

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

.REM 3

IDENTIFICATION

PRODUCT CODE: AC-8186C-MC  
PRODUCT NAME: CVKAACO LSI-11 BSC INST  
PRODUCT DATE: 01-NOVEMBER-1978  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: PERVEZ A. ZAKI  
MODIFIED BY: BARRY SUSSMAN 01-NOV-77  
BARRY SUSSMAN 01-NOV-78

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL            PDF            UNIBUS            MASSBUS  
DEC                DECLOS            DECTAPE

CONTENTS

53	
54	
55	
56	
57	1. ABSTRACT
58	2. REQUIREMENTS
59	2.1 EQUIPMENT
60	2.2 STORAGE
61	2.3 PRELIMINARY PROGRAMS
62	3. LOADING PROCEDURE
63	4. STARTING PROCEDURE
64	4.1 STARTING ADDRESS
65	4.2 PROGRAM AND/OR OPERATOR ACTION
66	5. OPERATING PROCEDURE
67	6. ERRORS
68	6.1 ERROR REPORTING
69	6.2 ERROR RECOVERY
70	7. RESTRICTIONS
71	8. MISCELLANEOUS
72	8.1 EXECUTION TIME
73	8.2 STACK POINTER
74	8.3 PASS COUNTER
75	8.4 TEST NUMBER
76	8.5 POWER FAIL
77	9. PROGRAM DESCRIPTION
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  
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

1. ABSTRACT  
THIS PROGRAM TESTS THE LSI-11 BASIC INSTRUCTION SET  
IN ALL MODES. THE DIAGNOSTIC IS DESIGNED TO RUN UNDER  
BOTH APT. AND ACT. SYSTEMS

2. REQUIREMENTS

2.1 EQUIPMENT  
Lsi-11 STANDARD COMPUTER  
AND 4K OF MEMORY

2.2 STORAGE  
PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17500

2.3 PRELIMINARY PROGRAMS  
NONE

3. LOADING PROCEDURE  
USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 STARTING ADDRESS  
AFTER LOADING THE PROGRAM IT SHOULD ALWAYS BE STARTED AT 200.  
IF IT IS DESIRED TO SAVE THE PASS COUNTER THEN CLEAR THE  
LOCATION \$TFSTN [I.E. LOCATION 102] AND RESTART FROM 450 OTHERWISE  
THE PROGRAM CAN BE RESTARTED AT 200. IF IT IS DESIRED TO GO TO A TEST  
OTHER THAN TEST # 0 THEN PLACE THE TEST NUMBER IN LOCATION \$TESTN  
AND RESTART THE PROGRAM AT 450. IN WHICH CASE THE PROGRAM WILL HALT  
AT LOCATION 464 AND WILL WAIT FOR THE OPERATOR TO PLACE THE  
STARTING ADDRESS OF THE DESIRED TEST IN PC (R7) AND TYPE A P.

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

4.2 PROGRAM AND/OR OPERATOR ACTION

- 1) PLACE LTC SWITCH IN OFF POSITION.
- 2) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 3) TYPE 200G [ THERE ARE NO SWITCH SETTINGS AND THIS DIAGNOSTIC DOES NOT USE SOFTWARE SWITCH LOCATION \$SWREG ]
- 4) THE PROGRAM WILL LOOP AND 'END PASS' WILL BE TYPED AFTER THE FIRST PASS AND THEN EVERY 377 PASSES. HOWEVER TYPE OUT WILL BE SUPPRESSED IF BIT 5 OF LOCATION \$ENVM IS HIGH
- 5) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

5. OPERATING PROCEDURE

5.1 OPERATING MODE:

AN 8 BIT BYTE \$ENVM [I.E. LOCATION 117] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE \$ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN LOCATION 116.

5.2 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

6. ERRORS

6.1 ERROR REPORTING

ON FINDING AN ERROR THE PROCESSOR WILL COME TO A HALT AFTER PLACING THE ERROR NUMBER IN LOCATION \$FATAL [I.E. LOCATION 100]. IN MOST CASES THE COMMENTS BESIDES THE HALTS TELL WHAT WAS BEING CHECKED. IN SOME CASES THE TEST CAN GET TO A HALT VIA 2 WAYS:

- 1) WRONG TEST SEQUENCE
- 2) ERROR IN ACTUAL TEST

WHEN A HALT DOES OCCUR IT IS RECOMMENDED THAT THE TEST SEQUENCE LOCATION [I.E. LOCATION 102] BE CHECKED TO VERIFY THAT IT MATCHES THE PRESENT TEST NUMBER. IF IT DOESN'T, THEN THE HALT OCCURED BECAUSE THE TEST SEQUENCE WAS NOT CORRECT OTHERWISE THE HALT IS DUE TO AN ERROR IN THE TEST.

202 6.2 ERROR RECOVERY  
203  
204 RESTART AT 200 OR 450 (SEE SEC 4.1)  
205  
206  
207 7. RESTRICTIONS  
208  
209 NONE  
210  
211  
212  
213 8. MISCELLANEOUS  
214  
215  
216  
217 8.1 EXECUTION TIME  
218  
219 EXECUTION TIME OF THE DIAGNOSTIC IS LESS THAN A SECOND, FIRST  
220 "END PASS" WILL BE TYPED OUT WITHIN A SECOND AND EVERY COSECUTIVE  
221 "END PASS" WILL BE TYPED OUT WITHIN 20 SECONDS (SEE SEC 4.2)  
222  
223 WHEN RUNNING UNDER APT IN A SCRIPT, THE FIRST PASS RUN TIME  
224 AND SUBSEQUENT PASS RUN TIMES ARE ONE (1) SECOND.  
225  
226  
227 8.2 STACK POINTER  
228  
229 STACK IS INITIALLY SET TO 450  
230  
231  
232  
233 8.3 PASS COUNT  
234  
235 A 16 BIT LOCATION "SPASS" (I.E. LOCATION 104) IS USED TO KEEP  
236 PASS COUNT. IT CAN BE CLEARED BY RESTARTING THE PROGRAM AT 200  
237  
238  
239  
240 8.4 TEST NUMBER  
241  
242 A 16 BIT LOCATION "STEST" (I.E. LOCATION 102) IS USED TO KEEP TRACK  
243 OF THE TEST NUMBER, UPPER BYTE OF THIS LOCATION GIVES THE ITERATION  
244 NUMBER AND THE LOWER BYTE THE TEST THAT WAS BEING EXECUTED  
245  
246  
247 8.5 POWER FAIL  
248  
249 THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. TO USE,  
250 START THE DIAGNOSTIC AS USUAL AND POWER DOWN THEN UP AT ANY TIME.  
251 THE PROGRAM SHOULD TYPE "POWER" AND RESTART AT 450 WITH TEST # 0  
252 HOWEVER THE DIAGNOSTIC WILL NOT RECOVER IF IT IS STORED IN A  
253 MEMORY NOT CAPABLE OF HOLDING DATA WITH POWER DOWN  
254  
255  
256  
257 9. PROGRAM DESCRIPTION

258  
259  
260  
261  
262  
263  
264

THIS PROGRAM TESTS ALL THE BASIC INSTRUCTIONS OF THE LSI-11 (EXCEPT TRAP-TYPE) WHICH INCLUDES CONTROL CHIP, DATA CHIP, MICROMS, PLA, AND OTHER CIRCUITRY ON THE LSI-11 CPU MODULE. TRAP DIAGNOSTIC SHOULD ALSO BE RUN TO MAKE SURE THAT THE BASIC LSI-11 IS FUNCTIONAL THIS DIAGNOSTIC DOES NOT MAKE A PASS WITH T-BIT SET.

%

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

```
.ABS  
;; LSI-11 MACRO INSTRUCTION EXERCISER  
.NLIST MC,MD,CND  
.LIST ME  
.TITLE CVKAAC  
;*COPYRIGHT (C) 1975, 1978  
;*DIGITAL EQUIPMENT CORP.  
;*MAYNARD, MASS. 01754  
;*.  
;*PROGRAM BY PERVEZ ZAKI  
;*.  
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.  
;*.  
$*N 1  
$SWR-16000C ;;:ALT ON ERROR, LOOP ON TEST, INHIBIT EPROR TYP0UT
```

```

290          .=0
291
292          ::*****
293
294          :      TRAP CATCHERS OF .+2 AND HALT IN LOCATIONS 0 THRU 776 [IT IS NLISTED]
295
296          .SBTTL ACT11 HOOKS
297
298          ::*****
299          :HOOKS REQUIRED BY ACT11
300          $SVPCC=          :SAVE PC
301          .=46
302          $ENDAD          ::1)SET LOC.46 TO ADDRESS OF $END.D IN .$EOP
303          .=52
304          .WORD 0          ::2)SET LOC.52 TO ZERO
305          .-$SVPCC        :: RESTORE PC
306
307
308
309
310
311
312
313
314
315
316
317
318
319          000076          . 76
320          000000          R0      =%0
321          000001          R1      =%1
322          000002          R2      =%2
323          000003          R3      =%3
324          000004          R4      =%4
325          000005          R5      =%5
326          000006          R6      =%6
327          000006          SP      =%6
328          000007          PC      =%7
329          000254          CLNZ     =254
330          000001          ERRNM    =1
331          000260          NOP1     =260
332          000263          SEVC     =263
333          000273          SENVC    =273
334          000000          $TN      =0
335          000004          .TYPE    =IC'

```



```
336 .SBTTL APT MAILBOX-E*ABLE
337
338
339
340 000076
341 000076 000000
342 000100 000000
343 000102 000000
344 000104 000000
345 000106 000000
346 000110 000000
347 000112 000000
348 000114 000000
349 000116
350 000116 000
351 000117 000
352 000120 000000
353 000122 000000
354 000124 000000
355
356
357
358
359
360
361 000126
362
363 .SBTTL APT PARAMETER BLOCK
364
365
366
367
368 000126
369 000024
370 000024 000200
371 000044 000044
372 000044 000126
373 000126
374
375
376
377
378 000126
379 000126 000000
380 000130 000076
381 000132 000001
382 000134 000001
383 000136 000000
384 000140 000014

*****
.EVEN
$MAIL: :: APT MAILBOX
$MSGTY: .WORD AMSGTY :: MESSAGE TYPE CODE
$FATAL: .WORD AFATAL :: FATAL ERROR NUMBER
$TESTN: .WORD ATESTN :: TEST NUMBER
$PASS: .WORD APASS :: PASS COUNT
$DEVCT: .WORD ADEVCT :: DEVICE COUNT
$UNIT: .WORD AUNIT :: I/O UNIT NUMBER
$MSGAD: .WORD AMSGAD :: MESSAGE ADDRESS
$MSGLG: .WORD AMSGLG :: MESSAGE LENGTH
$ETABLE: :: APT ENVIRONMENT TABLE
$ENV: .BYTE AENV :: ENVIRONMENT BYTE
$ENVM: .BYTE AENVM :: ENVIRONMENT MODE BITS
$SWREG: .WORD ASWREG :: APT SWITCH REGISTER
$USWR: .WORD AUSWR :: USER SWITCHES
$CPUOP: .WORD ACPUOP :: CPU TYPE, OPTIONS
*
* BIT 15-11=CPU TYPE
* 11/04=01,11/05-02,11/20=03,11/40=04,11/45=05
* 11/70=06,PDQ=07,0-10
*
* BIT 10=REAL TIME CLOCK
* BIT 9=FLOATING POINT PROCESSOR
* BIT 8=MEMORY MANAGEMENT
*
$*END:
.MEXIT
.SBTTL APT PARAMETER BLOCK
*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
*****
.$X= :: SAVE CURRENT LOCATION
.=24 :: SET POWER FAIL TO POINT TO START OF PROGRAM
200 :: FOR APT START UP
.=44 :: POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR :: POINT TO APT HEADER BLOCK
.=.$X :: RESET LOCATION COUNTER
*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0 :: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL :: ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM: .WORD 1 :: RUN TIME OF LONGEST TEST
$PASTM: .WORD 1 :: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD :: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
.WORD $*END-$MAIL/? :: ENTH MAILBOX-ETABLE (WORDS)
```

