

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

TITLE HA1NDEC+11-DCKBR+4
COPYRIGHT 1975 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
PROGRAM BY BRUCE BURGESS

OPERATIONAL SWITCH SETTINGS

```

|
| SWITCH USE
|-----|-----
| 15 NALY ON ERROR
| 16 LOOP ON TEST
| 13 INHIBIT ERROR TYPEOUTS
| 11 INHIBIT ITERATIONS
| 10 0 = BELL ON PASS COMPLETE
| 1 1 = BELL ON ERROR
| 9 LOOP ON ERROR
| 0 LOOP ON TEST IN SW<T16>
|SPECIAL USER TYPE SWITCH SW<12>
|
|IF SET INDICATES USER INPUT
|IF CLEAR INDICATES PROGRAM FIND
|SPECIAL KEY1 DISABLE SWITCH SW<00>
|
|IF SET INDICATES DON'T USE IF PRESENT
|IF CLEAR INDICATES ALLOW USE IF PRESENT

```

BASIC DEFINITIONS

```

|*****
|INITIAL ADDRESS OF THE STACK POINTER
|STACK= 1100
|*****
|EQUIV ENT,HLT BASIC DEFINITION OF ERROR CALL
|EQUIV IOY,SCOPE BASIC DEFINITION OF SCOPE CALL
|PC= 177776 IPROCESSOR STATUS WORD
|EQUIV PS,PSM
|SMR= 177576 ISWITCH REGISTER
|DISPLAY=SMR

```

REGISTER DEFINITION

```

|R0= X0 IGENERAL REGISTER
|R1= X1 IGENERAL REGISTER
|R2= X2 IGENERAL REGISTER
|R3= X3 IGENERAL REGISTER
|R4= X4 IGENERAL REGISTER
|R5= X5 IGENERAL REGISTER
|R6= X6 IGENERAL REGISTER
|R7= X7 IGENERAL REGISTER
|EQUIV R6,SP ISTACK POINTER
|EQUIV R7,PC IPROGRAM COUNTER

```

SWITCH DEFINITION

SW15= 100000

55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

|SW14= 40000
|SW13= 20000
|SW12= 10000
|SW11= 4000
|SW10= 2000
|SW09= 1000
|SW08= 400
|SW07= 200
|SW06= 100
|SW05= 40
|SW04= 20
|SW03= 10
|SW02= 4
|SW01= 2
|SW00= 1
|EQUIV SW09,SW9
|EQUIV SW08,SW8
|EQUIV SW07,SW7
|EQUIV SW06,SW6
|EQUIV SW05,SW5
|EQUIV SW04,SW4
|EQUIV SW03,SW3
|EQUIV SW02,SW2
|EQUIV SW01,SW1
|EQUIV SW00,SW0

```

MISCELLANEOUS BIT ASSIGNMENT

```

|BIT15= 100000
|BIT14= 40000
|BIT13= 20000
|BIT12= 10000
|BIT11= 4000
|BIT10= 2000
|BIT09= 1000
|BIT08= 400
|BIT07= 200
|BIT06= 100
|BIT05= 40
|BIT04= 20
|BIT03= 10
|BIT02= 4
|BIT01= 2
|BIT00= 1
|EQUIV BIT09,BIT9
|EQUIV BIT08,BIT8
|EQUIV BIT07,BIT7
|EQUIV BIT06,BIT6
|EQUIV BIT05,BIT5
|EQUIV BIT04,BIT4
|EQUIV BIT03,BIT3
|EQUIV BIT02,BIT2
|EQUIV BIT01,BIT1
|EQUIV BIT00,BIT0

```



```

169
170
171          001100          I*****
172          ,P1100
173
174          ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
175          THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
176          PROTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
177          PROTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
178
179          FCALL:
180          1) USING A TRAP INSTRUCTION
181          I          TYPE          ,MESADR          !MESADR IS FIRST ADDRESS OF AN ASCII STRING
182          JOR
183          I          TYPE
184          J          MESADR
185
186          1d) USING A JSR INSTRUCTION
187          I          MOV          PS,*(SP)          !PUSH PROCESSOR STATUS WORD ON THE STACK
188          J          JSR          PC,$TYPE          !CALL TYPE ROUTINE
189          I          MESADDR          !FIRST ADDRESS OF MESSAGE
190
191          01PS: 177564          !TTY PRINTER STATUS REG. ADDRESS
192          01PB: 177566          !TTY PRINTER BUFFER REG. ADDRESS
193          $NULL: .BYTE 0          !CONTAINS NULL CHARACTER FOR FILLS
194          $FILLS: .BYTE 2          !CONTAINS # OF FILLER CHARACTERS REQUIRED
195          01PFLG: .BYTE 0          !TERMINAL AVAILABLE FLAG (0=YES)
196          !RESERVED
197
198          01TYPE: TSYB          $TFPLG          !IS THERE A TERMINAL?
199          BEQ 68          !OR IF YES
200          HALT          !HALT HERE IF NO TERMINAL
201          OR 78          !LEAVE
202          MOV R0,*(SP)          !SAVE R0
203          MOV @2(SP),R0          !GET ADDRESS OF ASCII STRING
204          MOV @R0,*(SP)          !PUSH CHARACTER TO BE TYPED ONTO STACK
205          BNE 25          !OR IF IT ISN'T THE TERMINATOR
206          TST (SP)+          !IF TERMINATOR POP IT OFF THE STACK
207          MOV (SP),R0          !RESTORE R0
208          ADD @2,*(SP)          !ADJUST RETURN PC
209          RTI          !RETURN
210          JSR PC,58          !GO TYPE THIS CHARACTER
211          CMPB @12,(SP)+          !CHECK IF THE CHAR. TYPED WAS A LINE FEED
212          BNE 15          !GO GET NEXT CHAR. IF NOT LINE FEED
213          MOV $NULL,-(SP)          !GET # OF FILLER CHARS. NEEDED
214          DECB 1(SP)          !AND THE NULL CHAR.
215          BLY 38          !DOES A NULL NEED TO BE TYPED?
216          JSR PC,58          !OR IF NO--GO POP THE NULL OFF OF STACK
217          OR 45          !GO TYPE A NULL
218          TSTB @6TPS          !LOOP
219          SPL 58          !WAIT UNTIL PRINTER IS READY
220          MOVB 2(SP),@5TB          !LOAD CHAN TO BE TYPED INTO DATA REG.
221          RTS PC          !
222          .BLKB 8;          !RESERVE SOME MORE CORE FOR OVERLAY CAPABILITIES

```

```

223          I*****
224          !COMMON TAGS
225          ,=1300
226
227          001300          !=1300
228
229          001300 000000          $PASS: .WORD 0          !CONTAINS PASS COUNT
230          001302 000000          $1STNMI: .WORD 0          !CONTAINS THE TEST NUMBER
231          001304 000000          $1CNT: .WORD 0          !CONTAINS SUBTEST ITERATION COUNT
232          001306 000000          $LPADR: .WORD 0          !CONTAINS SCOPE LOOP ADDRESS
233          001310 000000          $LPERR: .WORD 0          !CONTAINS SCOPE RETURN FOR ERRORS
234          001312 000000          $ERTTL: .WORD 0          !CONTAINS TOTAL ERRORS DETECTED
235          001314 000          $ERFLG: .BYTE 0          !CONTAINS ERROR FLAG
236          001315 000          !RESERVED--NOT TO BE USED
237          001316 000000          !RESERVED--NOT TO BE USED
238          001322 000          $ITEMB: .BYTE 0          !CONTAINS ITEM CONTROL BYTE
239          001323 000          !RESERVED--NOT TO BE USED
240          001324 000000          $MLTAD: .WORD 0          !CONTAINS PC OF LAST MLT INSTRUCTION
241          001326 000000          $GDADR: .WORD 0          !CONTAINS ADDRESS OF 'GOOD' DATA
242          001330 000000          $BDADR: .WORD 0          !CONTAINS ADDRESS OF 'BAD' DATA
243          001332 000000          $GDATI: .WORD 0          !CONTAINS 'GOOD' DATA
244          001334 000000          $BDATI: .WORD 0          !CONTAINS 'BAD' DATA
245          001336 000000          $REGADI: .WORD 0          !CONTAINS THE ADDRESS FROM
246          !WHICH (*REGO) WAS OBTAINED
247          001340 000000          $REG0: .WORD 0          !CONTAINS ((SREGAD)+0)
248          001342 000000          $REG1: .WORD 0          !CONTAINS ((SREGAD)+2)
249          001344 000000          $REG2: .WORD 0          !CONTAINS ((SREGAD)+4)
250          001346 000000          $REG3: .WORD 0          !CONTAINS ((SREGAD)+6)
251          001350 000000          $REG4: .WORD 0          !CONTAINS ((SREGAD)+10)
252          001352 000000          $REG5: .WORD 0          !CONTAINS ((SREGAD)+12)
253          001354 000000          $REG6: .WORD 0          !CONTAINS ((SREGAD)+14)
254          001356 000000          $REG7: .WORD 0          !CONTAINS ((SREGAD)+16)
255          001360 000000          $REG10: .WORD 0          !CONTAINS ((SREGAD)+20)
256          001362 000000          $REG11: .WORD 0          !CONTAINS ((SREGAD)+22)
257          001364 000000          $REG12: .WORD 0          !CONTAINS ((SREGAD)+24)
258          001366 000000          $IMPO: .WORD 0          !USER DEFINED
259          001370 000000          $IMP1: .WORD 0          !USER DEFINED
260          001372 000000          $IMP2: .WORD 0          !USER DEFINED
261          001374 000000          $IMP3: .WORD 0          !USER DEFINED
262          001376 000000          $IMP4: .WORD 0          !USER DEFINED
263          001400 000000          $IMP5: .WORD 0          !USER DEFINED
264          001402 000000          $IMP6: .WORD 0          !USER DEFINED
265          001404 000000          $IMP7: .WORD 0          !USER DEFINED
266          001406 000000          $IMP10: .WORD 0          !USER DEFINED
267          001410 000000          $IMP11: .WORD 0          !USER DEFINED
268          001412 000000          $IMP12: .WORD 0          !USER DEFINED
269          !THE FOLLOWING TAG(S) ARE USER DEFINED
270          $IMPAD: .WORD 0
271          $SETAC: .WORD 0
272          $SETB: .WORD 0          !THESE LOCATIONS CONTAIN THE
273          $SET1: .WORD 0          !APPROPRIATE OFFSET VALUES
274          $SET2: .WORD 0          !FOR THE PARTY CONTROL
275          $SET3: .WORD 0          !REGISTER WHEN MEMORY
276          $SET4: .WORD 0          !MANAGEMENT IS ENABLED

```

277	001432	000000	ENETS1	WORD	0	;DURING PROGRAM EXECUTION	
278	001434	000000	ENETS6	WORD	0		
279	001436	000000	ENETS7	WORD	0		
280	001440	000000	ENETS10	WORD	0		
281	001442	000000	ENETS11	WORD	0		
282	001444	000000	ENETS12	WORD	0		
283	001446	000000	NICRAD	WORD	0		
284	001450	000000	NIER0	WORD	0		;THESE LOCATIONS CONTAIN THE ;APPROPRIATE INTERLEAVE FACTORS ;FOR THE PARITY CONTROL ;REGISTERS (IF ANY)
285	001452	000000	NIER1	WORD	0		
286	001454	000000	NIER2	WORD	0		
287	001456	000000	NIER3	WORD	0		
288	001460	000000	NIER4	WORD	0		
289	001462	000000	NIER5	WORD	0		
290	001464	000000	NIER6	WORD	0		
291	001466	000000	NIER7	WORD	0		
292	001470	000000	NIER10	WORD	0		
293	001472	000000	NIER11	WORD	0		
294	001474	000000	NIER12	WORD	0		
295	001476	000000	NKSTK1	WORD	0		
296			;END OF USER DEFINED TAG(S)				
297							

298			;THE FOLLOWING TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.		
299			;THIS INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN		
300			;LOCATION SYSTEMS. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.		
301			;NOTE1: IF SYSTEM IS 0 THE ONLY PERTINENT DATA IS (SHLTAD).		
302			;NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:		
303					
304			E	EM	;POINTS TO THE ERROR MESSAGE
305			H	DH	;POINTS TO THE DATA HEADEN
306			D	DT	;POINTS TO THE DATA
307			F	DF	;POINTS TO THE DATA FORHAI
308					
309					
310	001500		ERR00:		
311			;NOTE: ALL NUMBERS ARE TYPED AS 6-DIGIT OCTAL NUMBERS		
312					
313			ITEM 1		
314	001500	013752	EH1		;TEST DIDN'T ABORT
315	001502	015005	DH5		;PROGRAM PC
316					;REGISTER UNDER TEST
317					;EXPECTED ABORT PC
318	001504	015142	DT5		;SHLTAD,PARITY,&GDDAT
319	001506	000000	0		
320			ITEM 2		
321	001510	013776	EM2		;FATAL ERROR TO PROGRAM
322	001512	014470	DH1		;PROGRAM PC
323					;REGISTER UNDER TEST
324	001514	015106	DT1		;SHLTAD, PARITY
325	001516	000000	0		
326			ITEM 3		
327	001520	014027	EM3		;ABORTED INCORRECTLY
328	001522	014604	DH4		;PROGRAM PC
329					;REGISTER UNDER TEST
330					;EXPECTED BITS 5 THRU 11
331					;ACTUAL BITS 5 THRU 11
332					;EXPECTED ABORT PC
333					;ACTUAL ABORT PC
334	001524	015124	DT4		;SHLTAD,PARITY,&GDAOR,&BDAOR,&GDDAT,&BDDAT
335	001526	000000	0		
336			ITEM 4		
337	001530	014059	EM4		;NO PARITY MEMORY FOUND BELOW 28K
338	001532	014540	DH2		;REGISTER UNDER TEST
339	001534	015114	DT2		;PARITY
340	001536	000000	0		
341			ITEM 5		
342	001540	014120	EM5		;RESET DOESN'T WORK
343	001542	014470	DH1		;PROGRAM PC
344					;REGISTER UNDER TEST
345	001544	015106	DT1		;SHLTAD, PARITY
346	001546	000000	0		
347			ITEM 6		
348	001550	014149	EM6		;USER SELECTED REGISTER NOT PRESENT
349	001552	014566	DH3		;PROGRAM PC
350	001554	015120	DT3		;SHLTAD
351	001556	000000	0		


```

431
432
433
434
435
436
437
438
439 001620 000114
440 001622 000000
441 001624 000000
442
443
444
445 001626 000000
446
447 001630 000000
448
449
450
451 001632 000000
452
453
454
455
456 001634 000000
457
458
459
460
461
462
463
464 001636 000000
465
466
467 001640 000000
468
469 001642 000000
470
471
472
473
474
475 001644
476 001644 000003 000001
477 001650 000006 000002
478 001654 000011 000003
479 001660 000014 000004
480 001664 000017 000005
481 001670 000022 000006
482 001674 000025 000007
483 001700 000030 000010
484 001704 000000

```

/*****
/ MISCELLANEOUS COMMON PARITY VARIABLES AND FLAGS
/*****
/ PARITY INTERRUPT VECTOR ADDRESS
/ CONTAINS PARITY REGISTER IN USE
/ FLAG TO INDICATE TO 'CHECKLOC'
/ ROUTINE THAT A PS OR PC FETCH
/ FOR A ZONE ABORT WAS DONE
/ 0 = NO; 1 = YES
/ INDICATES PARITY TYPE
/ 0 = CORE; 1 = NOS
/ INDICATES USER SELECTION OF
/ PARITY REGISTER
/ 0 = PROGRAM FIND
/ 1 = USER SELECTION
/ CONTAINS THE NUMBER OF CONSEC-
/ UTIVE LOCATIONS TO BE TESTED
/ DURING PROGRAM TABULATION TO
/ COVER CASES OF MEMORY INTER-
/ LEAVING
/ INCLUDE PAGE 1 ADDRESS
/ FOR CURRENT MEMORY ADDRESS FOR
/ RESTORATION DURING
/ RUNNING UP PROGRAM. IT IS USED IF
/ WE HAVE CHECKED CONSECUTIVE
/ LOCATIONS WITHOUT AN ABORT BEFORE
/ GOING TO NEXT OFFSET WHICH WILL
/ PUT US IN ANOTHER BANK
/ CONTAINS THE NO. OF ABORTS
/ ENCOUNTERED IN DETERMINATION OF
/ AN INTERLEAVE FACTOR
/ CONTAINS A BASE ADDRESS OR A
/ CURRENT MEMORY ADDRESS USED IN
/ FLAG TO INDICATE PROCESSOR
/ 0 = 11/40; 1 = 11/40
/ PARITY TABLE CREATION

/ THE FOLLOWING TABLE IS USED TO DETERMINE THE
/ INTERLEAVE FACTOR FOR THE CONTROL REGISTERS
/ INTERLEAVE TABLE:
/ 3,,1 15 ABORTS ON 3 CONSECUTIVE LOCS. = 1 WAY LEAVE
/ 6,,2 15 ABORTS ON 6 CONSECUTIVE LOCS. = 2 WAY LEAVE
/ 9,,3 15 ABORTS ON 9 CONSECUTIVE LOCS. = 3 WAY LEAVE
/ 12,,4 15 ABORTS ON 12 CONSECUTIVE LOCS. = 4 WAY LEAVE
/ 15,,5 15 ABORTS ON 15 CONSECUTIVE LOCS. = 5 WAY LEAVE
/ 18,,6 15 ABORTS ON 18 CONSECUTIVE LOCS. = 6 WAY LEAVE
/ 21,,7 15 ABORTS ON 21 CONSECUTIVE LOCS. = 7 WAY LEAVE
/ 24,,8 15 ABORTS ON 24 CONSECUTIVE LOCS. = 8 WAY LEAVE
/ 0 END OF TABLE TERMINATOR

