 012700 001704' INC. R1 ;ADJUST ZERO-RELATIVE NUMBER
1690 010372 012702 000001 MOV. #ASURK,R0 ;POINT TO CONVERSION WORK AREA
1691 010376 MOV. #1,R2 ;KEEP ZEROS
CALL. $CBDSG ;CONVERT TEST NUMBER TO ASCII

```

```

1692. ;
1693 ; TEST NUMBER.
1694 ;
1695 010402 012705 002151' MOV #PRINT,R5 ;POINT TO PRINT LINE
1696 010406 116725 171275 MOVB ASURK+3,(R5)+ ;MOVE A DIGIT OF TEST NUMBER
1697 010412 116725 171272 MOVB ASURK+4,(R5)+
1698 010416 005205 INC R5 ;BUMP PRINT LINE POINTER
1699 ;
1700 ; PASS
1701 ;
1702 010420 CALL PASSC ; CONVERT NUMBER OF PASSES
1703 010424 005205 INC R5
1704 ;
1705 010426 CALL MEMNAM ; MOVE MEMORY NAME TO OUTPUT BUFFER
1706 010432 005205 INC R5
1707 ;
1708 ; ADDRESS
1709 ;
1710 010434 016701 171216 MOV PREADD,R1 ;LOAD ERROR ADDRESS
1711 010440 CALL UNPK ; CONVERT TO PRINTABLE CHARS
1712 010444 005205 INC R5 ;BUMP PRINT LINE POINTER
1713 ;
1714 ; TEST PATTERN
1715 ;
1716 010446 016701 171176 MOV CKDATA,R1 ;LOAD TEST PATTERN
1717 010452 CALL UNPK ; CONVERT
1718 010456 CALL CONSOL ; WRITE LINE TO CONSOL
1719 010462 000167 000212 JMP REST ;EXIT AST
1720 ;
1721 ;
1722 ;
1723 ; PRINT TEST PATTERN ALONE
1724 ;
1725 ;
1726 010466 CAST:
1727 010466 012705 002151' MOV #PRINT,R5
1728 010472 016701 171152 MOV CKDATA,R1 ;LOAD TEST PATTERN
1729 010476 CALL UNPK ; CONVERT TO ASCII
1730 010502 CALL CONSOL ; WRITE TO TT0
1731 010506 000474 BR REST
1732 ;
1733 ;
1734 ;
1735 ; PRINT OUT TEST NUMBER ALONE
1736 ;
1737 ;
1738 010510 TAST:
1739 ;
1740 ; DERIVE TEST NUMBER FROM THE POSITION OF THE CURRENT
1741 ; MEMORY TEST CONTROL ROUTINE ADDRESS IN THE CURRENT JUMP
1742 ; TABLE
1743 ;
1744 ; GET THE OFFSET FROM THE BEGINNING OF THE TABLE, DIVIDE
1745 ; BY THE NUMBER OF MEMORIES IN THE TABLE, THE QUOTIENT
1746 ; IS THE TEST NUMBER (ZERO-RELATIVE)
1747 ;
1748 010510 012700 000116 MOV #((MNT*MEMS)/20) ;TOTAL NUMBER OF ROUTINES

```

```

1749 010514 016701 167476          MOV.   MTCNT,R1          ;LOAD CURRENT COUNT.
1750 010520 160100          3$:   SUB.   R1,R0          ;GET CURRENT POSITION.
1751 010522 012701 000000          MOV.   #NMEMS,R1       ;NUMBER OF MEMORIES.
1752.
1753 010526          4$:   CALL.  $DIV          ;DIVIDE FOR TEST NUMBER.
1754 010532 010001          MOV.   R0,R1          ;PREPARE FOR CONVERSION.
1755 010534 005201          INC.   R1              ;ADJUST ZERO RELATIVE NUMBER.
1756 010536 012700 001704*        MOV.   #ASWRK,R0       ;POINT TO CONVERSION WORK AREA.
1757 010542 012702 000001          MOV.   #1,R2          ;KEEP ZEROS.
1758 010546          CALL.  $CBDSG         ;CONVERT TEST NUMBER TO ASCII.
1759.
1760 010552 012705 002151*        MOV.   #PRINT,R5       ;POINT TO PRINT LINE.
1761 010556 012700 001711*        MOV.   #TMSG,R0        ;POINT TO TEST.
1762 010562 012701 000005          MOV.   #5,R1          ;NUMBER OF CHARS IN STRING.
1763 010566 112025          MOV.   (R0)+,(R5)+     ;MOVE STRING TO PRINT LINE.
1764 010570 005301          5$:   DEC.   R1
1765 010572 001375          BNE.   5$
1766.
1767 010574 116725 171107          MOV.   ASWRK+3,(R5)+   ;MOVE A DIGIT OF TEST NUMBER.
1768 010600 116725 171104          MOV.   ASWRK+4,(R5)+
1769 010604          CALL.  CONSOL         ;PRINT TEST NUMBER.
1770 010610 000433          BR     REST           ;EXIT AST.
1771.
1772.
1773.
1774.
1775.
1776.
1777.
1778.
1779.
1780 010612.          EAST:
1781 010612.          CALL.  HLTST          ;WRITE EXIT MESSAGE.
1782 010616 032767 000100 167204    BIT.   #ALLTST,BASE    ;ALL TESTS
1783 010624 001025          BNE.   REST           ;YES, PASS NUMBER ALREADY PRINTED.
1784.
1785.
1786.
1787.
1788.
1789 010626          PRINT:
1790 010626 012700 002151*        MOV.   #PRINT,R0       ;POINT TO PRINT LINE.
1791 010632 012701 000116          MOV.   #78.,R1        ;NUMBER OF CHAR POSITIONS.
1792 010636 112720 000040          1$:   MOV.   #40,(R0)+     ;CLEAR THE LINE.
1793 010642 005301          DEC.   R1
1794 010644 001374          BNE.   1$
1795.
1796 010646 012705 002151*        MOV.   #PRINT,R5       ;POINT TO PRINT LINE.
1797 010652 012700 003774*        MOV.   #PMSG2,R0       ;POINT TO NUMBER OF PASSES.
1798 010656 012701 000014          MOV.   #PM2LN,R1      ;LENGTH OF STRING.
1799 010662 112025          2$:   MOV.   (R0)+,(R5)+     ;MOVE STRING TO PRINT LINE.
1800 010664 005301          DEC.   R1
1801 010666 001375          BNE.   2$
1802.
1803 010670          CALL.  PASSC          ;CONVERT NUMBER OF PASSES.
1804 010674          CALL.  CONSOL         ;WRITE MESSAGE.
1805.

```

```

1806
1807 010700          ; REST:  RESTOR:  R0,R1,R2,R3,R4,R5
1808
1809 010714  122767  000120  167066          CMPB:  #'P,ASTWRD:  ;PRINT: #, PASSES AND CONTINUE.
1810 010722  001414          BEQ:    10$
1811 010724  122767  000103  167056          CMPB:  #'C,ASTWRD:  ;PRINT: TEST PATTERN AND CONTINUE.
1812 010732  001410          BEQ:    10$
1813 010734  122767  000124  167046          CMPB:  #'T,ASTWRD:  ;PRINT: TEST NUMBER.
1814 010742  001404          BEQ:    10$
1815 010744  122767  000127  167036          CMPB:  #'W,ASTWRD:  ;PRINT: WHERE AND CONTINUE.
1816 010752  001003          BNE:    1$
1817
1818 010754          ; 10$:  ASTX$S.
1819
1820 010762  012701  002151  ; 1$:  MOV:    #'PRINT,R1          ;POINT TO PRINT LINE
1821 010766  012700  000116          MOV:    #'78.,R0          ;NUMBER OF CHARS.
1822 010772  112721  000040          2$:  MOVB   #'40,(R1)+        ;CLEAR PRINT LINE.
1823 010776  005300          DEC:    R0
1824 011000  001374          BNE:    2$
1825 011002  112767  000015  171141          MOVB:  #'15,PRINT:      ;WRITE OUT ONE CR+LF
1826 011010  112767  000012  171134          MOVB:  #'12,PRINT+1
1827 011016          CALL:  CONSOL.
1828
1829 011022  016737  166760  000274          MOV:    OLDVEC,@#274    ;RESTORE ORIGINAL VECTOR CONTENTS.
1830 011030          EXIT$S.

```

```

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. 011036      ;
1874. 011036      ;
1875. 011042.      ;
1876. 011046 103472. ;
1877. 011050      ;
1878. 011054 103003 ;
1879. 011056      ;
1880. 011062. 000755 ;
1881.      ;
1882.      ;
1883.      ;
1884. 011064 026766 166732 000002. 1$:
1885. 011072. 103003 ;
1886. 011074      ;
1887. 011100 000756 ;
1888.      ;
SUBRTN FOR MEMORY LIMITS.

INPUT:
2(SP) ABSOLUTE LOWER LIMITS.
4(SP) ABSOLUTE UPPER LIMITS.
6(SP) ADDRESS OF PROMPT ROUTINE.

OUTPUT:
(SP) CURRENT WORKING LOWER LIMITS.
2(SP) CURRENT WORKING UPPER LIMITS.

WORK FIELDS USED:
LOWER.
UPPER.

PROMPT FOR LIMITS. A <CR> RESPONSE MEANS TO TAKE
THE ABSOLUTE LOWER AND UPPER LIMITS AND RETURN
THEM ON THE STACK. OTHERWISE IF THE RESPONSE IS IN
THE FORM:
>0000 000A.

THIS ROUTINE CONVERTS THE FIRST VALUE AND COMPARES IT
AGAINST THE MEMORY'S ABSOLUTE LOWER LIMITS AT 2(SP). IF
THE NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A
TEMPORARY WORK FIELD. THE ROUTINE THEN CHECKS THE COMMAND
LINE FOR THE UPPER LIMITS, CONVERTS THEM, AND COMPARES
THEM AGAINST THE MEMORY'S ABSOLUTE UPPER LIMITS. IF THE
NEW LIMITS ARE IN RANGE, THEY ARE PLACED IN A TEMPORARY
WORK AREA. THE ROUTINE THEN COMPARES THE NEW UPPER LIMITS
WITH THE NEW LOWER LIMITS. IF THE NEW UPPER LIMITS ARE
EQUAL TO OR GREATER THAN THE NEW LOWER LIMITS, BOTH NEW
VALUES ARE PLACED ON THE STACK. THE ROUTINE THAT CALLED
LIMITS WILL TAKE THESE VALUES OFF THE STACK AND PLACE
THEM IN THE "CURRENT LIMITS TABLE". DURING THE MEMORY
TEST CYCLE, THE LIMITS FROM THIS TABLE ARE MADE AVAILABLE
TO THE MEMORY TEST ROUTINES.

LIMITS:
CALL @6(SP) ;PROMPT FOR LIMITS.
CALL FIND ;FIRST FIND A NUMBER
BCS LIMX2. ;NO OVERRIDES, LEAVE LIMITS ALONE.
CALL PACK ;CONVERT LOWER LIMIT TO BINARY
BCC 1$ ;VALUE OK, CONTINUE.
CALL ERR4
BR LIMITS. ;TRY AGAIN

CHECK LOWER LIMITS.

CMP BINWD,2(SP) ;COMPARE LOWER LIMITS.
BHIS 2$ ;OK, CONTINUE.
CALL ERR4
BR LIMITS. ;TRY AGAIN
    
```

```

1889 ; ALSO CHECK NEW LOWER LIMITS AGAINST ABSOLUTE UPPER
1890 ; LIMITS ON STACK. AN ERROR HERE WOULD SHOW UP BELOW
1891 ; BUT IT IS MORE CORRECT TO REPORT AN ERROR IN LOWER
1892 ; LIMITS IF THE NEW LOWER LIMITS ARE HIGHER THAN THE
1893 ; ABSOLUTE UPPER LIMITS.
1894 ;
1895 011102 026766 166714 000004 2$: CMP BINWD,4(SP) ; IS NEW LOW LIMIT GT UPPER LIMIT
1896 011110 101403 BLOS 20$ ; NO. NEW LOWER LIMIT IS OK
1897 011112 CALL ERR4
1898 011116 000747 BR LIMITS
1899 ;
1900 ; SAVE NEW LOWER LIMITS. FIND NEW UPPER LIMITS IN COMMAND
1901 ; LINE.
1902 ;
1903 011120 016767 166676 166676 20$: MOV BINWD,LOWER ; MOVE IN NEW LOW LIMITS
1904 011126 CALL FIND ; FIND UPPER LIMITS IN COMMAND LINE
1905 011132 103003 BCC 3$ ; OK, CONTINUE
1906 011134 CALL ERR5
1907 011140 000736 BR LIMITS ; START OVER
1908 ;
1909 011142 3$: CALL PACK ; CONVERT UPPER LIMITS
1910 011146 103003 BCC 4$ ; OK, CONTINUE
1911 011150 CALL ERR5
1912 011154 000730 BR LIMITS ; START OVER
1913 ;
1914 ; CHECK NEW UPPER LIMITS.
1915 ;
1916 011156 026766 166640 000004 4$: CMP BINWD,4(SP) ; COMPARE UPPER LIMITS
1917 011164 101403 BLOS 5$ ; OK, CONTINUE
1918 011166 CALL ERR5
1919 011172 000721 BR LIMITS ; START OVER
1920 ;
1921 ; COMPARE NEW LOWER LIMITS WITH NEW UPPER LIMITS.
1922 ;
1923 011174 016767 166622 166624 5$: MOV BINWD,UPPER ; PLACE NEW UPPER LIMIT ON STACK
1924 011202 026767 166616 166616 CMP LOWER,UPPER ; IS UPPER LIMIT GT LOWER LIMIT
1925 011210 101403 BLOS LIMX ; YES, EXIT
1926 011212 CALL ERR5
1927 011216 000707 BR LIMITS ; TRY AGAIN
1928 ;
1929 011220 016766 166600 000002 LIMX: MOV LOWER,2(SP) ; PUT NEW LOWER LIMITS ON STACK
1930 011226 016766 166574 000004 MOV UPPER,4(SP) ; PUT NEW UPPER LIMITS ON STACK
1931 011234 LIMX2: RETURN

```



```

1933 ;
1934 ;
1935 ; SCAN: A TABLE FOR A VALID COMMAND/MNEMONIC.
1936 ;
1937 ; INPUT:
1938 ; R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
1939 ; R1 -> CHAR STRING IN GCML COMMAND LINE.
1940 ; R2 -> TOP OF COMMAND TABLE.
1941 ;
1942 ; OUTPUT:
1943 ; R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
1944 ; R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
1945 ; R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
1946 ;
1947 ;
1948 011236 SCAN:
1949 011236 010346 MOV R3, -(SP) ;SAVE R3
1950 011240 010046 MOV R0, -(SP) ;SAVE # ENTRIES.
1951 011242 010146 MOV R1, -(SP) ;SAVE POINTER TO BEGINNING OF STRING.
1952 ;
1953 011244 011601 FNOUT1: MOV (SP), R1 ;POINT TO NON-BLANK IN COMMAND LINE.
1954 011246 012703 000002 MOV #2, R3 ;NUMBER OF CHARS IN NON-BLANK FIELD.
1955 011252 122122 FNIN1: CMPB (R1)+, (R2)+ ;DOES COMMAND LINE MATCH TABLE ENTRY?
1956 011254 001003 BNE FNOUT2 ;NO, TRY NEXT TABLE ENTRY.
1957 011256 005303 DEC R3 ;SUB FROM LOOP COUNT
1958 011260 001374 BNE FNIN1
1959 011262 000410 BR FNMTCH ;COMMAND FOUND IN TABLE.
1960 011264 000302 FNOUT2: ADD R2, R2 ;ADD # UNCOMPARED CHARS TO POINTER.
1961 011266 005202 INC R2 ;THEN ADJUST TO NEXT TABLE ENTRY.
1962 011270 005300 DEC R0 ;SUB FROM OUTER LOOP COUNT.
1963 011272 001364 BNE FNOUT1 ;TRY AGAIN
1964 011274 012601 MOV (SP)+, R1 ;RESTORE POINTER TO COMMAND LINE.
1965 ; MOV (SP)+, R0 ;RESTORE R0
1966 011276 012603 MOV (SP)+, R3 ;RESTORE R3
1967 011300 000261 SEC ;COMMAND NOT IN TABLE.
1968 011302 RETURN.
1969 ;
1970 011304 010201 FNMTCH: MOV R2, R1 ;POINT R1 AT RTN ADDR IN TABLE
1971 011306 062706 000002 ADD #2, SP ;POINT TO INCOMING R0 ON STACK
1972 011312 012602 MOV (SP)+, R2 ;GET TOTAL # TABLE ENTRIES.
1973 011314 160002 SUB R0, R2 ;GET POSITION OF MATCHED ENTRY
1974 011316 010200 MOV R2, R0 ;PUT IN R0 FOR RETURN
1975 011320 012603 MOV (SP)+, R3 ;RESTORE R3
1976 011322 000241 CLC
1977 011324 RETURN.

```



```

2034 ; CONVERT AN ASCII HEX VALUE FROM THE COMMAND LINE INTO BINARY.
2035 ; LEGAL STRINGS CONTAIN FROM 1 TO 4 CHARACTERS.
2036 ;
2037 ; INPUT:
2038 ; R0 = NUMBER OF CHARACTERS IN ASCII STRING.
2039 ; R1 -> STRING
2040 ;
2041 ; OUTPUT:
2042 ; THE FIELD 'BINWD' CONTAINS THE CONVERTED VALUE.
2043 ;
2044 ;
2045 011444 ; PACK:
2046 011444 005067 166352 CLR BINWD ;CLEAR DESTINATION.
2047 011450 005046 CLR -(SP) ;CLEAR FOR COND CODE INDICATOR
2048 011452 SAVE R0,R1,R2,R3,R4,R5
2049 ;
2050 ; DETERMINE THE CONVERSION FACTOR (POWER OF 16) FOR
2051 ; THE LEFTMOST ASCII CHARACTER.
2052 ;
2053 011466 022700 000004 CMP #4,R0 ;UPPER LIMIT FOR HEX DIGITS.
2054 011472 002455 BLT PSEXC ;ERROR EXIT.
2055 011474 010002 MOV R0,R2 ;NUMBER OF CHARS CONTROLS LOOP
2056 011476 022700 000004 CMP #4,R0 ;4 CHARS?
2057 011502 001003 BNE 1$
2058 011504 012700 010000 MOV #4096,,R0 ;HEX CONVERSION FACTOR FOR HIGH ORDER CHAR.
2059 011510 000416 BR 4$ ;ENTER LOOP.
2060 011512 022700 000003 1$: CMP #3,R0 ;3 CHARS?
2061 011516 001003 BNE 2$
2062 011520 012700 000400 MOV #256,,R0 ;CONVERSION FACTOR FOR HIGH ORDER CHAR.
2063 011524 000410 BR 4$ ;ENTER LOOP.
2064 011526 022700 000002 2$: CMP #2,R0 ;2 CHARS?
2065 011532 001003 BNE 3$
2066 011534 012700 000020 MOV #16,,R0 ;CONVERSION FACTOR FOR HIGH ORDER CHAR.
2067 011540 000402 BR 4$
2068 011542 012700 000001 3$: MOV #1,R0 ;1 CHAR.
2069 ;
2070 ; MULTIPLY EACH CHARACTER'S VALUE BY ITS CONVERSION
2071 ; FACTOR. THE CONVERSION FACTOR IS REDUCED BY A POWER
2072 ; OF 16 AS THE ASCII STRING IS SCANNED FROM LEFT TO RIGHT.
2073 ;
2074 011546 010105 4$: MOV R1,R5 ;MOVE INPUT POINTER TO R5
2075 011550 112503 HLOOP: MOVB (R5)+,R3 ;GET ASCII VALUE INTO A REG.
2076 011552 012704 000424' MOVB #TRTBL,R4 ;POINT TO TRANSLATE TABLE.
2077 011556 060304 ADD R3,R4 ;ADD VALUE OF CHARACTER.
2078 011560 111401 MOVB (R4),R1 ;MOVE BINARY VALUE TO A REG.
2079 011562 022704 000504' CMP #TRTBL+60,R4 ;WAS INPUT CHAR ZERO
2080 011566 001402 BEQ 1$ ;YES, THIS IS OK.
2081 011570 105701 TSTB R1 ;WAS TABLE POSITION EMPTY.
2082 011572 001415 BEQ PSEXC ;YES, TRANSLATION UNSUCCESSFUL
2083 011574 010046 1$: MOVB (R4),R1 ;NSAVE FOR DIVISION LATER.
2084 011576 CALL $MUL ;MULT BY 16 TO SOME POWER.
2085 011602 060167 166214 ADD R1,BINWD ;ACCUUM CONVERTED VALUE.
2086 011606 012600 MOVB (SP)+,R0 ;RELOAD FACTOR.
2087 011610 012701 000020 MOVB #16,,R1 ;LOAD DIVISOR.
2088 011614 CALL $DIV ;REDUCE FACTOR.
2089 011620 005302 DEC R2 ;SUB FROM LOOP COUNT.
2090 011622 001352 BNE HLOOP

```

```
2091 011624 000403          BR      PCLCX          ;AND·EXIT·
2092          ;
2093 011626 012766 177777 000014 PSECK: MOV      #-1,12.(SP)      ;INDICATE·ERROR·
2094 011634          PCLCX: RESTOR R0,R1,R2,R3,R4,R5
2095          ;
2096 011650 005726          TST      <SP>+          ;TEST·CC·INDICATOR·
2097 011652 002402          BLT      1$            ;ERROR·
2098 011654 000241          CLC
2099 011656 000401          BR      PACKX
2100 011660 000261          1$: SEC
2101 011662          PACKX: RETURN
```

```

2103 ;
2104 ;
2105 ; READ A RECORD (BLOCK)
2106 ;
2107 ; FILE NAME BLOCK PRE-INITIALIZED
2108 ;
2109 ; OUTPUT:
2110 ; C-BIT CLEAR - GOOD READ
2111 ; C-BIT SET - ERROR ON READ
2112 ;
2113 ;
2114 011664 GET:
2115 011664 READ$ #INFDB,..#VIRT.#EFN.1.#STAT
2116 011732 103005 BCC 1$
2117 011734 CALL ERR5
2118 011740 CALL ERNAME ; TELL WHICH FILE WAS IN ERROR
2119 011744 000421 BR GETSX
2120 ;
2121 011746 1$: WTSE$ #EFN.1
2122 ;
2123 011760 CLEF$ #EFN.1
2124 011772 105767 166014 TSTB STAT
2125 011776 003006 BGT GETCX ; GOOD COMPLETION
2126 012000 CALL ERR5
2127 012004 CALL ERNAME ; TELL WHICH FILE WAS IN ERROR
2128 ;
2129 012010 000261 GETSX: SEC
2130 012012 000403 BR GETX
2131 012014 005267 166022 GETCX: INC VIRT+2 ; INC BLOCK COUNTER
2132 012020 000241 CLC
2133 012022 GETX: RETURN

```