385		000126				.-SAPTHD	
386	000126					ADR:	
387		000130				.=ADR+2	
388	000130					ADR1:	
389		000132				.=ADR1+2	
390	000132					ADR2:	
391		000134				.=ADR2+2	
392	000134					DUMMY:	
393		000136				.=DUMMY+2	
394	000136					TEMP:	
395		000140				.=TEMP+2	
396	000140					TEMP1:	
397		000142				.=TEMP1+2	
398	000142					TEMP2:	
399		000144				.=TEMP2+2	
400	000144	177564				TPS:	177564
401	000146	177566				TPB:	177566
402	000150	006402				MARK:	2
403	000152	005015	042440	042116	MARK2:		
404	000160	050040	051501	000123	ENDPAS:	.ASCIZ	<15><12>' END PASS''
405							
406	000166	005015	047520	042527	POWER:	.EVEN	
407	000174	000122				.ASCIZ	<15><12>/POWER/
408							
409	000176	000023				.EVEN	
410						.BLKW	19.
411		000020					
412	000020	017004				.-20	
413	000022	000000				TYPE	
414						0	
415							
416							

: OUTPUT TTY STATUS REGISTER  
: OUTPUT BUFFER

```

417
418
419           ;STARTING OF THE PROGRAM
420           ;-----
421
422           . =200
423 000200 012737 016754 000024      MOV    #PWRDN,@#24      ; SERVICE POWER DOWN ROUTINE ON ANY FUTURE POWER DOWN
424 000206 012700 000116              MOV    #SETABLE,R0
425 000212 005040              2$:   CLR    -(R0)           ; START CLEANING THE STACK
426 000214 020027 000076              CMP    R0,#$MAIL      ; FOR INITIALIZATION
427 000220 101374              BHI   2$
428 000222 000167 000222              JMP    START
429           ;
430           ;-----
431
432           . 450
433
434 000450 012706 000450      START: MOV    #START,SP      ; SET THE STACK POINTER
435 000454 012705 000102      MOV    #TESTN,R5     ; PLACE THE ADDRESS OF LOCATION $TESTN IN R5
436 000460 005715              TST    (R5)           ; CHECK THE SEQUENCE COUNTER
437 000462 001401              BEQ   NOBIT           ; IF THIS IS THE STARTING OF THE TEST THEN
438                                ; GO TO NOBIT TEST
439 000464 000000              HALT                  ; OTHERWISE HALT AND WAIT FOR THE OPERATOR
440                                ; TO START AT THE PROPER TEST NUMBER
441

```

442  
443  
444  
445  
446 000466  
447 000466 021527 000000  
448 000472 001017  
449 000474 005215  
450 000476 000257  
451 000500 103414  
452 000502 102413  
453 000504 001412  
454 000506 100411  
455 000510 000260  
456 000512 103407  
457 000514 102406  
458 000516 001405  
459 000520 100404  
460 000522 002403  
461 000524 003402  
462 000526 101401  
463 000530 101004  
464 000532  
465 000532 012745 000001  
466 000536 005245  
467 000540 000000  
468 000542 102000  
469  
470  
471  
472  
473  
474  
475 000544  
476 000544 021527 000001  
477 000550 001012  
478 000552 005215  
479 000554 000270  
480 000556 100007  
481 000560 001406  
482 000562 002005  
483 000564 003004  
484 000566 103403  
485 000570 101402  
486 000572 103401  
487 000574 003404  
488 000576  
489 000576 012745 000002  
490 000602 005245  
491 000604 000000  
492 000606 001000

\*\*\*\*\*  
: \*TEST: 0 CHECK BRANCH INSTRUCTIONS WITH ZERO CONDITION CODES  
: \*\*\*\*\*

NBIT:           CMP       (R5),#0  
                  BNE       CC0           ; IF IN WRONG SEQUENCE GO TO HALT AT END OF THE TEST  
1\$:               INC       (R5)           ; ZERO CONDITION CODES, NZVC=0000  
                  CCC  
                  BCS       CC0  
                  BVS       CC0  
                  BEQ       CC0  
                  BMI       CC0  
                  NOP1           ; CHECK NOP1 INSTRUCTION I.E. OP-CODE 260  
                  BCS       CC0  
                  BVS       CC0  
                  BEQ       CC0  
                  BMI       CC0  
                  BLT       CC0  
                  BLE       CC0  
                  BLOS       CC0  
                  BHI       ENDCC0  
CC0:              MOV       #1, -(R5)  
                  INC       -(R5)  
                  HALT           ; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE  
ENDCC0:          BVC       NBIT

\*\*\*\*\*  
: \*TEST: 1 CHECK BRANCH INSTRUCTIONS WITH N BIT SET  
: \*\*\*\*\*

NBIT:           CMP       (R5),#1  
                  BNE       CC1           ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
1\$:               INC       (R5)           ; NBIT IS SET, NZVC=1000  
                  SEN  
                  BPL       CC1  
                  BEQ       CC1  
                  BGE       CC1  
                  BGT       CC1  
                  BCS       CC1  
                  BLOS       CC1  
                  BLO       CC1  
                  BLE       ENDCC1  
CC1:              MOV       #2, -(R5)  
                  INC       -(R5)  
                  HALT           ; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE  
ENDCC1:          BNE       VBIT

493  
494  
495  
496  
497 000610  
498 000610 021527 000002  
499 000614 001014  
500 000616 005215  
501 000620 000270  
502 000622 000262  
503 000624 102010  
504 000626 001407  
505 000630 100006  
506 000632 103405  
507 000634 002404  
508 000636 003403  
509 000640 101402  
510 000642 103401  
511 000644 003004  
512 000646  
513 000646 012745 000003  
514 000652 005245  
515 000654 000000  
516 000656 002000  
517  
518  
519

```
*****  
*TEST: 2 CHECK BRANCH INSTRUCTIONS WITH N&V BITS SET  
*****  
VBIT:  CMP (R5),#2  
      BNE CC2 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
1$:  INC (R5)  
      SEN  
      SEV ; V AND N BIT SET, NZVC = 1010  
      BVC CC2  
      BEQ CC2  
      BPL CC2  
      BCS CC2  
      BLT CC2  
      BLE CC2  
      BLOS CC2  
      BLO CC2  
      BGT ENDCC2  
CC2:  MOV #3, -(R5)  
      INC -(R5)  
      HALT ; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE  
ENDCC2: BGE CBIT
```

520  
521  
522  
523 000660  
524 000660 021527 000003  
525 000664 001013  
526 000666 005215  
527 000670 000270  
528 000672 000262  
529 000674 000261  
530 000676 001406  
531 000700 100005  
532 000702 102004  
533 000704 002403  
534 000706 003402  
535 000710 101001  
536 000712 002004  
537 000714  
538 000714 012745 000004  
539 000720 005245  
540 000722 000000  
541

```
*****  
*TEST: 3 CHECK BRANCH INSTRUCTIONS WITH N,V&C BITS SET  
*****  
CBIT:  CMP (R5),#3  
      BNE CC3 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
1$:  INC (R5)  
      SEN  
      SEV ; C, V, AND N BITS ARE SET, NZVC=1011  
      SEC  
      BEQ CC3  
      BPL CC3  
      BVC CC3  
      BLT CC3  
      BLE CC3  
      BHI CC3  
      BGE ZBIT  
CC3:  MOV #4, -(R5)  
      INC -(R5)  
      HALT ; ONE OF THE ABOVE BRANCHES FAILED  
      ; OR WRONG SEQUENCE
```

```

542
543
544
545
546 000724
547 000724 021527 000004
548 000730 001015
549 000732 005215
550 000734 000270
551 000736 000262
552 000740 000261
553 000742 000264
554 000744 001007
555 000746 100006
556 000750 102005
557 000752 103004
558 000754 002403
559 000756 003002
560 000760 101001
561 000762 001404
562 000764
563 000764 012745 000005
564 000770 005245
565 000772 000000
566
567
568
569
570
571
572
573 000774
574 000774 021527 000005
575 001000 001014
576 001002 005215
577 001004 000277
578 001006 100011
579 001010 001010
580 001012 102007
581 001014 103006
582 001016 000240
583 001020 100004
584 001022 001003
585 001024 102002
586 001026 103001
587 001030 101404
588 001032
589 001032 012745 000006
590 001036 005245
591 001040 000000

```

```

*****
TEST: 4 CHECK BRANCH INSTRUCTIONS WITH N,Z,V&C BITS SET
*****

```

```

ZBIT:
      CMP      (R5),#4
      BNE     CC4           ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC     (R5)
      SEN
      SEV
      SEC
      SEZ           ; ALL BITS SET, NZVC=1111
      BNE     CC4
      BPL     CC4
      BVC     CC4
      BCC     CC4
      BLT     CC4
      BGT     CC4
      BHI     CC4
      BEQ     YESCC
CC4:  MOV     #5,-(R5)
      INC     -(R5)
      HALT           ; ONE OF THE ABOVE BRANCHES FAILED
                       ; OR WRONG SEQUENCE

```

```

*****
TEST: 5 CHECK BRANCH INSTRUCTIONS WITH ALL THE CONDITION CODES SET
*****

```

```

YESCC:
      CMP      (R5),#5
      BNE     CC6           ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC     (R5)
      SCC           ; NZVC=1111
      BPL     CC6
      BNE     CC6
      BVC     CC6
      BCC     CC6
      NOP           ; CHECK NOP INSTRUCTION
      BPL     CC6
      BNE     CC6
      BVC     CC6
      BCC     CC6
      BLOS    NOTCC
CC6:  MOV     #6,-(R5)
      INC     -(R5)
      HALT           ; ONE OR A BRANCH FAILED, OR WRONG SEQUENCE

```

592  
593  
594  
595  
596 001042  
597 001042 021527 000006  
598 001046 001013  
599 001050 005215  
600 001052 000277  
601 001054 000241  
602 001056 103407  
603 001060 000242  
604 001062 102405  
605 001064 000244  
606 001066 001403  
607 001070 000250  
608 001072 100401  
609 001074 101004  
610 001076  
611 001076 012745 000007  
612 001102 005245  
613 001104 000000  
614 001106 100000  
615  
616  
617  
618  
619  
620  
621  
622 001110  
623 001110 021527 000007  
624 001114 001404  
625 001116 012745 000010  
626 001122 005245  
627 001124 000000  
628 001126 005215  
629 001130 000416  
630 001132 012745 000011  
631 001136 005245  
632 001140 000000  
633 001142 000404  
634 001144 012745 000012  
635 001150 005245  
636 001152 000000  
637 001154 000411  
638 001156 012745 000013  
639 001162 005245  
640 001164 000000  
641 001166 000765  
642 001170 012745 000014  
643 001174 005245  
644 001176 000000  
645 001200 000400

\*\*\*\*\*  
\*TEST: 6 CLEAR THE CONDITION CODES  
\*\*\*\*\*

NOTCC:

CMP (R5),#6  
BNE CC5 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
SCC ; NZVC=1111  
CLC ; NZVC=1110  
BCS CC5 ; NZVC=1100  
CLV ; NZVC=1100  
BVS CC5 ; NZVC=1000  
CLZ ; NZVC=0000  
BEQ CC5  
CLN  
BMI CC5  
BHI ENDCC5

CC5:

MOV #7, -(R5)  
INC -(R5)  
HALT ; ONE OF THE ABOVE CLEARS FAILED OR WRONG SEQUENCE  
ENDCC5: BPL BRANCH

\*\*\*\*\*  
\*TEST: 7 CHECK FORWARD AND BACKWARD BRANCHES.  
\*\*\*\*\*

BRANCH:

CMP (R5),#7  
BEQ 1\$ ; IF IN WRONG SEQUENCE GO TO HLT  
MOV #10, -(R5)  
INC -(R5)  
HALT  
1\$: INC (R5)  
BR 4\$ ; CHECK BRANCH FORWARD AND BACKWARD  
MOV #11, -(R5)  
INC -(R5)  
HALT ; FORWARD BRANCH FAILED  
2\$: BR 3\$  
MOV #12, -(R5)  
INC -(R5)  
HALT ; FORWARD BRANCH FAILED  
3\$: BR 5\$  
MOV #13, -(R5)  
INC -(R5)  
HALT ; FORWARD BRANCH FAILED  
4\$: BR 2\$  
MOV #14, -(R5)  
INC -(R5)  
HALT ; BACKWARD BRANCH FAILED  
5\$: BR IMP1

```

646
647
648
649
650 001202
651 001202 021527 000010
652 001206 001033
653 001210 005215
654 001212 012700 001232
655 001216 000277
656 001220 000110
657 001222 012745 000015
658 001226 005245
659 001230 000000
660 001232
661 001232 100003
662 001234 001002
663 001236 102001
664 001240 103404
665 001242
666 001242 012745 000016
667 001246 005245
668 001250 000000
669 001252 020027 001232
670 001256 001404
671 001260 012745 000017
672 001264 005245
673 001266 000000
674 001270 012700 001306
675 001274 000110
676 001276
677 001276 012745 000020
678 001302 005245
679 001304 000000