```

485
486 001706
487 001706 012706 001100
488 001712 012737 012062 000000
489 001720 012737 000340 000022
490 001726 005067 177358
491 001732 012737 012034 000030
492 001740 012737 000340 000032
493 001746 012737 013556 000034
494 001754 012737 000340 000036
495 001762 012737 013606 000024
496 001770 012737 000340 000026
497 001776 005067 177276
498 002002 005067 177276
499 002008 005067 010276
500 002012 105067 177276
501 002016 005067 177276
502 002022 005067 010746
503
504 002026 005037 001630
505
506
507 002032 005037 002304
508 002036 005037 001420
509 002042 005037 001422
510 002046 005037 001424
511 002052 005037 001426
512 002056 005037 001430
513 002062 005037 001432
514 002066 005037 001434
515 002072 005037 001436
516 002076 005037 001440
517 002102 005037 001442
518 002106 005037 001444
519 002112 005037 001450
520 002116 005037 001452
521 002122 005037 001454
522 002126 005037 001456
523 002132 005037 001460
524 002136 005037 001462
525 002142 005037 001464
526 002146 005037 001466
527 002152 005037 001470
528 002156 005037 001472
529 002162 005037 001474
530 002166 005037 001636
531
532 002172 005037 001642
533 002176 013746 000004
534 002202 013746 000010
535 002206 012737 002226 000010
536 002214 012737 000340 000012
537 002222 000237
538 002224 000003

```

BEGIN:
MOV @STACK,SP ;SETUP THE STACK POINTER
MOV @SCOPE,@I01VEC ;I01 VECTOR FOR SCOPE ROUTINE
MOV @300,@I01VEC+2 ;LEVEL 7
CLR \$Y5NHM ;INITIALIZE THE TEST NUMBER
MOV @MLT,@M1TVEC ;INIT VECTOR FOR MLT(ERROR) ROUTINE
MOV @320,@M1TVEC+2 ;LEVEL 7
MOV @STRAP,@I01RVEC ;TRAP VECTOR FOR TRAP CALLS
MOV @340,@I01RVEC+2 ;LEVEL 7
MOV @360,@PWRVEC ;POWER FAILURE VECTOR
MOV @380,@PWRVEC+2 ;LEVEL 7
CLR \$PASS ;CLEAR THE PASS COUNT
CLR \$CNT ;INITIALIZE THE ITERATION COUNTER
CLR \$TIMES ;INITIALIZE NUMBER OF ITERATIONS
CLR \$ERFLG ;CLEAR THE ERROR FLAG
CLR \$ERRCNT ;CLEAR THE ERROR COUNT
CLR \$ESCAPE ;CLEAR THE ESCAPE ON ERROR ADDRESS

CLR @USERTYPE ;SET USER SELECTION INDICATOR
;TO ZERO INDICATING PROGRAM
;TABULATION
CLR @SKT11 ;CLEAR K11 PRESENCE FLAG
CLR @SSET0 ;CLEAR THE OFFSET
CLR @SSET1 ;TABLE LOCATIONS FOR
;THE K11 OPTION
CLR @SSET2
CLR @SSET3
CLR @SSET4
CLR @SSET5
CLR @SSET6
CLR @SSET7
CLR @SSET10
CLR @SSET11
CLR @SSET12

CLR @ENTER0 ;CLEAR THE INTERLEAVE TABLE
;ENTRY LOCATIONS
CLR @ENTER1
CLR @ENTER2
CLR @ENTER3
CLR @ENTER4
CLR @ENTER5
CLR @ENTER6
CLR @ENTER7
CLR @ENTER10
CLR @ENTER11
CLR @ENTER12
CLR @LEAFcnt ;CLEAR NO. OF ABORTS PER NO. OF
;CONSECUTIVE LOCS, TESTED LOC.
CLR @CPUAC ;CLEAR PROCESSOR INDICATOR FLAG
MOV @4,,(SP) ;SAVE CONTENTS OF LOC. 4
MOV @10,,(SP) ;SAVE CONTENTS OF LOC. 10
MOV @18,@RESVEC ;SET UP FOR 'SPL' TRAP ADDRESS
MOV @300,@RESVEC+2 ;SET UP FOR 'SPL' TRAP PS
SPL 7 ;ATTEMPT TO SET A PRIORITY LEVEL
BR 26 ;BRANCH INDICATING WE ARE ON AN

```

530
540 002200 002026 13: CMP (SP)+,(SP)+ 75145 PROCESSOR
541 002200 002027 001000 INC #PCURD IRESET THE STACK FROM TRAP
542 ISET FLAG INDICATING WE ARE ON
543 IAN 11748 PROCESSOR
544 002200 002027 000010 20: MOV (SP)+,#R10 IRESTORE CONTENTS OF LOC. 10
545 002200 002027 000010 CLR #R10 IRESTORE TRAPCATCHER LOC. 12
546 002200 002027 000000 MOV #KTYTIMEOUT,#KERRVEC ISET UP K1 TIMEOUT ADDRESS
547 002200 002027 000000 MOV #R0,#KERRVEC+2 ISET UP K1 TIMEOUT P0
548 002200 002027 175710 TST #R0 IKY11 ARE YOU THERE?
549 IYES - INITIALIZE IT IN CASE
550 IUSER DOESNT WANT IT
551 002270 013700 177570 MOV #RSHR,R0 IGET SHR CONTENTS
552 002270 006300 R0 IMOVE BITS 6 TO BIT 07 POSITION
553 002270 105700 TST R0 IKY11 PRESENT (OBVIOUSLY) IF
554 IWE REACH THIS INSTRUCTION
555 IDOES USER WANT IT?
556 002300 100012 BMI GO IBRANCH IF NO
557 002300 005327 (PC)+ IYES = SET KY11 FLAG
558 002300 000000 BRTY11 ICONTAINS A *1 IF KY11 OPTION
559 IIS PRESENT
560 002300 004737 013704 JSR PC,#R5SIZE ISEE HOW MUCH MEMORY IS AVAILABLE
561 002310 005077 175712 CLR #KPAR0 ISELECT PAGE 0 OFFSEY REGISTER
562 002310 005277 175000 INC #SR0 IWORK ON MEMORY MANAGEMENT
563 002320 000001 BR OR ISKIP NEXT INSTRUCTION
564 002320 002026 KTYTIMEOUT: CMP (SP)+,(SP)+ IRESET THE STACK FROM TIMEOUT
565 IKY11 NOT PRESENT, THEREFORE
566 IONLY GO BELOW 20K
567 002320 012637 000000 GO: MOV (SP)+,#R4 IRESTORE CONTENTS OF LOC. 0
568 002320 005077 000000 CLR #R6 IRESTORE CONTENTS OF LOC. 6
569 002330 004337 011000 JSR R3,#R5INITIALIZE ISET UP TO BEGIN TESTING
570 002340 016702 177000 MOV #RHPAD,R3 ISET UP FOR MEMORY TABLE CREATION
571 002350 016702 176700 MOV #RREGID,R2 ISET UP FOR PARITY TABLE CREATION
572 002350 016700 177000 MOV #RSETAD,R0 ISET UP FOR OFFSET TABLE CREATION
573 ITHIS TABLE ONLY HAS EFFECT IF
574 IMEMORY MGMT IS TURNED ON
575 ISET UP FOR INTERLEAVE TABLE
576 ICREATION (8 = MAY INTERLEAVE
577 ICAPABILITY EXISTS)
578
579 *****
580 ILET'S DETERMINE IF SEVERAL REGISTERS EXIST, FOR EXAMPLE,
581 I
582 I 172100 GOVERNING CORE MEMORY 0 = 0K
583 I 172102 GOVERNING BOS MEMORY 8 = 16K
584 I 172112 GOVERNING CORE MEMORY 40 = 56K
585 I
586 IIF WE WANT TO PRESELECT ONE OF THEM OR CREATE A TABLE OF ALL THOSE
587 IAVAILABLE AND CARRY ON TESTING FROM THE TABLE
588 I
589 INOTE: SEE DOCUMENT CONCERNING TABLE APPEARANCES AS A
590 IFUNCTION OF MEMORY MANAGEMENT (KY11 OPTION) BEING
591 IENABLED OR DISABLED DURING PROGRAM EXECUTION
592 I
593 *****

```

```

593 002360 012737 010000 177570 BIT #BIT12,#RWR IDOES THE USER WISH TO SELECT THE
594 IREGISTER?
595 002370 001000 BEQ FINDONE IBRANCH IF NO
596 002370 000000 MSGTY: TYPE ,,+4 ITYPE ASCII STRING
597 002370 100000 002000 BR 64# IGET OVER THE ASCII
598 002370 000033 I.ASCIIZ <15><12>"TYPE THE REGISTER YOU WANT & HIT CARRIAGE RETURN "
599
600 002400 60: ACCEPT,#R00 IPICK UP THE DESIRED REGISTER
601 002400 100000 001300 FROM THE TELETYPE AND STORE
602 IIN FIRST TABLE LOCATION
603 ISET FLAG INDICATING USER SELECTION
604 002470 005237 001630 INC #USERSTYP ISAVE CONTENTS OF LOC. 4
605 002470 013700 000000 MOV #R4,-(SP) ISKIP THE NEXT INSTRUCTIONS
606 002500 000000 BR NEXT1 IMOVE 1ST POSSIBLE PARITY
607 002500 012712 172100 FINDONE: MOV #172100,(R2) IREGISTER INTO #R00
608 IREGISTER INTO #R00
609 002510 013700 000000 MOV #R4,-(SP) IPUSH CONTENTS OF LOC.4 ONTO STACK
610 002510 012700 001600 NEXT1: MOV #INTERTABLE,R4 IINITIALIZE INTERLEAVE TABLE
611 Ipointer
612 002520 005737 002300 TST #SKY11 IKY11 ARE YOU THERE?
613 002520 001000 BEQ 50 IBRANCH IF NO
614 002520 005077 175000 CLR #KPAR1 IRESET PAGE 1 ADDRESS REGISTER
615 IBEFORE TESTING NEXT PARITY
616 ICONTROL REGISTER
617 002530 012737 003100 50: MOV #NOREG,#R4 ISET PARITY TIMEOUT VECTOR SERVICE ADDRESS
618 002540 002712 172130 CMP #172130,(R2) IIS THE ADDRESS IN BOUNDS?
619 002540 100000 BPL 125 IBRANCH IF YES
620 002540 000137 003170 JMP #NOREG IOTHERWISE = TERMINATE TABLE
621 002550 005772 000000 10: TST #R2 IYES = IS THIS REGISTER PRESENT?
622 002550 004737 003200 JSR R7,#RPARTY ICHECK OUT FOR FATAL ERRORS
623
624 I
625 IWE HAVE CHECKED OUT THE REGISTER AND FOUND IT TO BE WORKING PROPERLY
626 INOW WE WILL FIND ITS ASSOCIATED PARITY MEMORY, IF IT EXISTS!!
627 I
628 I
629 002560 012737 003110 000000 MOV #PARCORE,#R4 ISET MEMORY TIMEOUT VECTOR
630 ISERVICE ADDRESS
631 002570 012737 013700 001600 MOV #13700,#MEMHAD ISET UP A STARTING ADDRESS
632 002570 011037 001630 MOV (R4),#BLKCNT ISET A COUNTER FOR CONSECUTIVE
633 ILOCATION CHECKS TO COVER MEMORY
634 IINTERLEAVING
635 002600 005737 002300 TST #SKY11 I SHOULD I LOOK ABOVE 20K?
636 002600 100013 IS IBRANCH IF NO
637 002610 002737 010000 001600 ADD #10000,#MEMHAD ISTEP UP TO A PAGE 1 BASE ADDRESS
638 IIF MEMORY MANAGEMENT TURNED ON
639 002610 013737 001600 001630 20: MOV #MEMHAD,#RESTOREBASE ISAVE PAGE 1 BASE ADDRESS
640 002620 002710 000100 ADD #100,(R0) ISET UP AN OFFSET FOR KPAR1
641 002630 011077 175370 MOV (R0),#KPAR1 ISET OFFSET IN PAGE 1 REGISTER
642 002630 000000 BR 50 ISKIP NEXT 2 INSTRUCTIONS
643 002630 002737 000000 10: ADD #R00,#MEMHAD ISTEP UP TO NEXT BANK
644 002640 013737 001600 001630 MOV #MEMHAD,#RESTOREBASE ISAVE INITIAL MEMORY ADDRESS
645 002650 005777 176700 TST #MEMHAD IIS THIS MEMORY AVAILABLE?
646 002650 013713 001600 MOV #MEMHAD,(R3) IYES = STORE THIS MEMORY LOCATION

```

647	002060	004137	003020	JSR	R7,PARCORE1	FROM LET'S SEE IF IT'S PARITY
648						MEMORY CORRESPONDING TO THE
649						PARITY REGISTER HAVE FOUND
650	002066	005737	003030	TST	@BKTI1	BT1) ARE YOU THERE?
651	002072	005037	001F32	BPL	00	BRANCH IF NO
652	002078	005037	001F32	DEC	@BLKCNT	DECREASE CONSECUTIVE
653						LOCATION COUNTER
654	002700	005737	001632	TRY	@BLKCNT	ARE WE DONE CHECKING
655						CONSECUTIVE LOCATIONS?
656	002704	001000		BEG	00	BRANCH IF YES
657	002706	002737	000002	ADD	@2,PARHEAD	STEP UP 1 LOCATION
658	002714	000756		BR	00	GO BACK TO TEST WITH THIS
659						LOCATION
660	002716	012737	001634	MOV	@RSTOREBASE,PARHEAD	RESTORE PAGE 1 BASE ADDRESS
661						BEFORE GOING BACK TO INCREASE
662						OFFSET
663	002720	002704	000004	ADD	@4,R4	STEP TABLE POINTER UP FOR
664						NEXT VALUE OF CONSECUTIVE
665						LOCATION TO BE CHECKED
666	002730	005714		TST	(R4)	ARE THERE ANY MORE?
667	002732	001400		BEG	100	BRANCH IF NO
668	002734	011437	001632	MOV	(R4),@BLKCNT	STORE THIS VALUE OF CONSECUTIVE
669						LOCATION CHECKS
670	002740	005037	001636	CLR	@LEAFONT	CLEAR INTERLEAVE VALUE HOLDER
671						BEFORE RETESTING
672	002744	000742		BR	00	GO BACK TO TEST WITH THIS VALUE
673						FOR CONSECUTIVE LOCATIONS
674	002746	012704	001644	MOV	@INTERTABLE,R4	INITIALIZE INTERLEAVE TABLE
675						POINTER
676	002752	011437	001632	MOV	(R4),@BLKCNT	RESET THE CONSECUTIVE
677						LOCATION COUNTER
678	002756	005037	001636	CLR	@LEAFONT	CLEAR INTERLEAVE VALUE HOLDER
679						BEFORE RETESTING
680	002762	000720		BR	00	GO BACK TO INCREASE OFFSET
681						AND TEST
682	002764	002737	107700	CMP	@157700,PARHEAD	ARE WE UP TO 26K YET?
683	002772	001400		BEG	00	BRANCH IF YES
684	002774	005037	001632	DEC	@BLKCNT	DECREASE CONSECUTIVE
685						LOCATION COUNTER
686	003000	005737	001632	TST	@BLKCNT	ARE WE DONE CHECKING CONSEC-
687						UTIVE LOCATIONS?
688	003004	001404		BEG	00	BRANCH IF YES
689	003006	002737	000002	ADD	@2,PARHEAD	STEP UP 1 LOCATION
690	003014	000716		BR	00	GO BACK TO TEST WITH THIS
691						LOCATION
692	003016	012737	001634	MOV	@RSTOREBASE,PARHEAD	RESTORE INITIAL MEMORY
693						ADDRESS BEFORE GOING BACK TO
694						STEP UP TO NEXT BANK
695	003024	002704	000004	ADD	@4,R4	STEP TABLE POINTER UP FOR
696						NEXT VALUE OF CONSECUTIVE
697						LOCATIONS TO BE CHECKED
698	003030	005714		TST	(R4)	ARE THERE ANY MORE?
699	003032	001400		BEG	110	BRANCH IF NO
700	003034	011437	001632	MOV	(R4),@BLKCNT	STORE THIS VALUE OF

701	003040	005037	001636	CLR	@LEAFONT	CONSECUTIVE LOCATION CHECKS
702						CLEAR INTERLEAVE VALUE HOLDER
703						BEFORE RETESTING
704	003044	000702		BR	00	GO BACK TO STEP UP TO THE
705						NEXT BANK TO CONDUCT TESTING
706	003046	012704	001644	MOV	@INTERTABLE,R4	INITIALIZE INTERLEAVE
707						TABLE POINTER
708	003052	011437	001632	MOV	(R4),@BLKCNT	RESET THE CONSECUTIVE
709						LOCATION COUNTER
710	003056	005037	001636	CLR	@LEAFONT	CLEAR INTERLEAVE VALUE HOLDER
711						BEFORE RETESTING
712	003062	000665		BR	10	GO BACK TO STEP UP TO NEXT
713						BANK TO CONDUCT TESTING
714	003064	011237	001622	MOV	(R2),@PARITY	STORE THE BAD REGISTER WITH
715						NO PARITY MEMORY
716	003070	104000		HLT	+4	NO PARITY MEMORY FOUND
717						BELOW 28K!!!!!!!
718	003072	005737	001630	TST	@USERTYPE	DID USER SELECT REGISTER?
719	003076	001402		BEG	30	BRANCH IF NO
720	003100	000137	002372	JMP	@MSGTYP	GO BACK TO RETYPE MESSAGE FOR
721						USER RESPONSE
722	003104	002712	000002	ADD	@2,(R2)	PLACE NEXT POSSIBLE REGISTER
723						INTO SAME TABLE LOCATION
724	003110	000601		BR	NEXT1	GO BACK TO TEST THIS REGISTER
725	003112	002626		PARCORE:	CMP (SP)+,(SP)+	RESET STACK FROM MEMORY TIMEOUT
726	003114	011237	001622	MOV	(R2),@PARITY	STORE THE REGISTER THAT EN-
727						OUNTERED A POSSIBLE HOLE IN
728						MEMORY
729	003120	104007		HLT	+7	IS A POSSIBLE HOLE IN MEMORY EXISTS
730						WITH NO PARITY BELOW IT!!!!!!!
731	003122	005737	001630	TST	@USERTYPE	DID USER SELECT REGISTER?
732	003126	001402		BEG	40	BRANCH IF NO
733	003130	000137	002372	JMP	@MSGTYP	GO BACK TO RETYPE MESSAGE FOR
734						USER RESPONSE
735	003134	002712	000002	ADD	@2,(R2)	PLACE NEXT POSSIBLE REGISTER
736						INTO SAME TABLE LOCATION
737	003140	000137	002514	JMP	@NEXT1	GO BACK TO TEST THIS REGISTER
738	003144	002626		NUREG:	CMP (SP)+,(SP)+	RESET STACK FROM REGISTER TIMEOUT
739	003146	005737	001630	TST	@USERTYPE	DID THE USER SELECT THE REGISTER?
740	003152	001403		BEG	10	BRANCH IF NO
741	003154	104006		HLT	+6	YES = USER SELECTED REGISTER NOT
742						PRESENT ON SYSTEM
743	003156	000137	002372	JMP	@MSGTYP	GO BACK TO RETYPE MESSAGE
744	003162	002712	000002	ADD	@2,(R2)	STEP UP TO NEXT PARITY REGISTER
745						AT SAME TABLE LOCATION
746	003166	000137	002514	JMP	@NEXT1	PREVIOUS PARITY REGISTER NOT
747						PRESENT - SEE IF THE NEXT ONE IS
748	003172	012637	000004	NUHORE:	MOV (SP)+,004	RESTORE CONTENTS OF LOC. 4
749	003176	005012		CLR	(R2)	ALL DONE TABLE CREATION
750						END IT WITH A 'G'
751	003200	000137	003020	JMP	@START	START RUNNING PROGRAM WITH
752						TABLE CONTENTS
753						*****
754						F