```

2135
2136
2137 012024
2138 012024
2139
2140 012030 012700 002151'
2141 012034 016701 172440
2142 012040
2143 012044 016701 172432
2144 012050
2145
2146 012054
2147 012060
2148 012064
2149
2150
2151
2152
2153 012066
2154 012066
2155 012072 016601 000006
2156 012076 012700 000023
2157 012102
2158 012106 012700 001765'
2159 012112 060100
2160 012114 012701 000023
2161 012120 112025
2162 012122 005301
2163 012124 001375
2164
2165
2166
2167 012126 016601 000006
2168 012132 006301
2169 012134 026161 000740' 000724'
2170 012142 001424
2171 012144 016700 165662
2172 012150 166100 000724'
2173 012154 162705 000003
2174 012160 112725 000127
2175 012164 112725 000117
2176 012170 112725 000122
2177 012174 112725 000104
2178 012200 112725 000050
2179 012204 116025 000625'
2180 012210 112725 000051
2181 012214
2182 012214
2183 012220 012616
2184 012222

```

ERNAME:

 SAVE R0,R1

 MOV #PRINT,R0 ;R0 -> PRINT-LINE

 MOV INDNB+N,FNAM,R1 ;R1 = RAD50 WORD

 CALL \$CSTA ;CONVERT FIRST WORD

 MOV INDNB+N,FNAM+2,R1 ;SECOND WORD

 CALL \$CSTA

 CALL CONSOL

 RESTOR R0,R1

 RETURN

 MOVE MEMORY NAME TO OUTPUT BUFFER

 MEMNAM:

 SAVE R0,R1

 MOV 6(SP),R1 ;GET REMAINDER

 MOV #19,R0 ;LENGTH OF MESSAGE STRING IN TABLE

 CALL \$MUL

 MOV #MFTBL,R0 ;POINT TO TABLE OF MESSAGES

 ADD R1,R0 ;POINT TO REG THAT FAILED

 MOV #19,R1 ;NUMBER OF CHARS IN NAME

 10\$: MOVB (R0)+(R5)+ ;MOVE NAME TO PRINT-LINE

 DEC R1

 BNE 10\$

 APPEND SUBSCRIPT IF MULTIWORD MEMORY

 MOV 6(SP),R1 ;LOAD REMAINDER

 ASL R1 ;GET TABLE OFFSET

 CMP CDHIGH(R1),CDLOW(R1) ;GET HIGH MEMORY SELECT CODE

 BEQ 20\$;BRANCH IF SINGLE WORD MEMORY

 MOV CODE,R0 ;GET MEMORY SELECT CODE

 SUB CDLOW(R1),R0 ;SUBTRACT BASE SELECT CODE

 SUB #3,R5 ;BACKSPACE POINTER

 MOVB #'W,(R5)+

 MOVB #'O,(R5)+

 MOVB #'R,(R5)+

 MOVB #'D,(R5)+

 MOVB #'C,(R5)+ ;MOVE LEFT PARENTHESIS TO BUFFER

 MOVB TRTBL2+1(R0),(R5)+ ;MOVE SUBSCRIPT DIGIT

 MOVB #'',(R5)+ ;MOVE RIGHT PARENTHESIS

 20\$: RESTOR R0,R1

 MOV (SP)+,@SP ;GET RETURN ADDRESS AND FIX STACK

 RETURN

```

2186 ;
2187 ;
2188 ;
2189 ; CONVERT A VALUE FROM BINARY TO PRINTABLE FORM.
2190 ; R1 = WORD TO BE CONVERTED.
2191 ; R5 -> PRINT LINE.
2192 ;
2193 UNPK:
2194   SAVE  R0,R1,R2
2195 ;
2196   ADD  #4,R5 ; DO LAST CHAR FIRST.
2197   MOV  #4,R2 ; NUMBER OF HEX DIGITS FOR A WORD
2198   MOV  R1,R0 ; SUBRTN EXPECTS DIVIDEND IN R0
2199   MOV  #16,R1 ; LOAD DIVIDOR
2200   CALL $DIV
2201   MOV  #TRTBL2,R3 ; POINT TO TRANSLATE TABLE
2202   ADD  R1,R3 ; ADD 4 BIT VALUE
2203   MOVB (R3),-(R5) ; MOVE CHAR TO PRINT LINE
2204   DEC  R2 ; DEC INNER LOOP COUNT
2205   BNE  1$
2206   ADD  #5,R5 ; BUMP PRINT LINE POINTER
2207 ;
2208 UNPKX:
2209   RESTOR R0,R1,R2
2210   RETURN

```


2269 012456 012604
2270 012460 012603
2271 012462
2272
2273
2274 012464

MOV. (SP)+.R4
MOV. (SP)+.R3
PASSX: RETURN
:
:
ASCNST: .BLKW 5

:RETURN:

:CONVERSION STORAGE AREA:

2276
 2277
 2278
 2279
 2280
 2281 012476
 2282 012476 012700 000120
 2283 012502 012701 002267
 2284 012506 122741 000040
 2285 012512 001003
 2286 012514 005300
 2287 012516 001373
 2288 012520 000436
 2289
 2290 012522
 2291
 2292
 2293 012600 012701 002151
 2294 012604 112721 000040
 2295 012610 005300
 2296 012612 001374
 2297 012614
 2298
 2299 012616

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

:
: WRITE A PRINT LINE TO TT0
:
:
CONSOL:
MOV #0,R0 ;PRINT BUFFER BYTE COUNT
MOV #PRINT+78,R1 ;POINT PAST END OF BUFFER
1$: CMPB #40,-(R1) ;LOOK FOR A NON-BLANK
BNE 2$ ;OK WRITE LINE
DEC R0 ;DEC CHAR COUNT
BNE 1$
BR ABEND2 ;NO NON-BLANKS?
2$: QIOW$ #IO,WVB,#LUN,TT,#EFN,1, #STAT, <#PRINT-2,R0>,ABEND2
:
:
MOV #PRINT,R1 ;POINT TO STRING
4$: MOVB #40,(R1)+ ;CLEAR LINE TO BLANKS
DEC R0 ;DEC LOOP COUNT
BNE 4$
RETURN
:
ABEND2: ABRT$ #MYSELF
  
```

```

2301      ;
2302      ;
2303      ;      WRITE TO TT0 AND PROMPT.
2304      ;
2305      ;
2306 012650 005267 165142  ENDTST: INC.  ERWORD.      : 'TEST(S) ENDED'
2307 012654 005267 165136  HLTST: INC.  ERWORD.      : 'TEST(S) HALTED'
2308 012660 005267 165132  STOP: INC.  ERWORD.      : 'ENTER ANY CHARACTER TO STOP TEST(S)'
2309 012664 005267 165126  BEGTST: INC. ERWORD.      : 'TEST(S) STARTED'
2310 012670 005267 165122  OUT1: INC.  ERWORD.      : 'SUBDOCUMENT PROCESSOR DIAGNOSTICS'
2311 012674 005267 165116  ERR10: INC. ERWORD.      : 'TEST 6 INCOMPATABLE WITH MEMORY LIMITS.'
2312 012700 005267 165112  ERR9: INC.  ERWORD.      : 'ERROR: NO SELECTIONS. EXIT.'
2313 012704 005267 165106  ERR00: INC. ERWORD.      : 'ERROR ON READ'
2314 012710 005267 165102  ERR0: INC.  ERWORD.      : 'INVALID ERROR OPTION'
2315 012714 005267 165076  ERR7: INC.  ERWORD.      : 'INVALID LOOP OPTION'
2316 012720 005267 165072  ERR60: INC. ERWORD.      : 'INVALID TEST PATTERN'
2317 012724 005267 165066  ERR6: INC.  ERWORD.      : 'INVALID TEST NUMBER'
2318 012730 005267 165062  ERR50: INC. ERWORD.      : 'ILLEGAL ODD ADDRESS'
2319 012734 005267 165056  ERR5: INC.  ERWORD.      : 'INVALID UPPER MEMORY LIMITS'
2320 012740 005267 165052  ERR4: INC.  ERWORD.      : 'INVALID LOWER MEMORY LIMITS'
2321 012744 005267 165046  ERR3: INC.  ERWORD.      : 'INVALID MEMORY MNEMONIC'
2322 012750 005267 165042  ERR2: INC.  ERWORD.      : 'INVALID RESPONSE'
2323      ;
2324      ;      ALL THE FOLLOWING INCLUDE A PROMPT.
2325      ;
2326 012754  NESTOP:
2327 012754 005267 165036  ERDPT: INC.  ERWORD.      : 'ENTER ERROR CONTROL'
2328 012760 005267 165032  LPTST: INC.  ERWORD.      : 'LOOP ON TEST(S)?'
2329 012764 005267 165026  PMT10: INC. ERWORD.      : 'ENTER PATTERN FOR TEST 10'
2330 012770 005267 165022  PMT3: INC.  ERWORD.      : 'ENTER PATTERN FOR TEST 3'
2331 012774 005267 165016  SELMT: INC. ERWORD.      : 'SELECT MEMORY TEST(S)'
2332 013000  PMT1:
2333 013000 005267 165012  PMT0L: INC. ERWORD.      : 'ENTER LIMITS FOR QEX MEMORY'
2334 013004 005267 165006  PMT0R: INC. ERWORD.      : 'ENTER LIMITS FOR QLB REFERENCE PAGE'
2335 013010 005267 165002  PMT0B: INC. ERWORD.      : 'ENTER LIMITS FOR QLB PAGE'
2336 013014 005267 164776  PMTSR: INC. ERWORD.      : 'ENTER LIMITS FOR SUBREAD MEMORY'
2337 013020 005267 164772  PMT01: INC. ERWORD.      : 'ENTER LIMITS FOR SIDMEM PAGE 1'
2338 013024 005267 164766  PMT02: INC. ERWORD.      : 'ENTER LIMITS FOR SIDMEM PAGE 2'
2339 013030 005267 164762  MEMSEL: INC. ERWORD.      : 'SELECT MEMORIES'
2340 013034 005267 164756  ALLSEL: INC. ERWORD.      : 'ALL TESTS, ALL MEMORIES, FULL RANGE'
2341      000015  NEST:  =.  <.-NESTOP>/4
2342      ;
2343      ;      USE THE INDEX ERWORD TO COUNT UP FROM THE BOTTOM OF THE
2344      ;      MESSAGE TABLE, FIND THE END OF THE MESSAGE FIRST, THEN
2345      ;      THE BEGINNING, THEN GET THE LENGTH.
2346      ;
2347      ;
2348 013040 010246  MOV.  R2, -(SP)      : DON'T WRECK R2.
2349 013042 016702. 164750  MOV.  ERWORD, R2.    : LOAD LOOP COUNT.
2350 013046 012701 003773*  MOV.  #ASCIZ, R1     : POINT TO END OF MESSAGE TABLE
2351 013052 105741  1$:  TSTB. -(R1)      : LOOK FOR END OF MESSAGE.
2352 013054 001376  BNE.  1$             :
2353 013056 005302.  DEC.  R2.           : LOOP COUNT.
2354 013060 001374  BNE.  1$             : BACK UP ANOTHER MESSAGE.
2355 013062 010100  MOV.  R1, R0         : SAVE POINTER TO END OF MESSAGE.
2356 013064 105741  2$:  TSTB. -(R1)      : BACK UP TO BEGINNING OF MESSAGE.
2357 013066 001376  BNE.  2$

```

```

2358 013070 005201      INC      R1              ;BUMP TO FIRST CHAR OF MESSAGE
2359 013072 160100      SUB      R1,R0          ;R0 NOW = MESSAGE LENGTH
2360 013074 012602      MOV      (SP)+,R2      ;RESTORE R2
2361      ;
2362 013076      QIOW#$  #IO,WVB,#LUN,TT,#EFN,1, #STAT, <R1,R0>,ABEND
2363      ;
2364 013152      CLEF#$  #EFN,1
2365 013164 105767 164622    TSTB    STAT           ;GOOD RETURN
2366 013170 003433      BLE     ABEND          ;NO
2367      ;
2368      ;
2369      ;
2370 013172 022767 000015 164616    GMP     #NEST,ERWORD   ;PROMPT WITH MESSAGE
2371 013200 002424      BLT     TTX            ;NO, JUST EXIT
2372      ;
2373 013202 012700 000056*    MOV     #GCMBUF,R0     ;POINT TO GCML BUFFER
2374 013206 012701 000051      MOV     #41,,R1        ;NUMBER OF WORDS
2375 013212 005020      3$:    CLR     (R0)+         ;CLEAR BUFFER
2376 013214 005301      DEC     R1
2377 013216 001375      BNE     3$
2378      ;
2379 013220      GCML$  #GCMBLK
2380 013234 103411      BCS     ABEND
2381 013236 016067 000146 164734    MOV     G,CMLD(R0),GCMLEN ;SAVE LENGTH
2382 013244 012767 000056* 164730    MOV     #GCMBUF,GCMPNT  ;INITIALIZE COMMAND BUFFER POINTER
2383 013252 005067 164540      TTX:   CLR     ERWORD   ;CLEAR ERROR NUMBER INDICATOR
2384 013256      RETURN
2385      ;
2386 013260      ABEND: ABRT#$ #MYSELF
2387      .END  START
    
```

ABEND 013260R
ABEND2 012616R
ALL 004712R
ALLMEM 000644R
ALLSEL 013034R
ALLTST 000100
ALUCKE 040000
ALUOE 004000
AMSG 001723R
APLACE 000034R
ASCIZ 003773R
ASCNST 012464R
AST 010236R
ASTWRD 000010R
ASWRK 001704R
A01 010000
BASE 000030RG
BEGTST 012664R
BINWD 000022RG
BITVAL 000000
BIT0 000001
BIT1 000002
BIT10 002000
BIT11 004000
BIT12 010000
BIT13 020000
BIT14 040000
BIT15 100000
BIT2 000004
BIT3 000010
BIT4 000020
BIT5 000040
BIT6 000100
BIT7 000200
BIT8 000400
BIT9 001000
BPTISR 007516R
BYTE0 000000
BYTE1 000001
BYTE10 000012
BYTE11 000013
BYTE12 000014
BYTE13 000015
BYTE14 000016
BYTE15 000017
BYTE16 000020
BYTE17 000021
BYTE18 000023
BYTE19 000023
BYTE2 000002
BYTE20 000024
BYTE21 000025
BYTE22 000026
BYTE23 000027
BYTE24 000030
BYTE25 000031
BYTE26 000032

BYTE27 000033
BYTE28 000034
BYTE29 000035
BYTE3 000003
BYTE30 000036
BYTE31 000037
BYTE32 000040
BYTE33 000041
BYTE34 000042
BYTE35 000043
BYTE36 000044
BYTE37 000045
BYTE38 000046
BYTE39 000047
BYTE4 000004
BYTE40 000050
BYTE41 000051
BYTE42 000052
BYTE43 000053
BYTE44 000054
BYTE45 000055
BYTE46 000056
BYTE47 000057
BYTE48 000060
BYTE49 000061
BYTE5 000005
BYTE50 000062
BYTE51 000063
BYTE52 000064
BYTE53 000065
BYTE54 000066
BYTE55 000067
BYTE56 000070
BYTE57 000071
BYTE58 000072
BYTE59 000073
BYTE6 000006
BYTE60 000074
BYTE61 000075
BYTE62 000076
BYTE63 000077
BYTE64 000100
BYTE65 000101
BYTE66 000102
BYTE67 000103
BYTE68 000104
BYTE69 000105
BYTE7 000007
BYTE70 000106
BYTE71 000107
BYTE72 000110
BYTE73 000111
BYTE74 000112
BYTE75 000113
BYTE76 000114
BYTE77 000115
BYTE78 000116

BYTE79 000117
BYTE8 000010
BYTE80 000120
BYTE81 000121
BYTE82 000122
BYTE83 000123
BYTE84 000124
BYTE85 000125
BYTE86 000126
BYTE87 000127
BYTE88 000130
BYTE89 000131
BYTE9 000011
BYTE90 000132
BYTE91 000133
BYTE92 000134
BYTE93 000135
BYTE94 000136
BYTE95 000137
BYTE96 000140
BYTE97 000141
BYTE98 000142
BYTE99 000143
BYTVAL 000144
CAST 010466R
CBKALL 001000
CBKCLK 000400
CDHIGH 000740RG
CDLOW 000724RG
CHECK 005704R
CHECK0 006022R
CKDATA 001650RG
CK2 001652RG
CK3 001654RG
CMILUN 000002
CNOBRE 100000
CODE 000032RG
CONSOL 012476R
CPCCEN 010000
CPREAD 040000
CPURTE 020000
CSABRD 000004
CSECCI 100000
CSOE 000040
CSR1 000572RG
CSWRTE 000100
CURLIM 001474R
DATA1 000036RG
DBR.RD 000001
DB\$CFF 0001457
DB\$SPT 000026
DB\$TPC 000023
DISPGS 100000
DMAWR 000005
DMARRD 000003
DMARWR 000004
DOUBLE 012334R

EAST 010612R
EFN.1 000001
EFN.3 000003 G
EMSG 001734R
ENBR 010000
ENLNL 000014
ENDOF 004010R
ENDTST 012650R
ERLIM 000020R
ERNAME 012024R
EROPT 012754R
ERRPMT 006202R
ERRADD 001660RG
ERRCT 001662RG
ERROR 000400
ERR10 012674R
ERR2 012750R
ERR3 012744R
ERR4 012740R
ERR5 012734R
ERR50 012730R
ERR6 012724R
ERR60 012720R
ERR7 012714R
ERR8 012710R
ERR80 012704R
ERR9 012700R
ERWORD 000016R
ERW1 001664RG
ERW2 001666RG
ERW3 001670RG
ERW4 001672RG
FD.CCL ***** GX
FD.REC ***** GX
FD.RWM ***** GX
FD.TTY ***** GX
FIND 011326R
FIRST 001000 G
FNIN1 011252R
FNMTCH 011304R
FNOUT1 011244R
FNOUT2 011264R
FSECX 011436R
FVER 000044RG
F.ACTL 000076
F.ALOC 000040
F.BBFS 000062
F.BDB 000070
F.BGBC 000057
F.BKDN 000026
F.BKDS 000020
F.BKEF 000050
F.BKP1 000051
F.BKST 000024
F.BKVB 000064
F.CHR 000075
F.CNTG 000034

F.DFNB 000046
F.DSPT 000044
F.DVNM 000134
F.EFBK 000010
F.EFN 000050
F.EOBB 000032
F.ERR 000052
F.FACC 000043
F.FFBY 000014
F.FNAM 000110
F.FNB 000102
F.FTYR 000116
F.FVER 000120
F.HIBK 000004
F.LUN 000042
F.MBCT 000054
F.MBC1 000055
F.MBFG 000056
F.NRBD 000024
F.NREC 000030
F.OVBS 000030
F.RACC 000016
F.RATT 000001
F.RCNM 000034
F.RCTL 000017
F.RSIZ 000002
F.RTYP 000000
F.SEON 000100
F.SPEN 000072
F.SPUN 000074
F.STBK 000036
F.UNIT 000136
F.URBD 000020
F.VBN 000064
F.VBSZ 000060
GCMBLK 004024R
GCMBUF 000056R
GCMLEN 000200R
GCMPNT 000202R
GET 011664RG
GETCX 012014R
GETSX 012010R
GETX 012022R
GE.BIF 177775
GE.CLO 000004
GE.COM 000001
GE.CON 000020
GE.EOP 177766
GE.IND 000002
GE.IOR 177777
GE.LC 000010
GE.MDE 177774
GE.OPR 177776
GE.RBG 177730
GE.SIZ 000040
G.CMLD 000146
G.DPRM 000160

SYMBOL TABLE

G.ERR = 000140	MMSL = 005046R	PLR.EN = 000200	Q#MNC = 140000	S.FNTY = 000004
G.ISIZ = 000020	MMWRTE = 000010	PMPT10 = 005620R	Q#MR = 000052	S.FITYP = 000002
G.LPDL = 000060	MNAMSZ = 000023	PMSG = 001716R	Q#MRP = 000040	S.NFEN = 000020
G.MODE = 000141	MNOBRE = 100000	PMSG2 = 003774R	Q#MRP2 = 000240	S1 = 000020
G.PSDS = 000142	MOVE = 004770R	PMTQL = 013000R	Q#MSC = 040000	S2 = 000040
G.SIZE = 000224	MREN1 = 000001	PMTQR = 013004R	Q#MSET = 000004	TA = 007206R
HALT = 000200	MREN2 = 020000	PMTQ0 = 013010R	Q#MSP = 100000	TAST = 010510R
HLOOP = 011550R	MSEL = 000212R	PMTQ1 = 013020R	Q#NCLK = 176000	TB = 007224R
HLTTST = 012654R	MSYN = 000040	PMTQ2 = 013024R	Q#PP = 000100	TC = 007350R
HRL0 = ***** GX	MT = 000015	PMTQ3 = 013014R	Q#PPSW = 000320	TCDADD = 001604R
INDNB = 004472RG	MTBL = 000674R	PMT1 = 013000R	Q#PP2 = 000300	TCSPD = ***** GX
INFDB = 004332RG	MTBLN = 000014	PMT10 = 012764R	Q#QHLT = 000013	TCSPU = ***** GX
INLUN = 000003	HTCNT = 000216R	PMT3 = 012770R	Q#QL = 000043	TCUADD = 001620R
IO.ATA = ***** GX	HTERR = 005440R	FM2LN = 000014	Q#QLA = 000053	TD = 007454R
IO.WVB = ***** GX	MTJUMP = 007466R	PPCR = ***** GX	Q#QLB = 000054	TDADDR = 001634R
JMPMT = 006664R	MTMAIN = 006442R	FREADD = 001656RG	Q#QLR = 000001	TDNUL = 007464R
LASTJ = 006732R	MTPMT = 005610R	PRINT = 002151RG	Q#QW = 000042	TD#CTR = 176370
LBPP = ***** GX	MTPNT = 000214R	PSECK = 011626R	Q#RDCD = 000005	TD#CTW = 176360
LCOUNT = 000046RG	MTREF = 000754R	QB = 000004	Q#RDMD = 000006	TD#INL = 004000
LCS = 001700RG	MTRT = 005426R	QR = 000002	Q#REBK = 001000	TD#MEM = 000270
LIMITS = 011036R	MTSET = 006326R	QR#CR1 = 176420	Q#RNC = 006000	TD#0AR = 176344
LIMM1 = 005172R	MTSL = 005376R	QR#CR2 = 176422	Q#RSC = 004000	TD#0TR = 176346
LIMNUM = 000014	MTSUB = 001240R	QR#LBR = 176424	Q#RSET = 000010	TD#QRD = 000274
LIMREF = 001210R	MT10 = 000054R	QX = 000001	Q#SM = 100000	TD#SW = 176376
LIMX = 011220R	MYSELF = 000000R	Q#ATTN = 000100	Q#SP = 000120	TD#TAR = 176372
LIMX2 = 011234R	MZREL = 005464R	Q#BCL = 000001	Q#SP2 = 000340	TD#TAW = 176362
LMM = 001674RG	N = 000144	Q#CCCP = 000040	REGSTR = 000020	TD#TDW = 176374
LOC.EN = 000100	NEST = 000015	Q#CHB = 000000	REST = 010700R	TD#TDW = 176364
LOC.WA = 040000	NESTOP = 012754R	Q#CHRL = 000200	RGQ.EN = 000200	TEST10 = 000010
LOC.WB = 100000	NMEMS = 000006	Q#CLR = 000040	RGQ.VA = 020000	TEST3 = 000002
LOOP = 000001	NREGS = 000000	Q#CNC = 030000	RMSG = 001746R	TEST6 = 000004
LOOPCT = 000210R	NXTCNT = 000222R	Q#CP = 000060	RT = 000000	THSG = 001711R
LOWER = 000024R	NXTPNT = 000220R	Q#CPCC = 000010	RT3 = 000052R	TROCT = 000224R
LPRMPT = 006054R	N.DID = 000024	Q#CP2 = 000260	SCAN = 011236R	TRTBL = 000424R
LPTST = 012750R	N.DVNM = 000032	Q#CSC = 010000	SELMT = 012774R	TRTBL2 = 000624R
LSTACK = 006706R	N.FID = 000000	Q#CSEL = 000360	SEQ.CI = 000010	TSKTCB = 000004RG
LUN.TT = 000001	N.FNAM = 000006	Q#CSET = 000002	SR = 000010	TTX = 013252R
MAREN1 = 000001	N.FITYP = 000014	Q#CSP = 020000	STADDR = 001524R	T#AD = 000020
MAREN2 = 004000	N.FVER = 000016	Q#DMA = 000001	START = 004530R	T#BA = 000002
MARL0D = 010000	N.NEXT = 000022	Q#ENBK = 040000	STAT = 000012RG	T#BD = 000010
MAROUT = 000002	N.STAT = 000020	Q#ENOP = 020000	STOP = 012660R	T#BSO = 100000
MAR.LO = 002000	N.UNIT = 000034	Q#FAL = 004000	STRMEM = 000027	T#BT = 000020
MAR.OU = 000040	OLDVEC = 000006RG	Q#FC = 000045	STUFSP = ***** GX	T#BTAR = 000030
MBKALL = 001000	OUT1 = 012670R	Q#FO = 000044	S#CLR = 000000	T#BTDR = 002000
MBKCLK = 000400	PACK = 011444R	Q#FP = 000046	S#LA = 000001	T#CD = 000100
MEMERR = 007632RG	PACKX = 011662R	Q#HBF = 000002	S#QB = 000005	T#CLK = 002000
MEMNAM = 012066R	PAR\$\$\$ = 000027	Q#ICP = 000006	S#QR = 000006	T#DISK = 000200
MEMOFF = 006664R	PASS = 000206R	Q#IHB = 000003	S#QX = 000004	T#DRD = 000004
MEMORY = 000040	PASSC = 012304R	Q#IHR = 000002	S#SR = 000007	T#EMEM = 010000
MEMS = 005300R	PASSH = 000204R	Q#IHRP = 000007	S#S1 = 000010	T#FSAA = 000000
MENSEL = 013030R	PASSX = 012462R	Q#LBD = 001000	S#S2 = 000014	T#FSAB = 000004
MEMTOP = 005100R	PAST = 010626R	Q#LBDP = 001001	S.BFHD = 000020	T#FSAC = 000014
MEMX = 010220R	PCLCX = 011634R	Q#LBP = 000001	S.FATT = 000016	T#FSB2 = 000010
MFTBL = 001765R	PLB = 000010	Q#LDCD = 000003	S.FDB = 000140	T#IBAR = 000024
MMADRD = 000100	PLC = 000020	Q#LDMD = 000004	S.FNAM = 000006	T#IBE = 020000
MMLEFT = 000002	PLD = 000030	Q#LDPP = 002000	S.FNB = 000036	T#IBF = 040000
MMOE = 000004	PLRWR = 000200	Q#LHP = 010000	S.FNSL = 000017	