```

```

:*****
:*TEST: 10 CHECK JMP INSTRUCTIONS FOR MODE 1
:*****

```

```

JMP1:
      CMP      (R5),#10
      BNE     ENDJP1      ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF THE TEST
1$:   INC      (R5)
      MOV     #2$,R0      ; TEST JUMP INSTRUCTION MODE 1
      SCC
      JMP     (R0)
      MOV     #15,-(R5)
      INC     -(R5)
      HALT      ; JUMP INSTRUCTION FAILED
2$:   BPL     3$
      BNE     3$
      BVC     3$
      BCS     4$
3$:   MOV     #16,-(R5)
      INC     -(R5)
      HALT      ; WRONG CC
4$:   CMP     R0,#2$
      BEQ     5$          ; CONTINUE IF R0 IS OK
      MOV     #17,-(R5)
      INC     -(R5)
      HALT
5$:   MOV     #JMP2,R0      ; TEST JUMP INSTRUCTION MODE 1
      JMP     (R0)
ENDJP1:
      MOV     #20,-(R5)
      INC     -(R5)
      HALT      ; JUMP INSTRUCTION FAILED OR WRONG SEQUENCE

```

```

680
681
682
683
684
685
686 001306
687 001306 021527 000011
688 001312 001073
689 001314 005215
690 001316 012700 001336
691 001322 000277
692 001324 000120
693 001326 012745 000021
694 001332 005245
695 001334 000000
696 001336
697 001336 100003
698 001340 001002
699 001342 102001
700 001344 103404
701 001346

```

```

:*****
:*TEST: 11 CHECK JMP INSTRUCTIONS FOR MODES 2 AND 3
:*****

```

```

JMP2:
      CMP     (R5),#11
      BNE     ENDJP3      ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF TEST
      INC     (R5)
      MOV     #3$,R0      ; TEST JUMP INSTRUCTION MODE 2
      SCC
      JMP     (R0)+
      MOV     #21,-(R5)
      INC     -(R5)
      HALT      ; JUMP INSTRUCTION FAILED
3$:   BPL     4$
      BNE     4$
      BVC     4$
      BCS     5$
4$:

```



CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 18  
 CVKAAC.P11 09-OCT-78 08:58 111

CHECK JMP INSTRUCTIONS FOR MODES 2 AND 3

SEQ 0017

702	001346	012745	000022			MOV	#22, -(R5)	
703	001352	005245				INC	-(R5)	
704	001354	000000				HALT		: WRONG CC
705	001356	020027	001340	5\$:		CMP	R0, #3\$+2	: IS THERE AUTO INC.?
706	001362	001404				BEQ	6\$	
707	001364	012745	000023			MOV	#23, -(R5)	
708	001370	005245				INC	-(R5)	
709	001372	000000				HALT		: MODE 2 FAILED FOR JMP INSTRUCTION
710	001374	012700	001412	6\$:		MOV	#JMP3, R0	: TEST JUMP INSTRUCTION MODE 2
711	001400	000120				JMP	(R0)+	
712	001402	012745	000024			MOV	#24, -(R5)	
713	001406	005245				INC	-(R5)	
714	001410	000000				HALT		: JUMP INSTRUCTION FAILED
715								
716	001412	012767	001446	176516	JMP3:	MOV	#3\$, TEMP	: TEST JUMP INSTRUCTION MODE 3
717	001420	012767	001466	176512		MOV	#4\$, TEMP+2	:
718	001426	012700	000136			MOV	#TEMP, R0	
719	001432	000277				SCC		
720	001434	000130				JMP	@(R0)+	:
721	001436	012745	000025			MOV	#25, -(R5)	
722	001442	005245				INC	-(R5)	
723	001444	000000				HALT		: JUMP INSTRUCTION FAILED
724	001446	027067	000000	000012	3\$:	CMP	@(R0), 4\$	: IS THERE AUTO INC.?
725	001454	001404				BEQ	4\$	
726	001456	012745	000026			MOV	#26, -(R5)	
727	001462	005245				INC	-(R5)	
728	001464	000000				HALT		: JMP INSTRUCTION FAILED IN MODE 2
729	001466	012767	001512	176442	4\$:	MOV	#JMP4, TEMP	: TEST JUMP INSTRUCTION MODE 3
730	001474	012700	000136			MOV	#TEMP, R0	
731	001500	000130				JMP	@(R0)+	
732	001502				ENDJP3:			
733	001502	012745	000027			MOV	#27, -(R5)	
734	001506	005245				INC	-(R5)	
735	001510	000000				HALT		: JUMP ERROR OR WRONG SEQUENCE

```

736
737
738
739
740 001512
741 001512 021527 000012
742 001516 001075
743 001520 005215
744 001522 012700 001544
745 001526 000277
746 001530 000140
747 001532 012745 000030
748 001536 005245
749 001540 000000
750 001542 000404
751 001544
752 001544 012745 000031
753 001550 005245
754 001552 000000
755 001554 022700 001542
756 001560 001404
757 001562 012745 000032
758 001566 005245
759 001570 000000
760 001572 012700 001612
761 001576 000140
762 001600 012745 000033
763 001604 005245
764 001606 000000
765
766 001610 012767 001642 176322
767 001616 012700 000140
768 001622 012767 001652 176306
769 001630 000150
770 001632 012745 000034
771 001636 005245
772 001640 000000
773 001642
774 001642 012745 000035
775 001646 005245
776 001650 000000
777 001652 022700 000136
778 001656 001404
779 001660 012745 000036
780 001664 005245
781 001666 000000
782 001670 012767 001642 176242
783 001676 012700 000140
784 001702 012767 001722 176226
785 001710 000150
786 001712
787 001712 012745 000037
788 001716 005245
789 001720 000000

```

```

*****
:TEST: 12 TEST JUMP INSTRUCTION FOR MODE 4, 5
*****

```

```

JMP4:
  CMP      (R5),#12
  BNE     ENDJP5      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #3$,R0      ; TEST JUMP INSTRUCTION MODE 4
  SCC
  JMP     -(R0)
  MOV     #30,-(R5)
  INC     -(R5)
  HALT
  BR      4$          ; JUMP INSTRUCTION FAILED
                          ; JUMP SHOULD LAND HERE
3$:
  MOV     #31,-(R5)
  INC     -(R5)
  HALT
                          ; NO AUTO DECREMENT FROM JMP4
4$:
  CMP     #3$-2,R0    ; CHECK R0
  BEQ     5$
  MOV     #32,-(R5)
  INC     -(R5)
  HALT
5$:
  MOV     #JMP5+2,R0  ; TEST JUMP INSTRUCTION MODE 4
  JMP     -(R0)
  MOV     #33,-(R5)
  INC     -(R5)
  HALT
                          ; JUMP INSTRUCTION FAILED
JMP5:
  MOV     #3$,TEMP1   ; TEST JUMP INSTRUCTION MODE 5
  MOV     #TEMP1,R0
  MOV     #4$,TEMP1-2
  JMP     @-(R0)
  MOV     #34,-(R5)
  INC     -(R5)
  HALT
                          ; JUMP INSTRUCTION FAILED
5$:
  MOV     #35,-(R5)
  INC     -(R5)
  HALT
                          ; ERROR, NO AUTO DECREMENT
4$:
  CMP     #TEMP1-2,R0 ; CHECK R0
  BEQ     5$
  MOV     #36,-(R5)
  INC     -(R5)
  HALT
                          ; JUMP INSTRUCTION FAILED IN MODE 5
5$:
  MOV     #3$,TEMP1   ; TEST JUMP INSTRUCTION MODE 5
  MOV     #TEMP1,R0
  MOV     #JMP6,TEMP1-2
  JMP     @-(R0)
ENDJP5:
  MOV     #37,-(R5)
  INC     -(R5)
  HALT
                          ; JUMP ERROR OR WRONG SEQUENCE

```

```

*****

```

```

792
793
794
795 001722
796 001722 021527 000013
797 001726 001071
798 00173C 005215
799 001732 012703 001760
800 001736 000163 177772
801 001742 012745 000040
802 001746 005245
803 001750 000000
804 001752 020327 001760
805 001756 001404
806 001760 012745 000041
807 001764 005245
808 001766 000000
809
810 001770 000167 000010
811 001774 012745 000042
812 002000 005245
813 002002 000000
814 002004 012703 002024
815 002010 000163 000000
816 002014 012745 000043
817 002020 005245
818 002022 000000
819
820 002024 012703 000136
821 002030 012713 002050
822 002034 000173 000000
823 002040 012745 000044
824 002044 005245
825 002046 000000
826 002050 012713 002074
827 002054 012700 000132
828 002060 000170 000004
829 002064 012745 000045
830 002070 005245
831 002072 000000
832 002074 012767 002122 176034
833 002102 012700 000136
834 002106 000170 000000
835 002112
836 002112 012745 000046
837 002116 005245
838 002120 000000

```

```

:*TEST: 13 TEST JMP INSTRUCTION FOR MODE 6 AND 7
:*****

```

```

JMP6:
      CMP      (R5),#13
      BNE     ENDJP7      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC     (R5)
      MOV     #1$+6,R3
      JMP     -6(R3)
      MOV     #40,-(R5)
      INC     -(R5)
      HALT
1$:   CMP     R3,#1$+6    ; JUMP INSTRUCTION FAILED
      BEQ     2$         ; CHECK R3
      MOV     #41,-(R5)
      INC     -(R5)
      HALT
2$:   JMP     3$-4(PC)   ; WRONG VALUE IN REGISTER AFTER JUMP MODE 6
      MOV     #42,-(R5)  ; OR JUMP INSTRUCTION FAILED IN MODE 6
      INC     -(R5)      ; TEST JUMP INSTRUCTION MODE 6
      HALT
3$:   MOV     #JMP7,R3   ; JUMP INSTRUCTION FAILED
      JMP     0(R3)      ; JUMP SHOULD LAND HERE
      MOV     #43,-(R5)
      INC     -(R5)
      HALT
      ; JUMP INSTRUCTION FAILED

JMP7:
      MOV     #TEMP,R3
      MOV     #1$(R3)
      JMP     @R3
      MOV     #44,-(R5)
      INC     -(R5)
      HALT
1$:   MOV     #3$(R3)    ; JUMP INSTRUCTION FAILED
      MOV     #TEMP-4,R0 ; TEST JUMP INSTRUCTION MODE 7
      JMP     @4(R0)
      MOV     #45,-(R5)
      INC     -(R5)
      HALT
3$:   MOV     #JSRST,TEMP ; JUMP INSTRUCTION FAILED
      MOV     #TEMP,R0   ; CONTINUE
      JMP     @0(R0)

ENDJP7:
      MOV     #46,-(R5)
      INC     -(R5)
      HALT
      ; JUMP ERROR OR SEQUENCE ERROR

```

```

839
840
841
842
843 002122
844 002122 021527 000014
845 002126 001177
846 002130 005215
847 002132 012706 000450
848 002136 000277
849 002140 004767 000026
850 002144
851 002144 012745 000047
852 002150 005245
853 002152 000000
854 002154 022706 000450
855 002160 001441
856 002162 012745 000050
857 002166 005245
858 002170 000000
859 002172
860 002172 000003
861 002174 001002
862 002176 102001
863 002200 103404
864 002202
865 002202 012745 000051
866 002206 005245
867 002210 000000
868 002212 022706 000446
869 002216 001404
870 002220 012745 000052
871 002224 005245
872 002226 000000
873 002230 022716 002144
874 002234 001404
875 002236 012745 000053
876 002242 005245
877 002244 000000
878
879 002246 012716 002154
880 002252 000207
881 002254 012745 000054
882 002260 005245
883 002262 000000
884 002264 010546
885 002266 016746 175642
886 002272 016746 175636
887 002276 016746 175646
888 002302 010503
889 002304 004467 000130
890 002310
891 002310 012745 000055
892 002314 005245
893 002316 000000
894 002320