```

755 /THE FOLLOWING ROUTINE WILL CREATE A 2 LOCATION MEMORY MAP AT
756 /THE HIGH END OF A 1K CORE SECTION. THIS 2 LOCATION MAP WILL
757 /INITIALLY BE USED TO DETERMINE WHERE/IF PARITY MEMORY
758 /PRESIDES AND LATER FOR SUBSEQUENT PROGRAM TESTING OF A REGISTER
759 /
760 /*****
761 003204 102700 000376 COMPUT: SUB #376,R0 /DROP DOWN SO AS NOT TO
762 /DESTROY ABS LOADER
763 003210 010001 MOV R0,R1 /R1 CONTAINS BEGINNING ADDRESS
764 /OF MEMORY MAP
765 003212 000000 CMP R0,(R0)+ /STEP R0 TO NEXT ADDRESS
766 003214 010011 MOV R0,R1 /1ST MEMORY LOCATION
767 003216 011110 MOV #R1,R0 /2ND MEMORY LOCATION
768 003220 000203 RTS R3 /RETURN TO TEST A DATI
769 /WITH CONTENTS OF THESE 2 LOCS.
770 /
771 /*****
772 /
773 /THIS ROUTINE WILL CHECK IF THE PARITY REGISTER IS STATICALLY IN
774 /GOOD OPERATION FOR TESTING TO BE CONDUCTED
775 /
776 /*****
777 003222 011267 176374 PARTST: MOV (R2),PARITY /GET PARITY REGISTER TO BE USED
778 /
779 /TEST 1 SET BITS (USED) OF PARITY REGISTER
780 /*****
781 003226 000004 TST1: SCOPE
782 003230 002777 000001 176364 BIS #BIT0,@PARITY /DID IT SET?
783 003234 002777 000001 176356 BIT #BIT0,@PARITY /YES
784 003244 001001 BNE ,+4 /NO - FATAL ERROR TO PROGRAM!
785 003246 100002 HLT +2
786 /*****
787 /TEST 2 CLEAR BITS (USED) OF PARITY REGISTER
788 /*****
789 003250 000004 TST2: SCOPE
790 003252 002777 000001 176342 BIS #BIT0,@PARITY /DID IT CLEAR?
791 003254 002777 000001 176334 BIT #BIT0,@PARITY /YES
792 003256 001001 BEO ,+4 /NO - FATAL ERROR TO PROGRAM!
793 003270 100002 HLT +2
794 /*****
795 /TEST 3 SET AND CLEAR BITS (USED) OF PARITY REGISTER
796 /*****
797 003272 000004 TST3: SCOPE
798 003274 002777 000004 176320 BIS #BIT2,@PARITY /DID IT SET?
799 003302 002777 000004 176312 BIT #BIT2,@PARITY /YES
800 003310 001001 BNE ,+4 /NO - FATAL ERROR TO PROGRAM!
801 003312 100002 HLT +2
802 003314 002777 000004 176300 BIS #BIT2,@PARITY /DID IT CLEAR?
803 003322 002777 000004 176272 BIT #BIT2,@PARITY /YES
804 003330 001001 BEO ,+4 /NO - FATAL ERROR TO PROGRAM!
805 003332 100002 HLT +2
806 /*****
807 /TEST 4 TEST RESET ON BITS 0, 2 AND 15
808 /*****

```

```

809 003334 000004 TST4: SCOPE
810 003336 005737 002304 TST #0SKY11 /KTY11 ON?
811 003342 100015 BHI WHICH1 /BRANCH IF YES AND DON'T DO
812 /THIS TEST BECAUSE THE 'RESET'
813 /WILL CLOBBER SEGMENTATION
814 003344 002777 100005 176250 BIS #100005,@PARITY /EXPECT BITS 0, 2 AND 15 TO CLEAR
815 003352 000005 RESET /DID THEY CLEAR?
816 003354 002777 100005 176240 BIT #100005,@PARITY /YES
817 003362 001004 BEO ,+12 /NO - CLEAR OUT REGISTER AS A
818 003364 002777 100005 176230 SIC #100005,@PARITY /PRECAUTION
819 /RESET DOESN'T WORK
820 003372 100005 HLT +5
821 /*****
822 /TEST 5 WHICH OPTION IS ABOUT TO BE TESTED
823 /*****
824 003374 000004 TST5: SCOPE
825 /
826 003376 002777 007740 176216 WHICH1: BIS #7740,@PARITY /IS AN OLD MS11 OPTION
827 /WITH NO ADDRESS BITS
828 /ABOUT TO BE TESTED?
829 003404 002777 007740 176210 BIT #7740,@PARITY /ADDRESS BITS ABLE TO BE SET?
830 003412 001002 BEO 15 /BRANCH IF NO INDICATING MS11
831 003414 000004 SCOPE
832 003416 000207 RTS R7 /RETURN TO NORMAL FLOW
833 003420 000237 001626 1ST: INC #MSREGFLAG /SET FLAG INDICATING MS11 OPTION
834 /WITH NO ADDRESS BITS
835 /
836 003424 000004 SCOPE
837 003426 000207 RTS R7 /RETURN TO NORMAL FLOW
838 /
839 /*****
840 /
841 /THE FOLLOWING ROUTINE WILL TAKE EACH 1K BANK OF MEMORY
842 /THAT IS AVAILABLE AND PERFORM A DATI IN IT
843 /TO DETERMINE IF PARITY EXISTS THERE. THIS ROUTINE IS
844 /ONLY USED DURING TABLE CREATION
845 /
846 /*****
847 003430 010546 ABORT: MOV R5,*(SP) /SAVE R5 CONTENTS ON STACK
848 003432 010046 MOV R0,*(SP) /SAVE R0 CONTENTS ON STACK
849 003434 011300 MOV (R3),R0 /GET THE MEMORY LOCATION
850 /JUST DETERMINED
851 003436 004337 003204 JSR R3,@COMPUT /COMPUTE AN AREA IN THIS BANK
852 /FOR DETERMINING PARITY MEMORY
853 003442 011267 176150 MOV (R2),PARITY /GET THE PARITY REGISTER JUST
854 /FOUND AND TEST WITH IT
855 /
856 /TEST A DATI IN THIS BANK
857 /*****
858 /
859 /
860 003446 012705 011450 /
861 003452 004015 JSR #R0,(R5) /SET UP SERVICE ROUTINE ADDRESS
862 003454 003500 ONETRY /SET UP PARITY VECTOR SERVICE
/ROUTINE ADDRESS

```

```

043 003456 011100      MOV      @R1,R0      /SET UP FOR A DATO
044 011400 010010      MOV      @R0,R0      /ADD THE DATO
045 003460 010020      MOV      @R0,(R0)    /ADD A DATO
046 003460 042777 000000 176130  BIC      @012101010,0PARITY /WRITE NORMAL AND DISABLE
047 003470 012600      MOV      @R0,R0      /RESTORE R0 CONTENTS
048 003470 012600      MOV      @R0,R0      /RESTORE R5 CONTENTS
049 003476 042207      RTS      R7          /NOT PARITY MEMORY
050                          /RETURN TO TEST AT NEXT
051                          /INCREMENT
052 003500 042777 000000 176110  ONETRY: BIC      @012101010,0PARITY /WE HAVE PARITY MEMORY * PROCEED
053 043500 016600 000000      MOV      4(SR),R0    /RESTORE R0 CONTENTS
054 003510 016600 000000      MOV      6(SR),R5    /RESTORE R5 CONTENTS
055 003510 042737 001430      INC      @LEAFENT    /INCREMENT INTERLEAVE COUNTER
056 003520 042737 000000 001630  CMP      @3,@LEAFENT /IS ABORTS REACHED?
057 003520 041400      BEQ      10         /BRANCH IF YES
058 003530 042706 000010      ADD      @10,@P      /BYPASS JUNK ON STACK
059 003530 042607      RTS      R7          /RETURN TO TEST AT NEXT INCREMENT
060 003540 042620 10:      CMP      @P,@(SP)+   /POP STACK BACK FROM PARITY ABORT
061 003540 012600      MOV      @R0,R0      /RESTORE R0 CONTENTS
062 003540 012600      MOV      @R0,R0      /RESTORE R5 CONTENTS
063 003546 042720      TST      @P+         /POP STACK ONCE FOR ABORT ROUTINE
064                          /ENTRY
065 003550 045737 001630      TST      @SUBERTYPE+ /IS USER TYPE IN REGISTER
066 003550 011400      BEQ      20         /BRANCH IF NO
067 003556 016410 000000      MOV      2(R0),(R5)  /SET INTERLEAVE VALUE INTO
068                          /TABLE
069 003560 040137 003520      JMP      @START     /AND LOCK ON THE USER SELECTED
070                          /REGISTER FOR TESTING
071 003566 045720 20:      TST      (R5)+      /USER DIDN'T SELECT - SO STEP UP
072                          /TO NEXT MEMORY TABLE LOCATION
073 003570 045720      TST      (R0)+      /STEP UP TO NEXT OFFSET TABLE
074                          /LOCATION - THIS TABLE WILL ONLY
075                          /BE APPLICABLE IF MEMORY MGMT
076                          /IS TURNED ON
077 003570 012212      MOV      (R2)+,(R2)  /SET NEXT POSSIBLE REGISTER INTO
078 003570 042712 000000      ADD      @2,(R2)     /NEXT REGISTER TABLE LOCATION
079 003600 016425 000000      MOV      2(R0),(R5)+ /SET INTERLEAVE VALUE INTO
080                          /TABLE
081 003600 045037 001630      CLR      @LEAFENT    /RESET NO. OF ABORTS COUNTER
082 003610 045037 001626      CLR      @ABORTFLAG /CLEAR PARITY TYPE INDICATOR
083 003610 040137 002514      JMP      @NEXT1     /GO BACK TO CHECK NEXT POSSIBLE
084                          /PARITY REGISTER
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
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

```

```

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

```

```

971
972 002676 004337 003200 JSR R3,#PCOMPUT WITH THE PARITY REGISTER
973 / COMPUTE AN AREA IN THIS BANK
974 003702 010137 001476 MOV R1,#PCNSWTR / FOR TESTING
975 / SET UP A NEW STACK POINTER
976 / FOR STACK OPERATIONS IN CASE
977 / AND PARITY MEMORY RESIDES IN
978 003706 004737 003376 JSR PC,#NWHIGH / LOWER OK
979 / DETERMINE IF WE ARE ABOUT TO
980 003712 012705 011450 MOV @VCCSET,R0 / TEST AN OLD NOS DESIGN!
981 / SET UP SERVICE ROUTINE ADDRESS
982 / *****
983 / TEST 6 TEST (ADDRESS) SMO,DM3 MOV INSTRUCTION
984 003716 000004 / *****
985 / TEST 6 SCOPE
986 / *****
987 / 11/40 *** ROM STATE 221 ***
988 003720 004015 JSR R0,(R5) / SET UP PARITY VECTOR SERVICE
989 003722 003752 A / ROUTINE ADDRESS
990 003724 011100 MOV @R1,R0 / SET UP FUR DATO
991 003726 010010 MOV R0,#R0 / DO THE DATO
992 003730 012737 003740 001332 MOV @,+10,@NSDDAT / STORE THE PC THAT SHOULD
993 / BE PUSHED ON THE STACK
994 / IF A PARITY ABORT OCCURS
995 003736 010030 MOV R0,@(R0)+ / DO A DATI
996 003740 002777 000004 175654 BIC @BIT2,@PARITY / WRITE NORMAL FOR EMT CALL
997 003746 100001 HLT +1 / DIDN'T ABORT
998 003750 000010 BR +22 / GO TO NEXT TEST
999 003752 002777 000005 175602 A1 BIC @BIT2|BIT0,@PARITY / WRITE NORMAL AND DISABLE
1000 003760 004037 011550 JSR R0,@CHECKLOC / CHECK FOR GOOD ABORT
1001 003764 100003 HLT +3 / ABORTED INCORRECTLY
1002 003766 012706 001100 MOV @STACK,SP / RESET THE STACK
1003 / *****
1004 / TEST 7 TEST (ADDRESS) SMO,DM5 MOV INSTRUCTION
1005 / *****
1006 / TEST 7 SCOPE
1007 / *****
1008 / 11/45 *** ROM STATE 231 ***
1009 / *****
1010 / 11/40 *** ROM STATE 207 ***
1011 003774 004015 JSR R0,(R5) / SET UP PARITY VECTOR SERVICE
1012 003776 100020 AB / ROUTINE ADDRESS
1013 004000 011100 MOV @R1,R0 / SET UP FUR DATO
1014 004002 010020 MOV R0,(R0)+ / DO THE DATO
1015 004004 012737 004014 001332 MOV @,+10,@NSDDAT / STORE THE PC THAT SHOULD
1016 / BE PUSHED ON THE STACK
1017 / IF A PARITY ABORT OCCURS
1018 004012 010050 MOV R0,@(R0) / DO A DATI
1019 004014 002777 000004 175600 BIC @BIT2,@PARITY / WRITE NORMAL FOR EMT CALL
1020 004022 100001 HLT +1 / DIDN'T ABORT
1021 004024 000010 BR +22 / GO TO NEXT TEST
1022 004026 002777 000005 175566 A01 BIC @BIT2|BIT0,@PARITY / WRITE NORMAL AND DISABLE
1023 004034 004037 011550 JSR R0,@CHECKLOC / CHECK FOR GOOD ABORT
1024 004042 100003 HLT +3 / ABORTED INCORRECTLY
1025 004044 012706 001100 MOV @STACK,SP / RESET THE STACK
    
```

```

1025 / *****
1026 / TEST 10 TEST (DATA) SMI,DM6 MOV INSTRUCTION
1027 / *****
1028 004046 000004 / *****
1029 / TEST 10 SCOPE
1030 / *****
1031 / 11/40 *** ROM STATE 206 ***
1032 004050 004015 JSR R0,(R5) / SET UP PARITY VECTOR SERVICE
1033 004052 004104 A1 / ROUTINE ADDRESS
1034 004054 011100 MOV @R1,R0 / SET UP FUR DATO
1035 004056 010010 MOV R0,#R0 / DO THE DATO
1036 004060 012737 004070 001332 MOV @,+10,@NSDDAT / STORE THE PC THAT SHOULD
1037 / BE PUSHED ON THE STACK
1038 / IF A PARITY ABORT OCCURS
1039 004066 011060 MOV @R0,+2(R0) / DO A DATI
1040 004072 002777 000004 175522 BIC @BIT2,@PARITY / WRITE NORMAL FOR EMT CALL
1041 004100 100001 HLT +1 / DIDN'T ABORT
1042 004102 000010 BR +22 / GO TO NEXT TEST
1043 004104 002777 000005 175510 A11 BIC @BIT2|BIT0,@PARITY / WRITE NORMAL AND DISABLE
1044 004112 004037 011550 JSR R0,@CHECKLOC / CHECK FOR GOOD ABORT
1045 004116 100003 HLT +3 / ABORTED INCORRECTLY
1046 004120 012706 001100 MOV @STACK,SP / RESET THE STACK
1047 / *****
1048 / TEST 11 TEST (ADDRESS) SMO,DM7 MOV INSTRUCTION
1049 / *****
1050 004124 000004 / *****
1051 / TEST 11 SCOPE
1052 / *****
1053 / 11/45 *** ROM STATE 231 ***
1054 / *****
1055 / 11/40 *** ROM STATE 207 ***
1056 004126 004015 JSR R0,(R5) / SET UP PARITY VECTOR SERVICE
1057 004130 004162 A2 / ROUTINE ADDRESS
1058 004132 011100 MOV @R1,R0 / SET UP FUR A DATO
1059 004134 010020 MOV R0,(R0)+ / DO THE DATO
1060 004136 012737 004150 001332 MOV @,+12,@NSDDAT / STORE THE PC THAT SHOULD
1061 / BE PUSHED ON THE STACK
1062 / IF A PARITY ABORT OCCURS
1063 004144 010070 MOV R0,@(R0) / DO A DATI
1064 004150 002777 000004 175444 BIC @BIT2,@PARITY / WRITE NORMAL FOR EMT CALL
1065 004156 100001 HLT +1 / DIDN'T ABORT
1066 004160 000010 BR +22 / GO TO NEXT TEST
1067 004162 002777 000005 175032 A21 BIC @BIT2|BIT0,@PARITY / WRITE NORMAL AND DISABLE
1068 004170 004037 011550 JSR R0,@CHECKLOC / CHECK FOR GOOD ABORT
1069 004174 100003 HLT +3 / ABORTED INCORRECTLY
1070 004176 012706 001100 MOV @STACK,SP / RESET THE STACK
1071 / *****
1072 / TEST 12 TEST (DATA) SMO,DM2 CMP INSTRUCTION
1073 / *****
1074 / TEST 12 SCOPE
1075 / *****
1076 / 11/40 *** ROM STATE 267 ***
1077 004204 004015 JSR R0,(R5) / SET UP PARITY VECTOR SERVICE
1078 004206 004236 B / ROUTINE ADDRESS
1079 004210 011100 MOV @R1,R0 / SET UP FUR DATO
    