SYMBOL TABLE

T\$ICD = 000040	T8 = 007122R	WORD30 = 000074	WORD61 = 000172	WORD91 = 000266
T\$MODE = 004000	T9 = 007152R	WORD31 = 000076	WORD62 = 000174	WORD92 = 000270
T\$OB = 000036	UBD:IN = 000020	WORD32 = 000100	WORD63 = 000176	WORD93 = 000272
T\$OBE = 004000	UNMSG = 001760R	WORD33 = 000102	WORD64 = 000200	WORD94 = 000274
T\$OBF = 010000	UNPK = 012224R	WORD34 = 000104	WORD65 = 000202	WORD95 = 000276
T\$OBRA = 000034	UNPKX = 012274R	WORD35 = 000106	WORD66 = 000204	WORD96 = 000300
T\$OBWA = 000032	UPPER = 000026R	WORD36 = 000110	WORD67 = 000206	WORD97 = 000302
T\$OUTA = 100000	VIRT = 000040RG	WORD37 = 000112	WORD68 = 000210	WORD98 = 000304
T\$RBD0 = 000200	WAST = 010336R	WORD38 = 000114	WORD69 = 000212	WORD99 = 000306
T\$RNB = 000040	WCOUNT = 000050RG	WORD39 = 000116	WORD7 = 000016	WRDVAL = 000310
T\$RSET = 040000	WORD0 = 000000	WORD4 = 000010	WORD70 = 000214	XTREAD = 001000
T\$SC = 000022	WORD1 = 000002	WORD40 = 000120	WORD71 = 000216	XTWRTE = 000400
T\$SCLK = 020000	WORD10 = 000024	WORD41 = 000122	WORD72 = 000220	\$CBDSC = ***** GX.
T\$SEG1 = 000000	WORD11 = 000026	WORD42 = 000124	WORD73 = 000222	\$CDTB = ***** GX.
T\$SEG2 = 000001	WORD12 = 000030	WORD43 = 000126	WORD74 = 000224	\$CEFI = ***** GX.
T\$SEG3 = 000002	WORD13 = 000032	WORD44 = 000130	WORD75 = 000226	\$CSTA = ***** GX.
T\$SO = 000001	WORD14 = 000034	WORD45 = 000132	WORD76 = 000230	\$DDIV = ***** GX.
T\$UBUS = 100000	WORD15 = 000036	WORD46 = 000134	WORD77 = 000232	\$DIV = ***** GX.
T\$ICLK = 000400	WORD16 = 000040	WORD47 = 000136	WORD78 = 000234	\$DRDSE = ***** GX.
T\$BBEN = 000020	WORD17 = 000042	WORD48 = 000140	WORD79 = 000236	\$IUL = ***** GX.
T1 = 006742R	WORD18 = 000044	WORD49 = 000142	WORD8 = 000020	\$TKTCB = ***** GX.
T1ADDR = 001540R	WORD19 = 000046	WORD5 = 000012	WORD80 = 000240	\$\$\$ = 004204R
T1SP = ***** GX.	WORD2 = 000004	WORD50 = 000144	WORD81 = 000242	\$\$\$ARG = 000002
T2 = 006752R	WORD20 = 000050	WORD51 = 000146	WORD82 = 000244	\$\$\$T1 = 000067
T3 = 006766R	WORD21 = 000052	WORD52 = 000150	WORD83 = 000246	\$\$\$T2 = 000027
T4 = 007004R	WORD22 = 000054	WORD53 = 000152	WORD84 = 000250	.FINIT = ***** G.
T5 = 007022R	WORD23 = 000056	WORD54 = 000154	WORD85 = 000252	.FSRCB = ***** G.
T6 = 007052R	WORD24 = 000060	WORD55 = 000156	WORD86 = 000254	.GCML1 = ***** G.
T6ADDR = 001554R	WORD25 = 000062	WORD56 = 000160	WORD87 = 000256	.READ = ***** G.
T6CHK = 005662R	WORD26 = 000064	WORD57 = 000162	WORD88 = 000260	...PC1 = 004332R
T6SP = ***** GX.	WORD27 = 000066	WORD58 = 000164	WORD89 = 000262	...PC2 = 004506R
T7 = 007112R	WORD28 = 000070	WORD59 = 000166	WORD9 = 000022	...PC3 = 004332R
T7ADDR = 001570R	WORD29 = 000072	WORD6 = 000014	WORD90 = 000264	...TPC = 000020
T7SP = ***** GX.	WORD3 = 000006	WORD60 = 000170		

.ABS. 000000 000
013312 001
\$\$FSR1 001020 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 9394 WORDS (37 PAGES)
DYNAMIC MEMORY: 10196 WORDS (39 PAGES)
ELAPSED TIME: 00:02:13
SMT, SMT/SP=C20, L3IM, C20, L3SMT

```

1          .TITLE--SPTTEST....
2 000000   .PSECT--SPTTEST.
3          ;
4          ;
5          ;
6          ;
7          ;
8          ;
9          ;
10 000000  STUFSP::
11 000000  016667  000002  000000G.  MOV.    2(SP),PREADD.  ;WORKING ADDRESS.
12 000006  1$:  CALL.   WSP.          ;WRITE QLB REF MEMORY.
13 000012  005267  000000G.  INC.   PREADD.       ;BUMP ADDRESS.
14 000016  026667  000004  000000G.  CMP.   4(SP),PREADD.  ;FINISHED?.
15 000024  103370  ;          BHIS.   1$.          ;NO.
16          ;
17 000026  012746  000000  ;          MOV.   #$CLR,-(SP)  ;CLEAR PPS.
18 000032  ;          CALL.  SPCR
19 000036  016667  000002  000000G.  MOV.   2(SP),PREADD.  ;WORKING ADDRESS.
20 000044  2$:  CALL.   CSP.          ;READ AND COMPARE QLB REF MEMORY.
21 000050  005267  000000G.  INC.   PREADD.       ;BUMP ADDRESS.
22 000054  026667  000004  000000G.  CMP.   4(SP),PREADD.  ;FINISHED?.
23 000062  103370  ;          BHIS.   2$.          ;NO.
24          ;
25 000064  012746  000000  ;          MOV.   #$CLR,-(SP)  ;CLEAR SPS.
26 000070  ;          CALL.  SPCR
27 000074  ;          RETURN.

```



```

29      ;
30      ;
31      ;      TEST-01
32      ;      WRITE MEMORY ADDRESS INTO MEMORY LOCATION.
33      ;
34      ;
35      000076      TISP::
36      000076      016667      000002      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS.
37      000104      016667      000002      000000G      MOV      2(SP),CKDATA      ;TEST PATTERN = ADDRESS.
38      000112      1$:      CALL      WSP      ;WRITE QLB REF MEMORY.
39      000116      005267      000000G      INC      CKDATA      ;BUMP TEST COUNTER
40      000122      005267      000000G      INC      PREADD      ;BUMP ADDRESS.
41      000126      026667      000004      000000G      CMP      4(SP),PREADD      ;FINISHED ?
42      000134      103366      BHS      1$      ;NO
43      ;
44      000136      012746      000000      MOV      #$CLR,-(SP)      ;CLEAR SPS.
45      000142      CALL      SPCR
46      000146      016667      000002      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS.
47      000154      016667      000002      000000G      MOV      2(SP),CKDATA      ;TEST PATTERN = ADDRESS.
48      000162      2$:      CALL      CSP      ;READ AND COMPARE QLB REF MEMORY.
49      000166      005267      000000G      INC      CKDATA      ;BUMP TEST COUNTER
50      000172      005267      000000G      INC      PREADD      ;BUMP ADDRESS.
51      000176      026667      000004      000000G      CMP      4(SP),PREADD      ;FINISHED ?
52      000204      103366      BHS      2$      ;NO
53      ;
54      000206      012746      000000      MOV      #$CLR,-(SP)      ;CLEAR SPS.
55      000212      CALL      SPCR
56      000216      RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

58
59
60
61
62
63
64 000220
65 000220 012767 177777 000000G.
66 000226 012702 000012
67 000232 016667 000002 000000G. 10$:
68 000240 1$:
69 000244 062767 000002 000000G.
70 000252 026667 000004 000000G.
71 000260 103367
72 000262 005302
73 000264 001362
74
75
76
77
78 000266
79 000266 012746 000000
80 000272
81 000276 005067 000000G.
82 000302 016667 000002 000000G.
83 000310 005267 000000G.
84 000314 1$:
85 000320 062767 000002 000000G.
86 000326 026667 000004 000000G.
87 000334 103367
88
89 000336 012746 000000
90 000342
91 000346

:
: TEST-06
: CROSS-TALK TEST
:
: T6SP::
MOV. #+1,CKDATA. ;SET TEST PATTERN =X'FFFF'
MOV. #10,R2. ;SET LOOP COUNT
MOV. 2(SP),PREADD. ;WORKING ADDRESS
CALL. WSP. ;WRITE QLB REF MEMORY
ADD. #2,PREADD. ;SKIP ONE ADDRESS
CMP. 4(SP),PREADD. ;FINISHED?
BHS. 1$ ;NO
DEC. R2. ;SUB FROM LOOP COUNT
BNE. 10$

:
: READ ZEROS FROM THE MEMORY LOCATIONS INTO WHICH ONES
: WERE NOT WRITTEN
:
: R6Z:
MOV. #S$CLR,-(SP) ;CLEAR SPS
CALL. SPCR
CLR. CKDATA. ;SET TEST PATTERN = 0
MOV. 2(SP),PREADD. ;WORKING ADDRESS
INC. PREADD. ;BUMP START ADDRESS
CALL. CSP. ;READ AND COMPARE QLB REF MEMORY
ADD. #2,PREADD. ;SKIP ONE ADDRESS
CMP. 4(SP),PREADD. ;FINISHED?
BHS. 1$ ;NO
MOV. #S$CLR,-(SP) ;CLEAR SPS
CALL. SPCR
RETURN

```

```

93      ;
94      ;
95      ;      TEST-07
96      ;      WRITE-COMPLEMENT-OF-MEMORY-ADDRESS-INTO-MEMORY-LOCATION.
97      ;
98      ;
99      000350      T7SP::
100     000350      016667      000002      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS.
101     000356      016602      000002      MOV      2(SP),R2      ;TEST-PATTERN-=-ADDRESS.
102     000362      005102      1$:      COM      R2      ;GET-ADDRESS-COMPLEMENT.
103     000364      010267      000000G      MOV      R2,CKDATA      ;SET-TEST-PATTERN.
104     000370      CALL      WSP      ;WRITE-QLB-REF-MEMORY.
105     000374      005267      000000G      INC      PREADD      ;BUMP-ADDRESS.
106     000400      016702      000000G      MOV      PREADD,R2      ;SET-UP-FOR-NEXT-TIME.
107     000404      026667      000004      000000G      CMP      4(SP),PREADD      ;FINISHED-?.
108     000412      103363      BHIS     1$      ;NO.
109
110     000414      012746      000000      MOV      #S$CLR,-(SP)      ;CLEAR-SPS.
111     000420      CALL     SPCR
112     000424      016667      000002      000000G      MOV      2(SP),PREADD      ;WORKING ADDRESS.
113     000432      016602      000002      MOV      2(SP),R2      ;TEST-PATTERN-=-ADDRESS.
114     000436      005102      2$:      COM      R2      ;GET-ADDRESS-COMPLEMENT.
115     000440      010267      000000G      MOV      R2,CKDATA      ;SET-TEST-PATTERN.
116     000444      CALL     CSP      ;READ-AND-COMPARE-QLB-REF-MEMORY.
117     000450      005267      000000G      INC      PREADD      ;BUMP-ADDRESS.
118     000454      016702      000000G      MOV      PREADD,R2      ;SET-UP-FOR-NEXT-TIME.
119     000460      026667      000004      000000G      CMP      4(SP),PREADD      ;FINISHED-?.
120     000466      103363      BHIS     2$      ;NO.
121
122     000470      012746      000000      MOV      #S$CLR,-(SP)      ;CLEAR-SPS.
123     000474      CALL     SPCR
124     000500      RETURN

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

126
127
128
129
130
131
132
133
134 000502.
135 000502. 016667 000002 000000G.
136 000510 016767 000000G 000000G 1$:
137 000516
138 000522. 016767 000000G 000000G.
139 000530
140 000534 005267 000000G.
141 000540 026667 000004 000000G.
142 000546 103360
143
144 000550 012746 000000
145 000554
146 000560
147
148
149
150
151 000562.
152 000562. 016667 000004 000000G.
153 000570 016767 000000G 000000G 1$:
154 000576
155 000602. 016767 000000G 000000G.
156 000610
157 000614 162767 000001 000000G.
158 000622. 026667 000002 000000G.
159 000630 003757
160
161 000632. 012746 000000
162 000636
163 000642.

;
; TEST 12.
; LOOK FORWARD, LOOK BEHIND ADDRESSING TEST.
;
;
; READ FROM TOP OF MEMORY DOWN, THEN WRITE.
;
; TCSPD::
; MOV 2(SP),PREADD. ;WORKING ADDRESS.
; MOV CK2,CKDATA. ;TEST PATTERN FOR READ.
; CALL CSP. ;CHECK MEMORY LOCATION.
; MOV CK3,CKDATA. ;TEST PATTERN FOR WRITE.
; CALL WSP. ;WRITE QLB REF MEMORY.
; INC PREADD. ;BUMP ADDRESS.
; CMP 4(SP),PREADD. ;FINISHED?.
; BHIS 1$. ;NO.
;
; MOV #S$CLR,-(SP) ;CLEAR SPS.
; CALL SPCR
; RETURN.
;
; TEST 12.
; READ FROM BOTTOM OF MEMORY UP, THEN WRITE.
;
; TCSPU::
; MOV 4(SP),PREADD. ;WORKING ADDRESS = END ADDRESS.
; MOV CK2,CKDATA. ;TEST PATTERN FOR READ.
; CALL CSP. ;CHECK MEMORY LOCATION.
; MOV CK3,CKDATA. ;TEST PATTERN FOR WRITE.
; CALL WSP. ;WRITE MEMORY LOCATION.
; SUB #1,PREADD. ;BACK UP 1
; CMP 2(SP),PREADD. ;FINISHED?.
; BLE 1$. ;NO.
;
; MOV #S$CLR,-(SP) ;CLEAR SPS.
; CALL SPCR
; RETURN.

```

```

165 ;
166 ;
167 ; WRITE QLB REF MEMORY
168 ;
169 ;
170 000644 WSP:
171 000644 012746 000001 MOV #S$LA,-(SP) ;SELECT MEMORY
172 000650 CALL SPCR ;
173 000654 016746 000000G MOV PREADD,-(SP) ;MEMORY ADDRESS
174 000660 CALL LBSP ;
175 000664 016746 000000G MOV CODE,-(SP) ;SELECT PARTICULAR MEMORY
176 000670 CALL SPCR ;
177 000674 016746 000000G MOV CKDATA,-(SP) ;TEST PATTERN
178 000700 CALL LBSP ;
179 000704 RETURN ;
180 ;
181 ;
182 ; READ AND COMPARE QLB REF MEMORY
183 ;
184 ;
185 000706 CSP:
186 000706 012746 000001 MOV #S$LA,-(SP) ;SELECT MEMORY
187 000712 CALL SPCR ;
188 000716 016746 000000G MOV PREADD,-(SP) ;MEMORY ADDRESS
189 000722 CALL LBSP ;
190 000726 016746 000000G MOV CODE,-(SP) ;SELECT PARTICULAR MEMORY
191 000732 CALL SPCR ;
192 000736 CALL SPLB ;
193 000742 012667 000000G MOV (SP)+,ERW1 ;UNLOAD WORD FROM QLB REF
194 ;
195 000746 022767 000004 000000G CMP #S$QX,CODE ;IS THIS QEX MEMORY?
196 000754 001003 BNE 20$ ;BRANCH IF NOT
197 000756 052767 007777 000000G BIS #7777,CKDATA ;LOWER 12 BITS ALWAYS ONES
198 000764 20$:
199 000764 026767 000000G 000000G CMP CKDATA,ERW1 ;SAME AS PATTERN WRITTEN
200 000772 001410 BEQ 1$ ;YES EXIT
201 000774 016767 000000G 000000G MOV PREADD,ERRADD ;ADDRESS OF ERROR
202 001002 012767 000001 000000G MOV #1,ERRCT ;NUMBER OF WORDS TO PRINT
203 001010 CALL MEMERR ;GO TO ERROR ROUTINE
204 001014 1$:
205 ; RETURN ;
206 000001 ; .END

```

ALUCKE = 040000	BYTE42 = 000052	BYTE94 = 000136	PLC = 000020	Q#QW = 000042
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	PLD = 000030	Q#RDCD = 000005
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	PLRWR = 000200	Q#RDMO = 000006
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	PREADD = ***** GX	Q#REBK = 001000
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	QR#CR1 = 176420	Q#RNC = 006000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	QR#CR2 = 176422	Q#RSC = 004000
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	QR#LBR = 176424	Q#RSET = 000010
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q#ATTN = 000100	Q#SM = 100000
BIT12 = 010000	BYTE5 = 000005	CBKCLK = 000400	Q#BCL = 000001	Q#SP = 000120
BIT13 = 020000	BYTE50 = 000062	CKDATA = ***** GX	Q#CCCP = 000040	Q#SP2 = 000340
BIT14 = 040000	BYTE51 = 000063	CK2 = ***** GX	Q#CHB = 000400	RG0,EN = 000200
BIT15 = 100000	BYTE52 = 000064	CK3 = ***** GX	Q#CHRL = 000200	RG0,VA = 020000
BIT2 = 000004	BYTE53 = 000065	CNOBRE = 100000	Q#CLR = 000040	RGZ = 000266R 002
BIT3 = 000010	BYTE54 = 000066	CODE = ***** GX	Q#CNC = 030000	SEQ,CI = 000010
BIT4 = 000020	BYTE55 = 000067	CPCCEN = 010000	Q#CP = 000060	SPCR = ***** GX
BIT5 = 000040	BYTE56 = 000070	CPREAD = 040000	Q#CPCC = 000010	SPLB = ***** GX
BIT6 = 000100	BYTE57 = 000071	CPWRTE = 020000	Q#CP2 = 000260	STUFSP = 000000RG 002
BIT7 = 000200	BYTE58 = 000072	CSADRD = 000004	Q#CSC = 010000	S#CLR = 000000
BIT8 = 000400	BYTE59 = 000073	CSEQCI = 100000	Q#CSEL = 000360	S#SLA = 000001
BIT9 = 001000	BYTE6 = 000006	CSDI = 000040	Q#CSET = 000002	S#QB = 000005
BYTE0 = 000000	BYTE60 = 000074	CSP = 000706R 002	Q#CSP = 020000	S#QR = 000006
BYTE1 = 000001	BYTE61 = 000075	CSWRTE = 000100	Q#DMA = 000001	S#QX = 000004
BYTE10 = 000012	BYTE62 = 000076	DBR, RD = 000001	Q#ENBK = 040000	S#SR = 000007
BYTE11 = 000013	BYTE63 = 000077	DB#CPP = 001457	Q#ENOP = 020000	S#S1 = 000010
BYTE12 = 000014	BYTE64 = 000100	DB#SPT = 000026	Q#FAL = 004000	S#S2 = 000014
BYTE13 = 000015	BYTE65 = 000101	DB#TPC = 000023	Q#FC = 000045	TCSPD = 000502RG 002
BYTE14 = 000016	BYTE66 = 000102	DISPGS = 100000	Q#FO = 000044	TCSPU = 000562RG 002
BYTE15 = 000017	BYTE67 = 000103	DMAAWR = 000005	Q#FP = 000046	TD#CTR = 176370
BYTE16 = 000020	BYTE68 = 000104	DMARRD = 000003	Q#HBF = 000002	TD#CTW = 176360
BYTE17 = 000021	BYTE69 = 000105	DMARWR = 000004	Q#ICP = 000006	TD#INL = 004000
BYTE18 = 000022	BYTE7 = 000007	ENBR = 010000	Q#IHB = 000003	TD#MEM = 000270
BYTE19 = 000023	BYTE70 = 000106	ERRADD = ***** GX	Q#IHRL = 000002	TD#OAR = 176344
BYTE2 = 000002	BYTE71 = 000107	ERRCT = ***** GX	Q#IMRP = 000007	TD#OTR = 176346
BYTE20 = 000024	BYTE72 = 000110	ERW1 = ***** GX	Q#LBD = 001000	TD#QRD = 000274
BYTE21 = 000025	BYTE73 = 000111	LBSF = ***** GX	Q#LBDP = 001001	TD#SW = 176376
BYTE22 = 000026	BYTE74 = 000112	LOC,EN = 000100	Q#LBP = 000001	TD#TAR = 176372
BYTE23 = 000027	BYTE75 = 000113	LOC,WA = 040000	Q#LDCD = 000003	TD#TAW = 176362
BYTE24 = 000030	BYTE76 = 000114	LOC,WB = 100000	Q#LDMO = 000004	TD#TDR = 176374
BYTE25 = 000031	BYTE77 = 000115	MAREN1 = 000001	Q#LDPP = 002000	TD#TDW = 176364
BYTE26 = 000032	BYTE78 = 000116	MAREN2 = 004000	Q#LHP = 010000	T#AD = 000020
BYTE27 = 000033	BYTE79 = 000117	MARLOD = 010000	Q#MNC = 140000	T#BA = 000002
BYTE28 = 000034	BYTE8 = 000010	MAROUT = 000002	Q#MR = 000052	T#BD = 000010
BYTE29 = 000035	BYTE80 = 000120	MAR,LO = 002000	Q#MRP = 000040	T#BSO = 100000
BYTE3 = 000003	BYTE81 = 000121	MAR,OU = 000040	Q#MRP2 = 000240	T#BT = 000020
BYTE30 = 000036	BYTE82 = 000122	MBKALL = 001000	Q#MSC = 040000	T#BTAR = 000030
BYTE31 = 000037	BYTE83 = 000123	MBKCLK = 000400	Q#MSET = 000004	T#BTD = 002000
BYTE32 = 000040	BYTE84 = 000124	MEMERR = ***** GX	Q#MSP = 100000	T#CD = 000100
BYTE33 = 000041	BYTE85 = 000125	M#ADRD = 000100	Q#NCLK = 176000	T#CLK = 002000
BYTE34 = 000042	BYTE86 = 000126	M#LEFT = 000002	Q#PP = 000100	T#DLSK = 000200
BYTE35 = 000043	BYTE87 = 000127	M#ODE = 000004	Q#PPSW = 000320	T#DRD = 000004
BYTE36 = 000044	BYTE88 = 000130	M#WRTE = 000010	Q#PP2 = 000300	T#EMEM = 010000
BYTE37 = 000045	BYTE89 = 000131	M#OBRE = 100000	Q#QHLT = 000013	T#FSAA = 000000
BYTE38 = 000046	BYTE9 = 000011	MREN1 = 000001	Q#QW = 000042	T#FSAB = 000004
BYTE39 = 000047	BYTE90 = 000132	MREN2 = 020000	Q#OLB = 000054	T#FSAC = 000014
BYTE4 = 000004	BYTE91 = 000133	MSYN = 000040	Q#OLR = 000000	T#FSB24 = 000010
BYTE40 = 000050	BYTE92 = 000134	N = 000144		T#IB = 000026
BYTE41 = 000051	BYTE93 = 000135	PLB = 000010		T#IBAR = 000024