```

```

*****
*TEST: 14 CHECK JSR AND MARK INSTRUCTIONS
*****
JSR1ST:
CMP (R5),#14
BNF ENDJSR ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF THE TEST
INC (R5)
MOV #START,SP ; SET UP STACK POINTER.
SCL PC,3$
1$:
MOV #47, -(R5)
INC -(R5)
HALT ; JSR INSTRUCTION FAILED
2$:
CMP #START,SP ; HAS SP BEEN RESTORED?
BEQ JSRM
MOV #50, -(R5)
INC -(R5)
HALT ; SP WAS NOT RESTORED BY RTS INSTRUCTION
3$:
BPL 4$
BNE 4$
BVC 4$
BCS 5$
4$:
MOV #51, -(R5)
INC -(R5)
HALT ; WRONG CC
5$:
CMP #START-2,SP ; WAS THE SP EFFECTED?
BEQ 6$
MOV #52, -(R5)
INC -(R5)
HALT ; WRONG SP AFTER EXECUTION OF JSR INSTRUCTION
6$:
CMP #1$, (SP) ; IS THE RETURN ADDRESS =1$
BEQ 7$
MOV #53, -(R5)
INC -(R5)
HALT ; SP DID NOT HAVE CORRECT RETURN ADDRESS
7$:
MOV #2$, (SP) ; AFTER EXECUTION OF JSR INSTRUCTION
RTS PC ; SET 2$ AS THE RETURN ADDRESS
MOV #54, -(R5)
INC -(R5)
JSRM:
MOV R5, -(SP) ; RTS INSTRUCTION FAILED
MOV DUMMY, -(SP) ; MOV R5 TO STACK
MOV DUMMY, -(SP)
MOV MARK2, -(SP) ; STORE MARK 2 ON THE STACK.
MOV R5, R3 ; SAVE R5 IN R3
JSR R4, 10$
1$:
MOV #55, -(R5)
INC -(R5)
HALT ; JSR INSTRUCTION FAILED
2$:

```

895	002320	100003		BPL	3\$	
896	002322	001002		BNE	3\$	
897	002324	102001		BVC	3\$	
898	002326	103404		BCS	4\$	
899	002330					
900	002330	012743	000056	MOV	#56,-(R3)	
901	002334	005243		INC	-(R3)	
902	002336	000000		HALT		; WRONG CC
903	002340	022705	000102	4\$: CMP	#\$TESTN,R5	
904	002344	001404		BEQ	5\$	
905	002346	012743	000057	MOV	#57,-(R3)	
906	002352	005243		INC	-(R3)	
907	002354	000000		HALT		; MARK INSTRUCTION FAILED
908	002356	022706	000450	5\$: CMP	#START,SP	
909	002362	001404		BEQ	6\$	
910	002364	012745	000060	MOV	#60,-(R5)	
911	002370	005245		INC	-(R5)	
912	002372	000000		HALT		; MARK INSTRUCTION FAILED
913	002374	012701	002502	6\$: MOV	#12\$,R1	; PLACE THE ADDRESS OF 12\$ IN R1
914	002400	004011		JSR	RO,(R1)	; GO TO TAG 12\$
915	002402					
916	002402	012745	000061	MOV	#61,-(R5)	
917	002406	005245		INC	-(R5)	
918	002410	000000		HALT		; JSR INSTRUCTION FAILED
919	002412	012745	000062	MOV	#62,-(R5)	
920	002416	005245		INC	-(R5)	
921	002420	000000		HALT		; RTS BROUGHT THE PROGRAM BACK IN WRONG
922						; PLACE
923	002422	022706	000450	8\$: CMP	#START,SP	
924	002426	001443		BEQ	REGS	
925	002430	012745	000063	MOV	#63,-(R5)	
926	002434	005245		INC	-(R5)	
927	002436	000000		HALT		; STACK POINTER WAS NOT RESET
928						
929	002440	020427	002310	10\$: CMP	R4,#1\$	; IS THE RETURN ADDRESS -1\$ ?
930	002444	001404		BEQ	11\$	
931	002446	012745	000064	MOV	#64,-(R5)	
932	002452	005245		INC	-(R5)	
933	002454	000000		HALT		; WRONG RETURN ADDRESS IN LINKAGE REGISTER R4
934	002456	010605		1 \$: MOV	SP,R5	
935	002460	005725		TST	(R5)+	; SET UP ADDRESS IN R5 AT MARK 2 INSTRUCTION
936	002462	012716	002320	MOV	#2\$, (SP)	; SET RETURN ADDRESS =2\$
937	002466	000277		SCC		
938	002470	000205		RTS	R5	; RETURN USING R5 AND IN-TURN USING MARK INSTRUCTION
939	002472	012745	000065	MOV	#65,-(R5)	
940	002476	005245		INC	-(R5)	
941	002500	000000		HALT		; RTS INSTRUCTION FAILED
942						
943	002502	020027	002402	12\$: CMP	RO,#7\$	; DOES RO CONTAIN THE RETURN ADDRESS?
944						
945	002506	001404		BEQ	13\$	
946	002510	012745	000066	MOV	#66,-(R5)	
947	002514	005245		INC	-(R5)	
948	002516	000000		HALT		; WRONG RETURN ADDRESS IN LINKAGE REGISTER RO
949	002520	012700	002422	14\$: MOV	#2\$,RO	; SET RETURN ADDRESS AT 8\$
950	002524	000200		RTS	RC	

951 002526  
952 002526 012745 000067  
953 002532 005245  
954 002534 000000  
955  
956  
957  
958  
959  
960 002536  
961 002536 021527 000015  
962 002542 001034  
963 002544 005215  
964 002546 010667 175364  
965 002552 012700 000001  
966 002556 012701 000004  
967 002562 012702 000020  
968 002566 012703 000100  
969 002572 012704 000400  
970 002576 005006  
971 002600 060006  
972 002602 060106  
973 002604 060206  
974 002606 060306  
975 002610 060406  
976 002612 060506  
977 002614 022706 000627  
978 002620 001003  
979 002622 016706 175310  
980 002626 000406  
981 002630 016706 175302  
982 002634  
983 002634 012745 000070  
984 002640 005245  
985 002642 000000

ENDJSR:

MOV #67, -(R5)  
INC -(R5)  
HALT

; RTS INSTRUCTION FAILED OR SEQUENCE ERROR

\*\*\*\*\*  
\*TEST: 15 CHECK REGISTER SELECTION  
\*\*\*\*\*

REGS:

CMP (R5), #15  
BNE EREGS  
INC (R5)  
MOV R6, TEMP  
MOV #1, R0  
MOV #4, R1  
MOV #20, R2  
MOV #100, R3  
MOV #400, R4  
CLR R6  
ADD R0, R6  
ADD R1, R6  
ADD R2, R6  
ADD R3, R6  
ADD R4, R6  
ADD R5, R6  
CMP #TESTN+525, R6  
BNE 1\$  
MOV TEMP, R6  
BR TSTB0  
MOV TEMP, R6

; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST

; SAVE THE STACK POINTER  
; LOAD THE REGISTERS

; ADD UP THE REGISTERS

; CHECK IT  
; FAILED  
; RESTORE STACK POINTER  
; CONTINUE  
; RESTORE STACK POINTER

1\$:  
EREGS:

MOV #70, -(R5)  
INC -(R5)  
HALT

; REGISTER SELECTION FAILURE OR SEQUENCE ERROR

```

986
987
988
989
990
991
992
993
994 002644
995 002644 021527 000016
996 002650 001404
997 002652 012745 000071
998 002656 005245
999 002660 000000
1000 002662 005215
1001 002664 000277
1002 002666 105000
1003 002670 004737 017150
1004 002674 105700
1005 002676 004737 017150
1006 002702 112701 000377
1007 002706 004737 017236
1008 002712 105701
1009 002714 004737 017236
1010
1011
1012
1013
1014
1015
1016
1017 002720
1018 002720 021527 000017
1019 002724 001051
1020 002726 005215
1021 002730 000277
1022 002732 152702 000377
1023 002736 004737 017256
1024 002742 122702 000377
1025 002746 001404
1026 002750 012745 000072
1027 002754 005245
1028 002756 000000
1029 002760 112700 000077
1030 002764 120002
1031 002766 100004
1032 002770 012745 000073
1033 002774 005245
1034 002776 000000
1035 003000 120200
1036 003002 100404
1037 003004 012745 000074
1038 003010 005245
1039 003012 000000
1040 003014 112702 000377
1041 003020 122702 000377

```

-----  
CHECK BYTE INSTRUCTIONS, DESTINATION MODE 0 ONLY  
-----

\*\*\*\*\*  
: \*TEST: 16 NEW INSTRUCTIONS USED IN THIS SECTION ARE TSTB, CLRB, MOVB  
: \*\*\*\*\*

```

TSTB0:
      CMP      (R5),#16
      BEQ     2$      ; IF IN WRONG SEQUENCE GO TO HLT BELOW
      MOV     #71, -(R5)
      INC     -(R5)
      HALT    ; PROGRAM IS IN WRONG SEQUENCE
2$:   INC     (R5)
      SCC
      CLRB   R0      ; CLEAR THE REGISTER
      JSR   PC,@#SCC4 ; CHECK FOR CC = 4
      TSTB  RC      ; CHECK IT
      JSR   PC,@#SCC4 ; CHECK FOR CC = 4
      MOVB  #377,R1 ; LOAD THE REGISTER
      JSR   PC,@#SCC10 ; CHECK FOR CC = 10
      TSTB  R1      ; CHECK IT
      JSR   PC,@#SCC10 ; CHECK FOR CC = 10

```

\*\*\*\*\*  
: \*TEST: 17 NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB  
: \*\*\*\*\*

```

(CMPB0:
      CMP      (R5),#17
      BNE     ECMPB0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$:   INC     (R5)
      SCC
      BISB   #377,R2 ; LOAD REGISTER
      JSR   PC,@#SCC11 ; CHECK FOR CC = 11
      CMPB  #377,R2 ; CHECK COMPARE
      BEQ     2$      ; CONTINUE IF OK
      MOV     #72, -(R5)
      INC     -(R5)
      HALT    ; BISB OR CMPB INSTRUCTION FAILED
2$:   MOVB   #77,R0
      CMPB  R0,R2 ; CHECK IT AGAIN
      BPL     3$      ; CONTINUE IF OK
      MOV     #73, -(R5)
      INC     -(R5)
      HALT    ; CMPB INSTRUCTION FAILED [WRONG CC]
3$:   CMPB  R2,R0 ; ONCE MORE
      BMI     4$      ; CONTINUE IF OK
      MOV     #74, -(R5)
      INC     -(R5)
      HALT    ; CMPB INSTRUCTION FAILED [WRONG CC]
4$:   MOVB  #377,R2
      CMPB  #377,R2 ; LOAD REGISTER, SIGN EXTEND
      ; CHECK IF BYTE INSTRUCTION

```

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 25  
 CVKAAC.P11 09-OCT-78 08:58 T17

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB

SEQ 0024

1042	003024	001404		BEO	5\$		; CONTINUE IF OK
1043	003026	012745	000075	MOV	#75, -(R5)		
1044	003032	005245		INC	-(R5)		
1045	003034	000000		HALT			; CMPB BECAME CMP INSTRUCTION
1046	003036	112702	000377	MOVB	#377, R2		; LOAD REGISTER, SIGN EXTEND
1047	003042	120227	000377	CMPB	R2, #377		; CHECK IF BYTE INSTRUCTION
1048	003046	001404		BEO	BICB0		; CONTINUE IF OK
1049	003050						
1050	00305C	012745	000076	MOV	#76, -(R5)		
1051	003054	005245		INC	-(R5)		
1052	003056	000000		HALT			; WRONG CC OR WRONG SEQUENCE

ECMPB0:



053  
 1054  
 1055  
 1056  
 1057 003060  
 1058 003060 C21527 000020  
 1059 003064 001404  
 1060 003066 012745 000077  
 1061 003072 005245  
 1062 003074 000000  
 1063 003076 005215  
 1064 003100 112703 000377  
 1065 003104 112700 000252  
 1066 003110 000277  
 1067 003112 140003  
 1068 003114 004737 017066  
 1069 003120 130003  
 1070 003122 001404  
 1071 003124 012745 000100  
 1072 003130 005245  
 1073 003132 000000  
 1074 003134 132703 000125  
 1075 003140 004737 017066  
 1076 003144 150003  
 1077 003146 100404  
 1078 003150 012745 000101  
 1079 003154 005245  
 1080 003156 000000  
 1081 003160 142703 000177  
 1082 003164 004737 017256  
 1083 003170 132703 000377  
 1084 003174 004737 017256  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092 003200  
 1093 003200 021527 000021  
 1094 003204 001404  
 1095 003206 012745 000102  
 1096 003212 005245  
 1097 003214 000000  
 1098 003216 005215  
 1099 003220 112704 000177  
 1100 003224 000261  
 1101 003226 105204  
 1102 003230 004737 017320  
 1103 003234 112704 000376  
 1104 003240 105204  
 1105 003242 004737 017256  
 1106 003246 105204  
 1107 003250 004737 017172  
 1108 003254 105204

\*\*\*\*\*  
 : \*TEST: 20 NEW INSTRUCTIONS USED IN THIS SECTION ARE BICB, BITB  
 : \*\*\*\*\*

BICB0:  
 CMP (R5),#20  
 BEQ 2\$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
 MOV #77,-(R5)  
 INC -(R5)  
 HALT ; PROGRAM IS IN WRONG SEQUENCE  
 2\$: INC (R5)  
 MOVB #377,R3 ; LOAD REGISTER  
 MOVB #252,R0 ; PLACE #252 IN R0  
 JCC  
 BICB R0,R3 ; CLEAR EVERY OTHER BIT  
 JSR PC,@#5CC1 ; CHECK FOR CC = 1  
 BITB R0,R3 ; CHECK IT  
 BEQ 4\$ ; CONTINUE IF OK  
 MOV #100,-(R5)  
 INC -(R5)  
 HALT ; BICB OR BITB INSTRUCTION FAILED  
 4\$: BITB #125,R3 ; CHECK IT  
 JSR PC,@#5CC1 ; CHECK FOR CC = 1  
 BISB R0,R3 ; SET THE BITS THAT WERE CLEARED  
 BMI 6\$  
 MOV #101,-(R5)  
 INC -(R5)  
 HALT ; BISB INSTRUCTION FAILED  
 6\$: BICB #177,R3 ; CLEAR ALL THE BITS EXCEPT FOR SIGN  
 JSR PC,@#5CC11 ; CHECK FOR CC = 11  
 BITB #377,R3 ; CHECK IT  
 JSR PC,@#5CC11 ; CHECK FOR CC = 11

\*\*\*\*\*  
 : \*TEST: 21 NEW INSTRUCTIONS USED IN THIS SECTION ARE INCB, DECB  
 : \*\*\*\*\*

INCB0:  
 CMP (R5),#21  
 BEQ 1\$ ; IF IN WRONG SEQUENCE GO TO HLT  
 MOV #102,-(R5)  
 INC -(R5)  
 HALT ; PROGRAM IS IN WRONG SEQUENCE  
 1\$: INC (R5)  
 MOVB #177,R4 ; R4 = 177  
 SEC  
 INCB R4 ; ADD ONES INTO REG. 4  
 JSR PC,@#5CC13 ; CHECK FOR CC = 13  
 MOVB #376,R4  
 INCB R4  
 JSR PC,@#5CC11 ; CHECK FOR CC = 11  
 JSR PC,@#5CC5 ; CHECK FOR CC = 5  
 INCB R4

1109	003256	004737	017066		JSR	PC, @W\$CC1		; CHECK FOR CC = 1
1110	003262	122704	000001		CMPB	#1, R4		; CHECK IT
1111	003266	001404			BEQ	Z\$		; CONTINUE IF OK
1112	003270	012745	000103		MOV	#103, -(R5)		
1113	003274	005245			INC	-(R5)		
1114	003276	000000			HALT			; INCB INSTRUCTION FAILED
1115	003300	000261		2\$:	SEC			
1116	003302	105304			DECB	R4		; SUBTRACT ONES FROM REG. 4
1117	003304	004737	017172		JSR	PC, @W\$CC5		; CHECK FOR CC = 5
1118	003310	105304			DECB	R4		
1119	003312	004737	017256		JSR	PC, @W\$CC11		; CHECK FOR CC = 11
1120	003316	012704	000200		MOV	#200, R4		
1121	003322	105304			DECB	R4		
1122	003324	004737	017126		JSR	PC, @W\$CC3		; CHECK FOR CC = 3
1123	003330	105304			DECB	R4		
1124	003332	004737	017066		JSR	PC, @W\$CC1		; CHECK FOR CC = 1

1125 .....  
1126 : \*TEST: 22 NEW INSTRUCTION IN THIS SECTION IS COMB  
1127 : .....  
1128

```

1129 003336 COMBO:
1130 003336 021527 000022      CMP      (R5),#22
1131 003342 001404           BEQ      1$                ; IF IN WRONG SEQUENCE GO TO HLT
1132 003344 012745 000104      MOV      #104,-(R5)
1133 003350 005245           INC      -(R5)
1134 003352 000000           HALT
1135 003354 005215           1$:      INC      (R5)                ; PROGRAM IS IN WRONG SEQUENCE
1136 003356 112703 000252      MOV      #252,R3          ; LOAD EVERY OTHER BIT
1137 003362 000277           SCC
1138 003364 105103           COMB     R3                ; 1'S COMPLEMENT
1139 003366 004737 017066      JSR      PC,@#SCC1        ; CHECK FOR CC = 1
1140 003372 122703 000125      CMPB    #125,R3          ; CHECK IT
1141 003376 001404           BEQ      2$                ; CONTINUE IF OK
1142 003400 012745 000105      MOV      #105,-(R5)
1143 003404 005245           INC      -(R5)
1144 003406 000000           HALT
1145 003410 000277           2$:      SCC                ; COMB INSTRUCTION FAILED
1146 003412 105103           COMB     R3                ; COMPLEMENT BACK
1147 003414 004737 017256      JSR      PC,@#SCC11       ; CHECK FOR CC = 11
1148 003420 122703 000252      CMPB    #252,R3          ; CHECK IT
1149 003424 001404           BEQ      3$                ; CONTINUE IF OK
1150 003426 012745 000106      MOV      #106,-(R5)
1151 003432 005245           INC      -(R5)
1152 003434 000000           HALT
1153 003436 012703 000377      3$:      MOV      #377,R3          ; COMB INSTRUCTION FAILED
1154 003442 000277           SCC
1155 003444 105103           COMB     R3                ; CHECK FOR CC = 5
1156 003446 004737 017172      JSR      PC,@#SCC5
1157
1158
1159

```

1160 .....  
1161 : \*TEST: 23 NEW INSTRUCTION IN THIS SECTION IS NEGB  
1162 : .....  
1163

```

1164 003452 NEGB0:
1165 003452 021527 000023      CMP      (R5),#23
1166 003456 001025           BNE     ENEGB0            ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1167 003460 005215           1$:      INC      (R5)
1168 003462 112700 000001      MOV      #1,R0            ; LOAD THE REGISTER
1169 003466 105400           NEGB    R0                ; 2'S COMPLEMENT
1170 003470 004737 017256      JSR      PC,@#SCC11       ; CHECK FOR CC = 11
1171 003474 122700 000377      CMPB    #377,R0          ; CHECK IT
1172 003500 001404           BEQ      2$                ; CONTINUE IF OK
1173 003502 012745 000107      MOV      #107,-(R5)
1174 003506 005245           INC      -(R5)
1175 003510 000000           HALT
1176 003512 012700 000200      2$:      MOV      #200,R0          ; NEGB INSTRUCTION FAILED
1177 003516 105400           NEGB    R0                ; 2'S COMPLEMENT
1178 003520 004737 017320      JSR      PC,@#SCC13       ; CHECK FOR CC = 13
1179 003524 122700 000200      CMPB    #200,R0          ; CHECK IT
1180 003530 001404           BEQ      ROLBC            ; CONTINUE IF OK

```

T23 NEW INSTRUCTION IN THIS SECTION IS NEGB

SEQ 0028

18 003532  
1182 003532 012745 000110  
1183 003536 005245  
1184 003540 000000

ENEGB0:

MOV #110,-(R5)  
INC -(R5)  
HALT

; WRONG RESULT IN R0 OR WRONG SEQUENCE

```

1185
1186
1187
1188
1189 003542
1190 003542 021527 000024
1191 003546 001026
1192 003550 005215
1193 003552 112701 000040
1194 003556 000257
1195 003560 106101
1196 003562 106101
1197 003564 004737 017300
1198 003570 122701 000200
1199 003574 001404
1200 003576 012745 000111
1201 003602 005245
1202 003604 000000
1203 003606 106101
1204 003610 004737 017214
1205 003614 106101
1206 003616 122701 000001
1207 003622 001404
1208 003624
1209 003624 012745 000112
1210 003630 005245
1211 003632 000000
1212
1213
1214
1215
1216
1217
1218 003634
1219 003634 021527 000025
1220 003640 001026
1221 003642 005215
1222 003644 112702 000004
1223 003650 000257
1224 003652 106002
1225 003654 106002
1226 003656 122702 000001
1227 003662 001404
1228 003664 012745 000113
1229 003670 005245
1230 003672 000000
1231 003674 106002
1232 003676 004737 017214
1233 003702 106002
1234 003704 004737 017300
1235 003710 122702 000200
1236 003714 001404
1237 003716
1238 003716 012745 000114
1239 003722 005245
1240 003724 000000

```

```

:*****
:*TEST: 24 NEW INSTRUCTION IN THIS SECTION IS ROLB
:*****

```

```

ROLB0:
  CMP      (R5),#24
  BNE     EROLB0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOVB    #40,R1      ; LOAD REGISTER
  CCC     ; CLEAR FLAGS
  ROLB    R1          ; SHIFT
  R1      ;
  JSR     PC,@#5CC12  ; CHECK FOR CC = 12
  CMPB    #200,R1     ; CHECK IT
  BEQ     1$         ; CONTINUE IF OK
  MOV     #111,-(R5)
  INC     -(R5)
  HALT    ; ROLB INSTRUCTION FAILED
1$:
  ROLB    R1          ; SHIFT
  JSR     PC,@#5CC7   ; CHECK FOR CC = 7
  ROLB    R1          ; SHIFT
  CMPB    #1,R1      ; CHECK IT
  BEQ     RORB0      ; CONTINUE IF OK
EROLB0:
  MOV     #112,-(R5)
  INC     -(R5)
  HALT    ; WRONG RESULT IN R1 OR WRONG SEQUENCE

```

```

:*****
:*TEST: 25 NEW INSTRUCTION IN THIS SECTION IS RORB
:*****

```

```

RORB0:
  CMP      (R5),#25
  BNE     ERORB0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOVB    #4,R2      ; LOAD REGISTER
  CCC     ; CLEAR FLAGS
  RORB    R2          ; SHIFT
  R2      ;
  JSR     PC,@#5CC7   ; CHECK IT
  CMPB    #1,R2      ; CONTINUE IF OK
  BEQ     1$
  MOV     #113,-(R5)
  INC     -(R5)
  HALT    ; RORB INSTRUCTION FAILED
1$:
  RORB    R2          ; SHIFT
  JSR     PC,@#5CC7   ; CHECK FOR CC = 7
  RORB    R2          ; SHIFT
  JSR     PC,@#5CC12  ; CHECK FOR CC = 12
  CMPB    #200,R2     ; CHECK IT
  BEQ     ASLB0      ; CONTINUE IF OK
ERORB0:
  MOV     #114,-(R5)
  INC     -(R5)
  HALT