```

```
1079 004010 010010      MOV    R0,R0          I/DO THE DATO
1080 004010 012737 004020 001332      MOV    R,+10,0000DAT I/STORE THE PC THAT SHOULD
1081                                     I/BE PUSHEW ON YK STACK
1082                                     I/IF A PARITY ABORT OCCURS
1083 004020 020020      CMP    R0,(R0)+      I/DO A DATIP, DATI
1084 004020 042777 000000 175070      BIC   @BIT2,@PARITY  I/WRITE NORMAL FOR ENT CALL
1085 004020 100001      HLT   +1             I/IDONT ABORT
1086 004020 000410      BR    +22           I/GO TO NEXT TEST
1087 004020 042777 000005 175356 B1    BIC   @BIT2IBIT0,@PARITY I/WRITE NORMAL AND DISABE
1088 004020 040037 011550      JSR   R0,@CHECKLOC  I/CHECK FOR GOOD ABORT
1089 004020 100003      HLT   +3            I/ABORTED INCORRECTLY
1090 004020 012706 001100      MOV    @STACK,SP    I/RESET THE STACK
1091                                     I/*****
1092 I/TEST 13      TEST (DATA) SMO,DMS CMP INSTRUCTION
1093 I/*****
1094 004020 000004      TBT13: SCOPE
1095 I/
1096 I/
1097 I/
1098 I/          11/40 *** ROM STATE 267 ***
1099 JSR    R0,(R5)      I/SET UP PARITY VECTOR SERVICE
1100 BR    @ROUTINE     I/ROUTINE ADDRESS
1101 MOV    @R1,R0      I/SET UP FOR DATO
1102 MOV    R0,(R0)+    I/DO THE DATO
1103 004020 012737 004030 001332      MOV    R,+10,0000DAT I/STORE THE PC THAT SHOULD
1104                                     I/BE PUSHEW ON THE STACK
1105                                     I/IF A PARITY ABORT OCCURS
1106 004020 042777 000004 175310      CMP    R0,(R0)
1107 004020 100001      HLT   +1             I/WRITE NORMAL FOR ENT CALL
1108 004020 000410      BR    +22           I/IDONT ABORT
1109 004020 042777 000005 175302 B0    BIC   @BIT2IBIT0,@PARITY I/WRITE NORMAL AND DISABE
1110 004020 040037 011550      JSR   R0,@CHECKLOC  I/CHECK FOR GOOD ABORT
1111 004020 100003      HLT   +3            I/ABORTED INCORRECTLY
1112 004020 012706 001100      MOV    @STACK,SP    I/RESET THE STACK
1113                                     I/*****
1114 I/TEST 14      TEST (DATA) SMO,DMS CMP INSTRUCTION
1115 I/*****
1116 004020 000004      TBT14: SCOPE
1117 I/
1118 I/
1119 I/          11/40 *** ROM STATE 267 ***
1120 JSR    R0,(R5)      I/SET UP PARITY VECTOR SERVICE
1121 BR    @ROUTINE     I/ROUTINE ADDRESS
1122 MOV    @R1,R0      I/SET UP FOR DATO
1123 MOV    R0,(R0)+    I/DO THE DATO
1124 004020 012737 004030 001332      MOV    R,+10,0000DAT I/STORE THE PC THAT SHOULD
1125                                     I/BE PUSHEW ON THE STACK
1126                                     I/IF A PARITY ABORT OCCURS
1127 004020 042777 177776      CMP    R0,@R0
1128 004020 042777 000004 175236      BIC   @BIT2,@PARITY  I/DO A DATIP, DATI
1129 004020 100001      HLT   +1             I/WRITE NORMAL FOR ENT CALL
1130 004020 000410      BR    +22           I/IDONT ABORT
1131 004020 042777 000005 175024 B11   BIC   @BIT2IBIT0,@PARITY I/WRITE NORMAL AND DISABE
1132 004020 040037 011550      JSR   R0,@CHECKLOC  I/CHECK FOR GOOD ABORT
```

```
1133 004020 104003      HLT   +3            I/ABORTED INCORRECTLY
1134 004020 012706 001100      MOV    @STACK,SP    I/RESET THE STACK
1135                                     I/*****
1136 I/TEST 15      TEST (DATA) SMO,DMS CMP INSTRUCTION
1137 I/*****
1138 004020 000004      TBT15: SCOPE
1139 I/
1140 I/
1141 I/          11/45 *** ROM STATE 175 ***
1142 JSR    R0,(R5)      I/SET UP PARITY VECTOR SERVICE
1143 BR    @ROUTINE     I/ROUTINE ADDRESS
1144 MOV    @R1,R0      I/SET UP FOR DATO
1145 MOV    R0,R0
1146 004020 012737 004030 001332      MOV    R,+10,0000DAT I/STORE THE PC THAT SHOULD
1147                                     I/BE PUSHEW ON THE STACK
1148                                     I/IF A PARITY ABORT OCCURS
1149 004020 042777 000000 175162      CMP    R0,R0
1150 004020 042777 000004 175162      BIC   @BIT2,@PARITY  I/DO A DATIP, DATI
1151 004020 100001      HLT   +1             I/WRITE NORMAL FOR ENT CALL
1152 004020 000410      BR    +22           I/IDONT ABORT
1153 004020 042777 000005 175150 B21   BIC   @BIT2IBIT0,@PARITY I/WRITE NORMAL AND DISABE
1154 004020 040037 011550      JSR   R0,@CHECKLOC  I/CHECK FOR GOOD ABORT
1155 004020 100003      HLT   +3            I/ABORTED INCORRECTLY
1156 004020 012706 001100      MOV    @STACK,SP    I/RESET THE STACK
1157                                     I/*****
1158 I/TEST 16      TEST (DATA) SMO,DMS CMP INSTRUCTION
1159 I/*****
1160 004020 000004      TBT16: SCOPE
1161 I/
1162 I/
1163 I/          11/45 *** ROM STATE 177 ***
1164 JSR    R0,(R5)      I/SET UP PARITY VECTOR SERVICE
1165 BR    @ROUTINE     I/ROUTINE ADDRESS
1166 MOV    @R1,R0      I/SET UP FOR DATO
1167 MOV    R0,(R0)+    I/DO THE DATO
1168 004020 042777 000004 175116      BIC   @BIT2,@PARITY  I/WRITE NORMAL
1169 004020 011110      MOV    @R1,R0      I/WRITE ADDRESS NORMAL (DATI)
1170 004020 042777 000004 175105      BIC   @BIT2,@PARITY  I/WRITE OTHER PARITY
1171 004020 012737 004020 001332      MOV    R,+10,0000DAT I/STORE THE PC THAT SHOULD
1172                                     I/BE PUSHEW ON THE STACK
1173                                     I/IF A PARITY ABORT OCCURS
1174 004020 042777 000000 175070      CMP    R0,(R0)+    I/DO A DATIP, DATI
1175 004020 042777 000004 175070      BIC   @BIT2,@PARITY  I/WRITE NORMAL FOR ENT CALL
1176 004020 100001      HLT   +1             I/IDONT ABORT
1177 004020 000410      BR    +22           I/GO TO NEXT TEST
1178 004020 042777 000005 175056 B31   BIC   @BIT2IBIT0,@PARITY I/WRITE NORMAL AND DISABE
1179 004020 040037 011550      JSR   R0,@CHECKLOC  I/CHECK FOR GOOD ABORT
1180 004020 100003      HLT   +3            I/ABORTED INCORRECTLY
1181 004020 012706 001100      MOV    @STACK,SP    I/RESET THE STACK
1182                                     I/*****
1183 I/TEST 17      TEST (ADDRESS) SMO,DMS CMP INSTRUCTION
1184 I/*****
1185 004020 000004      TBT17: SCOPE
1186 I/
1187 I/          11/45 *** ROM STATE 221 ***
```

```
1167 /
1168 /
1169 004560 004015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
1170 004560 004018 B4 /ROUTINE ADDRESS
1171 004560 011100 MOV R01,R0 /SET UP FOR A DAYO
1172 004566 010010 MOV R0,0R0 /DO THE DAYO
1173 004570 012737 004003 001332 MOV R,+10,0R0DDAT /STORE THE PC THAT SHOULD
/BE PUSHED ON THE STACK
1174 /
1175 /
1176 004576 040030 CMP R0,(R0)+ /IF A PARITY ABORT OCCURS
/DO A DATA
1177 004600 042777 000004 175014 BIC 0BIT2,0PARITY /WRITE NORMAL FOR ENT CALL
1178 004606 104001 HLT +1 /DIDN'T ABORT
1179 004610 000410 BR +22 /GO TO NEXT TEST
1200 004612 042777 000005 175002 04: BIC 0BIT2IBIT0,0PARITY /WRITE NORMAL AND DISABLE
1201 004620 004037 011550 JSR R0,0CHECKLOC /CHECK FOR GOOD ABORT
1202 004624 104003 HLT +3 /ABORTED INCORRECTLY
1203 004626 012706 001100 MOV 0STACK,SP /RESET THE STACK
/*****
/TEST 20 TEST (DATA) SM0,DMS CMP INSTRUCTION
/*****
TST20: SCOPE
/
/ 11/05 *** ROM STATE 177 ***
/
/
1210 /
1211 11/06 *** ROM STATE 267 ***
1212 004634 004015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
1213 004636 004706 B5 /ROUTINE ADDRESS
1214 004640 011100 MOV R01,R0 /SET UP FOR A DAYO
1215 004642 042777 000004 174752 BIC 0BIT2,0PARITY /WRITE NORMAL
1216 004650 040000 177776 MOV R0,(R0) /DO THE DAYO
1217 004654 042777 000004 174740 BIS 0BIT0,0PARITY /WRITE OTHER PARITY
1218 004662 011110 MOV R01,0R0 /DO THE DAYO
1219 004664 012737 004674 001332 MOV R,+10,0R0DDAT /STORE THE PC THAT SHOULD
/BE PUSHED ON THE STACK
1220 /
1221 /
1222 004672 040030 CMP R0,(R0) /IF A PARITY ABORT OCCURS
/DO A DATA, DAYIP
1223 004674 042777 000004 174720 BIC 0BIT2,0PARITY /WRITE NORMAL FOR ENT CALL
1224 004702 104001 HLT +1 /DIDN'T ABORT
1225 004704 000410 BR +22 /GO TO NEXT TEST
1226 004706 042777 000005 174706 04: BIC 0BIT2IBIT0,0PARITY /WRITE NORMAL AND DISABLE
1227 004714 004037 011550 JSR R0,0CHECKLOC /CHECK FOR GOOD ABORT
1228 004722 104003 HLT +3 /ABORTED INCORRECTLY
1229 004724 012706 001100 MOV 0STACK,SP /RESET THE STACK
/*****
/TEST 21 TEST (ADDRESS) SM0,DMS CMP INSTRUCTION
/*****
TST21: SCOPE
/
/ 11/45 *** ROM STATE 251 ***
/
/
1236 11/46 *** ROM STATE 264 ***
1237 004730 004015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
1238 004732 004766 B6 /ROUTINE ADDRESS
1239 004734 011100 MOV R01,R0 /SET UP FOR A DAYO
1240 004736 010010 MOV R0,0R0 /DO THE DAYO
1241 004740 002700 ADD 02,R0
```

```
1241 004744 012737 004754 001332 MOV R,+10,0R0DDAT /STORE THE PC THAT SHOULD
/BE PUSHED ON THE STACK
1242 /
1243 /
1244 004752 040030 CMP R0,(R0) /IF A PARITY ABORT OCCURS
/DO A DATA
1245 004754 042777 000004 174640 BIC 0BIT2,0PARITY /WRITE NORMAL FOR ENT CALL
1246 004762 104001 HLT +1 /DIDN'T ABORT
1247 004764 000410 BR +22 /GO TO NEXT TEST
1248 004766 042777 000005 174626 04: BIC 0BIT2IBIT0,0PARITY /WRITE NORMAL AND DISABLE
1249 004774 004037 011550 JSR R0,0CHECKLOC /CHECK FOR GOOD ABORT
1250 005000 104003 HLT +3 /ABORTED INCORRECTLY
1251 005002 012706 001100 MOV 0STACK,SP /RESET THE STACK
/*****
/TEST 22 TEST (DATA) SM2,DMS CMP INSTRUCTION
/*****
TST22: SCOPE
/
/ 11/45 *** ROM STATE 27 ***
/
/
1259 11/46 *** ROM STATE 250 ***
1260 005010 004015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
1261 005012 005042 C /ROUTINE ADDRESS
1262 005014 011100 MOV R01,R0 /SET UP FOR A DAYO
1263 005016 010010 MOV R0,0R0 /DO THE DAYO
1264 005020 012737 005030 001332 MOV R,+10,0R0DDAT /STORE THE PC THAT SHOULD
/BE PUSHED ON THE STACK
1265 /
1266 /
1267 005026 042000 CMP (R0)+,R0 /IF A PARITY ABORT OCCURS
/DO A DATA
1268 005030 042777 002004 174564 BIC 0BIT2,0PARITY /WRITE NORMAL FOR ENT CALL
1269 005036 104001 HLT +1 /DIDN'T ABORT
1270 005040 000410 BR +22 /GO TO NEXT TEST
1271 005042 042777 000005 174552 04: BIC 0BIT2IBIT0,0PARITY /WRITE NORMAL AND DISABLE
1272 005050 004037 011550 JSR R0,0CHECKLOC /CHECK FOR GOOD ABORT
1273 005054 104003 HLT +3 /ABORTED INCORRECTLY
1274 005056 012706 001100 MOV 0STACK,SP /RESET THE STACK
/*****
/TEST 23 TEST (DATA) SM4,DMS CMP INSTRUCTION
/*****
TST23: SCOPE
```

1276

11/25 **** ROM STATE 27 ****

1279

1280

1281 005064 004015

1282 005066 005116

1283 005070 011100

1284 005072 016020

I

I

11/40 **** ROM STATE 280 ****

JSR R0,(R5)

CC

MOV @R1,R0

MOV R0,(R0)*

ISET UP PARITY VECTOR SERVICE

IROUTINE ADDRESS

ISET UP FOR DATA

IADD THE DATA

```
1209 005074 012737 005100 001332 MOV 0,+10,0050DAT STORE THE PC THAT SHOULD  
1210 1746 BE PUSHED ON THE STACK  
1211 1767 IF A PARITY ABORT OCCURS  
1212 005102 004000 000000 174510 CHP *(R0),R0 ADD A DATA  
1213 005104 002777 000004 174510 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1214 005112 104001 HLT #1 IDONT ABORT  
1215 005114 004010 BR +22 GO TO NEXT TEST  
1216 005116 002777 000004 174076 C# BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1217 005124 004037 011550 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1218 005130 104003 HLT #3 ABORTED INCORRECTLY  
1219 005132 012700 001100 MOV #STACK,SP RESET THE STACK  
1220 *****  
1221 ITEST 20 TEST (DATA) S#5,D#0 CMP INSTRUCTION  
1222 *****  
1223 TST20 SCOPE  
1224 I  
1225 11/45 *** ROM STATE 146 ***  
1226 I  
1227 I  
1228 11/40 *** ROM STATE 250 ***  
1229 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1230 C1 ROUTINE ADDRESS  
1231 MOV #R1,R0 ISET UP FOR DATA  
1232 MOV R0,(R0)* ADD THE DATA (DATA OTHER PARITY)  
1233 BIC #B12,0PARITY WRITE NORMAL  
1234 MOV #R1,R0 ADD A DATA (ADDRESS NORMAL)  
1235 BIC #B12,0PARITY WRITE OTHER PARITY  
1236 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1237 HLT #3 ABORTED INCORRECTLY  
1238 MOV #STACK,SP RESET THE STACK  
1239 *****  
1240 ITEST 25 TEST (DATA) S#5,D#0 CMP INSTRUCTION  
1241 *****  
1242 TST25 SCOPE  
1243 I  
1244 11/45 *** ROM STATE 146 ***  
1245 I  
1246 I  
1247 11/40 *** ROM STATE 250 ***  
1248 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1249 C2 ROUTINE ADDRESS  
1250 MOV #R1,R0 ISET UP FOR DATA  
1251 BIC #B12,0PARITY WRITE NORMAL  
1252 MOV R0,(R0)* ADD A DATA (ADDRESS NORMAL)  
1253 BIC #B12,0PARITY WRITE OTHER PARITY  
1254 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1255 HLT #3 ABORTED INCORRECTLY  
1256 MOV #STACK,SP RESET THE STACK  
1257 *****  
1258 ITEST 26 TEST (DATA) S#1,D#0 CMP INSTRUCTION  
1259 *****  
1260 TST26 SCOPE  
1261 I  
1262 11/45 *** ROM STATE 27 ***  
1263 I  
1264 I  
1265 11/40 *** ROM STATE 250 ***  
1266 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1267 C3 ROUTINE ADDRESS  
1268 MOV #R1,R0 ISET UP FOR DATA  
1269 MOV R0,(R0)* ADD THE DATA  
1270 MOV 0,+10,0050DAT STORE THE PC THAT SHOULD  
1271 BE PUSHED ON THE STACK  
1272 IF A PARITY ABORT OCCURS  
1273 ADD A DATA  
1274 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1275 HLT #1 IDONT ABORT  
1276 BR +22 GO TO NEXT TEST  
1277 BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1278 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1279 HLT #3 ABORTED INCORRECTLY  
1280 MOV #STACK,SP RESET THE STACK  
1281 *****  
1282 ITEST 27 TEST (DATA) S#6,D#0 CMP INSTRUCTION  
1283 *****  
1284 TST27 SCOPE  
1285 I  
1286 11/45 *** ROM STATE 142 ***  
1287 I  
1288 I  
1289 11/40 *** ROM STATE 250 ***  
1290 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1291 C4 ROUTINE ADDRESS  
1292 MOV #R1,R0 ISET UP FOR DATA  
1293 MOV R0,(R0)* ADD THE DATA  
1294 MOV 0,+12,0050DAT STORE THE PC THAT SHOULD  
1295 BE PUSHED ON THE STACK  
1296 IF A PARITY ABORT OCCURS  
1297 ADD A DATA  
1298 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1299 HLT #1 IDONT ABORT  
1300 BR +22 GO TO NEXT TEST  
1301 BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1302 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1303 HLT #3 ABORTED INCORRECTLY  
1304 MOV #STACK,SP RESET THE STACK  
1305 *****  
1306 ITEST 30 TEST (ADDRESS) S#7,D#0 CMP INSTRUCTION  
1307 *****  
1308 TST30 SCOPE  
1309 I  
1310 11/45 *** ROM STATE 142 ***  
1311 I  
1312 I
```