T\$IBE = 020000	WORD0 = 000000	WORD32 = 000100	WORD56 = 000160	WORD8 = 000020
T\$IBF = 040000	WORD1 = 000002	WORD33 = 000102	WORD57 = 000162	WORD80 = 000240
T\$ICD = 000040	WORD10 = 000024	WORD34 = 000104	WORD58 = 000164	WORD81 = 000242
T\$MODE = 004000	WORD11 = 000026	WORD35 = 000106	WORD59 = 000166	WORD82 = 000244
T\$OB = 000036	WORD12 = 000030	WORD36 = 000110	WORD6 = 000014	WORD83 = 000246
T\$OBE = 004000	WORD13 = 000032	WORD37 = 000112	WORD60 = 000170	WORD84 = 000250
T\$OBF = 010000	WORD14 = 000034	WORD38 = 000114	WORD61 = 000172	WORD85 = 000252
T\$OBRA = 000034	WORD15 = 000036	WORD39 = 000116	WORD62 = 000174	WORD86 = 000254
T\$OBWA = 000032	WORD16 = 000040	WORD4 = 000010	WORD63 = 000176	WORD87 = 000256
T\$OUTA = 100000	WORD17 = 000042	WORD40 = 000120	WORD64 = 000200	WORD88 = 000260
T\$RBDQ = 000200	WORD18 = 000044	WORD41 = 000122	WORD65 = 000202	WORD89 = 000262
T\$RNB = 000040	WORD19 = 000046	WORD42 = 000124	WORD66 = 000204	WORD9 = 000022
T\$RSET = 040000	WORD2 = 000004	WORD43 = 000126	WORD67 = 000206	WORD90 = 000264
T\$SC = 000022	WORD20 = 000050	WORD44 = 000130	WORD68 = 000210	WORD91 = 000266
T\$SCLK = 020000	WORD21 = 000052	WORD45 = 000132	WORD69 = 000212	WORD92 = 000270
T\$SEG1 = 000000	WORD22 = 000054	WORD46 = 000134	WORD7 = 000016	WORD93 = 000272
T\$SEG2 = 000001	WORD23 = 000056	WORD47 = 000136	WORD70 = 000214	WORD94 = 000274
T\$SEG3 = 000002	WORD24 = 000060	WORD48 = 000140	WORD71 = 000216	WORD95 = 000276
T\$SO = 000001	WORD25 = 000062	WORD49 = 000142	WORD72 = 000220	WORD96 = 000300
T\$UBUS = 100000	WORD26 = 000064	WORD5 = 000012	WORD73 = 000222	WORD97 = 000302
T\$ICLK = 000400	WORD27 = 000066	WORD50 = 000144	WORD74 = 000224	WORD98 = 000304
T\$BEN = 000020	WORD28 = 000070	WORD51 = 000146	WORD75 = 000226	WORD99 = 000306
T1SP = 000076RG 002	WORD29 = 000072	WORD52 = 000150	WORD76 = 000230	WRDVAL = 000310
T6SP = 000220RG 002	WORD3 = 000006	WORD53 = 000152	WORD77 = 000232	WSP = 000644R 002
T7SP = 000350RG 002	WORD30 = 000074	WORD54 = 000154	WORD78 = 000234	XTREAD = 001000
UBD: IN = 000020	WORD31 = 000076	WORD55 = 000156	WORD79 = 000236	XTWRITE = 000400

. ABS. 000000 000
000000 001
SPTEST 001016 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3150 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:44
SPTEST, SPTEST /-SP=C[20, 1]IM, C[20, 1]SPTEST

```

1          .TITLE..$PSUB...
2 000000   .PSECT..SPSUB.
3          ;
4          ;
5          .MCALL..WTSE$S,CLEF$S.
6          EFN:3 = 3
7          ;
8          ;
9          ;
10         ;
11         ;
12         ;
13         ;
14         ;
15         ;
16         ;
17         ;
18         ;
19         ;
20 000000   LBSP::
21 000000   016667 000002 176424   MOV.    2(SP),DR#LBR.      ;MOVE DATA TO LOD BUS REG.
22 000006   012746 001001           MOV.    #<Q$LBD+Q$LBP>,-(SP) ;CLR DRIVE AND PULSE
23 000012   052716 000360           BIS.    #<Q#CSEL>,(SP)     ;CLR SELECTION BITS
24 000016   012746 176000           MOV.    #<Q#NCLK>,-(SP)   ;SET NO-CLOCKS
25 000022   052716 000340           BIS.    #Q$SP2,(SP)      ;SELECT SP
26 000026           CALL.   CSR1              ;
27         ;
28 000032   012746 006000           MOV.    #Q$RNC,-(SP)     ;CLEAR CP NO-CLOCK BITS
29 000036   012746 001000           MOV.    #Q$LBD,-(SP)     ;SET LOD BUS DRIVE
30 000042           CALL.   CSR1              ;
31         ;
32         ;
33         ;
34         ;
34 000046   012746 001001           MOV.    #<Q$LBD+Q$LBP>,-(SP) ;CLEAR DRIVE AND PULSE
35 000052   052716 000360           BIS.    #<Q#CSEL>,(SP)   ;CLR SELECTION BITS
36 000056   012746 176000           MOV.    #<Q#NCLK>,-(SP)   ;SET NO-CLOCKS
37 000062           CALL.   CSR1              ;
38         ;
39 000066   011666 000002           MOV.    (SP),2(SP)        ;MOVE RETURN ADDRESS DOWN STACK
40 000072   005726           TST.    (SP)+            ;POINT TO RETURN ADDRESS
41 000074           RETURN.

```



```

43      ;
44      ;
45      ;      DATA TRANSFER
46      ;      LOD-BUS REGISTER TO A DESTINATION ON THE SP-BUS
47      ;      SINGLE-CLOCK SEQUENCER ONLY.
48      ;
49      ;      INPUT:
50      ;      2(SP) DATA FOR PRE-SELECTED SP DESTINATION.
51      ;
52      ;
53 000076      ; LBSSC::
54 000076 016667 000002 176424      MOV. 2(SP),QR$LBR.      ;MOVE DATA TO LOD-BUS-REG.
55 000104 012746 001001      MOV.  *(Q$LBD+Q$LBP),-(SP)      ;CLEAR DRIVE AND PULSE.
56 000110 052716 000360      BIS.  #Q$CSEL,(SP)      ;CLR SELECTION BITS.
57 000114 012746 176000      MOV.  *(Q$NCLK),-(SP)      ;SET NO-CLOCKS.
58 000120 052716 000340      BIS.  #Q$SP2,(SP)      ;SELECT SP.
59 000124      CALL. CSR1      ;WRITE CONTROL REGISTER.
60      ;
61 000130 012746 006000      MOV.  #Q$RNC,-(SP)      ;CLEAR SP NO-CLOCK BITS.
62 000134 012746 005000      MOV.  *(Q$RSC+Q$LBD),-(SP)      ;SET SP CLOCK.
63 000140      CALL. CSR1      ;
64      ;
65      ;      DE-SELECTION
66      ;
67 000144 012746 001001      MOV.  *(Q$LBD+Q$LBP),-(SP)      ;CLEAR DRIVE AND PULSE.
68 000150 052716 000360      BIS.  #Q$CSEL,(SP)      ;CLR SELECTION BITS.
69 000154 012746 176000      MOV.  *(Q$NCLK),-(SP)      ;SET NO-CLOCKS.
70 000160      CALL. CSR1
71      ;
72 000164 011666 000002      MOV.  (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK.
73 000170 005726      TST.  (SP)+      ;POINT TO RETURN ADDRESS.
74 000172      RETURN.

```

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

```

76
77
78
79          :      DATA TRANSFER TO LOD BUS REG FROM SP
80          :
81          :      OUTPUT:
82          :      (SP)      DATA FROM PRE-SELECTED SP SOURCE
83          :
84          :
85          :      SPLB::
86          :
87          :
88          :
89          :
90          :
91          :
92          :
93          :
94          :
95          :
96          :
97          :
98          :
99          :
100         :
101         :
102         :
103         :
104         :
105         :
106         :
107         :
108         :
109         :
110         :
111         :
112         :
113         :
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         :
168         :
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         :
224         :
225         :
226         :
227         :
228         :
229         :
230         :
231         :
232         :
233         :
234         :
235         :
236         :
237         :
238         :
239         :
240         :
241         :
242         :
243         :
244         :
245         :
246         :
247         :
248         :
249         :
250         :
251         :
252         :
253         :
254         :
255         :
256         :
257         :
258         :
259         :
260         :
261         :
262         :
263         :
264         :
265         :
266         :
267         :
268         :
269         :
270         :
271         :
272         :
273         :
274         :
275         :
276         :
277         :
278         :
279         :
280         :
281         :
282         :
283         :
284         :
285         :
286         :
287         :
288         :
289         :
290         :
291         :
292         :
293         :
294         :
295         :
296         :
297         :
298         :
299         :
300         :
301         :
302         :
303         :
304         :
305         :
306         :
307         :
308         :
309         :
310         :
311         :
312         :
313         :
314         :
315         :
316         :
317         :
318         :
319         :
320         :
321         :
322         :
323         :
324         :
325         :
326         :
327         :
328         :
329         :
330         :
331         :
332         :
333         :
334         :
335         :
336         :
337         :
338         :
339         :
340         :
341         :
342         :
343         :
344         :
345         :
346         :
347         :
348         :
349         :
350         :
351         :
352         :
353         :
354         :
355         :
356         :
357         :
358         :
359         :
360         :
361         :
362         :
363         :
364         :
365         :
366         :
367         :
368         :
369         :
370         :
371         :
372         :
373         :
374         :
375         :
376         :
377         :
378         :
379         :
380         :
381         :
382         :
383         :
384         :
385         :
386         :
387         :
388         :
389         :
390         :
391         :
392         :
393         :
394         :
395         :
396         :
397         :
398         :
399         :
400         :
401         :
402         :
403         :
404         :
405         :
406         :
407         :
408         :
409         :
410         :
411         :
412         :
413         :
414         :
415         :
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         :
472         :
473         :
474         :
475         :
476         :
477         :
478         :
479         :
480         :
481         :
482         :
483         :
484         :
485         :
486         :
487         :
488         :
489         :
490         :
491         :
492         :
493         :
494         :
495         :
496         :
497         :
498         :
499         :
500         :
501         :
502         :
503         :
504         :
505         :
506         :
507         :
508         :
509         :
510         :
511         :
512         :
513         :
514         :
515         :
516         :
517         :
518         :
519         :
520         :
521         :
522         :
523         :
524         :
525         :
526         :
527         :
528         :
529         :
530         :
531         :
532         :
533         :
534         :
535         :
536         :
537         :
538         :
539         :
540         :
541         :
542         :
543         :
544         :
545         :
546         :
547         :
548         :
549         :
550         :
551         :
552         :
553         :
554         :
555         :
556         :
557         :
558         :
559         :
560         :
561         :
562         :
563         :
564         :
565         :
566         :
567         :
568         :
569         :
570         :
571         :
572         :
573         :
574         :
575         :
576         :
577         :
578         :
579         :
580         :
581         :
582         :
583         :
584         :
585         :
586         :
587         :
588         :
589         :
590         :
591         :
592         :
593         :
594         :
595         :
596         :
597         :
598         :
599         :
600         :
601         :
602         :
603         :
604         :
605         :
606         :
607         :
608         :
609         :
610         :
611         :
612         :
613         :
614         :
615         :
616         :
617         :
618         :
619         :
620         :
621         :
622         :
623         :
624         :
625         :
626         :
627         :
628         :
629         :
630         :
631         :
632         :
633         :
634         :
635         :
636         :
637         :
638         :
639         :
640         :
641         :
642         :
643         :
644         :
645         :
646         :
647         :
648         :
649         :
650         :
651         :
652         :
653         :
654         :
655         :
656         :
657         :
658         :
659         :
660         :
661         :
662         :
663         :
664         :
665         :
666         :
667         :
668         :
669         :
670         :
671         :
672         :
673         :
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         :
730         :
731         :
732         :
733         :
734         :
735         :
736         :
737         :
738         :
739         :
740         :
741         :
742         :
743         :
744         :
745         :
746         :
747         :
748         :
749         :
750         :
751         :
752         :
753         :
754         :
755         :
756         :
757         :
758         :
759         :
760         :
761         :
762         :
763         :
764         :
765         :
766         :
767         :
768         :
769         :
770         :
771         :
772         :
773         :
774         :
775         :
776         :
777         :
778         :
779         :
780         :
781         :
782         :
783         :
784         :
785         :
786         :
787         :
788         :
789         :
790         :
791         :
792         :
793         :
794         :
795         :
796         :
797         :
798         :
799         :
800         :
801         :
802         :
803         :
804         :
805         :
806         :
807         :
808         :
809         :
810         :
811         :
812         :
813         :
814         :
815         :
816         :
817         :
818         :
819         :
820         :
821         :
822         :
823         :
824         :
825         :
826         :
827         :
828         :
829         :
830         :
831         :
832         :
833         :
834         :
835         :
836         :
837         :
838         :
839         :
840         :
841         :
842         :
843         :
844         :
845         :
846         :
847         :
848         :
849         :
850         :
851         :
852         :
853         :
854         :
855         :
856         :
857         :
858         :
859         :
860         :
861         :
862         :
863         :
864         :
865         :
866         :
867         :
868         :
869         :
870         :
871         :
872         :
873         :
874         :
875         :
876         :
877         :
878         :
879         :
880         :
881         :
882         :
883         :
884         :
885         :
886         :
887         :
888         :
889         :
890         :
891         :
892         :
893         :
894         :
895         :
896         :
897         :
898         :
899         :
900         :
901         :
902         :
903         :
904         :
905         :
906         :
907         :
908         :
909         :
910         :
911         :
912         :
913         :
914         :
915         :
916         :
917         :
918         :
919         :
920         :
921         :
922         :
923         :
924         :
925         :
926         :
927         :
928         :
929         :
930         :
931         :
932         :
933         :
934         :
935         :
936         :
937         :
938         :
939         :
940         :
941         :
942         :
943         :
944         :
945         :
946         :
947         :
948         :
949         :
950         :
951         :
952         :
953         :
954         :
955         :
956         :
957         :
958         :
959         :
960         :
961         :
962         :
963         :
964         :
965         :
966         :
967         :
968         :
969         :
970         :
971         :
972         :
973         :
974         :
975         :
976         :
977         :
978         :
979         :
980         :
981         :
982         :
983         :
984         :
985         :
986         :
987         :
988         :
989         :
990         :
991         :
992         :
993         :
994         :
995         :
996         :
997         :
998         :
999         :
1000        :

```

```

97      ;
98      ;
99      ;      SP CONTROL REGISTER LOADING
100     ;
101     ;      INPUT:
102     ;      2(SP) BIT SETTING FOR SP CONTROL REGISTER
103     ;
104     ;
105     000242      SPCR::
106     000242      016667      000002      176424      MOV      2(SP),DR$LBR      ;CONTROL BITS DESTINED FOR SP
107     000250      012746      001001      MOV      *(Q$LBD+Q$LBP),-(SP)      ;CLEAR DRIVE AND PULSE
108     000254      052716      000360      BIS      #Q$CSEL,(SP)      ;CLR SELECTION BITS
109     000260      012746      000120      MOV      #Q$SP,-(SP)      ;SELECT SP
110     000264      CALL      CSR1      ;
111     ;
112     000270      005046      CLR      -(SP)      ;CLEAR NOTHING
113     000272      012746      000001      MOV      #Q$LBP,-(SP)      ;SET PULSE
114     000276      CALL      CSR1      ;
115     ;
116     000302      012746      000121      MOV      *(Q$SP+Q$LBP),-(SP)      ;CLEAR CR SELECTION AND PULSE
117     000306      005046      CLR      -(SP)      ;SET NOTHING
118     000310      CALL      CSR1      ;
119     ;
120     000314      011666      000002      MOV      (SP),2(SP)      ;MOVE RETURN ADDRESS DOWN STACK
121     000320      005726      TST      (SP)+      ;POINT TO RETURN ADDRESS
122     000322      RETURN      ;
123     ;
124     000001      .END

```

SYMBOL TABLE

ALUCKE = 000000	BYTE42 = 000052	BYTE94 = 000176	Q\$SM = 100000
ALUOE = 004000	BYTE43 = 000053	BYTE95 = 000137	Q\$ATTN = 000100
A01 = 010000	BYTE44 = 000054	BYTE96 = 000140	Q\$BCL = 000001
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	Q\$CCCP = 000040
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	Q\$CHB = 000400
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	Q\$CHRL = 000200
BIT10 = 002000	BYTE48 = 000060	BYTVAL = 000144	Q\$CLR = 000040
BIT11 = 004000	BYTE49 = 000061	CBKALL = 001000	Q\$CNC = 030000
BIT12 = 010000	BYTE50 = 000062	CBKCLK = 000400	Q\$CP = 000060
BIT13 = 020000	BYTE51 = 000063	CNOBRE = 100000	Q\$CPCC = 000010
BIT14 = 040000	BYTE52 = 000064	CPCCEN = 010000	Q\$CP2 = 000260
BIT15 = 100000	BYTE53 = 000065	CPREAD = 040000	Q\$CSC = 010000
BIT2 = 000004	BYTE54 = 000066	CPWRTE = 020000	Q\$CSEL = 000360
BIT3 = 000010	BYTE55 = 000067	CSADRD = 000004	Q\$CSET = 000002
BIT4 = 000020	BYTE56 = 000070	CSEQCI = 100000	Q\$CSP = 020000
BIT5 = 000040	BYTE57 = 000071	CSDOE = 000040	Q\$DMA = 000001
BIT6 = 000100	BYTE58 = 000072	CSR1 = ***** GX	Q\$ENBK = 176370
BIT7 = 000200	BYTE59 = 000073	CSWRTE = 000100	Q\$ENDP = 020000
BIT8 = 000400	BYTE60 = 000074	DBR.RD = 000001	Q\$FAL = 004000
BIT9 = 001000	BYTE61 = 000075	DB\$CPP = 001457	Q\$FC = 000045
BYTE0 = 000000	BYTE62 = 000076	DB\$SPT = 000026	Q\$FO = 000044
BYTE1 = 000001	BYTE63 = 000077	DB\$TPC = 000023	Q\$FP = 000046
BYTE10 = 000012	BYTE64 = 000100	DISPGS = 100000	Q\$HBF = 000002
BYTE11 = 000013	BYTE65 = 000101	DMALUR = 000005	Q\$ICP = 000006
BYTE12 = 000014	BYTE66 = 000102	DMARRD = 000003	Q\$IHB = 000003
BYTE13 = 000015	BYTE67 = 000103	DMARWR = 000004	Q\$IHRL = 000002
BYTE14 = 000016	BYTE68 = 000104	EFN.3 = 000003	Q\$IMRP = 000007
BYTE15 = 000017	BYTE69 = 000105	ENBR = 010000	Q\$ILBD = 001000
BYTE16 = 000020	BYTE70 = 000106	LBSP = 000000RG 002	Q\$ILBDP = 001001
BYTE17 = 000021	BYTE71 = 000107	LBSSC = 000076RG 002	Q\$ILBF = 000001
BYTE18 = 000022	BYTE72 = 000110	LOC.EN = 000100	Q\$ILDOD = 000003
BYTE19 = 000023	BYTE73 = 000111	LOC.WB = 100000	Q\$LDMD = 000004
BYTE2 = 000002	BYTE74 = 000112	MAREN1 = 000001	Q\$LDPP = 002000
BYTE20 = 000024	BYTE75 = 000113	MAREN2 = 004000	Q\$LHP = 010000
BYTE21 = 000025	BYTE76 = 000114	MARLOD = 010000	Q\$MNC = 140000
BYTE22 = 000026	BYTE77 = 000115	MAROUT = 000002	Q\$MR = 000052
BYTE23 = 000027	BYTE78 = 000116	MAR.LO = 002000	Q\$MRP = 000040
BYTE24 = 000030	BYTE79 = 000117	MAR.OU = 000040	Q\$MRP2 = 000240
BYTE25 = 000031	BYTE80 = 000120	MBKALL = 001000	Q\$MSC = 040000
BYTE26 = 000032	BYTE81 = 000121	MBKCLK = 000400	Q\$MSET = 000004
BYTE27 = 000033	BYTE82 = 000122	MMADDR = 000100	Q\$MSP = 100000
BYTE28 = 000034	BYTE83 = 000123	MMLFT = 000002	Q\$NCLK = 176000
BYTE29 = 000035	BYTE84 = 000124	MMOE = 000004	Q\$PP = 000100
BYTE3 = 000003	BYTE85 = 000125	MMWRTE = 000010	Q\$PPSWJ = 000320
BYTE30 = 000036	BYTE86 = 000126	MNOBRE = 100000	Q\$PP2 = 000300
BYTE31 = 000037	BYTE87 = 000127	MREN1 = 000001	Q\$QHLT = 000013
BYTE32 = 000040	BYTE88 = 000130	MREN2 = 020000	Q\$QL = 000043
BYTE33 = 000041	BYTE89 = 000131	MSYN = 000040	Q\$QLA = 000053
BYTE34 = 000042	BYTE90 = 000132	N = 000144	Q\$QLB = 000054
BYTE35 = 000043	BYTE91 = 000133	PLB = 000010	Q\$QLR = 000001
BYTE36 = 000044	BYTE92 = 000134	PLC = 000020	Q\$QW = 000042
BYTE37 = 000045	BYTE93 = 000135	PLD = 000030	Q\$RDCD = 000005
BYTE38 = 000046		PLRWJ = 000200	Q\$RDMD = 000006
BYTE39 = 000047		PLR.EN = 000200	Q\$REBK = 001000
BYTE4 = 000004		QR#CR1 = 176420	Q\$RNC = 006000
BYTE40 = 000050		QR#CR2 = 176422	Q\$RSC = 004000
BYTE41 = 000051			Q\$RSET = 000010
			Q\$SP = 000120
			Q\$SP2 = 000340
			RGD.EN = 000200
			RGD.VA = 020000
			SEQ.CI = 000010
			SPCR 000242RG 002
			SPLB 000174RG 002
			S\$CLR = 000000
			S\$LA = 000001
			S\$OB = 000005
			S\$OR = 000006
			S\$QX = 000004
			S\$SR = 000007
			S\$S1 = 000010
			S\$S2 = 000014
			TD\$CTR = 176370
			TD\$CTW = 176360
			TD\$INL = 004000
			TD\$MEM = 000270
			TD\$OAR = 176344
			TD\$OTR = 176346
			TD\$ORD = 000274
			TD\$SIW = 176376
			TD\$STAR = 176372
			TD\$TAW = 176362
			TD\$TDR = 176374
			TD\$TDW = 176364
			T\$AD = 000020
			T\$BA = 000002
			T\$BD = 000010
			T\$BSQ = 100000
			T\$BT = 000020
			T\$BTAR = 000030
			T\$BTD = 002000
			T\$CD = 000100
			T\$CLK = 002000
			T\$DISK = 000200
			T\$DRD = 000004
			T\$EMEM = 010000
			T\$FSA = 000000
			T\$FSAB = 000004
			T\$FSAC = 000014
			T\$FSB2 = 000010
			T\$IB = 000026
			T\$IBAR = 000024
			T\$IBE = 020000
			T\$IBF = 040000
			T\$ICD = 000040
			T\$MODE = 004000
			T\$OB = 000036
			T\$OBE = 004000
			T\$OBF = 010000
			T\$OBRA = 000034
			T\$OBWA = 000032
			T\$ODTA = 100000
			T\$RDDO = 000200