```

```

1241
1242
1243
1244
1245 003726
1246 003726 021527 000026
1247 003732 001404
1248 003734 012745 000115
1249 003740 005245
1250 003742 000000
1251 003744 005215
1252 003746 112703 000040
1253 003752 000257
1254 003754 106303
1255 003756 106303
1256 003760 004737 017300
1257 003764 122703 000200
1258 003770 001404
1259 003772 012745 000116
1260 003776 005245
1261 004000 000000
1262 004002 106303
1263 004004 004737 017214
1264 004010 106303
1265 004012 004737 017150
1266
1267
1268
1269
1270
1271 004016
1272 004016 021527 000027
1273 004022 001034
1274 004024 005215
1275 004026 112704 000004
1276 004032 000257
1277 004034 106204
1278 004036 106204
1279 004040 122704 000001
1280 004044 001404
1281 004046 012745 000117
1282 004052 005245
1283 004054 000000
1284 004056 106204
1285 004060 004737 017214
1286 004064 106204
1287 004066 004737 017150
1288 004072 112703 000202
1289 004076 106203
1290 004100 106203
1291 004102 004737 017256
1292 004106 122703 000340
1293 004112 001404
1294 004114
1295 004114 012745 000120
1296 004120 005245

```

```

:*****
: *TEST: 26 NEW INSTRUCTION IN THIS SECTION IS ASLB
:*****

```

```

ASLB0:
(CMP (R5),#26
BEQ 2$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW
MOV #115,-(R5)
INC -(R5)
HALT ; PROGRAM IS IN WRONG SEQUENCE
2$: INC (R5)
MOVB #40,R3 ; LOAD REGISTER
CCC ; CLEAR FLAGS
ASLB R3 ; SHIFT
ASLB R3
JSR PC,@#SCC12 ; CHECK FOR CC - 12
CMPB #200,R3 ; CHECK IT
BEQ 4$ ; CONTINUE IF OK
MOV #116,-(R5)
INC -(R5)
HALT ; ASLB INSTRUCTION FAILED
4$: ASLB R3 ; SHIFT
JSR PC,@#SCC7 ; CHECK FOR CC = 7
ASLB R3 ; SHIFT
JSR PC,@#SCC4 ; CHECK FOR CC = 4

```

```

:*****
: *TEST: 27 NEW INSTRUCTION IN THIS SECTION IS ASRB
:*****

```

```

ASRB0:
(CMP (R5),#27
BNE EASRB0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$: INC (R5)
MOVB #4,R4 ; LOAD REGISTER
CCC ; CLEAR FLAGS
ASRB R4 ; SHIFT
ASRB R4
CMPB #1,R4 ; CHECK IT
BEQ 2$ ; CONTINUE IF OK
MOV #117,-(R5)
INC -(R5)
HALT ; ASRB INSTRUCTION FAILED
2$: ASRB R4 ; SHIFT
JSR PC,@#SCC7 ; CHECK FOR CC = 7
ASRB R4 ; SHIFT
JSR PC,@#SCC4 ; CHECK FOR CC - 4
MOVB #202,R3 ; LOAD REGISTER
ASRB R3 ; SHIFT
ASRB R3
JSR PC,@#SCC11 ; CHECK FOR CC - 11
CMPB #340,R3 ; CHECK IT
BEQ ADCB0 ; CONTINUE IF OK
EASRB0:
MOV #120,-(R5)
INC -(R5)

```

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 32

CVKAAC.P11 09-OCT-78 08:58

T27

NEW INSTRUCTION IN THIS SECTION IS ASRB

SEQ 003

1297 004122 00000

HALT

1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353

004124  
004124 021527 000030  
004130 001404  
004132 012745 000121  
004136 005245  
004140 000000  
004142 005215  
004144 105000  
004146 000257  
004150 105500  
004152 004737 017150  
004156 000261  
004160 105500  
004162 000261  
004164 105500  
004166 004737 017046  
004172 122700 000002  
004176 001404  
004200 012745 000122  
004204 005245  
004206 000000  
004210 112700 000177  
004214 000261  
004216 105500  
004220 004737 017300  
004224 122700 000200  
004230 001404  
004232 012745 000123  
004236 005245  
004240 000000  
004242 112700 000377  
004246 000261  
004250 105500  
004252 004737 017172

\*\*\*\*\*  
 : \*TEST: 30 NEW INSTRUCTION IN THIS SECTION IS ADCB  
 : \*\*\*\*\*

```

ADCB0:
    CMP      (R5),#30
    BEQ     2$
    MOV     #121,-(R5)
    INC     -(R5)
    HALT
    ; IF IN WRONG SEQUENCE GO TO HLT BELOW
    ; PROGRAM IS IN WRONG SEQUENCE
2$:
    INC     (R5)
    CLRB   RO
    CCC
    ; CLEAR THE REGISTER
    ; CLEAR FLAGS
    ADCB   RO
    ; ADD C BIT 0
    JSR   PC,#$CC4
    ; CHECK FOR CC = 4
    SEC
    ; C=1
    ADCB   RO
    ; ADD C BIT-1
    SEC
    ; C=1
    ADCB   RO
    ; AGAIN
    JSR   PC,#$CC0
    ; CHECK FOR CC = 0
    CMPB  #2,RO
    ; CHECK IT
    BEQ   4$
    ; CONTINUE IF OK
    MOV     #1,-(R5)
    INC     -(R5)
    HALT
    ; ADCB INSTRUCTION FAILED
    ; LOAD LARGEST POSITIVE NUMBER
4$:
    MOVB   #177,RO
    SEC
    ; C=1
    ADCB   RO
    ; ADD C BIT=1
    JSR   PC,#$CC12
    ; CHECK FOR CC = 12
    CMPB  #200,RO
    ; CHECK IT
    BEQ   6$
    ; CONTINUE IF OK
    MOV     #123,-(R5)
    INC     -(R5)
    HALT
    ; ADCB INSTRUCTION FAILED
    ; LOAD -1
6$:
    MOVB   #377,RO
    SEC
    ; C=1
    ADCB   RO
    ; ADD C BIT=1
    JSR   PC,#$CC5
    ; CHECK FOR CC = 5
    
```

\*\*\*\*\*  
 : \*TEST: 31 NEW INSTRUCTION IN THIS SECTION IS SBCB  
 : \*\*\*\*\*

```

SBCB0:
    CMP      (R5),#31
    BEQ     1$
    MOV     #124,-(R5)
    INC     -(R5)
    HALT
    ; IF IN WRONG SEQUENC GO TO HLT BELOW
    ; PROGRAM IS IN WRONG SEQUENCE
    ; TEST IS IN WRONG SEQUENCE
1$:
    INC     (R5)
    MOVB   #3,R1
    CLRB   RO
    ; CLEAR REGISTER
    ; CLEAR FLAGS
    SBCB   R1
    ; SUBTRACT C BIT=0
    JSR   PC,#$CC0
    ; CHECK FOR CC = 0
    
```



VKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 34  
 VKAAC.P11 09-OCT-78 08:58 T31

NEW INSTRUCTION IN THIS SECTION IS SBCB

SEQ 0033

354	004312	122701	000003		CMPB	#3,R1		: CHECK IT
1355	004316	001404			BEQ	2\$		: CONTINUE IF OK
1356	004320	012745	000125		MOV	#125,-(R5)		
1357	004324	005245			INC	-(R5)		
1358	004326	000000			HALT			: SBCB INSTRUCTION FAILED
1359	004330	000261		2\$:	SEC			: C=1
1360	004332	105601			SBCB	R1		: SUBTRACT C BIT=1
1361	004334	000261			SEC			: C=1
1362	004336	105601			SBCB	R1		
1363	004340	004777	017046		JSR	PC,@#5CC0		: CHECK FOR CC = 0
1364	004344	122701	000001		CMPB	#1,R1		: CHECK IT
1365	004350	001404			BEQ	3\$		: CONTINUE IF OK
1366	004352	012745	000126		MOV	#126,-(R5)		
1367	004356	005245			INC	-(R5)		
1368	004360	000000			HALT			: SBCB INSTRUCTION FAILED
1369	004362	000261		3\$:	SEC			: C=1
1370	004364	105601			SBCB	R1		: SUBTRACT C BIT=1
1371	004366	004737	017150		JSR	PC,@#5CC4		: CHECK FOR CC = 4
1372	004372	000261			SEC			: C=1
1373	004374	105601			SBCB	R1		: SUBTRACT C BIT = 1
1374	004376	004737	017256		JSR	PC,@#5CC11		: CHECK FOR CC = 11
1375	004402	122701	000377		CMPB	#377,R1		: CHECK IT
1376	004406	001404			BEQ	4\$		: CONTINUE IF OK
1377	004410	012745	000127		MOV	#127,-(R5)		
1378	004414	005245			INC	-(R5)		
1379	004416	000000			HALT			: SBCB INSTRUCTION FAILED
1380	004420	112701	000200	4\$:	MOVB	#200,R1		: LOAD R1
1381	004424	000261			SEC			: C=1
1382	004426	105601			SBCB	R1		: SUBTRACT C BIT = 1
1383	004430	004737	017106		JSR	PC,@#5CC2		: CHECK FOR CC = 2

1384  
 1385  
 1386  
 1387  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393  
 1394  
 1395  
 1396 004434  
 1397 004434 021527 000032  
 1398 004440 001404  
 1399 004442 012745 000130  
 1400 004446 005245  
 1401 004450 000000  
 1402 004452 005215  
 1403 004454 000277  
 1404 004456 005000  
 1405 004460 004737 017150  
 1406 004464 005700  
 1407 004466 004737 017150  
 1408 004472 012704 177777  
 1409 004476 010401  
 1410 004500 004737 017236  
 1411 004504 005701  
 1412 004506 004737 017236  
 1413 004512 020401  
 1414 004514 001404  
 1415 004516 012745 000131  
 1416 004522 005245  
 1417 004524 000000  
 1418 004526 000263  
 1419 004530 010000  
 1420 004532 004767 012434

-----  
 CHECK WORD INSTRUCTIONS, DESTINATION MODE 0 ONLY  
 -----

\*\*\*\*\*  
 \*TEST: 32 NEW INSTRUCTIONS USED IN THIS SECTION ARE TST, CLR, MOV  
 \*\*\*\*\*

TST0:  
 CMP (R5),#32  
 BEQ 1\$ ; IF IN WRONG SEQUENCE GO TO HLT  
 MOV #130,-(R5)  
 INC -(R5)  
 HALT ; TEST IS IN WRONG SEQUENCE  
 1\$: INC (R5)  
 SCC  
 CLR R0 ; CLEAR THE REGISTER  
 JSR PC,#\$CC4 ; CHECK FOR CC = 4  
 R0 ; CHECK IT  
 TST R0 ; CHECK FOR CC = 4  
 JSR PC,#\$CC4 ; CHECK FOR CC = 4  
 MOV #177777,R4 ; LOAD THE REGISTER  
 MOV R4,R1  
 JSR PC,#\$CC10 ; CHECK FOR CC = 10  
 R1 ; CHECK IT  
 TST R1 ; CHECK FOR CC = 10  
 JSR PC,#\$CC10 ; CHECK FOR CC = 10  
 CMP R4,R1 ; CHECK R1 TO CONTAIN PROPER DATA  
 BEQ 2\$  
 MOV #131,-(R5)  
 INC -(R5)  
 HALT  
 2\$: SEVC ; SET V & C BITS  
 MOV R0,R0  
 JSR PC,\$CC5

\*\*\*\*\*  
 \*TEST: 33 NEW INSTRUCTIONS USED IN THIS SECTION ARE CMP, BIS  
 \*\*\*\*\*

1421  
 1422  
 1423  
 1424  
 1425  
 1426  
 1427  
 1428 004536  
 1429 004536 021527 000033  
 1430 004542 001126  
 1431 004544 005215  
 1432 004546 012700 177777  
 1433 004552 050002  
 1434 004554 004737 017236  
 1435 004560 020002  
 1436 004562 001404  
 1437 004564 012745 000132  
 1438 004570 005245  
 1439 004572 000000

CMP0:  
 CMP (R5),#33  
 BNE ECMP0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
 1\$: INC (R5)  
 MOV #177777,R0 ; LOAD REGISTER  
 BIS R0,R2 ; CHECK THE BIS INSTRUCTION  
 JSR PC,#\$CC10 ; CHECK FOR CC = 10  
 CMP R0,R2 ; CHECK COMPARE  
 BEQ 2\$ ; CONTINUE IF OK  
 MOV #132,-(R5)  
 INC -(R5)  
 HALT ; BIS OR CMP INSTRUCTION FAILED