```
1339 005272 042777 000004 174322 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1340 005300 104001 HLT #1 IDONT ABORT  
1341 005302 004010 BR +22 GO TO NEXT TEST  
1342 005304 042777 000005 174310 C# BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1343 005312 004037 011550 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1344 005316 104003 HLT #3 ABORTED INCORRECTLY  
1345 005320 012700 001100 MOV #STACK,SP RESET THE STACK  
1346 *****  
1347 ITEST 26 TEST (DATA) S#1,D#0 CMP INSTRUCTION  
1348 *****  
1349 TST26 SCOPE  
1350 I  
1351 11/45 *** ROM STATE 27 ***  
1352 I  
1353 I  
1354 11/40 *** ROM STATE 250 ***  
1355 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1356 C5 ROUTINE ADDRESS  
1357 MOV #R1,R0 ISET UP FOR DATA  
1358 MOV R0,(R0)* ADD THE DATA  
1359 MOV 0,+10,0050DAT STORE THE PC THAT SHOULD  
1360 BE PUSHED ON THE STACK  
1361 IF A PARITY ABORT OCCURS  
1362 ADD A DATA  
1363 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1364 HLT #1 IDONT ABORT  
1365 BR +22 GO TO NEXT TEST  
1366 BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1367 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1368 HLT #3 ABORTED INCORRECTLY  
1369 MOV #STACK,SP RESET THE STACK  
1370 *****  
1371 ITEST 27 TEST (DATA) S#6,D#0 CMP INSTRUCTION  
1372 *****  
1373 TST27 SCOPE  
1374 I  
1375 11/45 *** ROM STATE 142 ***  
1376 I  
1377 I  
1378 11/40 *** ROM STATE 250 ***  
1379 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE  
1380 C6 ROUTINE ADDRESS  
1381 MOV #R1,R0 ISET UP FOR DATA  
1382 MOV R0,(R0)* ADD THE DATA  
1383 MOV 0,+12,0050DAT STORE THE PC THAT SHOULD  
1384 BE PUSHED ON THE STACK  
1385 IF A PARITY ABORT OCCURS  
1386 ADD A DATA  
1387 BIC #B12,0PARITY WRITE NORMAL FOR EMT CALL  
1388 HLT #1 IDONT ABORT  
1389 BR +22 GO TO NEXT TEST  
1390 BIC #B12|B10,0PARITY WRITE NORMAL AND DISABLE  
1391 JSR R0,0CHECKLOC CHECK FOR GOOD ABORT  
1392 HLT #3 ABORTED INCORRECTLY  
1393 MOV #STACK,SP RESET THE STACK  
1394 *****  
1395 ITEST 30 TEST (ADDRESS) S#7,D#0 CMP INSTRUCTION  
1396 *****  
1397 TST30 SCOPE  
1398 I  
1399 11/45 *** ROM STATE 142 ***  
1400 I  
1401 I
```

```
1393 005456 000004          TEST1 SCOPE
1394          |
1395          |
1396          |
1397          |
1398 005442 000017          |
1399 005442 000018          |
1400 005440 011100          |
1401 005440 010000          |
1402 005470 012737 005402 001370  MOV     R0,(R0)+
1403          |
1404 005470 007000 177776      CHP     @-2(R0),R0
1405 005500 002777 000004 174112  BIC     @BIT2,@PARITY
1406 005510 100001          HLT     +1
1407 005510 000010          BR      +22
1408 005514 002777 000005 174100  C7:    BIC     @BIT2IBIT0,@PARITY
1409 005520 000037 011550      JSR     R0,@CHECKLOC
1410 005526 100003          HLT     +3
1411 005530 012700 001100      MOV     @STACK,SP
1412          |
1413          |*****|
1414          |TEST 31  TEST (ADDRESS) SHS,DMS  CMP INSTRUCTION|
1415          |*****|
1416          |
1417          |
1418          |
1419          |
1420 005530 000004          TEST1 SCOPE
1421          |
1422          |
1423          |
1424          |
1425          |
1426          |
1427 005536 000015          |
1428 005540 005570          |
1429 005540 011100          |
1430 005540 010001          |
1431 005546 000010          |
1432 005570 002777 000005 174020  C9:    BIC     @BIT2IBIT0,@PARITY
1433 005576 000037 011550      JSR     R0,@CHECKLOC
1434 005602 100003          HLT     +3
1435 005606 012700 001100      MOV     @STACK,SP
1436          |
1437          |*****|
1438          |TEST 32  TEST (ADDRESS) SHS,DMS  CMP INSTRUCTION|
1439          |*****|
1440          |
1441          |
1442          |
1443          |
1444          |
1445          |
1446          |
1447          |
1448          |
1449          |
1450          |
1451          |
1452          |
1453          |
1454          |
1455          |
1456          |
1457          |
1458          |
1459          |
1460          |
1461          |
1462          |
1463          |
1464          |
1465          |
1466          |
1467          |
1468          |
1469          |
1470          |
1471          |
1472          |
1473          |
1474          |
1475          |
1476          |
1477          |
1478          |
1479          |
1480          |
1481          |
1482          |
1483          |
1484          |
1485          |
1486          |
1487          |
1488          |
1489          |
1490          |
1491          |
1492          |
1493          |
1494          |
1495          |
1496          |
1497          |
1498          |
1499          |
1500          |
```

```
1447          |
1448 005630 005000          |
1449 005632 002777 000004 173702  CMP     @-(R0),R0
1450 005640 100001          BIC     @BIT2,@PARITY
1451 005642 000010          HLT     +1
1452 005646 002777 000005 173750  C7:    BR      +22
1453 005650 002777 000005 173750  C7:    BIC     @BIT2IBIT0,@PARITY
1454 005652 000037 011550      JSR     R0,@CHECKLOC
1455 005656 100003          HLT     +3
1456 005660 012700 001100      MOV     @STACK,SP
1457          |
1458          |*****|
1459          |TEST 33  TEST (DATA) SHY,DMS  CMP INSTRUCTION|
1460          |*****|
1461          |
1462          |
1463          |
1464          |
1465          |
1466          |
1467          |
1468          |
1469          |
1470          |
1471          |
1472          |
1473          |
1474          |
1475          |
1476          |
1477          |
1478          |
1479          |
1480          |
1481          |
1482          |
1483          |
1484          |
1485          |
1486          |
1487          |
1488          |
1489          |
1490          |
1491          |
1492          |
1493          |
1494          |
1495          |
1496          |
1497          |
1498          |
1499          |
1500          |
```



```

1609
1610
1611
1612 006012 013006 001476      MOV      #RNEWSTK,SP      /GET THE STACK IN PARITY
1613
1614 006016 017702 173224      MOV      ENTERAD,R4      /MEMORY AREA
1615
1616 006022 005302      DEC      R2              /GET THE INTERLEAVE FACTOR
1617
1618 006024 010546      MOV      R5,=(SP)        /FOR THIS CONTROLLER
1619 006026 005702      TST      R2              /CALCULATE NO. OF PARAMETERS
1620
1621 006030 001404      BEO      Z0              /TO BE PUSHED ON THE STACK
1622 006032 005302      DEC      R2              /ANY PARAMETERS TO BE
1623 006034 012706 000001      MOV      #1,=(SP)        /PUSHED ON THE STACK
1624 006040 000772      BR       B1              /BRANCH IF NO
1625 006042 002777 000004 173352 001  MOV      #B10,SPARITY    /SUBTRACT 1 FROM PARAMETER COUNT
1626 006050 017702 173172      MOV      ENTERAD,R4      /PUSH PARAMETER ONTO STACK
1627
1628 006054 005302      DEC      R2              /GO BACK TO SEE IF ANY MORE
1629
1630 006058 002777 000004 173352 001  MOV      #B10,SPARITY    /WRITE OTHER PARITY
1631
1632 006062 002777 000004 173352 001  MOV      ENTERAD,R4      /GET THE INTERLEAVE FACTOR
1633
1634 006066 005302      DEC      R2              /FOR THIS CONTROLLER
1635
1636 006070 002777 000004 173352 001  MOV      ENTERAD,R4      /CALCULATE NO. OF PARAMETERS
1637
1638 006074 010546      MOV      R2,=(SP)        /THAT HERE TO BE PUSHED ON THE
1639 006076 005702 001042      TST      #CPU00         /STACK
1640 006082 001002      BNE      Z0              /CALCULATE THE CORRESPONDING
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
  
```


```

/*****
/*****
/
/ THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS
/
/ 1ST PUSH = ADDRESS OF THE TAG 'VECSET' (OTHER PARITY)
/           THIS ADDRESS WOULD BE PLACED IN
/           R5 UPON COMPLETION OF 'RTS R5'
/ 2ND PUSH
/           :
/           : NO. OF PARAMETERS AS A FUNCTION
/           : OF MEMORY INTERLEAVING
/
/ NTH PUSH
/ NTH +1 PUSH = MARK INSTRUCTION (NORMAL)
/ LAST PUSH = OLD PC FROM THE 'JSR' (NORMAL)
/
/ NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO RESTORE
/       R5 CONTENTS (1ST PUSH)
/
/       WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
/       PROPERLY UPDATED, THUS GIVING,
/
/ 1ST PUSH = R5 FROM THE PARITY ERROR
/ 2ND PUSH = PC FROM THE PARITY ERROR
/
/*****
/*****
/*****
/*****
1695
1696
1697
1698 006346 012705 011450      MOV      #VECSET,R5      /RESTORE THE PARITY VECTOR
1699
1700
1701
1702
1703
1704 006352 000004      TST001  #CPU00         /SERVICE ADDRESS SETUP ROUTINE
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
  
```



```

1935
1936
1937
1938
1939
1940 007106 000004
1941
1942
1943
1944 007110 000015
1945 007112 007150
1946 007114 012726 000300
1947 007120 012706 007134
1948 007126 012737 007134 001332
1949
1950
1951 007132 000000
1952 007134 104001
1953 007136 000010
1954 007140 000077 000000 172400
1955 007146 000000
1956 007150 000077 000001 172400
1957 007154 000037 011550
1958 007162 100003
1959 007166 012706 001476
1960
1961
1962

```

```

*****
*****
*****
*****

```

```

//THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
//
//1ST PUSH = OLD PS FOR 'RTI' (OTHER PARITY)
//2ND PUSH = OLD PC FOR 'RTI' (OTHER PARITY)
//
//NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO 'POP'
//      THE OLD PS (1ST PUSH)
//
//      THE 2ND PUSH IS REWRITTEN NORMAL BEFORE
//      DURING THE 'RTI'
//
//      WHEN THE PARITY ERROR OCCURS THE STACK
//      POINTER IS PROPERLY UPDATED AND THE NEW
//      PS AND PC FROM THE RTI INSTRUCTION IS
//      PUSHED ONTO THE STACK, THUS GIVING,
//
//1ST PUSH = NEW PS FROM 'RTI' (NORMAL)
//2ND PUSH = NEW PC FROM 'RTI' (NORMAL)

```

```

//
//
//*****
//*****
//*****
//*****
1993
1994
1995
1996
1997
1998 007170 000004
1999
2000
2001
2002 007172 000015
2003 007174 007242
2004 007176 012746 000300
2005 007202 012746 007216
2006 007206 012737 007216 001332
2007
2008
2009 007214 000002
2010 007216 104001
2011 007220 000016
2012 007222 002777 000005 172372
2013 007230 012606
2014
2015 007232 002777 000001 172362
2016 007240 000002
2017 007242 002777 000001 172352
2018 007250 000037 011550
2019 007254 104003
2020 007256 012706 001100
2021
2022
2023
2024
2025
2026 007262 000004
2027
2028
2029
2030
2031 007264 000015
2032 007266 007320
2033 007270 011100
2034 007272 010020
2035 007274 012737 007306 001332
2036
2037
2038 007302 000000 177776
2039 007306 002777 000000 172306
2040 007314 104001

```

```
2041 007316 000010 BR +20 IGO TO NEXT TEST
2042 007318 000010 BIC #BIT2,0PARITY IWRITE NORMAL AND DISABLE
2043 007320 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2044 007322 000003 HLT +3 IABORTED INCORRECTLY
2045 007324 001100 MOV #STACK,SP IRESET THE STACK
I*****
I TEST 47 TEST DMI INTO INSTRUCTION
I*****
I TEST 48 SCOPE I1/45 *** ROM STATE 175 ***
I
I
I 11/40 *** ROM STATE 207 ***
2053 007302 000015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2054 007304 000015 R IROUTINE ADDRESS
2055 007306 001100 MOV #R1,R0 ISET UP FOR A DATA
2056 007308 000010 MOV R0,(R0) IDO THE DATA
2057 007310 001332 MOV #+10,#500DAT ISTORE THE PC THAT SHOULD
2058 IBE PUSHED ON THE STACK
2059 IIF A PARITY ABORT OCCURS
2060 IDO A DATA
2061 IWRITE NORMAL FOR EMT CALL
2062 IDON'T ABORT
2063 IGO TO NEXT TEST
2064 007314 000015 BIC #BIT2,0PARITY IWRITE NORMAL AND DISABLE
2065 007316 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2066 007318 000003 HLT +3 IABORTED INCORRECTLY
2067 007320 001100 MOV #STACK,SP IRESET THE STACK
I*****
I TEST 50 TEST DMS CLRD INSTRUCTION
I*****
I TEST 51 SCOPE I1/45 *** ROM STATE 221 ***
I
I
I 11/40 *** ROM STATE 264 ***
2075 007416 000015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2076 007418 000015 R IROUTINE ADDRESS
2077 007420 001100 MOV #R1,R0 ISET UP FOR A DATA
2078 007422 000010 MOV R0,(R0) IDO THE DATA
2079 007424 001332 MOV #+10,#500DAT ISTORE THE PC THAT SHOULD
2080 IBE PUSHED ON THE STACK
2081 IIF A PARITY ABORT OCCURS
2082 IDO A DATA
2083 IWRITE NORMAL FOR EMT CALL
2084 IDON'T ABORT
2085 IGO TO NEXT TEST
2086 007428 000015 BIC #BIT2,0PARITY IWRITE NORMAL AND DISABLE
2087 007430 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2088 007432 000003 HLT +3 IABORTED INCORRECTLY
2089 007434 001100 MOV #STACK,SP IRESET THE STACK
I*****
I TEST 51 TEST SMI SUBTRACT INSTRUCTION
I*****
I TEST 52 SCOPE I1/45 *** ROM STATE 27 ***
I
I
```