T#RNB = .000040	WORD19 = .000046	WORD4 = .000010	WORD60 = .000170	WORD81 = .000242
T#RSET = .040000	WORD2 = .000004	WORD40 = .000120	WORD61 = .000172	WORD82 = .000244
T#SC = .000022	WORD20 = .000050	WORD41 = .000122	WORD62 = .000174	WORD83 = .000246
T#SCLK = .020000	WORD21 = .000052	WORD42 = .000124	WORD63 = .000176	WORD84 = .000250
T#SEG1 = .000000	WORD22 = .000054	WORD43 = .000126	WORD64 = .000200	WORD85 = .000252
T#SEG2 = .000001	WORD23 = .000056	WORD44 = .000130	WORD65 = .000202	WORD86 = .000254
T#SEG3 = .000002	WORD24 = .000060	WORD45 = .000132	WORD66 = .000204	WORD87 = .000256
T#SO = .000001	WORD25 = .000062	WORD46 = .000134	WORD67 = .000206	WORD88 = .000260
T#UBUS = .100000	WORD26 = .000064	WORD47 = .000136	WORD68 = .000210	WORD89 = .000262
T#1CLK = .000400	WORD27 = .000066	WORD48 = .000140	WORD69 = .000212	WORD9 = .000022
T#BBEN = .000020	WORD28 = .000070	WORD49 = .000142	WORD7 = .000016	WORD90 = .000264
UBD:IN = .000020	WORD29 = .000072	WORDS = .000012	WORD70 = .000214	WORD91 = .000266
WORD0 = .000000	WORD3 = .000006	WORDS0 = .000144	WORD71 = .000216	WORD92 = .000270
WORD1 = .000002	WORD30 = .000074	WORDS1 = .000146	WORD72 = .000220	WORD93 = .000272
WORD10 = .000024	WORD31 = .000076	WORDS2 = .000150	WORD73 = .000222	WORD94 = .000274
WORD11 = .000026	WORD32 = .000100	WORDS3 = .000152	WORD74 = .000224	WORD95 = .000276
WORD12 = .000030	WORD33 = .000102	WORDS4 = .000154	WORD75 = .000226	WORD96 = .000300
WORD13 = .000032	WORD34 = .000104	WORDS5 = .000156	WORD76 = .000230	WORD97 = .000302
WORD14 = .000034	WORD35 = .000106	WORDS6 = .000160	WORD77 = .000232	WORD98 = .000304
WORD15 = .000036	WORD36 = .000110	WORDS7 = .000162	WORD78 = .000234	WORD99 = .000306
WORD16 = .000040	WORD37 = .000112	WORDS8 = .000164	WORD79 = .000236	WRDVAL = .000310
WORD17 = .000042	WORD38 = .000114	WORDS9 = .000166	WORD8 = .000020	XTREAD = .001000
WORD18 = .000044	WORD39 = .000116	WORD5 = .000014	WORD80 = .000240	XTWTE = .000400

. ABS: 000000 000
000000 001
SPSUB: 000324 002
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 3112 WORDS (13 PAGES)
DYNAMIC MEMORY: 3860 WORDS (14 PAGES)
ELAPSED TIME: 00:00:42
SPSUB,SPSUB/SP=C20,1JIM,C20,1JSPSUB

27-MAR-80 16:15

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

TASK NAME : ...SMT
 PARTITION NAME : GEN
 IDENTIFICATION : 08
 TASK UIC : [7,5]
 TASK PRIORITY : 100
 STACK LIMITS : 040232 040531 000300 00192
 PRG XFR ADDRESS : 045262
 TASK ATTRIBUTES : AL,CP
 TOTAL ADDRESS WINDOWS : 3
 TASK IMAGE SIZE : 6528 WORDS
 TASK ADDRESS LIMITS : 040000 071337
 R-W DISK BLK LIMITS : 000042 000073 000032 00026

*** ROOT SEGMENT: SMT

R/W MEM LIMITS : 040000 071337 031340 13024
 DISK BLK LIMITS : 000042 000073 000032 00026

MEMORY ALLOCATION SYNOPSIS:

SECTION	TITLE	IDENT	FILE
. BLK : (RW, I, LCL, REL, CON)	040532-020366	08438	
	040532-013312-05834	SMT	SMT.OBJ:1
CPSUB : (RW, I, LCL, REL, CON)	061120-000502-00322		
	061120-000502-00322	CPSUB	CPSUB.OBJ:1
MRPSUB : (RW, I, LCL, REL, CON)	061622-000506-00326		
	061622-000506-00326	MRPSUB	MRPSUB.OBJ:1
PPSUB : (RW, I, LCL, REL, CON)	062330-000766-00502		
	062330-000766-00502	PPSUB	PPSUB.OBJ:1
SPSUB : (RW, I, LCL, REL, CON)	063316-000324-00212		
	063316-000324-00212	SPSUB	SPSUB.OBJ:1
SPTST : (RW, I, LCL, REL, CON)	063642-001016-00526		
	063642-001016-00526	SPTST	SPTST.OBJ:1
\$\$FSR1 : (RW, D, GBL, REL, OVR)	064660-001020-00528		
	064660-001020-00528	SMT	SMT.OBJ:1
\$\$FSR2 : (RW, D, GBL, REL, CON)	065700-000104-00068		
\$\$RESL : (RW, I, LCL, REL, CON)	066004-003332-01754		
\$\$RESM : (RW, I, LCL, REL, CON)	132000-007656-04014		

GLOBAL SYMBOLS:

BASE	040562-R	CPCRA	061540-R	ERW3	042422-R	LBCSC	061364-R	LMM	042426-R	PRINT	042703-R	TCSPU	064424-R
BINWD	040554-R	CPLB	061462-R	ERW4	042424-R	LBMRP	061774-R	MEMERR	050364-R	SELPG	062656-R	TSKTCB	040536-R
CDHIGH	041472-R	CSR1	050324-R	FIRST	001000	LBMSC	062072-R	MRPCR	062240-R	SEDCS	061120-R	T1SP	063740-R
CDLOW	041456-R	DATA1	040570-R	FVER	040576-R	LBPP	062330-R	MRPCRA	062246-R	SEONM	061622-R	T6SP	064062-R
CKDATA	042402-R	EFN.3	000003	GET	052416-R	LBPSC	062426-R	MRPLB	062170-R	SPCR	063560-R	T7SP	064212-R
CK2	042404-R	ERRADD	042412-R	INDNB	045224-R	LBSP	063316-R	DLDVEC	040540-R	SPLB	063512-R	VIRT	040572-R
CK3	042406-R	ERRCT	042414-R	INFDN	045064-R	LBSSC	063414-R	PPCR	062574-R	STAT	040544-R	WCOUNT	040602-R
CODE	040564-R	ERW1	042416-R	IO.WVB	011000	LCCOUNT	040600-R	PPLB	062524-R	STUFSP	063642-R	WRTCS	061206-R
CPCR	061532-R	ERW2	042420-R	LBCP	061266-R	LCS	042432-R	PREADD	042410-R	WRTMM	063710-R	WRTMM	063710-R

Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

SMT.TSK:15 MEMORY-ALLOCATION MAP.TKB: PAGE:2
SMT: 27-MAR-80 16 Approved For Release 2005/07/12 : CIA-RDP85-00514R000200030001-2

\$GEFI . 005174 \$ESTA . 004226 \$DIV . 007145 \$DRDSE . 017134 \$MUL . 007116 \$TKTCB . 004026

*** TASK-BUILDER-STATISTICS:

TOTAL-WORK-FILE-REFERENCES: 22237.
WORK-FILE-READS: 0.
WORK-FILE-WRITES: 0.
SIZE-OF-CORE-POOL: 6634. WORDS (25. PAGES)
SIZE-OF-WORK-FILE: 2816. WORDS (11. PAGES)
ELAPSED-TIME:00:00:19

```
1 .TITLE .CONVRT.
2 ;
3 ;
4 CONVERT FILES FOR LOADING INTO THE HQR.
5 ;
6 TASK BUILD --
7 ;
8 TKB
9 CONVRT,CONVRT=CONVRT.
10 /
11 ASG=SY0:3:4
12 ASG=T1:1:2
13 PAR=GEN:40000:40000 (THIS LINE FOR 11/04 ONLY)
14 //
15 ;
16 ;
17 CONVERT *DASL* MICROCODE LISTING FILES TO HQR.
18 LOADABLE MICROCODE (MRP AND CP)
19 ;
20 CONVERT EDITABLE DATA MEMORY FILES TO LOADABLE FILES
21 MRP DATA MEMORY
22 CP DATA MEMORY
23 QEX WINDOW MEMORY
24 QEX LOCATION MEMORY
25 FAL POINTER MEMORY
26 FAL COUNTER MEMORY
27 QLB REFERENCE PAGE
28 QLB PAGE 0
29 QLB PAGE 1
30 QLB PAGE 2
31 QEX SUCCESS BIT MEMORY
32 SUBDOC REFERENCE PAGE MEMORY
33 SLB PAGE MEMORY
34 SUBREAD MEMORY
35 SUBID MEMORY 1
36 SUBID MEMORY 2
37 ;
38 ;
39 CONVERT PROMPTS FOR COMMAND LINE INPUT IN THE FORM OF
40 A MEMORY MNEMONIC AND AN OPTIONAL FILE VERSION NUMBER.
41 CONVRT OPENS A FILE WHOSE NAME IS IN THE FORM INXX.DAT
42 WHERE XX IS REPLACED BY THE MEMORY MNEMONIC AND CREATES
43 AN OUTPUT FILE WHOSE NAME IS IN THE FORM LDXX.DAT WITH
44 THE SAME SUBSTITUTION FOR XX.
45 ;
46 THE FORMATS OF THE INPUT FILES AND THE OUTPUT FILES VARY.
47 ;
48 THE MICROCODE INPUT FILES ARE THE LISTING FILE OUTPUTS
49 OF THE DASL MICROCODE ASSEMBLER RENAMED TO INMM.DAT AND
50 INCS.DAT FOR THE MRP AND CP, RESPECTIVELY.
51 ;
52 EACH LOCATION IN MRP MICROPGM MEMORY CONSISTS OF TWO
53 WORDS, A LEFT WORD AND A RIGHT WORD. CONVRT READS INMM.DAT,
54 FINDS THE ASCII EQUIVALENTS OF THE HEX MICROCODE FIELDS, AND
55 CONVERTS ALL OF THE LEFT COLUMN (WRITING BLOCKS TO LDMM.DAT
56 AS THEY ARE FILLED. CONVRT THEN CLOSSES INMM.DAT, REOPENS IT,
57 RELOCATES THE ASCII HEX CODE, AND CONVERTS THE RIGHT COLUMN
58 (AGAIN WRITING BLOCKS TO LDMM.DAT) WHEN THE CONVERSION IS
```



```

58 ; COMPLETE, CONVRT WRITES THE NUMBER OF WORDS IN A COLUMN INTO
59 ; THE RESERVED FIRST WORD OF THE FIRST BLOCK OF LDMM.DAT.
60 ;
61 ; EACH LOCATION IN CP CONTROL STORE CONSISTS OF FOUR WORDS.
62 ; SECTIONS 'A', 'B', 'C', AND 'D'. CONVRT READS INCS.DAT, FINDS
63 ; THE ASCII EQUIVALENTS OF THE HEX MICROCODE FIELDS, AND CONVERTS
64 ; ALL OF SECTION 'A' (WRITING BLOCKS TO LDOS.DAT AS THEY ARE FILLED).
65 ; CONVRT THEN CLOSES INCS.DAT, REOPENS IT, AND REPEATS THE PROCESS FOR
66 ; SECTIONS 'B', 'C', AND 'D'. WHEN CONVERSION IS COMPLETE, CONVRT
67 ; WRITES THE NUMBER OF WORDS IN A COLUMN INTO THE RESERVED FIRST
68 ; WORD OF THE FIRST BLOCK OF LDOS.DAT.
69 ;
70 ; FOR BOTH MICRO CODE FILES CONVRT PERFORMS ADDRESS CHECKING.
71 ; THAT IS, IF AN ADDRESS (OR MORE THAN ONE ADDRESS) IS SKIPPED
72 ; IN THE MICROCODE (WHICH IS THE LEGITIMATE RESULT OF THE USE
73 ; OF THE DASL KEYWORD 'LOCATION'), CONVRT FILLS IN THE INTERVENING
74 ; ADDRESSES WITH ZEROS. THIS ALLOWS THE LOADER OR HOROLS TO DO A
75 ; STRAIGHT SEQUENTIAL LOAD.
76 ;
77 ; ALL OTHER MEMORIES ARE ONE WORD WIDE. THE INXX FILES CONTAIN
78 ; ADDRESSES AND DATA:
79 ;
80 ; 0000 FFFF.
81 ; 0001 AAAA.
82 ; 0007 BBBB.
83 ;
84 ; FOR ALL FRAME 2 MEMORIES, CONVRT WRITES BOTH THE ADDRESS AND
85 ; THE DATA TO THE OUTPUT FILE. THE LOADER THEN PRE-CLEARs THE
86 ; ENTIRE MEMORY BEFORE LOADING DATA ONLY AT THE ADDRESSES IT
87 ; ENCOUNTERS IN THE LDXX FILE. CONVRT PLACES A COUNT OF THE
88 ; NUMBER OF ADDR/DATA PAIRS IN THE FILE INTO THE RESERVED
89 ; FIRST WORD OF THE FIRST BLOCK.
90 ;
91 ; FOR THE FRAME 1 MEMORIES (MRP DATA MEMORY AND CP DATA MEMORY)
92 ; CONVRT WRITES DATA ONLY TO THE OUTPUT FILE. CONVRT ALSO PERFORMS
93 ; SKIPPED ADDRESS CHECKING, WRITING ZEROS INTO ALL GAPS. THE COUNT
94 ; OF THE NUMBER OF WORDS GOES INTO THE FIRST WORD OF THE FIRST
95 ; BLOCK OF THE OUTPUT FILE. THESE TWO FILES (LDMD.DAT AND LDOD.DAT)
96 ; ARE DIFFERENT FROM THE FRAME 2 FILES BECAUSE THEIR LOADING IS
97 ; DONE BY DMA.
98 ;
99 ; EXIT FROM THE PROGRAM TAKES PLACE IN RESPONSE TO THE
100 ; COMMAND 'EX' ENTERED FROM THE TERMINAL.
101 ;
102 ;
103 ; .MCALL QIOW$,QIO$,EXIT$,ABRT$,GCML$,GCMLB$,FORSZ$,CLEF$,
104 ; .MCALL FDBDF$,FDRC$,FDBK$,FDOP$,NMBLK$,OPEN$,GET$
105 ; .MCALL OPEN$,OPEN$,READ$,WRITE$,WTSE$,CLOSE$
106 ;
107 ;
108 ; LUN,TT = 1 ;LUN FOR TTD
109 ; EFN,1 = 1 ;EVENT FLAG FOR TTD
110 ; CMILUN = 2 ;LUN FOR GCML
111 ; INLUN = 3 ;LUN FOR INPUT FILES
112 ; OUTLUN = 4 ;LUN FOR OUTPUT FILES
113 ; ALL = 1 ;CONVERT ALL FILES
114 ;

```

115						
116						
117	000000	012446	106144	MYSELF:	.RAD50	/CONVRT/
118	000004	035175	050500	IMM:	.RAD50	/INMM/
119	000010	035175	014400	IMD:	.RAD50	/INMD/
120	000014	035163	073300	ICS:	.RAD50	/INCS/
121	000020	035163	014400	ICD:	.RAD50	/INCD/
122	000024	045655	050500	LMM:	.RAD50	/LDMM/
123	000030	045655	014400	LMD:	.RAD50	/LDMD/
124	000034	045643	073300	LCS:	.RAD50	/LDCS/
125	000040	045643	014400	LCD:	.RAD50	/LDCD/
126	000044	035201	107700	IOW:	.RAD50	/INQW/
127	000050	045661	107700	LOW:	.RAD50	/LDQW/
128	000054	035201	045400	IOL:	.RAD50	/INQL/
129	000060	045661	045400	LQL:	.RAD50	/LDQL/
130	000064	035166	062000	IFP:	.RAD50	/INFP/
131	000070	045646	062000	LFP:	.RAD50	/LDFF/
132	000074	035166	011300	IFC:	.RAD50	/INFC/
133	000100	045646	011300	LFC:	.RAD50	/LDFF/
134	000104	035201	070200	IQR:	.RAD50	/INQR/
135	000110	045661	070200	LQR:	.RAD50	/LDQR/
136	000114	035201	135600	I00:	.RAD50	/INQ0/
137	000120	045661	135600	L00:	.RAD50	/LDQ0/
138	000124	035201	140700	I01:	.RAD50	/INQ1/
139	000130	045661	140700	L01:	.RAD50	/LDQ1/
140	000134	035201	144000	I02:	.RAD50	/INQ2/
141	000140	045661	144000	L02:	.RAD50	/LDQ2/
142	000144	035203	022600	ISF:	.RAD50	/INSF/
143	000150	045663	022600	LSF:	.RAD50	/LDSF/
144	000154	035203	135600	IS0:	.RAD50	/INS0/
145	000160	045663	135600	LS0:	.RAD50	/LDS0/
146	000164	035203	140700	IS1:	.RAD50	/INS1/
147	000170	045663	140700	LS1:	.RAD50	/LDS1/
148	000174	035203	144000	IS2:	.RAD50	/INS2/
149	000200	045663	144000	LS2:	.RAD50	/LDS2/
150	000204	035203	070200	ISR:	.RAD50	/INSR/
151	000210	045663	070200	LSR:	.RAD50	/LDSR/
152	000214	035201	113000	I0X:	.RAD50	/INQX/
153	000220	045661	113000	L0X:	.RAD50	/LDQX/
154				:		
155	000224	000000		VIRT:	.WORD	0
156	000226	000001			.WORD	1
157	000230			STAT:	.BLKW	2
158	000234	000000		FVER:	.WORD	0
159	000236	000000		ERWORD:	.WORD	0
160	000240	000000		BINWD:	.WORD	0
161	000242	000000		SELECT:	.WORD	0
162	000244	000000		SCOUNT:	.WORD	0
163	000246	000000		UWORD:	.WORD	0
164	000250	000000		LCOUNT:	.WORD	0
165	000252	000000		CURADD:	.WORD	0
166	000254	000000		PREADD:	.WORD	0
167	000256	000000		CURDAT:	.WORD	0
168	000260	000000		ALLPT:	.WORD	0
169	000262	000000		ALLCT:	.WORD	0
170	000264			INLINE:		
171	000264			GMBUF:	.BLKW	88

:MRP-MICROPROGRAM-MEMORY-INPUT-FILE
 :MRP-DATA-MEMORY-INPUT-FILE
 :CP-CONTROL-STORE-INPUT-FILE
 :CP-DATA-MEMORY-INPUT-FILE
 :MRP-MICROPROGRAM-MEMORY-OUTPUT-FILE
 :MRP-DATA-MEMORY-OUTPUT-FILE
 :CP-CONTROL-STORE-OUTPUT-FILE
 :CP-DATA-MEMORY-OUTPUT-FILE
 :QEX-WINDOW-MEMORY-INPUT-FILE
 :QEX-WINDOW-MEMORY-OUTPUT-FILE
 :QEX-LOCATION-MEMORY-INPUT-FILE
 :QEX-LOCATION-MEMORY-OUTPUT-FILE
 :FAL-POINTER-MEMORY-INPUT-FILE
 :FAL-POINTER-MEMORY-OUTPUT-FILE
 :FAL-COUNTER-MEMORY-INPUT-FILE
 :FAL-COUNTER-MEMORY-OUTPUT-FILE
 :QLB-REF-PAGE-INPUT-FILE
 :QLB-REF-PAGE-OUTPUT-FILE
 :QLB-PAGE-0-INPUT-FILE
 :QLB-PAGE-0-OUTPUT-FILE
 :QLB-PAGE-1-INPUT-FILE
 :QLB-PAGE-1-OUTPUT-FILE
 :QLB-PAGE-2-INPUT-FILE
 :QLB-PAGE-2-OUTPUT-FILE
 :SLB-REF-PAGE-INPUT-FILE
 :SLB-REF-PAGE-OUTPUT-FILE
 :SLB-PAGE-INPUT-FILE
 :SLB-PAGE-OUTPUT-FILE
 :SIDMEM-1-INPUT-FILE
 :SIDMEM-1-OUTPUT-FILE
 :SIDMEM-2-INPUT-FILE
 :SIDMEM-2-OUTPUT-FILE
 :SUB-READ-INPUT-FILE
 :SUB-READ-OUTPUT-FILE
 :QEX-SUCCESS-BIT-INPUT-FILE
 :QEX-SUCCESS-BIT-OUTPUT-FILE
 :VIRTUAL-BLOCK-NUMBER
 :IO-STATUS
 :FILE-VERSION-NUMBER
 :INDEX-VALUE-FOR-ERROR-MESSAGE-TABLE
 :TARGET-FOR-NUMERIC-CONVERSIONS-FROM-ASCII
 :ALL-PURPOSE-FLAG
 :WORK-COUNTER-FOR-COLUMNS-PROCESSED
 :MICROWORD-COLUMN-COUNT
 :COUNTER-FOR-NUMBER-OF-ITEMS-IN-A-COLUMN
 :CURRENT-DATA-ADDRESS
 :PREVIOUS-DATA-ADDRESS
 :CURRENT-DATA
 :-> TABLE-OF-ALL-MEMORY-ROUTINE-ADDRESSES
 :NUMBER-OF-MEMORIES
 :INPUT-LINE
 :TABLE-OF-ALL-MEMORY-ROUTINE-ADDRESSES

```

172 000524 000000          GCMLN: .WORD 0          :COMMAND LINE LENGTH
173 000526 000000          GCMPNT: .WORD 0        :COMMAND LINE POINTER
174 000530          OUTLIN: .BLKW 256.    :OUTPUT BLOCK
175          ;
176          ;
177          ;
178 001530          ;
179          TRTBL:
180 001610 000 001 002 . .BYTE 0,1,2,3,4,5,6,7,8,9.
181 001631 001631 013 014 . = TRTBL+101
182 001631 012 . .BYTE 10,11,12,13,14,15.
183 001730 001730 . = TRTBL+200
184          ;
185          ;
186          ;
187 001730 060 061 062 TRTBL2: .ASCII /0123456789ABCDEF/
188          .EVEN
189          ;
190          ;
191          ;
192          ;
193          ;
194          ;
195          ;
196 001750          FTBL:
197 001750 101 114 . .ASCII /AL/          :CONVERT ALL INPUT FILES
198 001752 003726 115 . .WORD AL
199 001754 115 . .ASCII /MM/          :MICROPGM MEMORY
200 001756 004026 104 . .WORD MRPMM
201 001760 115 . .ASCII /MD/          :DATA MEMORY
202 001762 004074 123 . .WORD MRPHD
203 001764 103 . .ASCII /CS/          :CONTROL STORE
204 001766 004134 104 . .WORD CPCPS
205 001770 103 . .ASCII /CD/          :CP DATA MEMORY
206 001772 004202 127 . .WORD CPCD
207 001774 121 . .ASCII /QW/          :QEX WINDOW MEMORY
208 001776 004242 114 . .WORD QW
209 002000 121 114 . .ASCII /QL/          :QEX LOCATION MEMORY
210 002002 004302 120 . .WORD QL
211 002004 106 . .ASCII /FP/          :FAL POINTER MEMORY
212 002006 004342 103 . .WORD FP
213 002010 106 . .ASCII /FC/          :FAL COUNTER MEMORY
214 002012 004402 122 . .WORD FC
215 002014 121 . .ASCII /QR/          :QLB REF PAGE
216 002016 004442 060 . .WORD QR
217 002020 121 060 . .ASCII /Q0/          :QLB PAGE 0
218 002022 004502 061 . .WORD Q0
219 002024 121 . .ASCII /Q1/          :QLB PAGE 1
220 002026 004542 062 . .WORD Q1
221 002030 121 . .ASCII /Q2/          :QLB PAGE 2
222 002032 004602 130 . .WORD Q2
223 002034 121 . .ASCII /QX/          :QEX SUCCESS BIT MEMORY
224 002036 004642 122 . .WORD QX
225 002040 123 . .ASCII /SR/          :SUBREAD MEMORY
226 002042 004702 106 . .WORD SR
227 002044 123 . .ASCII /SF/          :SLB REFERENCE PAGE MEMORY
228 002046 004742 . .WORD SF