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMP, BIS

SEQ 0035

440	004574	022702	000077	2\$:	CMP	#77,R2	: CHECK IT AGAIN
1441	004600	000004			BPL	3\$	: CONTINUE IF OK
1442	004602	012745	000133		MOV	#133,-(R5)	
1443	004606	005245			INC	-(R5)	
1444	004610	000000			HALT		: CMP INSTRUCTION FAILED [WRONG CC]
1445	004612	020227	000077	3\$:	CMP	R2,#77	: ONCE MORE
1446	004616	100404			BMI	BIC0	: CONTINUE IF OK
1447	004620			ECMP0:			
1448	004620	012745	000034		MOV	#134,-(R5)	
1449	004624	005245			INC	-(R5)	
1450	004626	000000			HALT		: WRONG CC OR WRONG SEQUENCE

1451  
1452  
1453  
1454  
1455 004630  
1456 004630 021527 000C34  
1457 004634 001053  
1458 004636 005215  
1459 004640 012703 177777  
1460 004644 012700 000136  
1461 004650 012710 125252  
1462 004654 000277  
1463 004656 041003  
1464 004660 004737 017066  
1465 004664 031003  
1466 004666 001404  
1467 004670 012745 000135  
1468 004674 005245  
1469 004676 000000  
1470 004700 032703 052525  
1471 004704 004737 017066  
1472 004710 052703 125252  
1473 004714 100404  
1474 004716 012745 000136  
1475 004722 005245  
1476 004724 000000  
1477 004726 042703 077777  
1478 004732 004737 017256  
1479 004736 012700 177777  
1480 004742 030003  
1481 004744 004737 017256  
1482 004750 000263  
1483 004752 040000  
1484 004754 004737 017172  
1485 004760 005700  
1486 004762 001404  
1487 004764  
1488 004764 012745 000137  
1489 004770 005245  
1490 004772 000000  
1491

\*\*\*\*\*  
\*TEST: 34 NEW INSTRUCTIONS USED IN THIS SECTION ARE BIC, BIT  
\*\*\*\*\*

BIC0:

CMP (R5),#34  
BNE EBIC0 ; IF IN WRONG SEQUENCE GO TO HLT ABOVE  
INC (R5)  
MOV #177777,R3 ; LOAD REGISTER  
MOV #TEMP,R0 ; PLACE THE ADDRESS OF LOCATION TEMP IN R0  
MOV #125252,(R0) ; SET (R0)  
SCC  
BIC (R0),R3 ; CLEAR EVERY OTHER BIT  
JSR PC,@#SCC1 ; CHECK FOR CC = 1  
BIT (R0),R3 ; CHECK IT  
BEQ 1\$ ; CONTINUE IF OK  
MOV #135,-(R5)  
INC -(R5)  
HALT ; BIC OR BIT INSTRUCTION FAILED  
1\$: BIT #52525,R3 ; CHECK IT  
JSR PC,@#SCC1 ; CHECK FOR CC = 1  
BIS #125252,R3 ; SET THE BITS THAT WERE CLEARED  
BMI 2\$ ; CONTINUE IF OK  
MOV #136,-(R5)  
INC -(R5)  
HALT ; BIT OR BIS INSTRUCTION FAILED  
2\$: BIC #77777,R3 ; CLEAR ALL THE BITS EXCEPT FOR SIGN  
JSR PC,@#SCC1 ; CHECK FOR CC = 11  
MOV #177777,R0  
BIT R0,R3 ; CHECK IT  
JSR PC,@#SCC11 ; CHECK FOR CC = 11  
SEVC ; SET V & C BITS  
BIC R0,R0  
JSR PC,@#SCC5 ; CHECK CC = 5  
TST R0 ; CHECK R0 TO CONTAIN 0  
BEQ INCO  
EBIC0:  
MOV #137,-(R5)  
INC -(R5)  
HALT ; BIC FAILED OR SEQUENCE ERROR

1492

```

493
1494
1495
1496
1497
1498 004774
1499 004774 021527 000035
1500 005000 001404
1501 005002 012745 000140
1502 005006 005245
1503 005010 000000
1504 005012 005215
1505 005014 012704 077777

```

```

*****
:*TEST: 35 NEW INSTRUCTIONS USED IN THIS SECTION ARE INC, DEC
*****

```

```

INCO:
      CMP      (R5),#35
      BEQ      2$
      MOV      #140,-(R5) ; IF IN WRONG SEQUENCE GO TO HLT BELOW
      INC      -(R5)
      HALT
2$:   INC      (R5) ; PROGRAM IS IN WRONG SEQUENCE
      MOV      #77777,R4 ; R4-77777

```

1506	005020	000261		SEC		
1507	005022	005204		INC	R4	: ADD ONES INTO REG. 4
1508	005024	004737	017320	JSR	PC, @R5CC13	: CHECK FOR CC = 13
1509	005030	012704	177776	MOV	#177776, R4	
1510	005034	005204		INC	R4	
1511	005036	004737	017256	JSR	PC, @R5CC11	: CHECK FOR CC = 11
1512	005042	005204		INC	R4	
1513	005044	004737	017172	JSR	PC, @R5CC5	: CHECK FOR CC = 5
1514	005050	005204		INC	R4	
1515	005052	004737	017066	JSR	PC, @R5CC1	: CHECK FOR CC = 1
1516	005056	022704	000001	CMP	#1, R4	: CHECK IT
1517	005062	001404		BEQ	4\$	: FAILED
1518	005064	012745	000141	MOV	#141, -(R5)	
1519	005070	005245		INC	-(R5)	
1520	005072	000000		HALT		: INC INSTRUCTION FAILED
1521	005074	000261		SEC		
1522	005076	005304		DEC	R4	: SUBTRACT ONES FROM REG. 4
1523	005100	004737	017172	JSR	PC, @R5CC5	: CHECK FOR CC = 5
1524	005104	005304		DEC	R4	
1525	005106	004737	017256	JSR	PC, @R5CC11	: CHECK FOR CC = 11
1526	005112	012704	100000	MOV	#100000, R4	
1527	005116	005304		DEC	R4	
1528	005120	004737	017126	JSR	PC, @R5CC3	: CHECK FOR CC = 3
1529	005124	005304		DEC	R4	
1530	005126	004737	017066	JSR	PC, @R5CC1	: CHECK FOR CC = 1

4\$:

```

1531
1532
1533
1534
1535 005132
1536 005132 021527 000036
1537 005136 001404
1538 005140 012745 000142
1539 005144 005245
1540 005146 000000
1541 005150 005215
1542 005152 012703 125252
1543 005156 000277
1544 005160 005103
1545 005162 004737 017256
1546 005166 022703 125252
1547 005172 001404
1548 005174 012745 000142
1549 005200 005245
1550 005202 000000
1551 005204 000277
1552 005206 005103
1553 005210 004737 017256
1554 005214 022703 125252
1555 005220 001404
1556 005222 012745 000144
1557 005226 005245
1558 005230 000000
1559 005232 012703 177777
1560 005236 000277
1561 005240 005103
1562 005242 004737 017172

```

```

*****
*TEST: 36 NEW INSTRUCTION IN THIS SECTION IS COM
*****

```

COMO:

```

CMP (R5),#36
BEQ 1$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW
MOV #142,-(R5)
INC -(R5)
HALT ; TEST IS IN WRONG SEQUENCE
1$: INC (R5)
MOV #125252,R3 ; LOAD EVERY OTHER BIT
SCC
COM R3 ; 1'S COMPLEMENT
SM PC,@$CC1 ; CHECK FOR CC = 1
M #125252,R3 ; CHECK IT
HALT ; CONTINUE IF OK
2$: SCC
COM R3 ; COMPLEMENT BACK
JSR PC,@$CC1 ; CHECK FOR CC = 1
CMP #125252,R3 ; CHECK IT
BEQ 3$ ; CONTINUE IF OK
MOV #144,-(R5)
INC -(R5)
HALT ; COM INSTRUCTION FAILED
3$: MOV #177777,R3
SCC
COM R3
JSR PC,@$CC5 ; CHECK FOR CC = 5

```

```

1563
1564
1565
1566
1567
1568
1569
1570 005246
1571 005246 021527 000037
1572 005252 001025
1573 005254 005215
1574 005256 012700 000001
1575 005262 005400
1576 005264 004737 017256
1577 005270 022700 177777
1578 005274 001404
1579 005276 012745 000145
1580 005302 005245
1581 005304 000000
1582 005306 012700 100000
1583 005312 005400
1584 005314 004737 017320
1585 005320 022700 100000
1586 005324 001404

```

```

*****
*TEST: 37 NEW INSTRUCTION IN THIS SECTION IS NEG
*****

```

NEGO:

```

CMP (R5),#37
BNE ENEG0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$: INC (R5)
MOV #1,R0 ; LOAD THE REGISTER
NEG R0 ; 2'S COMPLEMENT
JSR PC,@$CC11 ; CHECK FOR CC = 11
CMP #177777,R0 ; CHECK IT
BEQ 2$ ; CONTINUE IF OK
MOV #145,-(R5)
INC -(R5)
HALT ; NEG INSTRUCTION FAILED
2$: MOV #100000,R0
NEG R0 ; 2'S COMPLEMENT
JSR PC,@$CC13 ; CHECK FOR CC = 13
CMP #100000,R0 ; CHECK IT
BEQ ROLC ; CONTINUE IF OK

```



CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 42  
CVKAAC.P11 09-OCT-78 08:58 T37

NEW INSTRUCTION IN THIS SECTION IS NEG

SEQ 004

1587 005326  
1588 005326 012745 000146  
1589 005332 005245  
1590 005334 000000

ENEG0:

MOV #146, -(R5)  
INC -(R5)  
HALT

; WRONG RESULT IN R0 OR WRONG SEQUENCE

```

591
1592
1593
1594
1595 005336
1596 005336 021527 000040
1597 005342 001026
1598 005344 005215
1599 005346 012701 020000
1600 005352 000257
1601 005354 006101
1602 005356 006101
1603 005360 004737 017300
1604 005364 022701 100000
1605 005370 001404
1606 005372 012745 000147
1607 005376 005245
1608 005400 000000
1609 005402 006101
1610 005404 004737 017214
1611 005410 006101
1612 005412 022701 000001
1613 005416 001404
1614 005420
1615 005420 012745 000150
1616 005424 005245
1617 005426 000000
1618
1619
1620
1621
1622
1623
1624
1625 005430
1626 005430 021527 000041
1627 005434 001026
1628 005436 005215
1629 005440 012702 000004
1630 005444 000257
1631 005446 006002
1632 005450 006002
1633 005452 022702 000001
1634 005456 001404
1635 005460 012745 000151
1636 005464 005245
1637 005466 000000
1638 005470 006002
1639 005472 004737 017214
1640 005476 006002
1641 005500 004737 017300
1642 005504 022702 100000
1643 005510 001404
1644 005512
1645 005512 012745 000152
1646 005516 005245

```

```

:*****
:*TEST: 40 NEW INSTRUCTION IN THIS SECTION IS ROL
:*****

```

```

ROL0:
  CMP      (R5),#40
  BNE     ER0L0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #20000,R1  ; LOAD REGISTER
  CCC     ; CLEAR FLAGS
  ROL     R1         ; SHIFT
  ROL     R1
  JSR     PC,@#5CC12 ; CHECK FOR CC = 12
  CMP     #100000,R1 ; CHECK IT
  BEQ     1$        ; CONTINUE IF OK
  MOV     #147,-(R5)
  INC     -(R5)
  HALT
1$:
  ROL     R1         ; ROL INSTRUCTION FAILED
  JSR     PC,@#5CC7  ; CHECK FOR CC = 7
  ROL     R1         ; SHIFT
  CMP     #1,R1      ; CHECK IT
  BEQ     ROR0      ; CONTINUE IF OK
ER0L0:
  MOV     #150,-(R5)
  INC     -(R5)
  HALT              ; WRONG RESULT IN R1 OR WRONG SEQUENCE