```
2095 I
2096 I 11/40 *** ROM STATE 256 ***
2097 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2098 R IROUTINE ADDRESS
2099 MOV #R1,R0 ISET UP FOR A DATA
2100 MOV R0,(R0) IDO THE DATA
2101 MOV #+10,#500DAT ISTORE THE PC THAT SHOULD
2102 IBE PUSHED ON THE STACK
2103 IIF A PARITY ABORT OCCURS
2104 IDO A DATA
2105 IWRITE NORMAL FOR EMT CALL
2106 IDON'T ABORT
2107 IGO TO NEXT TEST
2108 007526 000015 BIC #BIT2,0PARITY IWRITE NORMAL AND DISABLE
2109 007528 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2110 007530 000003 HLT +3 IABORTED INCORRECTLY
2111 007532 001100 MOV #STACK,SP IRESET THE STACK
I*****
I TEST 52 TEST (DATA) SMS BISS INSTRUCTION
I*****
I TEST 53 SCOPE I1/45 *** ROM STATE 177 ***
I
I
I 11/40 *** ROM STATE 245 ***
2119 007550 000015 JSR R0,(R5) ISET UP PARITY VECTOR SERVICE
2120 007552 000015 R IROUTINE ADDRESS
2121 007554 001100 MOV #R1,R0 ISET UP FOR A DATA
2122 007556 000010 MOV R0,(R0) IDO THE DATA
2123 007558 001332 MOV #+10,#500DAT ISTORE THE PC THAT SHOULD
2124 IBE PUSHED ON THE STACK
2125 IIF A PARITY ABORT OCCURS
2126 IDO A DATA
2127 007572 000015 BIC #+2(R0),2(M0) IWRITE NORMAL FOR EMT CALL
2128 007574 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2129 007576 000003 HLT +3 IABORTED INCORRECTLY
2130 007578 001100 MOV #+22 IGO TO NEXT TEST
2131 007602 000015 BIC #BIT2,0PARITY IWRITE NORMAL AND DISABLE
2132 007604 000037 JSR R0,#CHECKLOC ICHECK FOR GOOD ABORT
2133 007606 000003 HLT +3 IABORTED INCORRECTLY
2134 007608 001100 MOV #STACK,SP IRESET THE STACK
I*****
I TEST 53 NEXT 4 TESTS ARE NON-INTERLEAVE AN DEPENDENT
I*****
I TEST 54 SCOPE I1/45 *** ROM STATE 177 ***
I
I
I 11/40 *** ROM STATE 245 ***
2137 007624 000004 CHM #17360,R1 IHAVE WE PARITY IN THE LOWER 4K?
2138 007626 001332 CHM #17360,R1 IAND IS THE SELECTED REGISTER
2139 IGOVERNING THE 4K AREA?
2140 ITHIS COMPARE IS IF THE RT11
2141 IISN'T ENABLED DURING PROGRAM
2142 IEXECUTION
2143 IYES = PROCEED TO NEXT TESTS
2144 007632 001100 BHI B5 IWHICH DEPEND ON PARITY IN LOWER 4K
2145 ITHE ABOVE COMPARE DIDN'T
2146 007634 000015 CHM #140,#500140 ICHECK - " DO THIS COMPARE TO
2147 ISEE IF IT WAS BECAUSE THE
```

2194
2195 007640 001401 BEQ 25
2196 007644 000004 BR 45
2197 007646 002177 000001 171072 201 CMP #1,ENTERAD
2198 007650 001405 BEQ 35
2199 007656 002107 000004 171016 431 ADD #4,SYSTNH
2200 007664 000137 010426 JMP #REDD+06
2201 007670 000237 001024 341 INC #PCPCORZONES

TABLE WAS CREATED WITH MEMORY
MANAGEMENT RUNDON
PROCEED TO NEXT TESTS IF THIS
ICMPARE LOCKS
IF THE REGISTER UNDER TEST IS NOT
ICONTROLLING THE LOWER ARITH
ICGO TO RETRIEVAL SYSTEM AND
IF JUMP OVER THE (4) 4K DEPENDENT
ITESTS
IF IS THIS CONTROLLER
IF INTERLEAVED?
IF BRANCH IF NO
IF SET SYSTEM TO PROPER VALUE
IF SINCE THE NEXT 4 TESTS WILL
IF BE SKIPPED
IF JUMP TO 1ST INDEX WORD TEST
IF SET FLAG INDICATING TO CHECKLOC
IF ROUTINE THAT PS AND PC FETCH
IF AND RED & YELLOW ZONE AREAS
IF ARE GOING TO BE TESTED

/ THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
/ 1ST PUSH = OLD PS FROM ERROR TRAP (NORMAL)
/ 2ND PUSH = OLD PC FROM ERROR TRAP (NORMAL)
/ NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PS
/ WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
/ ALTERED FROM THE ORIGINAL ERROR TRAP, THUS GIVING,
/ 1ST PUSH = OLD PS FROM ERROR TRAP (NORMAL)
/ 2ND PUSH = NEW PS FROM THE PARITY ERROR
/ 3RD PUSH = NEW PC FROM THE PARITY ERROR

2197
2198
2199
2200
2201
2202 007674 000004

/ TEST 5A TEST NEW 'PS' FETCH
/ *****
/ TEST 5A SCOPED

2203
2204
2205 007676 004019
2206 007700 007750 JSR R0,(R5)
2207 007702 002177 000004 171112 TB
2208 007710 012737 007750 000010 BIC #BIT2,@PARITY
2209 007716 012737 007750 001332 MOV #1,@SRDDAT
2210 007724 002777 000004 171670 BIS #BIT2,@PARITY
2211 007732 012737 000300 000012 MOV #340,@RESVEC+2
2212 007740 002777 000004 171654 BIC #BIT2,@PARITY
2213 007746 007600 Y000
2214 007750 104001 Y1 HLT +1
2215 007752 000406 BR .+16
2216 007754 002777 000001 171640 TWI BIC #BIT0,@PARITY
2217 007762 000037 011550 JSR R0,@CHECKLOC
2218 007766 104003 HLT +3
2219 007770 012706 001100 MOV #STACK,SP

11/45 *** ROM STATE 357 ***
11/40 *** ROM STATE 115 ***
ISET UP PARITY VECTOR SERVICE
IF ROUTINE ADDRESS
IF WRITE NORMAL
IF RESERVED INSTRUCTION TIMEOUT
IF VECTOR ADDRESS
IF STORE THE PC THAT SHOULD
IF BE PUSHED ON THE STACK
IF IF A PARITY ABORT OCCURS
IF WRITE OTHER PARITY
IF NEW 'PS' FOR ERROR TRAP
IF WRITE NORMAL
IF NON-RECOGNIZABLE OP-CODE
IF SHOULD ATTEMPT TO TRAP TO 'Y':
IF IDION'T ABORT
IF GO TO RESET THE STACK
IF DISABLE PARITY
IF CHECK FOR GOOD ABORT
IF ABORTED INCORRECTLY
IF RESET THE STACK

/ THE CONTENTS OF THE STACK FOR THE NEXT TEST ARE AS FOLLOWS:
/ 1ST PUSH = OLD PS FROM ERROR TRAP (NORMAL)
/ 2ND PUSH = OLD PC FROM ERROR TRAP (NORMAL)
/ NOTE: THE TEST SHOULD FAIL ON ATTEMPT TO FETCH THE NEW PC
/ WHEN THE PARITY ERROR OCCURS THE STACK POINTER IS
/ NOT ALTERED FROM THE ORIGINAL ERROR TRAP AND THE
/ PC FOR THE PARITY ERROR IS THE OLD PS, THUS GIVING,
/ 1ST PUSH = OLD PS FROM ERROR TRAP (NORMAL)
/ 2ND PUSH = OLD PS FROM ERROR TRAP (NORMAL)
/ IN OTHER WORDS THE ORIGINAL ERROR TRAP AND VECTOR IS LOST!!!

```

2365 010166 042777 000001 171330 V41 BIC #BIT0,0PARITY 1DISABLE PARITY
2366 010170 042787 000372 TST 00372 1STACK VIOLATION PICKED UP -
2367 010200 041300 UNE 10 1HAS THE PARITY ABORT?
2368 010200 040010 HLT *12 1BRANCH TO YES
2369 010200 040010 HLT *12 1STACK VIOLATION PICKED UP
2370 010200 040010 HLT *12 1BUT ABORT NOT RECOGNIZED
2371 010200 040010 BR YELLOW 1GO TO RESET STACK AND RESTORE
2372 010200 040010 BR YELLOW 1TIMEOUT VECTORS
2373 010200 040010 BR YELLOW 1RESET STACK BACK TO NORMAL
2374 010210 040010 BR YELLOW 1PUSH ABORT PC ONTO NORMAL
2375 010210 040010 BR YELLOW 1STACK OUT OF VIOLATION (NEED
2376 010210 040010 BR YELLOW 1FOR CHECKING PURPOSES
2377 010210 040010 BR YELLOW 1CHECK FOR GOOD ABORT
2378 010210 040010 BR YELLOW 1ABORTED INCORRECTLY
2379 010210 040010 BR YELLOW 1RESET STACK BACK TO NORMAL
2380 010210 040010 BR YELLOW 1RESTORE CONTENTS OF LOC. 4
2381 010210 040010 BR YELLOW 1RESTORE CONTENTS OF LOC. 6
2382 010210 040010 BR YELLOW 1RESET TRAPCATCHER LOCATION
2383 010210 040010 BR YELLOW 1FOR NEXT TEST
2384 010210 040010 BR YELLOW 1CLEAR BOTTOM LIMIT LOCATION
2385 010210 040010 BR YELLOW 1OF 'YELLOW' ZONE USING NORMAL
2386 010210 040010 BR YELLOW 1PARITY
2387
2388
2389

```

```

*****
|
| THE FOLLOWING TEST WILL/S COULD CAUSE A PARITY ABORT IN THE
| 'RED' ZONE. THE 'RED' ZONE IS THE AREA BEYOND THE 'YELLOW'
| ZONE DESCRIBED IN THE ABOVE TEST. I.E. LOCATIONS 336 ON
| DOWN COMPRISE THE 'RED' ZONE
|
| SINCE PARITY ERRORS HAVE HIGHER PRIORITY WE WILL BE LOOKING
| FOR THE PARITY ABORT TO OCCUR BEFORE THE STACK VIOLATION
| TRAP TO LOCATION 4.
|
| THE CONTENTS OF THE STACK AFTER EXECUTION OF THE NEXT TEST
| SHOULD BE AS FOLLOWS:
|
| LOC. 0 = PC FROM STACK VIOLATION
| LOC. 2 = PS FROM STACK VIOLATION
|
| NOTE: THE PA AND PC FROM THE STACK VIOLATION ARE IN LOC. 0 & 2
| BECAUSE THE STACK POINTER (R6) IS REPOSITIONED TO LOC. 4
| (BY HARDWARE)!!
|
*****

```

```

2409
2410
2411
2412
2413
2414 010252 000004
2415
2416
2417
2418 010254 000015 JSR R0,(R5) 1SET UP A PARITY VECTOR SERVICE
2419 010256 010336 V3 1ROUTINE ADDRESS
2420 010260 005037 000336 CLR 00336 1CLEAR 1ST LOCATION OF 'RED'
2421 010264 042777 000004 171330 BIC 0012,0PARITY 1WRITE NORMAL
2422 010272 012706 000376 MOV 0376,SP 1SET STACK IN 'YELLOW' AREA
2423 010276 013727 000004 MOV 000,PC 1SAVE CONTENTS OF LOC. 4
2424 010302 000000 HLDLOC4: .WORD 0 1IN NEXT LOCATION
2425 010304 012737 010342 000004 NOV 0V4,0ERRVEC 1ORIGINAL CONTENTS OF LOC. 4
2426 010304 012737 010342 000004 NOV 0V4,0ERRVEC 1GO HERE
2427 010312 012737 000340 000006 MOV 0340,0ERRVEC+2 1SET UP A TIMEOUT VECTOR SERVICE
2428 010320 012737 010332 001332 MOV 0+12,0SLODAT 1ROUTINE ADDRESS FOR STACK
2429 010326 005766 177740 TST =40(SP) 1VIOLATION
2430 010332 104010 HLT *10 1NEW PS ON TIMEOUT TRAP
2431 010334 000421 BR RED 1STORE THE PC THAT SHOULD
2432 010336 104011 V41 HLT *11 1BE PUSHED ON THE STACK
2433 010340 000417 BR RED 1IF A PARITY ABORT OCCURS
2434 010342 042777 000001 171252 V41 BIC 0012,0PARITY 1REFERENCE 1ST LOCATION OF 'RED'
2435 010350 005737 000372 TST 00372 1ZONE USING REGISTER 6
2436 010350 005737 000372 TST 00372 1THIS INSTRUCTION SHOULD CAUSE
2437 010350 005737 000372 TST 00372 1AN ABORT 1ST, THEN A STACK
2438 010350 005737 000372 TST 00372 1VIOLATION
2439 010350 005737 000372 TST 00372 1DIDN'T ABORT OR RECOGNIZE THE
2440 010350 005737 000372 TST 00372 1STACK VIOLATION
2441 010350 005737 000372 TST 00372 1GO TO RESET STACK AND RESTORE
2442 010350 005737 000372 TST 00372 1TIMEOUT VECTORS
2443 010350 005737 000372 TST 00372 1ABORTED BUT STACK VIOLATION
2444 010350 005737 000372 TST 00372 1NOT RECOGNIZED
2445 010350 005737 000372 TST 00372 1GO TO RESET STACK AND RESTORE
2446 010350 005737 000372 TST 00372 1TIMEOUT VECTORS
2447 010350 005737 000372 TST 00372 1DISABLE PARITY
2448 010350 005737 000372 TST 00372 1STACK VIOLATION PICKED UP -

```

```

2470
2471 010054 041370 RND 18
2472 010054 144610 HLT *18
2473
2474 010050 000000 OR R0
2475
2476 010057 012700 001100 IPI MOV *0STACK,SP
2477 010050 013700 000000 NOV *0370,-(R0)
2478
2479
2480 010072 004000 011000 JSR R0,*0CHECKLOC
2481 010076 100000 HLT *3
2482 010000 012700 001100 RFD1 MOV *0STACK,SP
2483 010000 013700 010000 *000000 *0HOLDLOC,*0R
2484 010110 015000 000000 CLR *06
2485 010010 000000 000000 CLR *0370
2486
2487 010000 000000 000000 CLR *0330
2488
2489 *****
2490 FIRST 02 TEST (INDEX WORD) SMG,DNG MOV INSTRUCTION
2491 *****
2492 010000 000000 TST001 SCOPE
2493 I
2494 I
2495 I
2496 010000 000000 I
2497 I
2498 I
2499 010000 000000 I
2500 010000 000000 I
2501 010000 000000 I
2502 010000 000000 I
2503 *****
2504 *****
2505 010000 000000 I
2506 010000 000000 I
2507 *****
2508 010000 000000 I
2509 010000 000000 I
2510 *****
2511 *****
2512 010000 000000 I
2513 *****
2514 *****
2515 010000 000000 I
2516 010000 000000 I
2517 010000 000000 I
2518 010000 000000 I
2519 010000 000000 I
2520 010000 000000 I
2521 010000 000000 I
2522 *****
2523 *****
    
```

```

2524 *****
2525 010000 000000 TST01 SCOPE
2526 I
2527 I
2528 I
2529 010000 000000 I
2530 010000 000000 I
2531 010000 000000 I
2532 010000 000000 I
2533 010000 000000 I
2534 *****
2535 *****
2536 010000 000000 I
2537 010000 000000 I
2538 *****
2539 010000 000000 I
2540 010000 000000 I
2541 *****
2542 *****
2543 010000 000000 I
2544 *****
2545 *****
2546 010000 000000 I
2547 010000 000000 I
2548 010000 000000 I
2549 010000 000000 I
2550 010000 000000 I
2551 010000 000000 I
2552 010000 000000 I
2553 *****
2554 *****
2555 *****
2556 010000 000000 TST02 SCOPE
2557 I
2558 I
2559 I
2560 010000 000000 I
2561 010000 000000 I
2562 010000 000000 I
2563 010000 000000 I
2564 010000 000000 I
2565 *****
2566 *****
2567 010000 000000 I
2568 010000 000000 I
2569 *****
2570 010000 000000 I
2571 010000 000000 I
2572 *****
2573 *****
2574 010000 000000 I
2575 *****
2576 *****
2577 010000 000000 I
    
```

```

2576 010714 100001 MLY +1 /ABORTED ABORT
2579 010714 100010 BR +22 /GO TO NEXT TEST
2580 010714 000001 170074 Y: BIC #BIT2,#PARITY /DISABLE PARITY
2581 010714 000000 011000 JSR R0,#CHECKLOC /CHECK FOR GOOD ABORT
2582 010714 100003 MLY +3 /ABORTED INCORRECTLY
2583 010714 012706 001100 MOV #STACK,SP /RESET THE STACK
/*****
/TEST 65 TEST DNE INSTRUCTION (=BOOK)
/*****
/TEST65 SCOPE
/
/ 11/45 *** ROM STATE 321 ***
/
/ 11/40 *** ROM STATE 1 ***
2591 010714 000015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
2592 010714 011050 Z0 /ROUTINE ADDRESS
2593 010714 002777 000000 170066 BIC #BIT2,#PARITY /WRITE NORMAL
2594 010714 010102 MOV R1,R2 /SET UP FOR A DATO
2595 010714 102702 000004 SUB #4,R2 /CALCULATE THE START ADDRESS FOR THIS TEST
2596 010714 012702 012700 MOV #12700,(R0)+ /MOVE THE INSTRUCTION
2597 010714 012702 012700 MOV #0-1,(R2)+ /MOVE A '0' INTO NEXT /PARITY MEMORY AREA LOCATION
2600 010714 012702 177777 MOV #0-1,(R2)+ /MOVE THE INSTRUCTION
2601 010714 012702 100001 MOV #100001,(R2)+ /MOVE '00' INTO NEXT PARITY /MEMORY AREA LOCATION
2602 010714 012702 100001 BIC #BIT2,#PARITY /WRITE OTHER PARITY
2603 010714 005737 001002 TST #CPU40 /ARE WE ON AN 11/40?
2604 010714 001405 BEQ ZS /BRANCH IF NO
2605 010714 012702 012700 MOV R2,#SGDDAT /SINCE THE PC WILL BE UPDATED /ON THE PARITY ABORT /STORE THIS PC THAT SHOULD /BE PUSHED ON THE STACK
2606 010714 012702 100001 MOV #240,(R2)+ /IF A PARITY ABORT OCCURS /SINCE THE PC IS NOT UPDATED /ON THE PARITY ABORT /MOVE A '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2607 010714 000004 BR 1$ /CONTINUE WITH TEST
2608 010714 012702 000240 20: MOV #240,(R2)+ /MOVE A '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2609 010714 010237 001332 MOV R2,#SGDDAT /STORE THE PC THAT SHOULD /BE PUSHED ON THE STACK
2610 010714 010237 001332 MOV R2,#SGDDAT /IF A PARITY ABORT OCCURS /WRITE NORMAL
2611 010714 002777 170560 10: BIC #BIT2,#PARITY /MOVE '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2612 010714 012712 000203 MOV #203,(R2) /PARITY MEMORY AREA LOCATION
2613 010714 004362 177710 JSR R3,#10(R2) /IN CASE WE DON'T ABORT
2614 010714 100001 MLY +1 /GO TO PARITY MEMORY AREA
2615 010714 000010 BR +22 /DON'T ABORT
2616 010714 002777 000001 170536 Z0: BIC #BIT2,#PARITY /GO TO NEXT TEST
2617 010714 004037 011550 JSR R0,#CHECKLOC /DISABLE PARITY /CHECK FOR GOOD ABORT
    