```

```

229 002050      123      060      .ASCII /S0/
230 002052      005002'   .WORD   S0           ;SLB PAGE MEMORY
231 002054      123      061      .ASCII /S1/
232 002056      005042'   .WORD   S1           ;SIDMEM 1
233 002060      123      062      .ASCII /S2/
234 002062      005110'   .WORD   S2           ;SIDMEM 2
235 002064      105      130      .ASCII /EX/         ;EXIT ROUTINES
236 002066      005156'   .WORD   EXIT
237          000024      FNUM:   = <.-FTBL/4>
238          ;
239          ;
240          ; ADDRESSES OF ALL CONVERSION ROUTINES EXCEPT THOSE FOR
241          ; MRP MICROPGM MEMORY AND CP CONTROL STORE
242 002070      ALLTBL:
243 002070      004074' 004202' 004242' .WORD   MRPMD,CPCD,QU,QL,FP,FC,QR,Q0,Q1,Q2
244          000012'   ALLNUM: = <.-ALLTBL>/2
245          ;
246          ;
247          ; PRINT LINE
248          ;
249 002114      015      012      .BYTE   15,12       ;PRECEDE PRINT LINE WITH CRLF
250 002116      PRINT:
251          .NLIST MEB
252          .REPT  78
253          .BYTE  40
254          .ENDR
255          ;
256          ;
257          ; TABLE OF MESSAGES
258          ;
259          ;
260 002234      000          .BYTE   0
261 002235      015      012      015 .BYTE   15,12,15,12
262 002241      105      130      111 .ASCII  /EXIT-HOR-CONVERSION PROGRAM/
263 002274      015      012      000 .BYTE   15,12,0
264 002277      015      012      015 .BYTE   15,12,15,12,15,12
265 002305      110      101      122 .ASCII  /HARDWARE QUERY RESOLVER CONVERSION PROGRAM/
266 002357      015      012      000 .BYTE   15,12,0
267 002362      015      012          .BYTE   15,12
268 002364      105      122      122 .ASCII  /ERROR ON WRITE/
269 002403      015      012          .BYTE   15,12
270 002405      111      116      126 .ASCII  /INVALID NUMERIC VALUE/
271 002433      015      012          .BYTE   15,12
272 002435      105      122      122 .ASCII  /ERROR ON READ/
273 002453      015      012          .BYTE   15,12
274 002455      111      116      126 .ASCII  /INVALID MEMORY MNEMONIC/
275 002505      015      012          .BYTE   15,12
276 002507      115      111      123 .ASCII  /MISSING OPERAND/
277 002527      015      012          .BYTE   15,12
278 002531      111      116      103 .ASCII  /INCORRECT CHARACTER COUNT/
279 002563      015      012          .BYTE   15,12
280 002565      123      105      114 .ASCII  /SELECT MEMORY OR EXIT/
281 002613      377          ASCII:  .BYTE   377
282          ;
283 002614      105      115      120 EMPTY: .ASCII  /EMPTY LINE IN INPUT FILE/
284          .EVEN
285          .LIST  REV

```

```
286          .NLIST- CND-
287          ;
288          ;
289          COMMAND-LINE MACRO-
290          ;
291          ;
292 002646 GCMBLK: GCMLB$ 2, .GCMBUF, CHILUN-
293          ;
294          INPUT-FILE- FDB-
295          ;
296 003154 INFDB:
297 003154   FDBDF$
298 003314   FDRC$A- ., INLINE, 160-
299 003314   FDOP$A- INLUN, ., INDNB-
300 003314 INDNB: NMBLK$ ., DAT
301          ;
302          OUTPUT-FILE- FDB-
303          ;
304 003352 OUTFDB:
305 003352   FDBDF$
306 003512   FDRC$A- FD, RWM-
307 003512   FDBK$A- OUTLIN, 512, ., ., STAT-
308 003512   FDOP$A- OUTLUN, ., OUTDNB
309 003512 OUTDNB: NMBLK$ ., DAT
310 003550   FSRSZ$ 1
```



```
369 003674 000727          BR      COM          :START OVER.
370          :
371          :      JUMP TO THE ROUTINE THAT GOVERNS THE COMMAND.
372          :
373 003676 116767 174336 177426 4#:  MOV#  BINWD,INDNB+N,FVER : INSERT FILE VERSION NUMBER.
374 003704 016767 174330 174322  MOV#  BINWD,FVER : SAVE VERSION NUMBER.
375 003712          :
376 003712 012767 000001 174326 5#:  MOV#  #1,UWORD : DEFAULT MEMORY SIZE = 1 WORD.
377 003720 012601          :      MOV#  (SP)+,R1 : LOAD ROUTINE ADDRESS.
378 003722 000171 000000          :      JMP#  @ (R1) : GO THERE.
```

```

380 ;
381 ;
382 ; CONVERT ALL INPUT FILES.
383 ;
384 ;
385 ; SET UP TO SEQUENCE THROUGH ALL CONVERSION ROUTINES.
386 ; NB. ALL CONVERSION ROUTINES RETURN TO THE TOP OF THE
387 ; COMMAND LOOP ('COM'). CONTROL WILL BE RE-ROUTED HERE
388 ; WITHOUT A PROMPT ONCE THE #ALL FLAG HAS BEEN SET.
389 ;
390 ; IF THE ORIGINAL COMMAND HAD A FILE VERSION NUMBER
391 ; FOLLOWING IT IN THE COMMAND LINE, THEN ALL INPUT
392 ; FILES WITH THAT VERSION NUMBER WILL BE CONVERTED.
393 ;
394 ; >AL 4
395 ;
396 003726 AL:
397 003726 052767 000001 174306 BIS #ALL,SELECT ;SET FLAG FOR CONVERTING ALL
398 003734 012767 002070 174316 MOV #ALLTBL,ALLPT ;INITIALIZE POINTER
399 003742 012767 000012 174312 MOV #ALLNUM,ALLCT ;INITIALIZE COUNT
400 ;
401 003750 ALL2:
402 003750 005767 174306 TST ALLCT ;FINISHED?
403 003754 001005 BNE 1$
404 003756 042767 000001 174256 BIC #ALL,SELECT ;CLEAR 'ALL' FLAG
405 003754 000167 177564 JMP COM ;RETURN TO MAIN LOOP
406 ;
407 003770 116767 174240 177334 1$: MOV FVER,INDNB+N.FVER ;INSERT FILE VERSION NUMBER
408 003776 012767 000001 174242 MOV #1,UWORD ;DEFAULT MEMORY SIZE = 1 WORD
409 004004 005367 174252 DEC ALLCT ;SUB FROM CONVERSION COUNT
410 004010 016701 174244 MOV ALLPT,R1 ;R1 -> CONVERSION ROUTINE ADDRESS
411 004014 062767 000002 174236 ADD #2,ALLPT ;BUMP FOR NEXT TIME
412 004022 000171 000000 JMP @R1 ;GO TO CONVERSION ROUTINE

```



```

414 ;
415 ;
416 ;
417 ;
418 ;
419 004026 ; MRPMM:
420 004026 016767 173752 177266 MOV. IMM,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
421 004034 016767 173746 177262 MOV. IMM+2,INDNB+N.FNAM+2.
422 004042 016767 173756 177450 MOV. LMM,OUTDNB+N.FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
423 004050 016767 173752 177444 MOV. LMM+2,OUTDNB+N.FNAM+2.
424 004056 012767 000002 174162 MOV. #2,UWORD. ;NUMBER OF 16-BIT SECTIONS IN UWORD.
425 004064 CALL. MICRO. ;CONVERT DASL LISTING FILE.
426 004070 000167 177460 JMP. COM.
427 ;
428 ;
429 ; MRP DATA MEMORY.
430 ;
431 ;
432 004074 ; MRPM:
433 004074 016767 173710 177220 MOV. IMD,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
434 004102 016767 173704 177214 MOV. IMD+2,INDNB+N.FNAM+2.
435 004110 016767 173714 177402 MOV. LMD,OUTDNB+N.FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
436 004116 016767 173710 177376 MOV. LMD+2,OUTDNB+N.FNAM+2.
437 004124 CALL. DMADAT. ;CONVERT EDITABLE DATA FILE.
438 004130 000167 177420 JMP. COM.
439 ;
440 ;
441 ; GP CONTROL STORE.
442 ;
443 ;
444 004134 ; CPCS:
445 004134 016767 173654 177160 MOV. ICS,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
446 004142 016767 173650 177154 MOV. ICS+2,INDNB+N.FNAM+2.
447 004150 016767 173660 177342 MOV. LCS,OUTDNB+N.FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
448 004156 016767 173654 177336 MOV. LCS+2,OUTDNB+N.FNAM+2.
449 004164 012767 000004 174054 MOV. #4,UWORD. ;NUMBER OF 16-BIT SECTIONS IN UWORD.
450 004172 CALL. MICRO. ;CONVERT DASL LISTING FILE.
451 004176 000167 177352 JMP. COM.
452 ;
453 ;
454 ; GP DATA MEMORY.
455 ;
456 ;
457 004202 ; CPCD:
458 004202 016767 173612 177112 MOV. ICD,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.
459 004210 016767 173606 177106 MOV. ICD+2,INDNB+N.FNAM+2.
460 004216 016767 173616 177274 MOV. LCD,OUTDNB+N.FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
461 004224 016767 173612 177270 MOV. LCD+2,OUTDNB+N.FNAM+2.
462 004232 CALL. DMADAT. ;CONVERT EDITABLE DATA FILE.
463 004236 000167 177312 JMP. COM.
464 ;
465 ;
466 ; QEX WINDOW MEMORY.
467 ;
468 ;
469 004242 ; QW:
470 004242 016767 173576 177052 MOV. IQW,INDNB+N.FNAM. ;PLACE FILE NAME INTO INPUT DNB.

```

471	004250	016767	173572	177046	MOV.	IQW+2, INDNB+N, FNAM.	
472	004256	016767	173566	177234	MOV.	LQW, OUTDNB+N, FNAM.	; PLACE FILE NAME INTO OUTPUT DNB.
473	004264	016767	173562	177230	MOV.	LQW+2, OUTDNB+N, FNAM+2.	
474	004272				CALL.	DATA	; CONVERT EDITABLE DATA FILE.
475	004276	000167	177252		JMP.	COM.	
476					:		
477					:		
478					:		
479					:		
480					:		
481	004302				:		
482	004302	016767	173546	177012	QL:		
483	004310	016767	173542	177006	MOV.	IQL, INDNB+N, FNAM.	; PLACE FILE NAME INTO INPUT DNB.
484	004316	016767	173536	177174	MOV.	IQL+2, INDNB+N, FNAM+2.	
485	004324	016767	173532	177170	MOV.	LQL, OUTDNB+N, FNAM.	; PLACE FILE NAME INTO OUTPUT DNB.
486	004332				MOV.	LQL+2, OUTDNB+N, FNAM+2.	
487	004336	000167	177212		CALL.	DATA	; CONVERT EDITABLE DATA FILE.
488					JMP.	COM.	
489					:		
490					:		
491					:		
492					:		
493	004342				FP:		
494	004342	016767	173516	176752	MOV.	IFP, INDNB+N, FNAM.	; PLACE FILE NAME INTO INPUT DNB.
495	004350	016767	173512	176746	MOV.	IFP+2, INDNB+N, FNAM+2.	
496	004356	016767	173506	177134	MOV.	LFP, OUTDNB+N, FNAM.	; PLACE FILE NAME INTO OUTPUT DNB.
497	004364	016767	173502	177130	MOV.	LFP+2, OUTDNB+N, FNAM+2.	
498	004372				CALL.	DATA	; CONVERT EDITABLE DATA FILE.
499	004376	000167	177152		JMP.	COM.	
500					:		
501					:		
502					:		
503					:		
504					:		
505	004402				FC:		
506	004402	016767	173466	176712	MOV.	IFC, INDNB+N, FNAM.	; PLACE FILE NAME INTO INPUT DNB.
507	004410	016767	173462	176706	MOV.	IFC+2, INDNB+N, FNAM+2.	
508	004416	016767	173456	177074	MOV.	LFC, OUTDNB+N, FNAM.	; PLACE FILE NAME INTO OUTPUT DNB.
509	004424	016767	173452	177070	MOV.	LFC+2, OUTDNB+N, FNAM+2.	
510	004432				CALL.	DATA	; CONVERT EDITABLE DATA FILE.
511	004436	000167	177112		JMP.	COM.	
512					:		
513					:		
514					:		
515					:		
516					:		
517	004442				QR:		
518	004442	016767	173436	176652	MOV.	IQR, INDNB+N, FNAM.	; PLACE FILE NAME INTO INPUT DNB.
519	004450	016767	173432	176646	MOV.	IQR+2, INDNB+N, FNAM+2.	
520	004456	016767	173426	177034	MOV.	LQR, OUTDNB+N, FNAM.	; PLACE FILE NAME INTO OUTPUT DNB.
521	004464	016767	173422	177030	MOV.	LQR+2, OUTDNB+N, FNAM+2.	
522	004472				CALL.	DATA	; CONVERT EDITABLE DATA FILE.
523	004476	000167	177052		JMP.	COM.	
524					:		
525					:		
526					:		
527					:		

```

528. ;
529 004502. ; Q0:
530 004502. 016767 173406 176612. MOV. I00, INDNB+N, FNAM. ;PLACE FILE NAME INTO INPUT DNB.
531 004510 016767 173402 176606 MOV. I00+2, INDNB+N, FNAM+2.
532 004516 016767 173376 176774 MOV. L00, OUTDNB+N, FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
533 004524 016767 173372 176770 MOV. L00+2, OUTDNB+N, FNAM+2.
534 004532. CALL. DATA ;CONVERT EDITABLE DATA FILE.
535 004536 000167 177012 JMP. COM.
536 ;
537 ;
538 ; QLB PAGE 1
539 ;
540 ;
541 004542. ; Q1:
542 004542. 016767 173356 176552. MOV. I01, INDNB+N, FNAM. ;PLACE FILE NAME INTO INPUT DNB.
543 004550 016767 173352 176546 MOV. I01+2, INDNB+N, FNAM+2.
544 004556 016767 173346 176734 MOV. L01, OUTDNB+N, FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
545 004564 016767 173342 176730 MOV. L01+2, OUTDNB+N, FNAM+2.
546 004572. CALL. DATA ;CONVERT EDITABLE DATA FILE.
547 004576 000167 176752 JMP. COM.
548 ;
549 ;
550 ; QLB PAGE 2.
551 ;
552 ;
553 004602. ; Q2:
554 004602. 016767 173326 176512. MOV. I02, INDNB+N, FNAM. ;PLACE FILE NAME INTO INPUT DNB.
555 004610 016767 173322 176506 MOV. I02+2, INDNB+N, FNAM+2.
556 004616 016767 173316 176674 MOV. L02, OUTDNB+N, FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
557 004624 016767 173312 176670 MOV. L02+2, OUTDNB+N, FNAM+2.
558 004632. CALL. DATA ;CONVERT EDITABLE DATA FILE.
559 004636 000167 176712 JMP. COM.
560 ;
561 ;
562 ; QEX SUCCESS BITS.
563 ;
564 ;
565 004642. ; QX:
566 004642. 016767 173346 176452. MOV. I0X, INDNB+N, FNAM. ;PLACE FILE NAME INTO INPUT DNB.
567 004650 016767 173342 176446 MOV. I0X+2, INDNB+N, FNAM+2.
568 004656 016767 173336 176634 MOV. L0X, OUTDNB+N, FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
569 004664 016767 173332 176630 MOV. L0X+2, OUTDNB+N, FNAM+2.
570 004672. CALL. DATA ;CONVERT EDITABLE DATA FILE.
571 004676 000167 176652 JMP. COM.
572 ;
573 ;
574 ; SUBREAD MEMORY.
575 ;
576 ;
577 004702. ; SR:
578 004702. 016767 173276 176412. MOV. ISR, INDNB+N, FNAM. ;PLACE FILE NAME INTO INPUT DNB.
579 004710 016767 173272 176406 MOV. ISR+2, INDNB+N, FNAM+2.
580 004716 016767 173266 176574 MOV. LSR, OUTDNB+N, FNAM. ;PLACE FILE NAME INTO OUTPUT DNB.
581 004724 016767 173262 176570 MOV. LSR+2, OUTDNB+N, FNAM+2.
582 004732. CALL. DATA ;CONVERT EDITABLE DATA FILE.
583 004736 000167 176612 JMP. COM.
584

```

```

585 ;
586 ; SLB REFERENCE PAGE
587 ;
588 ;
589 004742 SF:
590 004742 016767 173176 176352 MOV ISF,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
591 004750 016767 173172 176346 MOV ISF+2,INDNB+N,FNAM+2
592 004756 016767 173166 176534 MOV LSF,OUTDNB+N,FNAM ;PLACE FILE NAME INTO OUTPUT DNB
593 004764 016767 173162 176530 MOV LSF+2,OUTDNB+N,FNAM+2
594 004772 CALL DATA ;CONVERT EDITABLE DATA FILE
595 004776 000167 176552 JMP COM
596 ;
597 ;
598 ; SLB PAGE
599 ;
600 ;
601 005002 S0:
602 005002 016767 173146 176312 MOV IS0,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
603 005010 016767 173142 176306 MOV IS0+2,INDNB+N,FNAM+2
604 005016 016767 173136 176474 MOV LS0,OUTDNB+N,FNAM ;PLACE FILE NAME INTO OUTPUT DNB
605 005024 016767 173132 176470 MOV LS0+2,OUTDNB+N,FNAM+2
606 005032 CALL DATA ;CONVERT EDITABLE DATA FILE
607 005036 000167 176512 JMP COM
608 ;
609 ;
610 ; SIDMEM 1
611 ;
612 ;
613 005042 S1:
614 005042 016767 173116 176252 MOV IS1,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
615 005050 016767 173112 176246 MOV IS1+2,INDNB+N,FNAM+2
616 005056 016767 173106 176434 MOV LS1,OUTDNB+N,FNAM ;PLACE FILE NAME INTO OUTPUT DNB
617 005064 016767 173102 176430 MOV LS1+2,OUTDNB+N,FNAM+2
618 005072 012767 000003 173146 MOV #3,UWORD ;MEMORY SIZE = 3 WORDS
619 005100 CALL DATA ;CONVERT EDITABLE DATA FILE
620 005104 000167 176444 JMP COM
621 ;
622 ;
623 ; SIDMEM 2
624 ;
625 ;
626 005110 S2:
627 005110 016767 173060 176204 MOV IS2,INDNB+N,FNAM ;PLACE FILE NAME INTO INPUT DNB
628 005116 016767 173054 176200 MOV IS2+2,INDNB+N,FNAM+2
629 005124 016767 173050 176366 MOV LS2,OUTDNB+N,FNAM ;PLACE FILE NAME INTO OUTPUT DNB
630 005132 016767 173044 176362 MOV LS2+2,OUTDNB+N,FNAM+2
631 005140 012767 000003 173100 MOV #3,UWORD ;MEMORY SIZE = 3 WORDS
632 005146 CALL DATA ;CONVERT EDITABLE DATA FILE
633 005152 000167 176376 JMP COM
634 ;
635 ;
636 ; EXIT PROGRAM
637 ;
638 ;
639 005156 EXIT:
640 005156 CALL ENDTST ;END OF PROGRAM MESSAGE
641 005162 EXIT$

```

```

643 ;
644 ;
645 ; CONVERT DASL LISTING FILE INTO HQR LOADABLE FORM.
646 ;
647 ;
648 ; READ FILE AND LOOK FOR MICROCODE LOCATION COUNTER.
649 ; THEN CONVERT MICROCODE TO LOADABLE BINARY FORM.
650 ; CONVERT BY 16-BIT COLUMNS.
651 ;
652 ; INPUT:
653 ; FILE NAME BLOCK OF FDB FILLED IN (BOTH INPUT AND OUTPUT)
654 ; UWORD. NUMBER OF COLUMNS.
655 ;
656 ; OUTPUT:
657 ; FILE LDMM.DAT OR LDOS.DAT.
658 ;
659 ;
660 ; SET UP FOR CONVERSION. RESERVE FIRST WORD OF FIRST BLOCK.
661 ; OPEN INPUT FILE AND OUTPUT FILE.
662 ;
663 005170 ; MICRO:
664 005170 012767 000001 173030 MOV #1,VIRT+2. ; INITIALIZE BLOCK COUNTER.
665 005176 012767 000001 173040 MOV #1,SCOUNT. ; INIT COLUMN COUNTER
666 005204 005067 173040 CLR LCOUNT. ; CLEAR # WORDS IN COLUMN COUNTER.
667 005210 012767 177777 173036 MOV #*1,PREADD. ; SET "PREVIOUS ADDRESS" = -1
668 005216 012704 000377 MOV #<256,-1>,R4 ; ACCOUNT FOR FIRST (OPEN) WORD
669 005222 012705 000532 MOV #*OUTLN+2,R5 ; LEAVE FIRST WORD OPEN.
670 ;
671 005226 ; OPEN$R #INFDB.
672 ;
673 005244 ; OPEN$W #OUTFDB.
674 ;
675 ; LOOP TO FIND THE FIRST WORD OF ASCII HEX MICROCODE IN
676 ; THE INPUT FILE. LOOK FOR THE ADDRESS FIELD.
677 ;
678 005262 ; MTOP:
679 005262 GET$ #INFDB. ; READ DASL FILE.
680 005272 103006 BCC 1$ ; RECORD READ. CONTINUE.
681 005274 CALL ERR5 ; AN ERROR HERE ALSO.
682 005300 CALL ERRIN.
683 005304 000167 000476 JMP MEX. ; CLOSE FILES AND EXIT.
684 005310 122767 000060 172750 1$: CMPB #*0,INLINE+2. ; LOOK FOR MICROCODE START IN COL 2.
685 005316 001361 BNE MTOP ; GET NEXT RECORD.
686 ;
687 ; CONVERSION LOOP.
688 ; CONVERT ONE LINE OF MICROCODE AT A TIME.
689 ; FIRST MAKE SURE THAT THE LINE TRULY CONTAINS MICROCODE.
690 ;
691 005320 016767 175654 173176 MCLOOP: MOV INFDB+*,NRBD,GCMLN. ; GET LENGTH OF RECORD READ.
692 005326 022767 000002 173170 CMP #2,GCMLN. ; IS RECORD AT LEAST 2.
693 005334 002102 BGE MNEXT. ; NO. GET NEXT RECORD.
694 005336 162767 000002 173160 SUB #2,GCMLN. ; SUB FOR FIRST 2 CHARS.
695 005344 012767 000266 173154 MOV #*INLINE+2,GCMPNT. ; POINT PAST FIRST 2 CHARS.
696 005352 CALL FIND ; FIND FIRST NON-BLANK (LOC. COUNTER)
697 005356 103471 BCS MNEXT. ; NO "A" HERE. GET NEXT RECORD.
698 005360 022701 000266 CMP #*INLINE+2,R1 ; WAS LOC OF FIRST CHAR RETURNED?
699 005364 001066 BNE MNEXT. ; NO. GET NEXT RECORD

```

```

700 ;
701 ;
702 ; FOUND LOCATION COUNTER.
703 ; FIND AND CONVERT THE CORRECT MICROCODE WORD (DEPENDING
704 ; UPON THE COLUMN COUNT).
705 005366 CALL PACK ; CONVERT LOCATION COUNTER
706 005372 103463 BCS MNEXT ; NOT HEX, GET NEXT RECORD.
707 005374 016767 172640 172650 MOV BINWD,CURADD ; SAVE ADDRESS.
708 005402 016702 172636 MOV SCOUNT,R2 ; LOAD CURRENT COLUMN COUNT.
709 005406 4$: CALL FIND ; FIND MICROCODE WORD
710 005412 103453 BCS MNEXT ; NOTHING THERE; GET NEXT RECORD.
711 005414 005302 DEC R2 ; DETERMINE WHICH 16-BIT SECTION TO CONVERT.
712 005416 001373 BNE 4$
713 ;
714 ; FOUND CORRECT MICROCODE WORD. CONVERT IT.
715 ; MAKE UP ANY GAPS IN THE LOCATION COUNTER BY WRITING
716 ; ZEROS TO THE OUTPUT FILE.
717 ;
718 005420 CALL PACK ; CONVERT ASCII-HEX TO BINARY.
719 005424 103446 BCS MNEXT ; INVALID HEX VALUE, READ NEXT RECORD.
720 005426 005267 172616 CATCH1: INC LCOUNT ; COUNT NUMBER OF WORDS IN A COLUMN.
721 005432 005267 172616 INC PREADD ; BUMP PREVIOUS ADDRESS.
722 005436 026767 172612 172606 CMP PREADD,CURADD ; ANY GAPS?
723 005444 001417 BEQ 5$ ; NO.
724 005446 005025 CLR (R5)+ ; CLEAR WORD IN OUTPUT BUFFER.
725 005450 005304 DEC R4 ; OUTPUT BUFFER FULL?
726 005452 001365 BNE CATCH1 ; NO, CONTINUE.
727 005454 CALL WRITE ; WRITE OUTPUT FILE.
728 005460 103002 BCC 50$ ; OK, CONTINUE.
729 005462 000167 000320 JMP MEX ; ERROR.
730 ;
731 005466 005267 172534 50$: INC VIRT+2 ; WRITE NEXT BLOCK.
732 005472 012705 000530 MOV #OUTLN,R5 ; R5 -> OUTPUT BUFFER
733 005476 012704 000400 MOV #256,,R4 ; R4 = WORD COUNT.
734 005502 000751 BR CATCH1 ; MAKE UP GAPS.
735 ;
736 ; NO MORE GAPS
737 ; BINWD = MICROCODE WORD FOR OUTPUT.
738 ;
739 005504 016725 172530 5$: MOV BINWD,(R5)+ ; MOVE WORD TO OUTPUT BUFFER.
740 005510 005304 DEC R4 ; OUTPUT BUFFER FULL?
741 005512 001013 BNE MNEXT ; NO, CONTINUE.
742 ;
743 005514 CALL WRITE ; WRITE A BLOCK.
744 005520 103002 BCC 6$ ; WRITE SUCCESSFUL.
745 005522 000167 000260 JMP MEX.
746 005526 005267 172474 6$: INC VIRT+2 ; NEXT TIME WRITE NEXT BLOCK.
747 005532 012705 000530 MOV #OUTLN,R5 ; POINT TO START OF OUTPUT BUFFER.
748 005536 012704 000400 MOV #256,,R4 ; NUMBER OF WORDS IN BUFFER.
749 ;
750 ; GET NEXT RECORD.
751 ;
752 005542 MNEXT: GET# #INFD#
753 005552 103262 BCC MLOOP ; LOOK FOR NEXT HEX VALUE.
754 005554 122760 177766 000052 CMPB #-10,,F.ERR(R0) ; END OF FILE.
755 005562 001406 BEQ NXC0L ; YES, OPEN FOR NEXT COLUMN.
756 005564 CALL ERRS ; ELSE ERROR ON READ.