```

```

2632 011070 100003 MLY +3 /ABORTED INCORRECTLY
2633 011070 012706 001100 MOV #STACK,SP /RESET THE STACK
/*****
/TEST 64 TEST DNE INSTRUCTION (=BOOK)
/*****
/TEST64 SCOPE
/
/ 11/45 *** ROM STATE 322 ***
/
/ 11/40 *** ROM STATE 1 ***
2641 011100 000015 JSR R0,(R5) /SET UP PARITY VECTOR SERVICE
2642 011100 011214 Z1 /ROUTINE ADDRESS
2643 011100 002777 000004 170510 BIC #BIT2,#PARITY /WRITE NORMAL
2644 011112 010102 MOV R1,R2 /SET UP FOR A DATO
2645 011114 102702 000004 SUB #4,R2 /CALCULATE THE START ADDRESS /FOR THIS TEST
2646 011120 012702 012700 MOV #12700,(R0)+ /MOVE THE INSTRUCTION
2647 011124 012722 000000 MOV #0,(R2)+ /MOVE A '0' INTO NEXT PARITY /MEMORY AREA LOCATION
2648 011130 012722 001001 MOV #1001,(R2)+ /MOVE THE INSTRUCTION
2649 011134 002777 000004 170460 BIC #BIT2,#PARITY /MOVE '00' INTO NEXT /PARITY MEMORY AREA LOCATION
2650 011140 005737 001002 TST #CPU40 /WRITE OTHER PARITY
2651 011146 001405 BEQ ZS /ARE WE ON AN 11/40? /BRANCH IF NO
2652 011150 010237 001332 MOV R2,#SGDDAT /SINCE THE PC WILL BE UPDATED /ON THE PARITY ABORT /STORE THIS PC THAT SHOULD /BE PUSHED ON THE STACK
2653 011150 010237 001332 MOV R2,#SGDDAT /IF A PARITY ABORT OCCURS /SINCE THE PC WILL NOT BE /UPDATED ON THE PARITY ABORT /MOVE A '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2654 011160 000004 BR 1$ /CONTINUE WITH TEST
2655 011162 012722 000240 20: MOV #240,(R2)+ /MOVE A '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2656 011166 010237 001332 MOV R2,#SGDDAT /STORE THE PC THAT SHOULD /BE PUSHED ON THE STACK
2657 011172 002777 000004 170422 10: BIC #BIT2,#PARITY /IF A PARITY ABORT OCCURS /WRITE NORMAL
2658 011200 012712 000203 MOV #203,(R2) /MOVE '000' INTO NEXT /PARITY MEMORY AREA LOCATION
2659 011204 004362 177770 JSR R3,#10(R2) /IN CASE WE DON'T ABORT
2660 011210 100001 MLY +1 /GO TO PARITY MEMORY AREA
2661 011212 000010 BR +22 /DON'T ABORT
2662 011214 002777 000001 170400 Z1: BIC #BIT2,#PARITY /GO TO NEXT TEST
2663 011220 004037 011550 JSR R0,#CHECKLOC /CHECK FOR GOOD ABORT
2664 011226 100003 MLY +3 /ABORTED INCORRECTLY
2665 011230 012706 001100 MOV #STACK,SP /RESET THE STACK
/*****
/TEST 65 TEST DNE INSTRUCTION (=BOOK)
/*****
    
```

```
2636 *****  
2637 011374 000004 11745 *** ROM STATE 324 ***  
2638 /  
2639 /  
2640 /  
2641 011375 000015 11740 *** ROM STATE 1 ***  
2642 JSR RC,(R0) /WRITE NORMAL  
2643 ZP /WRITE NORMAL  
2644 BIC #BIT2,#PARITY /WRITE OTHER PARITY  
2645 MOV R1,R2 /AND BE ON AN 11700  
2646 SUB #4,R2 /BRANCH IF NO  
2647 /SINCE THE PC WILL BE UPDATED  
2648 MOV #12700,(R0)+ /FOR THE PARITY ABOUT  
2649 /STORE THIS PC THAT SHOULD  
2650 /BE PUSHED ON THE STACK  
2651 /IF A PARITY ABORT OCCURS  
2652 /SINCE THE PC WILL NOT BE  
2653 /UPDATED ON THE PARITY ABORT  
2654 MOV #1,(R2)+ /MOVE A '1' INTO NEXT PARITY  
2655 /MEMORY AREA LOCATION  
2656 /MOVE THE INSTRUCTION  
2657 /PCED, '4' INTO NEXT PARITY  
2658 /MEMORY AREA LOCATION  
2659 /WRITE OTHER PARITY  
2660 TBY /AND BE ON AN 11700  
2661 BCD R2 /BRANCH IF NO  
2662 /SINCE THE PC WILL BE UPDATED  
2663 /FOR THE PARITY ABOUT  
2664 /STORE THIS PC THAT SHOULD  
2665 /BE PUSHED ON THE STACK  
2666 /IF A PARITY ABORT OCCURS  
2667 /SINCE THE PC WILL NOT BE  
2668 /UPDATED ON THE PARITY ABORT  
2669 MOV #240,(R2)+ /MOVE A '000' INTO NEXT  
2670 /PARITY MEMORY AREA LOCATION  
2671 /CONTINUE WITH TEST  
2672 /MOVE A '000' INTO NEXT  
2673 /PARITY MEMORY AREA LOCATION  
2674 /STORE THE PC THAT SHOULD  
2675 /BE PUSHED ON THE STACK  
2676 /IF A PARITY ABORT OCCURS  
2677 /WRITE NORMAL  
2678 BIC #BIT2,#PARITY /WRITE OTHER PARITY  
2679 MOV #203,(R2) /MOVE '010' INTO NEXT  
2680 /PARITY MEMORY AREA LOCATION  
2681 /IF A PARITY ABORT  
2682 /GO TO PARITY MEMORY AREA  
2683 /IDENTIFY ABORT  
2684 /GO TO NEXT TEST  
2685 /ADJUSTABLE PARITY  
2686 /CHECK FOR GOOD ABORT  
2687 /ABORTED INCORRECTLY  
2688 /RESET THE STACK  
2689 *****  
2690 /TEST 66 END OF PROGRAM  
2691 *****  
2692 /TEST 67 SCOPE  
2693 BIC #BIT15(1),#R15,#PARITY /DISABLE ALL PARITY  
2694 /AND CLEAR ERROR BITS  
2695
```

```
2743 011462 005737 001630 TST #USERVFR /DID THE USER SELECT THE TEST  
2744 011460 001014 BNE 16 /BRANCH IF SO AND DON'T STEP  
2745 /UP THE TABLE  
2746 011410 002737 000002 001336 ADD #2,#SEPCAD /STEP UP TO NEXT REGISTER  
2747 011416 002737 000002 001414 AND #2,#STMPAD /STEP UP TO CORRESPONDING  
2748 /PARITY MEMORY  
2749 011424 002737 000002 001416 ADD #2,#SESEYAD /STEP UP TO NEXT OFFSET - THIS  
2750 /IS ONLY APPLICABLE IF MEMORY  
2751 /MIGHT BE BURNED ON  
2752 /STEP UP TO NEXT INTERLEAVE  
2753 /VALUE  
2754 011448 002737 000002 001444 ADD #2,#ENTERAD /STEP UP TO NEXT INTERLEAVE  
2755 /VALUE  
2756 011448 004337 011474 101 JSR R3,#FLAGCLR /CLEAR PERTINENT FLAGS  
2757 011444 000167 000320 JMP SEOP /GO TO RING-A-DING  
2758 /BEFORE REITERATING THE PROGRAM  
2759 *****  
2760 /ROUTINE FOR SETTING UP THE PARITY VECTOR SERVICE ADDRESS  
2761 /  
2762 011450 012077 170144 VECSET: MOV (R0)+,#INVEC /WRITE ADDRESS INTO LOCATION 114  
2763 011454 002777 000005 170140 BIC #BIT2(1),#PARITY /WRITE OTHER PARITY AND DISABLE  
2764 011462 000200 RTS R0 /RETURN TO TESTING  
2765 *****  
2766 /ROUTINE TO RESET AND GO BACK TO TABLE BEGINNING  
2767 /  
2768 *****  
2769 RESTART: JSR R3,#INITIALIZE  
2770 011464 004337 011506 JMP SEOP /GO TO RING-A-DING  
2771 011470 000167 000374 *****  
2772 /ROUTINE TO CLEAR PERTINENT FLAGS BEFORE PASSING INRU THE PROGRAM  
2773 /WITH ANOTHER TABLE ENTRY  
2774 /  
2775 *****  
2776 /FLAGCLR: CLR #PSPCORZONES /CLEAR PC, PC AND ZONES ABOUT  
2777 /  
2778 011474 005037 001674 CLR #MREGFLAG /FLAG  
2779 /CLEAR MS11 REGISTER PRESENCE  
2780 /FLAG  
2781 011504 000203 RTS R3 /RETURN  
2782 *****  
2783 /ROUTINE TO COMPLETELY REINITIALIZE BEFORE RESTARTING PROGRAM OVER  
2784 /AT THE BEGINNING OF THE TABLE  
2785 /  
2786 *****  
2787 INITIALIZE: CLR #PSPCORZONES /CLEAR PC, PC AND ZONES  
2788 /ABOUT FLAG  
2789 011504 005037 001674 CLR #MREGFLAG /CLEAR MS11 REGISTER PRESENCE  
2790 /FLAG  
2791 011512 005037 001676 CLR #MREGFLAG /FLAG  
2792 /RETURN  
2793 011516 012737 001340 001336 MOV #SEPC0,#MREGAD /MOVE FIRST REGISTER CONTAINER
```

```

2741                FINTO INHEAD WHICH IS USED TO
2742                POINT TO THE PARITY REGISTER
2743                TABLE
2744 011528 012737 001306 001438      MOV      02THRO,001INHEAD
2745                MOVE PARITY MEMORY CONTAINER
2746                FINTO STMPD WHICH IS USED TO
2747                POINT TO THE MEMORY LOCATION
2748                TABLE
2749 011532 012737 001408 001416      MOV      00SBTD,001STAD
2750                MOVE FIRST OFFSET CONTAINER
2751                FINTO 00S2AD WHICH IS USED TO
2752                POINT TO THE OFFSET LOCATION
2753                TABLE IF MEMORY HORT IS USED
2754 011540 012737 001450 001466      MOV      00YERR,00YERRAD
2755                MOVE 1ST INTERLEAVE CONTAINER
2756                FINTO 00INERAD WHICH IS USED
2757                TO POINT TO THE INTERLEAVE
2758                LOCATION TABLE
2759 011546 000200                RTS      R3
2760                RETURN
2761
2762 *****
2763
2764 ;SUBROUTINE TO CHECK THAT IF A TEST HAS ASSIGNED IT DID INCLUDE
2765 ;ABORT IN THE PROPER PLACE. IT IS QUITE UNDESIRABLE THAT A TEST
2766 ;FINAL SHOULD HAVE OCCURRED IN THE 160 LOCATION MAP AREA ABORTED
2767 ;BY THE RESULT OF THE INSTRUCTION THAT HAS TO CAUSE THE ABORT.
2768 ;THIS SUBROUTINE WILL FLAG SUCH OCCURRENCES. WITHOUT THIS CHECK
2769 ;THE PROGRAM WOULD APPEAR TO HAVE RUN PROPERLY.
2770
2771 ;BOTH THE CORRECT HIGH ORDER ERROR ADDRESS BITS AND PROPER PC
2772 ;PUSH ON THE STACK ARE LOOKED FOR AFTER A PARITY ABORT OCCURS.
2773
2774 *****
2775 CHECKLOC:      MOV      R4,*(SP)      ;SAVE R2 CONTENTS ON STACK
2776                MOV      R3,*(SP)      ;SAVE R3 CONTENTS ON STACK
2777                CLR      R2           ;SAVE R4 CONTENTS ON STACK
2778                CLR      R4           ;CLEAR ERROR ADDRESS BY COMPARE
2779                CLR      R4           ;REGISTERS IN CASE WE HAVE AN OLD
2780                ;MOS DESIGN THAT DOESN'T HAVE
2781                ;ADDRESS BITS
2782                TST      00000000     ;IS AN R311 OR R311 PARITY
2783                BNE     Y8           ;OPTION BEING TESTED?
2784                BNE     Y8           ;BRANCH IF R311 AND DON'T GO
2785                TST      00000000     ;ADDRESS BITS CHECKING
2786                BNE     Y8           ;ARE WE DOING A PC OR PC FETCH
2787                BNE     Y8           ;FOR ZONE ABORT YES?
2788                BNE     Y8           ;BRANCH IF NO
2789                SET     R2,0          ;SET THE PARITY REGISTER HIGH
2790                ;ORDER ADDRESS BITS VALUE
2791                ;(BITS 5 THRU 11) TO ZERO FOR
2792                ;THE PC OR PC FETCH ABORT TESTS
2793                ;GO CHECK IT AGAINST ACTUAL
2794                ;PARITY REGISTER VALUE
2795                ;MEMORY MANAGEMENT ONLY
2796                BNE     Y8           ;BRANCH IF NO
2797                ;PICK UP THE OFFSET VALUE - IT
    
```

```

2848
2849                ;SHOULD BE THE VALUE THAT WILL
2850                ;APPEAR IN THE PARITY REGISTER
2851                ;ERROR ADDRESS BITS
2852 011614 012703 017400 001      BR       Z8
2853                ;GO TO CHECK1
2854                ;GET A FIRST POSSIBLE ABORT
2855                ;LOCATION AREA
2856 011622 012702 000140      MOV      0140,R2
2857                ;GET THE FIRST POSSIBLE PARITY
2858                ;REGISTER HIGH ORDER ADDRESS
2859                ;BITS VALUE (BITS 5 THRU 11)
2860                ;IS THIS THE ABORT AREA BEING
2861                ;TESTED?
2862 011630 101027 001326 001      BHI     Z8
2863                ;NO - SEE IF IT'S THE NEXT ONE
2864                ;PROCEED TO SEE IF
2865                ;IT WAS A PROPER ABORT BY LOOKING
2866                ;AT THE HIGH ORDER ADDRESS BITS
2867                ;OF THE PARITY REGISTER
2868                ;(BITS 5 THRU 11)
2869                ;CLEAR ALL BITS EXCEPT 5 THRU 11
2870                ;ARE THE ERROR ADDRESS BITS
2871                ;WHAT THEY SHOULD BE?
2872                ;BRANCH IF NO
2873                ;STORE WHAT THE ADDRESS BITS
2874                ;SHOULD HAVE BEEN
2875                ;STORE WHAT THE ADDRESS BITS
2876                ;WERE
2877 011636 042704 170037      BIC     017023,R4
2878                ;RESTORE R4 CONTENTS
2879 011642 040402      CMP     R4,R2
2880                ;RESTORE R3 CONTENTS
2881                ;RESTORE R2 CONTENTS
2882 011644 001026      DNE     Z8
2883                ;BRANCH IF NO
2884 011646 010237 001326 7%      MOV      R2,000GDAUR
2885                ;STORE WHAT THE ADDRESS BITS
2886                ;SHOULD HAVE BEEN
2887                ;STORE WHAT THE ADDRESS BITS
2888                ;WERE
2889 011652 010437 001330      MOV      R4,000BDAUR
2890                ;RESTORE R4 CONTENTS
2891 011656 012604      MOV      (SP)+,R4
2892                ;RESTORE R3 CONTENTS
2893 011660 012603      MOV      (SP)+,R3
2894                ;RESTORE R2 CONTENTS
2895 011662 012602      MOV      (SP)+,R2
2896                ;RESTORE R2 CONTENTS
2897
2898 ;IF WE HAVE REACHED THIS PATH 1 OF 2 CONDITIONS EXIST --
2899 ;WE HAVE AN OLD MOS DESIGN WITH NO ADDRESS BITS, OR WE HAVE
2900 ;A NEW MOS DESIGN WITH ADDRESS BITS OK!!
2901
2902 011664 026637 000002 001332      CMP      2(SP),000000
2903                ;WAS THE CORRECT PC PUSHED
2904                ;ON THE STACK?
2905 011672 001404      BEQ     Z8
2906                ;BRANCH IF YES
2907 011674 016637 000002 001334      MOV      2(SP),000000
2908                ;SAVE INCORRECT PC FOR PRINTOUT
2909 011702 000200      RTS     R0
2910                ;GO BACK TO INDICATE BAD ABORT
2911 011704 005720 9%      TST     (R0)+
2912                ;STEP UP RETURN ADDRESS
2913                ;TO BYPASS THE ERROR HLT
2914 011706 000000      RTG     R0
2915                ;RETURN TO CONTINUE TESTING
2916 011710 002703 004000 3%      ADD     R0,0000,R3
2917                ;STEP UP TO THE NEXT POSSIBLE
2918                ;ABORT LOCATION AREA
2919 011714 002702 000040      ADD     040,R2
2920                ;CHANGE THE VALUE OF THE HIGH
2921                ;ORDER ADDRESS BITS VALUE TO BE
2922                ;CHECKED
2923
2924 011720 000742      BR      Z8
2925                ;GO BACK TO CHECK THIS ONE
2926 011722 010237 001326 4%      MOV      R2,000GDAUR
2927                ;STORE WHAT THE ADDRESS BITS
2928                ;SHOULD HAVE BEEN
2929 011726 010437 001330      MOV      R4,000BDAUR
2930                ;STORE WHAT THE
2931                ;ADDRESS BITS WERE
2932 011732 012604      MOV      (SP)+,R4
2933                ;RESTORE R4 CONTENTS
2934 011734 012603      MOV      (SP)+,R3
2935                ;RESTORE R3 CONTENTS
2936 011736 012602      MOV      (SP)+,R2
2937                ;RESTORE R2 CONTENTS
    
```

```

2902
2903
2904
2905
2906 011786 000000 000000 001330 CMP 2(SB),#00000000 /WAS THE CORRECT PC PUSHED
2907 /ON THE STACK
2908 011786 001800 BEO 100 /SEARCH IF YES
2909 011786 010000 000000 001330 MOV 2(SB),#000000 /SAVE INDEPLY PC FOR FRONT
2910 011786 000000 RTS 00 /FEED BACK TO INDICATE BAD ADDR
2911 011786 011757 167340 001330 1001 MOV 000001,00000000 /FEEDED TO BACK LAST THE
2912 /CORRECT PC WAS PUSHED ON STACK
2913 011786 000000 RTS 00 /FEED BACK TO INDICATE BAD ADDR
2914
    
```

```

2915
2916
2917
2918
2919
2920
2921
2922 011778 000000 167300 CLR STSNR /ZERO THE TEST NUMBER
2923 011778 000000 000300 CLR STSNR /ZERO THE NUMBER OF ITERATIONS
2924 012000 000000 167272 INC SPADR /INCREMENT THE PASS NUMBER
2925 012000 032157 000000 177570 BIT #0010,00000 /BREAK THE BELL?
2926 012015 001000 012050 BNE 48 /NO
2927 012015 100000 012050 TYPE #BELL /RING A BELL
2928 012020 013700 000000 401 MOV #002,00 /GET MONITOR ADDRESS
2929 012020 001411 BEO SDGAGN /IF NONE
2930 012030 002100 177777 000002 CMP #-1,2(RB)
2931 012030 001001 BNE SENDAD
2932 012040 000005 RESET
2933 012042 000710 SENDAD: JSR PC,(R0) /GO TO MONITOR
2934 012044 000000 NOP /SAVE ROOM
2935 012046 000000 NOP /FOR
2936 012050 000000 NOP /FACT11
2937 012052 000137 000000 SDGAGN: JHP #0START /RETURN
2938 012056 177607 000377 BELL: #ASCIZ <207><377><377>
    
```



```

3141 013026 012767 000000      MOV      (R0)+,R5      PICK UP 'ERROR MESSAGE' POINTER
3142 013028 012768      TYPE     E            TYPE 'ERROR MESSAGE'
3143 013029 000700 101      E            'ERROR MESSAGE' POINTER GOES HERE
3144 013030 100000 012763      TYPE     ,SCLRF      TYPE A CARRIAGE RETURN AND
3145                                ILINE FEED
3146 013032 012767 000000      MOV      (R0)+,R6      PICK UP 'DATA HEADER' POINTER
3147 013034 012768      BFD     32           IF 'R' ONLY TYPE
3148 013035 100000      TYPE     'DATA HEADER'
3149 013037 000000      'DATA HEADER' POINTER GOES HERE
3150 013038 100400 012762      TYPE     ,SCLRF      TYPE A CARRIAGE RETURN AND
3151                                ILINE FEED
3152 013040 012000      MOV      (R0)+,R6      PICK UP 'DATA POINTER'
3153 013042 012000      BNE     51           IF THERE IS DATA TO TYPE GO DO IT
3154 013044 012000      MOV      (SP)+,R0     RESTORE R0
3155 013046 100400 012763      TYPE     ,SCLRF      TYPE A CARRIAGE RETURN AND
3156                                HARD LINE FEED
3157 013072 000000      RTS     PC           RETURN TO TESTING
3158 013074      501
3159 013076 013046      MOV      0(R0)+,(SP)  'SAVE (R0)+ FOR TYPEOUT
3160                                TYPE DATA
3161 013078 000000 000246      JSR      R0,550001   I/O TYPE-OCTAL ASCII
3162 013080 000      ,BYTE 0            TYPE 6 DIGITS
3163 013082 001      ,BYTE 1            TYPE LEAVING ZEROS
3164 013084 005710 101      TST     (R0)         HAVE WE REACHED THE 'OF TERMINATOR
3165 013086 005710      BCC     45           YES - CLEAN UP FOR RETURN
3166 013088 100400 013116      TYPE     ,SP         TYPE 5 SPACES
3167 013090 000700      BR      55           FLOOR TILL 'OF TERMINATOR REACHED
3168 013116 000340 000000 601      ,ASCII / /
3169                                ,EVEN
    
```

```

3170 ]*****
3171 ]
3172 ]THE FOLLOWING ROUTINE WILL SIZE MEMORY
3173 ]
3174 ]PLSTLK WILL CONTAIN THE LAST BANK AS AN SAF (SEGMENT ADDRESS FIELD)
3175 ]
3176 ]THE SECOND ADDRESS FIELD CONTAINS THE 4 MOST SIGNIFICANT BYTS OF
3177 ]THE LAST ADDRESS OF THE LAST BANK FOUND
3178 ]
3179 ]*****
3180 013124 010000      SIZE:  MOV      R5,(SP)  'SAVE R0 CONTENTS
3181 013126 000000      MOV      SP,R5       'SAVE THE STACK POINTER
3182 013128 012703 017006      MOV      R3,PKPDR0   'GET THE FOLLOWING PAGE
3183 013130 010377 100000      MOV      R3,PKPDR1   'DESCRIPTOR REGISTERS TO
3184 013132 010377 100000      MOV      R3,PKPDR2   'READ/WRITE AND TRANSFER OF
3185 013134 010377 100000      MOV      R3,PKPDR3   '1000 (10) WORDS PER SEGMENT
3186 013136 010377 100000      MOV      R3,PKPDR4
3187 013138 010377 100000      MOV      R3,PKPDR5
3188 013140 010377 100000      MOV      R3,PKPDR6
3189 013142 010377 100000      MOV      R3,PKPDR7
3190 013144 000000      CLR     PKPAR0       'SET THE FOLLOWING PAGE
3191 013146 000000 100024      MOV      R20,PKPAR1  'ADDRESS REGISTERS TO THEIR
3192 013148 012777 000000 100020      MOV      R40,PKPAR2  'RESPECTIVE OFFSET VALUES
3193 013150 012777 000000 100014      MOV      R60,PKPAR3  'FOR REDUPLICATION PURPOSES
3194 013152 012777 001800 100010      MOV      R100,PKPAR4
3195 013154 012777 001800 100004      MOV      R120,PKPAR5
3196 013156 012777 001400 100000      MOV      R140,PKPAR6
3197 013158 012777 007600 100774      MOV      R760,PKPAR7
3198                                'THIS ONE'S THE I/O RECORD
3199                                'PAGE CONTAINING CONTROL STATUS
3200                                'REGISTERS, ETC.
3201 013252 016704 104766      MOV      RPAR6,R4    'GET ADDRESS OF PAGE 6 REGISTER
3202 013254 000000 104760      CLR     (R4)         'CLEAR THE REGISTER
3203 013256 005777 104720      INC     CSR0         'TURN ON MEMORY MANAGEMENT
3204 013258 010706      MOV      PC,(SP)     'MAKE K11 TIMEOUT SERVICE
3205 013260 000026      ADD     @SKTOUT-,(SP) 'ROUTINE ADDRESS POSITION
3206                                'INDEPENDENT
3207 013272 012637 000000      MOV      (SP)+,@ENRVEC 'SET FOR TIMEOUT
3208 013274 005737 103776 101      TST     @103776     'TRAP ON NON-EXISTENT MEMORY
3209 013276 002714 000000      ADD     R40,(R4)     'MAKE A 16 STEP
3210 013278 007714 104734      CMP     PKPAR7,(R4)  'LAST ONE?
3211 013280 003371      BGT     13           RNO - TRY IT!
3212 013282 011400      SATOUT: MOV      (R0),R0 'GET LAST BANK #1
3213 013284 102700 000000      SUB     R0,R0       'DROPP BACK
3214 013286 010706 000000 000004      MOV      R1,@ENRVEC  'SET FOR ERRORS
3215 013288 010506      MOV      R5,SP       'RESTORE THE STACK POINTER
3216 013290 010007 000010      MOV      R0,PLSTLK  'STORE THE SAF
3217 013292 005077 104602      CLR     CSR0       'TURN MEMORY MGMT OFF
3218 013294 012605 101      MOV      (SP)+,R5   'RESTORE R5
3219 013296 000000      RTS     PC         'RETURN TO NORMAL FLOW
3220 013346 000000      PLSTLKI  ,WORD 0    'CONTAINS THE SAF
3221 ]*****
3222 ]BINARY TO OCTAL (ASCII) AND TYPE
3223 ]%OCT--ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
    
```



```

3291
3292
3293
3294 013006 016286
3295 013006 016670 000002
3296 013006 016700
3297 013006 011000
3298 013070 016000 013070
3299 013070 002000
3300
3301
3302
3303
3304
3305 013070 001110
3306 013000 012456
3307 013000 012512
3308 013000 012310
    
```

```

*****
TRAP HANDLER
SIRAP:  MOV    RC,*(SP)      /SAVE RC
        MOV    P(CP),R6     /GET TRAP ADDRESS
        YST    *(R6)        /BACKUP BY 2
        MOVB   (R6),R6      /GET RIGHT BYTE OF TRAP
        MOV    SIRPAD(RV),R0 /INDEX TO TABLE
        RTS    R6           /GO TO ROUTINE

TRAP TABLE
:
:
:
SIRPAD: STYPE  /CALLTYPE      TRAP#0(104400)  TTY TYPEOUT ROUTINE
        SREADR /CALLGETCHR    TRAP#2(104402)  TTY TYPEIN CHARACTER ROUTINE
        SREADR /CALLGETYR    TRAP#4(104404)  TTY TYPEIN STRING ROUTINE
        SACCPT /CALLACCPTY    TRAP#6(104406)  READ AN OCTAL NUMBER FROM TTY
    
```

```

3309
3310
3311
3312 013006 012737 013006 000024
3313 013014 012737 000340 000026
3314 013020 010760
3315 013020 010760
3316 013026 012246
3317 013030 012501
3318 013032 010466
3319 013034 012500
3320 013036 012517 000070
3321 013002 012717 013000 000020
3322 013000 000000
3323 013050 000776
3324
3325
3326 013054 016706 000000
3327 013000 000000 000000
3328 013064 000000 000000
3329 013070 001375
3330 013072 012605
3331 013074 012604
3332 013076 012603
3333 013700 012602
3334 013702 012601
3335 013704 012600
3336 013705 012737 013006 000024
3337 013714 012737 000340 000026
3338 013722 104400 013700
3339 013726 012716 001700
3340 013732 000000
3341 013734 000000
3342 013736 000776
3343 013740 000000
3344 013742 000015 007520 042521
3345 013750 000122
3346
    
```

```

*****
POWER DOWN ROUTINE
SPWRDN: MOV    $SILLUP,0$PWRVEC    /SET UP LAST UP
        MOV    $00,0$PWRVEC+2     /PRIORITY
        MOV    R5,*(SP)           /PUSH R5 ON STACK
        MOV    R1,*(SP)           /PUSH R1 ON STACK
        MOV    R2,*(SP)           /PUSH R2 ON STACK
        MOV    R3,*(SP)           /PUSH R3 ON STACK
        MOV    R4,*(SP)           /PUSH R4 ON STACK
        MOV    R6,*(SP)           /PUSH R6 ON STACK
        MOV    SP,$SAVR6         /SAVE SP
        HALT    0$PWRUP,0$PWRVEC  /SET UP VECTOR
        BR     ,*2

*****
POWER UP ROUTINE
SPWRUP: MOV    $SAVR6,SP          /GET SP
        CLR    $SAVR6            /WAIT LOOP FOR THE TTY
        INC    $SAVR6            /WAIT FOR THE INC
        BNE    IS
        MOV    (SP)+,R5          /POPP STACK INTO R5
        MOV    (SP)+,R4          /POPP STACK INTO R4
        MOV    (SP)+,R3          /POPP STACK INTO R3
        MOV    (SP)+,R2          /POPP STACK INTO R2
        MOV    (SP)+,R1          /POPP STACK INTO R1
        MOV    (SP)+,R0          /POPP STACK INTO R0
        MOV    $SPWRDN,0$PWRVEC  /SET UP THE POWER DOWN VECTOR
        MOV    $30,0$PWRVEC+2    /PRIORITY
        TYPE   $POWER            /POWER FAIL MESSAGE
        MOV    $BEGIN,(SP)       /RESTART AT BEGIN
        RTS

$SILLUP: HALT
$BR     ,*2
$SAVR6: 0
$POWER: ,ASCIZ <15><12>"POWER"
,EVEN
    
```



```

3755 010030 009100 041500 040004
3436 010040 009504 010 000000
3437 010040 000 000000 000103
3438 010050 000000 000000 000000
3439 010060 000116 001100 000000
3440 010070 001500 001100 000000
3441 010074 001100 001117 000104
3442 010100 000000 000
3443 010100 000000
3444 010100 001500 001500 000000 0101
    WORD SMLTAD,PARITY,0
3445 010110 001500 001500 000000 0101
    WORD PARITY,0
3446 010120 001500 000000 0101
    WORD SMLTAD,0
3447 010120 001500 001500 001500 0101
    WORD SMLTAD,PARITY,SGADR,SGADR,SGDDAI,SGDDAT,0
3448 010130 001500 001500 001500 0101
    WORD SMLTAD,PARITY,SGDDAT,0
3449 010140 000000
3470 010160 000000 001500 001500 0101
    WORD SMLTAD,PARITY,SGDDAT,0
3471 010160 000000
3472
3473 000000
    END
    
```

A	003752	ABORT	003430	ACCEPT	100400	AG	000026
A1	004100	A2	004100	B	000230	BEGIN	001700
BIT0	000001	BIT00	000001	BIT01	000002	BIT02	000004
BIT03	000010	BIT04	000020	BIT05	000040	BIT06	000100
BIT07	000000	BIT08	000000	BIT09	001000	BIT1	000002
BIT10	000000	BIT11	000000	BIT12	000000	BIT13	000000
BIT14	000000	BIT15	000000	BIT16	000000	BIT17	000000
BIT18	000000	BIT19	000000	BIT1	000100	BIT77	000000
BIT6	000000	BIT9	001000	BLKCNT	000100	BYVECC	000010
B1	004310	B1	004370	B2	000000	B3	000000
B4	004610	B5	004700	B6	000000	C	000040
CHECKL	011550	COMPUT	003200	CPU40	001040	CC	000110
C1	000010	C2	000000	C3	000000	C4	000000
C5	000010	C6	000000	C7	000000	C8	000000
D	000000	D0	000000	D08	000000	D01	000140
DH1	014470	DH2	014540	DH3	014560	DH4	014600
DH5	014605	DISPLA	177570	DT1	015100	DT2	015110
DTS	015120	DT4	015120	DT5	015140	D0	000070
D1	006150	E	006320	EMVVECC	000000	EM1	013750
EM10	014050	EM11	014020	EM12	014400	EM2	013770
EM3	014020	EM4	014050	EM5	014120	EM6	014140
EM7	014210	ERRVEC	000000	E0	000000	E1	000000
E2	000010	F	006630	FINDDN	000000	FLAGSC	011470
F0	000000	F1	006770	G	007050	GETCHR	100400
GETSTR	100400	G0	000000	G0	007060	H	007140
HOLDLO	010000	H0	007150	H1	007130	H2	007240
K5	007210	H0	007220	INITIA	011500	INTERT	001640
INTVEC	001620	JOTVEC	000000	KPAR0	000230	KPAR1	000030
KPAR2	000230	KPAR3	000230	KPAR4	000240	KPAR5	000040
KPAR6	000240	KPAR7	000240	KPDR0	000210	KPDR1	000010
KPDR2	000210	KPDR3	000210	KPDR4	000220	KPDR5	000020
KPDR6	000220	KPDR7	000220	KTYIME	000220	L	000320
LEAFON	001630	N	007370	MEMAD	001000	MRK0	000320
MRK1	000500	MSGTYP	002370	MSREGF	001000	N	000000
NEWSTK	001470	NEXT1	000510	NN	007450	NOMORE	003170
NOREG	005140	NYERAD	001440	NTER0	001450	NYER1	001450
NYER10	001470	NYER11	001470	NTER12	001470	NYER2	001450
NYER3	001450	NYER4	001460	NTER5	001460	NYER6	001460
NYER7	001460	ONETHY	003500	P	007520	PARCDB	003110
PARITY	001620	PARTST	003220	PC	*X000007	PS	*177770
PSPCON	001620	PSW	*177770	PNRVEC	000000	R	007600
RED	010400	RESTAR	011460	RESTOR	001630	RESVEC	000010
R0	*X000000	R1	*X000001	R2	*X000002	R3	*X000003
R4	*X000004	R5	*X000005	R6	*X000006	R7	*X000007
SAVLOC	010120	SEAVEC	000250	SP	*X000008	SR0	000000
SR2	000000	STACK	*001100	START	003620	SWR	*177570
SW0	*000001	SW00	*000001	SW01	*000002	SW02	*000004
SW03	*000010	SW04	*000020	SW05	*000040	SW06	*000100
SW07	*000000	SW08	*000400	SW09	*001000	SW1	*000002
SW10	*000000	SW11	*000020	SW12	*000000	SW13	*000000
SW14	*000020	SW15	*100000	SW2	*000004	SW3	*000010
SW2	*000000	SW5	*000040	SW6	*000100	SW7	*000200
SW8	*000000	SW9	*001000	T	007750	TBITVE	000014
YK6	012622	YK8	012620	TRAPVE	000034	TRTVEC	000014