```

```

757 005570          CALL.  ERRIN.          ;PRINT NAME OF FILE.
758 005574 000167 000206 JMP.   MEX.           ;EXIT.
759                ;
760                ; TEST FOR ALL COLUMNS FINISHED.
761                ; IF NOT, CLOSE AND REOPEN INPUT FILE FOR NEXT COLUMN.
762                ;
763 005600 026767 172442 172436 NXCLOC: CMP.   UWORD,SCOUNT.  ;FINISHED ALL COLUMNS.
764 005606 001424          BEQ.   1$.           ;YES, WRITE LAST BLOCK.
765 005610 005267 172430          INC.   SCOUNT.      ;NEXT COLUMN.
766 005614 005067 172430          CLR.   LCOUNT.    ;COUNT NUMBER OF WORDS IN 1 COLUMN ONLY.
767 005620 012767 177777 172426 MOV.   *-1,PREADD.  ;SET PREVIOUS ADDRESS = -1
768 005626          CLOSE$. #INFDB.
769                ;
770 005636          OPEN$. #INFDB.
771 005654 000167 177402 JMP.   MTOP.
772                ;
773                ; WRITE LAST RECORD.
774                ;
775 005660 022704 000400          1$:  CMP.   #256.,R4   ;CURRENT RECORD EMPTY.
776 005664 001405          BEQ.   2$.           ;YES, DO NOT WRITE.
777 005666          CALL.  WRITE.      ;WRITE A BLOCK.
778 005672 103002.          BCC.   2$.
779 005674 000167 000106          JMP.   MEX.
780                ;
781                ;
782                ; ALL COLUMNS FINISHED.
783                ; REWRITE FIRST BLOCK.
784                ;
785                ;
786 005700          2$:  CLOSE$. #OUTFDB.  ;CLOSE OUTPUT FILE.
787                ;
788                ; READ FIRST BLOCK.
789                ;
790 005710          OPEN$. #OUTFDB.    ;OPEN FILE
791                ;
792 005726 012767 000001 172272. MOV.   #1,VIRT+2.  ;SET BLOCK NUMBER = 1
793 005734          CALL.  READ.       ;READ BLOCK 1
794 005740 103002.          BCC.   3$.           ;READ SUCCESSFUL.
795 005742 000167 000040          JMP.   MEX.
796                ;
797 005746          3$:  CLOSE$. #OUTFDB.  ;CLOSE AGAIN.
798                ;
799 005756          OPEN$. #OUTFDB.    ;OPEN FOR MODIFY.
800                ;
801 005774 016767 172250 172526 MOV.   LCOUNT,OUTLIN. ;PUT 'WORDS IN COLUMN' COUNT IN FIRST WORD.
802 006002.          CALL.  WRITE.    ;WRITE BLOCK 1
803                ;
804                ;
805                ; EXIT FOR GOOD.
806                ;
807                ;
808 006006          MEX:  CLOSE$. #OUTFDB.
809                ;
810 006016          CLOSE$. #INFDB.
811 006026 105067 175300          CLR$.  INDB+N.FVER.  ;CLEAR FILE VERSION NUMBER.
812 006032 000207          RTS.   PC.

```

```

814 ;
815 ;
816 ; CONVERT DATA MEMORY FILES TO HDR LOADABLE FORM.
817 ;
818 ;
819 ;
820 ; DATA FILES ARE IN THE FORM 'NNNN NNNN' WHERE THE FIRST
821 ; SET OF N'S IS THE LOCATION AND THE SECOND SET IS THE
822 ; DATA.
823 ;
824 ; INPUT:
825 ; INPUT AND OUTPUT FILE DATA NAME BLOCKS INITIALIZED.
826 ;
827 ; OUTPUT:
828 ; FILE WITH A NAME IN THE FORM LDXX.DAT.
829 ;
830 ;
831 ; SET UP FOR CONVERSION. RESERVE FIRST WORD OF FIRST BLOCK.
832 ; OPEN INPUT FILE AND OUTPUT FILE.
833 ;
834 ; DATA:
835 ;
836 ;
837 ;
838 ;
839 ;
840 ;
841 ;
842 ;
843 ;
844 ;
845 ;
846 ;
847 ;
848 ;
849 ;
850 ;
851 ;
852 ;
853 ;
854 ;
855 ;
856 ;
857 ;
858 ;
859 ;
860 ;
861 ;
862 ;
863 ;
864 ;
865 ;
866 ;
867 ;
868 ;
869 ;
870 ;

```

833 006034					MOV	#1,VIRT+2		; INITIALIZE BLOCK COUNTER
834 006034	012767	000001	172164		CLR	LCOUNT		; CLEAR ITEM COUNTER
835 006042	005067	172202			MOV	#<256,-2>,R4		; ACCOUNT FOR 2 EMPTY WORDS
836 006046	012704	000376			MOV	#OUTLIN+4,R5		; POINT PAST FIRST 2 WORDS
837 006052	012705	000534						
839 006056					OPEN#R	#INFDB		
841 006074					OPEN#W	#OUTFDB		
845 006112					READ	RECORD	OF	INPUT FILE
846 006112					DTOP:			
847 006122	103012				GET\$	#INFDB		; READ INPUT FILE
848 006124	122760	177766	000052		BCC	10\$; RECORD READ, CONTINUE
849 006132	001474				CMPB	#->10,,F,ERR(R0)		; END OF FILE
850 006134					BEQ	DLAST		; YES, WRITE LAST OUTPUT BUFFER
851 006140					CALL	ERRS		; AN ERROR HERE ALSO
852 006144	000167	000300			CALL	ERRIN		; PRINT NAME OF FILE
853					JMP	DEX		; CLOSE FILES AND EXIT
854 006150	016767	175024	172346	10\$:	MOV	INFDB+F,NRBD,GCMLN		; SET LENGTH OF RECORD READ
855 006156	012767	000264	172342		MOV	#INLINE,GCMPNT		; SAVE ADDRESS OF RECORD
856 006164	016767	172056	172052		MOV	UWORD,SCOUNT		; GET WORD COUNT OF MEMORY
857 006172	005267	172052			INC	LCOUNT		; INCREMENT NUMBER OF ITEMS
859								
860								
861 006176					CONVERSION LOOP:			
862 006176					DLOOP:			
863 006202	103010				CALL	FIND		; FIND FIELD VALUE
864 006204					BCC	10\$; OK, CONTINUE
865 006210					CALL	ERR2		; MISSING OPERAND (EMPTY LINE)
866 006214					CALL	ERRDA		; TELL THAT LINE WAS EMPTY
867 006220	000167	000224			CALL	ERRIN		; TELL WHICH FILE
868					JMP	DEX		
869								
870					CONVERT VALUE:			


```

871 006224          1$:  CALL  PACK          : CONVERT FIELD
872 006230 103010  BCC  2$          : OK, CONTINUE
873 006236          CALL  ERR6
874 006236          CALL  ERRDA          : PRINT OUT LINE IN ERROR
875 006242          CALL  ERRIN          : TELL WHICH FILE
876 006246 000167 000176 JMP  DEX
877          ;
878          ; MOVE VALUE TO OUTPUT RECORD
879          ;
880 006252          2$:
881 006252 016725 171762 MOV  BINWD, (R5)+ : MOVE WORD TO OUTPUT BUFFER
882 006256 005304      DEC  R4          : OUTPUT BUFFER FULL?
883 006260 001013      BNE  DNEXT        : NO, CONTINUE
884 006262          CALL  WRITE          : WRITE OUTPUT FILE
885 006266 103002      BCC  5$          : WRITE SUCCESSFUL
886 006270 000167 000154 JMP  DEX
887          ;
888 006274 005267 171726 5$:  INC  VIRT+2        : NEXT TIME WRITE NEXT BLOCK
889 006300 012705 000530* MOV  #OUTLIN, R5   : POINT TO START OF OUTPUT BUFFER
890 006304 012704 000400 MOV  #256, R4      : NUMBER OF WORDS IN BUFFER
891          ;
892          ; GET NEXT FIELD
893          ;
894 006310          DNEXT:
895 006310 005767 171730 TST  SCOUNT        : ANY MORE FIELDS LEFT?
896 006314 001676      BEQ  DTOP          : IF NOT, GET NEXT INPUT RECORD
897 006316 005367 171722 DEC  SCOUNT        : REDUCE FIELD COUNT
898 006322 000725      BR   DLOOP          : LOCATE NEXT FIELD
899          ;
900          ; WRITE LAST RECORD
901          ;
902 006324          DLAST:
903 006324 022704 000400 CMP  #256, R4      : IS OUTPUT BUFFER EMPTY?
904 006330 001405      BEQ  2$          : YES, NO LAST RECORD TO WRITE
905 006332          CALL  WRITE          : WRITE LAST RECORD
906 006336 103002      BCC  2$          : WRITE SUCCESSFUL
907 006340 000167 000104 JMP  DEX
908          ;
909          ; REWRITE FIRST BLOCK OF FILE
910          ;
911 006344          2$:  CLOSE$ #OUTFDB
912          ;
913          ; READ FIRST BLOCK
914          ;
915 006354          OPEN$R #OUTFDB          : OPEN FILE
916          ;
917 006372 012767 000001 171626 MOV  #1, VIRT+2   : SET BLOCK NUMBER
918 006400          CALL  READ          : READ BLOCK 1
919 006404 103001      BCC  6$          : READ BLOCK 1
920 006406 000420      BR   DEX          :
921          ;
922 006410          6$:  CLOSE$ #OUTFDB
923          ;
924 006420          OPEN$M #OUTFDB          : OPEN FOR APPEND
925          ;
926 006436 016767 171606 172066 MOV  LCOUNT, OUTLIN+2 : PUT TOTAL DATA WORDS IN SECOND WORD
927 006444          CALL  WRITE          : WRITE BLOCK

```

928.	:			
929	:			
930	:		EXIT FOR GOOD.	
931	:			
932 006450	DEX:	CLOSE\$	#OUTFDB.	
933	:			
934 006460		CLOSE\$	#INFDB.	
935 006470 105067 174636		CLRB.	INDNB+N.FVER.	: CLEAR FILE VERSION NUMBER.
936 006474 000207		RTS.	PC.	

```

938 ;
939 ;
940 ; CONVERT DATA MEMORY FILES INTO DMA LOADABLE FORM
941 ;
942 ;
943 ; DATA FILES ARE IN THE FORM 'NNNN NNNN' WHERE THE FIRST
944 ; SET OF N'S IS THE LOCATION AND THE SECOND SET IS THE
945 ; DATA
946 ;
947 ; DIFFERENCES BETWEEN THIS ROUTINE AND THE ROUTINE 'DATA'
948 ; ARE NOTED IN THE PROGRAM INTRODUCTORY NOTES.
949 ;
950 ;
951 ; SET UP FOR CONVERSION. RESERVE FIRST WORD OF FIRST BLOCK
952 ; OPEN INPUT FILE AND OUTPUT FILE.
953 ;
954 006476 DMADAT:
955 006476 012767 000001 171522 MOV #1,VIRT+2 ; INITIALIZE BLOCK COUNTER
956 006504 005067 171540 CLR LCOUNT ; CLEAR ITEM COUNTER
957 006510 012767 177777 171536 MOV #-1,PREADD ; START PREVIOUS ADDRESS AT -1
958 006516 012704 000377 MOV #<256,-1>,R4 ; ACCOUNT FOR 1 EMPTY WORDS
959 006522 012705 000532 MOV #OUTLIN+2,R5 ; POINT PAST FIRST WORD
960 ;
961 006526 OPEN$R #INFDB ;
962 ;
963 006544 OPEN$W #OUTFDB ;
964 ;
965 ; READ FIRST RECORD OF INPUT FILE.
966 ;
967 006562 DMATOP:
968 006562 GET$ #INFDB ; READ DASL FILE
969 006572 103006 BCC DMLOOP ; RECORD READ, CONTINUE
970 006574 CALL ERR5 ; AN ERROR HERE ALSO
971 006600 CALL ERRIN ; PRINT NAME OF FILE
972 006604 000167 000456 JMP DMEX ; CLOSE FILES AND EXIT
973 ;
974 ;
975 ; CONVERSION LOOP
976 ; LOCATE ADDRESS IN INPUT RECORD
977 006610 016767 174364 171706 DMLOOP: MOV INFDB+F,NRBD,GCMLN ; GET LENGTH OF RECORD READ
978 006616 012767 000264 171702 MOV #INLINE,GCMPNT ; SAVE ADDRESS OF RECORD
979 ;
980 006624 CALL FIND ; FIND LOCATION VALUE
981 006630 103010 BCC 1$ ; OK, CONTINUE
982 006632 CALL ERR2 ; MISSING OPERAND (EMPTY LINE)
983 006636 CALL ERROA ; TELL THAT LINE WAS EMPTY
984 006642 CALL ERRIN ; TELL WHICH FILE
985 006646 000167 000414 JMP DMEX
986 ;
987 ;
988 ; CONVERT ADDRESS
989 006652 1$: CALL PACK ; CONVERT ADDRESS
990 006656 103010 BCC 2$ ; OK, CONTINUE
991 006660 CALL ERR6 ;
992 006664 CALL ERROA ;
993 006670 CALL ERRIN ;
994 006674 000167 000366 JMP DMEX ; TELL WHICH FILE

```

```

995 ;
996 ;
997 ; SAVE ADDRESS (IN CASE OF CATCH-UP, SEE BELOW).
998 ; LOCATE DATA IN INPUT RECORD.
999 ;
1000 006700 016767 171334 171344 2#: MOV BINWD,CURADD ;MOVE ADDRESS.
1001 006706 CALL FIND ;FIND DATA VALUE.
1002 006712 103010 BCC 3# ;OK, CONTINUE.
1003 006720 CALL ERR2 ;MISSING OPERAND.
1004 006724 CALL ERRDA ;PRINT OUT LINE IN ERROR.
1005 006730 000167 000332 CALL ERRIN ;TELL WHICH FILE.
1006 JMP DMEX
1007 ;
1008 ; CONVERT DATA
1009 ;
1010 006734 3#: CALL PACK ;CONVERT ASCII-HEX TO BINARY.
1011 006740 103010 BCC 4# ;OK, CONTINUE.
1012 006742 CALL ERR6 ;INVALID NUMERIC VALUE.
1013 006752 CALL ERRDA ;PRINT OUT LINE IN ERROR.
1014 006756 000167 000304 CALL ERRIN ;TELL WHICH FILE.
1015 JMP DMEX
1016 ;
1017 ; SAVE DATA, MAKE UP ANY GAPS IN ADDRESS.
1018 ; EG. IF THE RECORDS ARE:
1019 ;
1020 ; 0000 FFFF.
1021 ; 0006 BBBB.
1022 ;
1023 ; AND THE SECOND RECORD HAS JUST BEEN READ,
1024 ; THE VALUE 0006 HAS BEEN SAVED IN CURADD AND THE BBBB
1025 ; WILL BE SAVED IN CURDAT. CONVRT PROCEEDS TO WRITE
1026 ; 0000 TO THE OUTPUT RECORD FOR THE ADDRESSES 0000
1027 ; 0002, 0003, 0004, AND 0005.
1028 ;
1029 006762 016767 171252 171266 4#: MOV BINWD,CURDAT ;SAVE DATA WORD.
1030 006770 005267 171254 CATCH: INC LCOUNT ;COUNT NUMBER OF WORDS.
1031 007000 026767 171250 171244 INC PREADD ;BUMP PREVIOUS ADDRESS.
1032 007006 001417 CMP PREADD,CURADD ;ANY GAPS?
1033 007010 005025 BEQ 2# ;NO.
1034 007012 005304 CLR (R5)+ ;CLEAR WORD IN OUTPUT BUFFER.
1035 007014 001365 DEC R4 ;OUTPUT BUFFER FULL?
1036 007016 BNE CATCH ;NO, CONTINUE.
1037 007022 103002 BCC WRITE ;WRITE OUTPUT FILE.
1038 007024 000167 000236 BCC 1# ;WRITE SUCCESSFUL.
1039 JMP DMEX
1040 ;
1041 007030 005267 171172 1#: INC VIRT+2 ;NEXT TIME WRITE NEXT BLOCK.
1042 007034 012705 000530 MOV #OUTLIN,R5 ;POINT TO START OF OUTPUT BUFFER.
1043 007040 012704 000400 MOV #256,,R4 ;NUMBER OF WORDS IN BUFFER.
1044 BR CATCH ;PLAY CATCH-UP.
1045 ;
1046 ; NO GAPS: MOVE DATA WORD TO OUTPUT RECORD.
1047 007046 016725 171204 2#: MOV CURDAT,(R5)+ ;MOVE WORD TO OUTPUT BUFFER.
1048 007052 005304 DEC R4 ;OUTPUT BUFFER FULL?
1049 007054 001013 BNE DMNEXT ;NO, CONTINUE.
1050 007056 CALL WRITE ;WRITE OUTPUT FILE.
1051 007062 103002 BCC 3# ;WRITE SUCCESSFUL.

```

```

1052 007064 000167 000176          JMP      DMEX
1053                                ;
1054 007070 005267 171132          3$: INC      VIRT+2          ;NEXT TIME WRITE NEXT BLOCK
1055 007074 012705 000530          MOV      #OUTLIN,R5        ;POINT TO START OF OUTPUT BUFFER
1056 007100 012704 000400          MOV      #256,R4          ;NUMBER OF WORDS IN BUFFER
1057                                ;
1058                                ;
1059                                ;
1060 007104                                ;
1061 007114 103235                                ;
1062 007116 122760 177766 000052  DMNEXT: GET$    #INFDB
1063 007124 001406                                BCC     DMLOOP            ;PROCESS NEXT RECORD
1064 007126                                CMPB    #-10,F.ERR(R0)    ;END OF FILE
1065 007132                                BEQ     4$                ;YES, WRITE LAST OUTPUT BUFFER
1066 007136 000167 000124          CALL    ERR5              ;ELSE ERROR ON READ
1067                                CALL    ERRIN             ;PRINT NAME OF FILE
1068                                JMP     DMEX
1069                                ;
1070                                ;
1071                                ;
1072 007142 022704 000400          4$: CMP      #256,R4        ;IS OUTPUT BUFFER EMPTY
1073 007146 001405                                BEQ     5$                ;YES, NO LAST RECORD TO WRITE
1074 007150                                CALL    WRITE             ;WRITE LAST RECORD
1075 007154 103002                                BCC     5$                ;WRITE SUCCESSFUL
1076 007156 000167 000104          JMP     DMEX
1077                                ;
1078 007162                                ;
1079                                ;
1080                                ;
1081                                ;
1082 007172                                ;
1083                                ;
1084 007210 012767 000001 171010          5$: MOV      #1,VIRT+2      ;SET BLOCK NUMBER = -1
1085 007216                                CALL    READ              ;READ BLOCK 1
1086 007222 103001                                BCC     6$                ;
1087 007224 000420                                BR      DMEX
1088                                ;
1089 007226                                ;
1090                                ;
1091 007236                                ;
1092                                ;
1093 007254 016767 170770 171246          6$: CLOSE$ #OUTFDB        ;OPEN FILE
1094 007262                                OPEN$R  #OUTFDB          ;OPEN FOR MODIFY
1095                                ;
1096                                ;
1097                                ;
1098                                ;
1099                                ;
1100 007266                                ;
1101                                ;
1102 007306 105067 174020          DMEX: CLOSE$ #OUTFDB
1103 007312 000207                                CLRB    INDB+H,FVER      ;CLEAR FILE VERSION NUMBER
                                RTS      PC

```

```

1105 ;
1106 ;
1107 ; READ-VIRTUAL BLOCK-1
1108 ;
1109 ; FILE-NAME-BLOCK-PRE-INITIALIZED
1110 ;
1111 ; OUTPUT:
1112 ; C-BIT-CLEAR - GOOD READ
1113 ; C-BIT-SET - ERROR-ON-READ
1114 ;
1115 ;
1116 007314 READ:
1117 007314 READ$ #OUTFDB,..#VIRT,#EFN.1,#STAT
1118 007362 103005 BCC 1$
1119 007364 CALL ERR5 ;ERROR-ON-READ
1120 007370 CALL ERRIN ;PRINT-NAME-OF-FILE
1121 007374 000424 BR 3$
1122 ;
1123 007376 1$: WTSE$S #EFN.1
1124 ;
1125 007410 CLEF$S #EFN.1
1126 007422 105767 170602 TSTB STAT
1127 007426 003005 BGT 2$
1128 007430 CALL ERR5 ;GOOD-COMPLETION
1129 007434 CALL ERRIN ;ERROR-ON-READ
1130 007440 000402 BR 3$ ;PRINT-NAME-OF-FILE
1131 ;
1132 007442 000241 2$: CLC
1133 007444 000401 BR 4$
1134 007446 000261 3$: SEC
1135 007450 4$: RETURN

```

```

1137      ;
1138      ;
1139      ;      WRITE VIRTUAL BLOCK
1140      ;
1141      ;      FILE NAME BLOCK PRE-INITIALIZED
1142      ;
1143      ;
1144      ;      OUTPUT:
1145      ;      C-BIT CLEAR      - GOOD WRITE
1146      ;      C-BIT SET       - ERROR ON WRITE
1147      ;
1148      ;      WRITE:
1149      ;      WRITE$  #OUTFDB...#VIRT.#EFN.1.#STAT
1150      ;      BCC     1$
1151      ;      CALL    ERR7                      ;ERROR ON WRITE
1152      ;      CALL    ERRROUT                  ;PRINT NAME OF FILE
1153      ;      BR      3$
1154      ;
1155      ;      1$:  WTSE$S  #EFN.1
1156      ;
1157      ;      CLEF$S  #EFN.1
1158      ;      TSTB   STAT
1159      ;      BGT    2$                      ;GOOD COMPLETION
1160      ;      CALL    ERR7                      ;ERROR ON WRITE
1161      ;      CALL    ERRROUT                  ;PRINT NAME OF FILE
1162      ;      BR      3$
1163      ;
1164      ;      2$:  CLC
1165      ;      BR      4$
1166      ;      3$:  SEC
1167      ;      4$:  RETURN

```

```

1169
1170
1171
1172
1173
1174 007610
1175 007610 012705 002116*
1176 007614 016701 173360
1177 007620 001005
1178 007622 012704 002614*
1179 007626 112425
1180 007630 001376
1181 007632 000405
1182
1183 007634 012704 000264*
1184 007640 112425
1185 007642 005301
1186 007644 001375
1187
1188 007646
1189 007652
1190
1191
1192
1193
1194
1195
1196 007654
1197 007654 012700 002116*
1198 007660 016701 173436
1199 007664
1200 007670 016701 173430
1201 007674
1202 007700 000412
1203
1204 007702
1205 007702 012700 002116*
1206 007706 016701 173606
1207 007712
1208 007716 016701 173600
1209 007722
1210
1211 007726
1212 007732

```

```

;
;
; PRINT INPUT RECORD IN WHICH THERE IS AN ERROR.
;
ERRDA:
MOV #PRINT,R5 ;POINT TO PRINT LINE
MOV INFDB+F,HRBD,R1 ;LOAD LENGTH OF INPUT LINE
BNE 2$ ;AT LEAST THERE WAS SOMETHING
MOV #EMPTY,R4 ;POINT TO MESSAGE
1$: MOVB (R4)+,(R5)+ ;NOTHING IN THE LINE
BNE 1$
BR 4$ ;PRINT LINE
;
2$: MOV #INLINE,R4 ;POINT TO INPUT LINE
3$: MOVB (R4)+,(R5)+
DEC R1
BNE 3$
;
4$: CALL CONSOL ;PRINT LINE IN ERROR
RETURN
;
;
; CONVERT NAME OF FILE IN ERROR FROM RAD-50 TO ASCII AND PRINT.
;
ERRIN:
MOV #PRINT,R0 ;R0 -> PRINT LINE
MOV INDNB+N,FNAM,R1 ;R1 = RAD-50 WORD
CALL %C5TA ;CONVERT FIRST WORD
MOV INDNB+N,FNAM+2,R1 ;SECOND WORD
CALL %C5TA
BR ERRX
;
ERROUT:
MOV #PRINT,R0 ;R0 -> PRINT LINE
MOV OUTDNB+N,FNAM,R1 ;R1 = RAD-50 WORD
CALL %C5TA ;CONVERT FIRST WORD
MOV OUTDNB+N,FNAM+2,R1 ;SECOND WORD
CALL %C5TA
;
ERRX: CALL CONSOL ;NOW PRINT LINE
RETURN

```



```

1214 ;
1215 ;
1216 ; SCAN A TABLE FOR A VALID COMMAND/MNEMONIC.
1217 ;
1218 ; INPUT:
1219 R0 = NUMBER OF ENTRIES IN COMMAND TABLE.
1220 R1 -> CHAR STRING IN GCML COMMAND LINE.
1221 R2 -> TOP OF COMMAND TABLE.
1222 ;
1223 ; OUTPUT:
1224 R1 -> ROUTINE THAT GOVERNS THE COMMAND (IF MATCH WAS MADE)
1225 R1 -> CHAR STRING IN COMMAND LINE (IF NO MATCH WAS MADE)
1226 R0 = RELATIVE POSITION OF MATCHED ENTRY IN TABLE.
1227 ;
1228 ;
1229 007734 SCAN:
1230 007734 010346 MOV R3, -(SP) ;SAVE R3
1231 007736 010046 MOV R0, -(SP) ;SAVE # ENTRIES
1232 007740 010146 MOV R1, -(SP) ;SAVE POINTER TO BEGINNING OF STRING
1233 ;
1234 007742 011601 FNOUT1: MOV (SP), R1 ;POINT TO NON-BLANK IN COMMAND LINE
1235 007744 012703 000002 MOV #2, R3 ;NUMBER OF CHARS IN NON-BLANK FIELD
1236 007750 122122 FNIN1: CMPB (R1)+, (R2)+ ;DOES COMMAND LINE MATCH TABLE ENTRY
1237 007752 001003 BNE FNOUT2 ;NO, TRY NEXT TABLE ENTRY
1238 007754 005303 DEC R3 ;SUB FROM LOOP COUNT
1239 007756 001374 BNE FNIN1
1240 007760 000411 BR FNMTCH ;COMMAND FOUND IN TABLE
1241 007762 060302 FNOUT2: ADD R3, R2 ;ADD # UNCOMPARED CHARS TO POINTER
1242 007764 005202 INC R2 ;THEN ADJUST TO NEXT TABLE ENTRY
1243 007766 005300 DEC R0 ;SUB FROM OUTER LOOP COUNT
1244 007770 001364 BNE FNOUT1 ;TRY AGAIN
1245 007772 012601 MOV (SP)+, R1 ;RESTORE POINTER TO COMMAND LINE
1246 007774 012600 MOV (SP)+, R0 ;RESTORE R0
1247 007776 012603 MOV (SP)+, R3 ;RESTORE R3
1248 010000 000261 SEC ;COMMAND NOT IN TABLE
1249 010002 RETURN
1250 ;
1251 010004 010201 FNMTCH: MOV R2, R1 ;POINT R1 AT RTN ADDR IN TABLE
1252 010006 062706 000002 ADD #2, SP ;POINT TO INCOMING R0 ON STACK
1253 010012 012602 MOV (SP)+, R2 ;GET TOTAL # TABLE ENTRIES
1254 010014 160002 SUB R0, R2 ;GET POSITION OF MATCHED ENTRY
1255 010016 010200 MOV R2, R0 ;PUT IN R0 FOR RETURN
1256 010020 012603 MOV (SP)+, R3 ;RESTORE R3
1257 010022 000241 CLC
1258 010024 RETURN

```


1372 010314		CALL	\$DIV		
1373 010320 005302		DEC	R2		:REDUCE FACTOR
1374 010322 001352		BNE	HLOOP		:SUB FROM LOOP COUNT
1375 010324 000403		BR	PCLCX		
1376					
1377 010326 012766 177777 000014		PSECDX:	MOV	#-1,12,(SP)	
1378 010334		PCLCX:	RESTOR	R0,R1,R2,R3,R4,R5	:SET COND CODE INDICATOR
1379					
1380 010350 005726			TST	(SP)+	
1381 010352 002402			BLT	1\$:GET COND CODE INDICATOR
1382 010354 000241			CLC		:DO SEC
1383 010356 000401			BR	PACKX	
1384 010360 000261		1\$:	SEC		:AND RETURN
1385 010362		PACKX:	RETURN		


```

1420 ;
1421 ;
1422 ; WRITE A LINE TO TT0
1423 ;
1424 ;
1425 010426 CONSOL:
1426 010426 SAVE R0,R1
1427 ;
1428 010432 012700 000120 MOV #0,R0 ;PRINT BUFFER BYTE COUNT
1429 010436 012701 002234 MOV #PRINT+70,R1 ;POINT PAST END OF BUFFER
1430 010442 122741 000040 1$: CMPB #40,-(R1) ;LOOK FOR A NON-BLANK
1431 010446 001003 BNE 2$ ;OK, WRITE LINE
1432 010450 005300 DEC R0 ;DEC CHAR COUNT
1433 010452 001373 BNE 1$
1434 010454 000440 BR ABEND2 ;NO NON-BLANKS?
1435 ;
1436 010456 2$: QIOW$ #TO,UWB,#LUN,TT,#EFN,1, #STAT, <#PRINT-2,R0>,ABEND2
1437 ;
1438 ;
1439 010534 012701 002116 MOV #PRINT,R1 ;POINT TO STRING
1440 010540 112721 000040 4$: MOVB #40,(R1)+ ;CLEAR LINE TO BLANKS
1441 010544 005300 DEC R0 ;DEC LOOP COUNT
1442 010546 001374 BNE 4$
1443 ;
1444 010550 RESTOR R0,R1
1445 010554 RETURN
1446 ;
1447 010556 ABEND2:: ABRT$ #MYSELF

```


ABEND = 011052R	BYTE32 = 000040	BYTE84 = 000124	EMPTY = 002614R	F.FTYP = 000116
ABEND2 = 010556RG	BYTE33 = 000041	BYTE85 = 000125	ENBR = 010000	F.FVER = 000120
AL = 003726R	BYTE34 = 000042	BYTE86 = 000126	ENDTST = 010610R	F.HIBK = 000004
ALL = 000001	BYTE35 = 000043	BYTE87 = 000127	ERRDA = 007610R	F.LUN = 000042
ALLCT = 000262R	BYTE36 = 000044	BYTE88 = 000130	ERRIN = 007654R	F.MBCT = 000054
ALLNUM = 000012	BYTE37 = 000045	BYTE89 = 000131	ERROUT = 007702R	F.MBC1 = 000055
ALLPT = 000260R	BYTE38 = 000046	BYTE9 = 000011	ERRX = 007726R	F.MBFG = 000056
ALLTBL = 0002070R	BYTE39 = 000047	BYTE90 = 000132	ERR1 = 010644R	F.NRBD = 000024
ALL2 = 003750R	BYTE4 = 000004	BYTE91 = 000133	ERR2 = 010640R	F.NREC = 000030
ALUCKE = 040000	BYTE40 = 000050	BYTE92 = 000134	ERR3 = 010634R	F.QVBS = 000030
ALUOE = 004000	BYTE41 = 000051	BYTE93 = 000135	ERR5 = 010630R	F.RACC = 000016
ASC12 = 002613R	BYTE42 = 000052	BYTE94 = 000136	ERR6 = 010624R	F.RATT = 000001
A01 = 010000	BYTE43 = 000053	BYTE95 = 000137	ERR7 = 010620R	F.RCNM = 000034
BINWD = 000240R	BYTE44 = 000054	BYTE96 = 000140	ERWORD = 000236R	F.RCTL = 000017
BITVAL = 000000	BYTE45 = 000055	BYTE97 = 000141	EXIT = 005156R	F.RSIZ = 000002
BIT0 = 000001	BYTE46 = 000056	BYTE98 = 000142	FC = 004402R	F.RTYP = 000000
BIT1 = 000002	BYTE47 = 000057	BYTE99 = 000143	FD.CCL = ***** GX	F.SEQN = 000100
BIT10 = 0002000	BYTE48 = 000060	BYTVAL = 000144	FD.REC = ***** GX	F.SPDV = 000072
BIT11 = 004000	BYTE49 = 000061	CATCH = 006770R	FD.RUM = ***** GX	F.SPUN = 000074
BIT12 = 010000	BYTE5 = 000005	CATCH1 = 005426R	FD.TTY = ***** GX	F.STBK = 000036
BIT13 = 020000	BYTE50 = 000062	CBKALL = 001000	FIND = 010026R	F.UNIT = 000136
BIT14 = 040000	BYTE51 = 000063	CBKCLK = 000400	FNIN1 = 007750R	F.URBD = 000020
BIT15 = 100000	BYTE52 = 000064	CMILUN = 000002	FNMTCH = 010004R	F.VBN = 000064
BIT2 = 000004	BYTE53 = 000065	CHOBRE = 100000	FNOUT1 = 007742R	F.VBSZ = 000060
BIT3 = 000010	BYTE54 = 000066	COM = 003554R	FNOUT2 = 007762R	GCMBLK = 002646R
BIT4 = 000020	BYTE55 = 000067	CONSOL = 010426R	FNUM = 000024	GCMBUF = 000264R
BIT5 = 000040	BYTE56 = 000070	CPCCEN = 010000	FO.MFY = ***** GX	GCMLEN = 000524R
BIT6 = 0000100	BYTE57 = 000071	CPCD = 004202R	FO.RD = ***** GX	GCMFNT = 000526R
BIT7 = 000200	BYTE58 = 000072	CPCS = 004134R	FO.URT = ***** GX	GE.BIF = 17775
BIT8 = 000400	BYTE59 = 000073	CPREAD = 040000	FP = 004342R	GE.CLO = 000004
BIT9 = 001000	BYTE6 = 000006	CPWRTE = 020000	FSECX = 010136R	GE.COM = 000001
BYTE0 = 000000	BYTE60 = 000074	CSADRD = 000004	FTBL = 001750R	GE.CON = 000020
BYTE1 = 000001	BYTE61 = 000075	CSEQCI = 100000	FVER = 000234R	GE.EOF = 177766
BYTE10 = 000012	BYTE62 = 000076	CSOE = 000040	F.ACTL = 000076	GE.IND = 000002
BYTE11 = 000013	BYTE63 = 000077	CSWRTE = 000100	F.ALCC = 000040	GE.LOR = 17777
BYTE12 = 000014	BYTE64 = 000100	CUPADD = 000252R	F.BBFS = 000062	GE.LC = 000010
BYTE13 = 000015	BYTE65 = 000101	CURDAT = 000256R	F.BDB = 000070	GE.MDE = 177774
BYTE14 = 000016	BYTE66 = 000102	DATA = 006034R	F.BGBC = 000057	GE.OPR = 177776
BYTE15 = 000017	BYTE67 = 000103	DBR.RD = 000001	F.BKDN = 000026	GE.RBG = 177730
BYTE16 = 000020	BYTE68 = 000104	DB*CPP = 001457	F.BKDS = 000020	GE.SIZ = 000040
BYTE17 = 000021	BYTE69 = 000105	DB*SPT = 000026	F.BKEF = 000050	GE.CMLD = 000146
BYTE18 = 000022	BYTE7 = 000007	DB*TPC = 000023	F.BKPI = 000051	G.DPRM = 000160
BYTE19 = 000023	BYTE70 = 000106	DEX = 006450R	F.BKST = 000024	G.ERR = 000140
BYTE2 = 000002	BYTE71 = 000107	DISPGS = 100000	F.BKVB = 000064	G.ISIZ = 000020
BYTE20 = 000024	BYTE72 = 000110	DLAST = 006324R	F.CHR = 000075	G.LPDL = 000060
BYTE21 = 000025	BYTE73 = 000111	DLOOP = 006176R	F.CNTG = 000034	G.MODE = 000141
BYTE22 = 000026	BYTE74 = 000112	DMAADR = 000005	F.DFNB = 000046	G.PSDS = 000142
BYTE23 = 000027	BYTE75 = 000113	DMADAT = 006476R	F.DSPT = 000044	G.SIZ = 000224
BYTE24 = 000030	BYTE76 = 000114	DMARRD = 000003	F.DVNM = 000134	HLOOP = 010250R
BYTE25 = 000031	BYTE77 = 000115	DMARUR = 000004	F.EFBK = 000010	ICD = 000020R
BYTE26 = 000032	BYTE78 = 000116	DMATOP = 006562R	F.EFN = 000050	ICS = 000014R
BYTE27 = 000033	BYTE79 = 000117	DMEX = 007266R	F.EQBB = 000032	IFC = 000074R
BYTE28 = 000034	BYTE8 = 000010	DMLOOP = 006610R	F.ERR = 000052	IFP = 000064R
BYTE29 = 000035	BYTE80 = 000120	DMNEXT = 007104R	F.FACC = 000043	IMD = 000010R
BYTE3 = 000003	BYTE81 = 000121	DNEXT = 006310R	F.FFBY = 000014	IMH = 000004R
BYTE30 = 000036	BYTE82 = 000122	DTOP = 006112R	F.FNAM = 000110	INDNB = 003314R
BYTE31 = 000037	BYTE83 = 000123	EFN.1 = 000001	F.FNB = 000102	INFDB = 003154R

INLINE = 000264R	MRPMD = 004074R	Q#CSP = 020000	SF = 004742R	T#IBE = 020000
INLUN = 000003	MRPMM = 004026R	Q#DMA = 000001	SR = 004702R	T#IBF = 040000
ID:WVB = ***** GX	MSYN = 000040	Q#ENBK = 040000	START = 003550R	T#ICD = 000040
IQL = 000054R	MTOP = 005262R	Q#ENDP = 020000	STAT = 000230R	T#MODE = 004000
IQR = 000104R	MYSELF = 000000R	Q#FAL = 004000	S#CLR = 000000	T#OB = 000036
IQU = 000044R	N = 000144	Q#FC = 000045	S#LA = 000001	T#OBE = 004000
IQX = 000214R	NEST = 000001	Q#FO = 000044	S#QB = 000005	T#ODF = 010000
IQQ = 000114R	NESTOP = 010650R	Q#FP = 000046	S#QR = 000006	T#OBRA = 000034
IQ1 = 000124R	NXTCOL = 005600R	Q#HBF = 000002	S#QX = 000004	T#OBWA = 000032
IQ2 = 000134R	N.DID = 000024	Q#HCP = 000006	S#SR = 000007	T#OUTA = 100000
ISF = 000144R	N.DVNM = 000032	Q#IHB = 000003	S#S1 = 000010	T#RBD0 = 000200
ISR = 000204R	N.FID = 000000	Q#IHR = 000002	S#S2 = 000014	T#RNB = 000040
IS0 = 000154R	N.FNAM = 000006	Q#IMRP = 000007	S.BFHD = 000020	T#RSET = 040000
IS1 = 000164R	N.FTYP = 000014	Q#LBD = 001000	S.FATT = 000016	T#SC = 000022
IS2 = 000174R	N.FYER = 000016	Q#LBDP = 001001	S.FDB = 000140	T#SCLK = 020000
LCD = 000040R	N.NEXT = 000022	Q#LBP = 000001	S.FHAM = 000006	T#SEG1 = 000000
LCCOUNT = 000250R	N.STAT = 000020	Q#LDCD = 000003	S.FNB = 000036	T#SEG2 = 000001
LCS = 000034R	N.UNIT = 000034	Q#LDMD = 000004	S.FNBW = 000017	T#SEG3 = 000002
LFC = 000100R	OUTDNB = 003512R	Q#LDPP = 002000	S.FNTY = 000004	T#SO = 000001
LFP = 000070R	OUTFDB = 003352R	Q#LHP = 010000	S.FTYF = 000002	T#UBUS = 100000
LMD = 000030R	OUTLIN = 000530R	Q#MNC = 140000	S.NFEN = 000020	T#1CLK = 000400
LMM = 000024R	OUTLUN = 000004	Q#MR = 000052	S0 = 005002R	T#BEN = 000020
LOC.EN = 000100	OUT1 = 010614R	Q#MRP = 000040	S1 = 005042R	UBD.IN = 000020
LOC.WA = 040000	PACK = 010144R	Q#MRP2 = 000240	S2 = 005110R	WORD = 000246R
LOC.WB = 100000	PACK0 = 010364R	Q#MSC = 040000	TD#CTR = 176370	VIRT = 000224R
LQL = 000060R	PACKX = 010362R	Q#MSET = 000004	TD#CTW = 176360	WORD0 = 000000
LQR = 000110R	PAR\$\$\$ = 000027	Q#MSL = 100000	TD#INL = 004000	WORD1 = 000002
LQW = 000050R	PCLCX = 010334R	Q#NCLK = 176000	TD#MEM = 000270	WORD10 = 000024
LQX = 000220R	PLB = 000010	Q#PP = 000100	TD#OR = 176344	WORD11 = 000026
LQ0 = 000120R	PLC = 000020	Q#PPSW = 000320	TD#OTR = 176346	WORD12 = 000030
LQ1 = 000130R	PLD = 000030	Q#PP2 = 000300	TD#ORD = 000274	WORD13 = 000032
LQ2 = 000140R	PLRWR = 000200	Q#QHLT = 000013	TD#SW = 176376	WORD14 = 000034
LSF = 000150R	PLR.EN = 000200	Q#QL = 000043	TD#STAR = 176372	WORD15 = 000036
LSR = 000210R	POX = 010422R	Q#QLA = 000053	TD#TAW = 176362	WORD16 = 000040
LS0 = 000160R	PREADD = 000254R	Q#QLB = 000054	TD#TDR = 176374	WORD17 = 000042
LS1 = 000170R	PRINT = 002116R	Q#QLR = 000001	TD#TDW = 176364	WORD18 = 000044
LS2 = 000200R	PSECC = 010326R	Q#QW = 000042	TRTBL = 001530R	WORD19 = 000046
LUN.TT = 000001	QL = 004302R	Q#RDCD = 000005	TRTBL2 = 001730R	WORD2 = 000004
MAREN1 = 000001	QR = 004442R	Q#RDMD = 000006	TTX = 011044R	WORD20 = 000050
MAREN2 = 004000	QR#CR1 = 176420	Q#REBK = 001000	T#AD = 000020	WORD21 = 000052
MARL0D = 010000	QR#CR2 = 176422	Q#RNC = 006000	T#BA = 000002	WORD22 = 000054
MAROUT = 000002	QR#LBR = 176424	Q#RSC = 004000	T#BD = 000010	WORD23 = 000056
MAR.LO = 002000	QW = 004242R	Q#RSET = 000010	T#BSO = 100000	WORD24 = 000060
MAR.OU = 000040	QX = 004642R	Q#SM = 100000	T#BT = 000020	WORD25 = 000062
MBKALL = 001000	Q#ATTN = 000100	Q#SP = 000120	T#BTAR = 000030	WORD26 = 000064
MBKCLK = 000400	Q#BCL = 000001	Q#SP2 = 000340	T#BTDR = 002000	WORD27 = 000066
MCL00P = 005320R	Q#CCCP = 000040	Q0 = 004502R	T#CD = 000100	WORD28 = 000070
MEX = 006006R	Q#CHB = 000400	Q1 = 004542R	T#EMEM = 010000	WORD29 = 000072
MICRO = 005170R	Q#CHRL = 000200	Q2 = 004602R	T#FSAA = 000000	WORD3 = 000006
MMADR0 = 000100	Q#CLR = 000040	READ = 007314R	T#FSAB = 000004	WORD30 = 000074
MMLEFT = 000002	Q#CNC = 030000	RGD.EN = 000200	T#FSAC = 000014	WORD31 = 000076
MMOE = 000004	Q#CPC = 000060	RGD.VA = 020000	T#FSB2 = 000010	WORD32 = 000100
MMWRITE = 000010	Q#CPC2 = 000210	SCAN = 007734R	T#IB = 000026	WORD33 = 000102
MNEXT = 005542R	Q#CSC = 010000	SCOUNT = 000244R	T#IBAR = 000024	WORD34 = 000104
MNOBRE = 100000	Q#CSEL = 000360	SELECT = 000242R		WORD35 = 000106
MREN1 = 000001	Q#CSET = 000002	SELMEM = 010650R		WORD36 = 000110
MREN2 = 020000		SEQ.CI = 000010		WORD37 = 000112

SYMBOL TABLE

WORD38= 000114	WORD55= 000156	WORD71= 000216	WORD88= 000260	\$CSTA = ***** GX
WORD39= 000116	WORD56= 000160	WORD72= 000220	WORD89= 000262	\$DIV = ***** GX
WORD4 = 000010	WORD57= 000162	WORD73= 000222	WORD9 = 000022	\$MUL = ***** GX
WORD40= 000120	WORD58= 000164	WORD74= 000224	WORD90= 000264	\$\$\$ = 003026R
WORD41= 000122	WORD59= 000166	WORD75= 000226	WORD91= 000266	\$\$\$ARG= 000002
WORD42= 000124	WORD6 = 000014	WORD76= 000230	WORD92= 000270	\$\$\$T1 = 000067
WORD43= 000126	WORD60= 000170	WORD77= 000232	WORD93= 000272	\$\$\$T2 = 000027
WORD44= 000130	WORD61= 000172	WORD78= 000234	WORD94= 000274	.CLOSE= ***** G
WORD45= 000132	WORD62= 000174	WORD79= 000236	WORD95= 000276	.FSRCB= ***** G
WORD46= 000134	WORD63= 000176	WORD8 = 000020	WORD96= 000300	.GCML1= ***** G
WORD47= 000136	WORD64= 000200	WORD80= 000240	WORD97= 000302	.GET = ***** G
WORD48= 000140	WORD65= 000202	WORD81= 000242	WORD98= 000304	.OPEN = ***** G
WORD49= 000142	WORD66= 000204	WORD82= 000244	WORD99= 000306	.READ = ***** G
WORD5 = 000012	WORD67= 000206	WORD83= 000246	WRDVAL= 000310	.WRITE= ***** G
WORD50= 000144	WORD68= 000210	WORD84= 000250	WRITE = 007452R	...PC1= 003352R
WORD51= 000146	WORD69= 000212	WORD85= 000252	XTREAD= 001000	...PC2= 003352R
WORD52= 000150	WORD7 = 000016	WORD86= 000254	XTWRITE= 000400	...PC3= 003352R
WORD53= 000152	WORD70= 000214	WORD87= 000256	\$COTB = ***** GX	...TPC= 000020
WORD54= 000154				

. ABS. 000000 000
 011104 001
 \$\$\$FSR1 001020 002
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 9224 WORDS. (.37 PAGES)
 DYNAMIC MEMORY: 10196 WORDS. (.39 PAGES)
 ELAPSED TIME: 00:02:11
 CONVRT, CONVRT, -SP=C20, IJIM, C20, IJCONVRT