```

```

:*****
:*TEST: 41 NEW INSTRUCTION IN THIS SECTION IS ROR
:*****

```

```

ROR0:
  CMP     (R5),#41
  BNE     EROR0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #4,R2      ; LOAD REGISTER
  CCC     ; CLEAR FLAGS
  ROR     R2         ; SHIFT
  ROR     R2
  CMP     #1,R2      ; CHECK IT
  BEQ     1$        ; CONTINUE IF OK
  MOV     #151,-(R5)
  INC     -(R5)
  HALT
1$:
  ROR     R2         ; ROR INSTRUCTION FAILED
  JSR     PC,@#5CC7  ; CHECK FOR CC = 7
  ROR     R2         ; SHIFT
  JSR     PC,@#5CC12 ; CHECK FOR CC = 12
  CMP     #100000,R2 ; CHECK IT
  BEQ     ASLO      ; CONTINUE IF OK
EROR0:
  MOV     #152,-(R5)
  INC     -(R5)

```

CVKAA MACY11 30A(1052) 09-OCT-78 08:59 PAGE 44  
CVKAA.P11 09-OCT-78 08:58 T41

NEW INSTRUCTION IN THIS SECTION IS ROR

SEQ 0042

1647 005520 000000

HALT

; WRONG RESULT IN R2 OR WRONG SEQUENCE

1648  
 1649  
 1650  
 1651  
 1652 005522  
 1653 005522 021527 000042  
 1654 005526 001404  
 1655 005530 012745 000153  
 1656 005534 005245  
 1657 005536 000000  
 1658 005540 005215  
 1659 005542 012703 020000  
 1660 005546 000257  
 1661 005550 006303  
 1662 005552 006303  
 1663 005554 004737 017300  
 1664 005560 022703 100000  
 1665 005564 001404  
 1666 005566 012745 000154  
 1667 005572 005245  
 1668 005574 000000  
 1669 005576 006303  
 1670 005600 004737 017214  
 1671 005604 006303  
 1672 005606 004737 017150  
 1673  
 1674  
 1675  
 1676  
 1677

\*\*\*\*\*  
 :\*TEST: 42 NEW INSTRUCTION IN THIS SECTION IS ASL  
 \*\*\*\*\*

ASL0:  
 CMP (R5),#42  
 BEQ 2\$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
 MOV #153,-(R5)  
 INC -(R5)  
 HALT ; PROGRAM IS IN WRONG SEQUENCE  
 2\$: INC (R5)  
 MOV #20000,R3 ; LOAD REGISTER  
 CCC ; CLEAR FLAGS  
 ASL R3 ; SHIFT  
 ASL R3 ;  
 JSR PC,@#5CC12 ; CHECK FOR CC = 12  
 CMP #100000,R3 ; CHECK IT  
 BEQ 4\$ ; CONTINUE IF OK  
 MOV #154,-(R5)  
 INC -(R5)  
 HALT ; ASL INSTRUCTION FAILED  
 4\$: ASL R3 ; SHIFT  
 JSR PC,@#5CC7 ; CHECK FOR CC = 7  
 ASL R3 ; SHIFT  
 JSR PC,@#5CC4 ; CHECK FOR CC = 4

\*\*\*\*\*  
 :\*TEST: 43 NEW INSTRUCTION IN THIS SECTION IS ASR  
 \*\*\*\*\*

1678 005612  
 1679 005612 021527 000043  
 1680 005616 001034  
 1681 005620 005215  
 1682 005622 012704 000004  
 1683 005626 000257  
 1684 005630 006204  
 1685 005632 006204  
 1686 005634 022704 000001  
 1687 005640 001404  
 1688 005642 012745 000155  
 1689 005646 005245  
 1690 005650 000000  
 1691 005652 006204  
 1692 005654 004737 017214  
 1693 005660 006204  
 1694 005662 004737 017150  
 1695 005666 012703 100002  
 1696 005672 006203  
 1697 005674 006203  
 1698 005676 004737 017256  
 1699 005702 022703 160000  
 1700 005706 001404  
 1701 005710  
 1702 005710 012745 000156  
 1703 005714 005245

ASR0:  
 CMP (R5),#43  
 BNE EASR0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
 1\$: INC (R5)  
 MOV #4,R4 ; LOAD REGISTER  
 CCC ; CLEAR FLAGS  
 ASR R4 ; SHIFT  
 ASR R4 ;  
 CMP #1,R4 ; CHECK IT  
 BEQ 2\$ ; CONTINUE IF OK  
 MOV #155,-(R5)  
 INC -(R5)  
 HALT ; ASR INSTRUCTION FAILED  
 2\$: ASR R4 ; SHIFT  
 JSR PC,@#5CC7 ; CHECK FOR CC = 7  
 ASR R4 ; SHIFT  
 JSR PC,@#5CC4 ; CHECK FOR CC = 4  
 MOV #100002,R3 ; LOAD REGISTER  
 ASR R3 ; SHIFT  
 ASR R3 ;  
 JSR PC,@#5CC11 ; CHECK FOR CC = 11  
 CMP #160000,R3 ; CHECK IT  
 BEQ ADC0 ; CONTINUE IF OK  
 EASR0:  
 MOV #156,-(R5)  
 INC -(R5)

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 46  
CVKAAC.P11 09-OCT-78 08:58 T43

NEW INSTRUCTION IN THIS SECTION IS ASR

SEQ 0045

1704 005716 000000

HALT

; WRONG RESULT IN R3 OR WRONG SEQUENCE

1705  
1706  
1707  
1708  
1709 005720  
1710 005720 021527 000044  
1711 005724 001404  
1712 005726 012745 000157  
1713 005732 005245  
1714 005734 000000  
1715 005736 005215  
1716 005740 005000  
1717 005742 000257  
1718 005744 005500  
1719 005746 004737 017150  
1720 005752 000261  
1721 005754 005500  
1722 005756 000261  
1723 005760 005500  
1724 005762 004737 017046  
1725 005766 022700 000002  
1726 005772 001404  
1727 005774 012745 000160  
1728 006000 005245  
1729 006002 000000  
1730 006004 012700 077777  
1731 006010 000261  
1732 006012 005500  
1733 006014 004737 017300  
1734 006020 022700 100000  
1735 006024 001404  
1736 006026 012745 000161  
1737 006032 005245  
1738 006034 000000  
1739 006036 012700 177777  
1740 006042 000261  
1741 006044 005500  
1742 006046 004737 017172

```
*****  
*TEST: 44 NEW INSTRUCTION IN THIS SECTION IS ADC  
*****  
ADCC:  
    CMP      (R5),#44  
    BEQ      2$  
    MOV      #157,-(R5) ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
    INC      -(R5)  
    HALT     ; PROGRAM IS IN WRONG SEQUENCE  
2$:  INC      (R5)  
    CLR      R0 ; CLEAR THE REGISTER  
    CCC     ; CLEAR FLAGS  
    ADC      R0 ; ADD C BIT = 0  
    JSR     PC,@#5CC4 ; CHECK FOR CC - 4  
    SEC     ; C=1  
    ADC      R0 ; ADD C BIT=1  
    SEC     ; C=1  
    ADC      RC ; AGAIN  
    JSR     PC,@#5CC0 ; CHECK FOR CC 0  
    CMP     #2,R0 ; CHECK IT  
    BEQ     4$ ; CONTINUE IF OK  
    MOV     #160,-(R5)  
    INC     -(R5)  
    HALT     ; ADC INSTRUCTION FAILED  
4$:  MOV     #77777,R0 ; LOAD LARGEST POSITIVE NUMBER  
    SEC     ; C=1  
    ADC     R0 ; ADD C BIT-1  
    JSR     PC,@#5CC12 ; CHECK FOR CC - 12  
    CMP     #100000,R0 ; CHECK IT  
    BEQ     6$ ; FAILED  
    MOV     #161,-(R5)  
    INC     -(R5)  
    HALT     ; ADC INSTRUCTION FAILED  
6$:  MOV     #-1,R0 ; LOAD -1  
    SEC     ; C=1  
    ADC     R0 ; ADD C BIT=1  
    JSR     PC,@#5CC5 ; CHECK FOR CC - 5
```

1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750 006052  
1751 006052 021527 000045  
1752 006056 001404  
1753 006060 012745 000162  
1754 006064 005245  
1755 006066 000000  
1756 006070 005215  
1757 006072 012701 000003  
1758 006076 000257  
1759 006100 005601  
1760 006102 004737 017046

```
*****  
*TEST: 45 NEW INSTRUCTION IN THIS SECTION IS SBC  
*****  
SBC0:  
    CMP      (R5),#45  
    BEQ      1$  
    MOV      #102,-(R5) ; IF IN WRONG SEQUENCE GO TO HLT  
    INC      -(R5)  
    HALT     ; TEST IS IN WRONG SEQUENCE  
1$:  INC      (R5)  
    MOV     #3,R1 ; LOAD REGISTER  
    CCC     ; CLEAR FLAGS  
    SBC     R1 ; SUBTRACT C BIT=0  
    JSR     PC,@#5CCC ; CHECK FOR CC - 0
```

NEW INSTRUCTION IN THIS SECTION IS SBC

SEQ 0047

1761	006106	022701	000003		CMP	#3,R1		: CHECK IT
1762	006112	001404			BEQ	2\$		: CONTINUE IF OK
1763	006114	012745	000163		MOV	#163,-(R5)		
1764	006120	005245			INC	-(R5)		
1765	006122	000000			HALT			: SBC INSTRUCTION FAILED
1766	006124	00026		2\$:	SEC			: C=1
1767	006126	005601			SBC	R1		: SUBTRACT C BIT=1
1768	006130	000261			SEC			: C=1
1769	006132	005601			SBC	R1		
1770	006134	004737	017046		JSR	PC,@#S00		: CHECK FOR CC = 0
1771	006140	022701	000001		CMP	#1,R1		: CHECK IT
1772	006144	001404			BEQ	3\$		: CONTINUE IF OK
1773	006146	012745	000164		MOV	#164,-(R5)		
1774	006152	005245			INC	-(R5)		
1775	006154	000000			HALT			: SBC INSTRUCTION FAILED
1776	006156	000261		3\$:	SEC			: C=1
1777	006160	005601			SBC	R1		: SUBTRACT C BIT=1
1778	006162	004737	017150		JSR	PC,@#S004		: CHECK FOR CC = 4
1779	006166	000261			SEC			: C=1
1780	006170	005601			SBC	R1		: SUBTRACT C BIT = 1
1781	006172	004737	017256		JSR	PC,@#S0011		: CHECK FOR CC = 11
1782	006176	022701	177777		CMP	#-1,R1		: CHECK IT
1783	006202	001404			BEQ	4\$		: CONTINUE IF F OK
1784	006204	012745	000165		MOV	#165,-(R5)		
1785	006210	005245			INC	-(R5)		
1786	006212	000000			HALT			: SBC INSTRUCTION FAILED
1787	006214	012701	100000	4\$:	MOV	#100000,R1		: LOAD R1
1788	006220	000261			SEC			: C=1
1789	006222	005601			SBC	R1		: SUBTRACT C BIT = 1
1790	006224	004737	017106		JSR	PC,@#S002		: CHECK FOR CC = 2

NEW INSTRUCTION IN THIS SECTION IS SXT

SEQ 0048

791  
1792  
1793  
1794  
1795 006230  
1796 006230 021527 000046  
1797 006234 001024  
1798 006236 005215  
1799 006240 005002  
1800 006242 000277  
1801 006244 000254  
1802 006246 006702  
1803 006250 004737 017172  
1804 006254 005702  
1805 006256 001404  
1806 006260 012745 000166  
1807 006264 005245  
1808 006266 000000  
1809 006270 000273  
1810 006272 006702  
1811 006274 004737 017256  
1812 006300 022702 177777  
1813 006304 001404  
1814 006306  
1815 006306 012745 000167  
1816 006312 005245  
1817 006314 000000

\*\*\*\*\*  
\*TEST: 46 NEW INSTRUCTION IN THIS SECTION IS SXT  
\*\*\*\*\*

SXT0:  
1\$: CMP (R5),#46  
BNE ESXT0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
CLR R2 ; CLEAR REGISTER  
SCC  
CLN  
SXT R2 ; SIGN EXTEND  
JSR PC,@#SCC5 ; CHECK FOR CC = 5  
TST R2 ; REG. 2 SHOULD STILL BE 0  
BEQ 2\$ ; CONTINUE IF OK  
MOV #166,-(R5)  
INC -(R5)  
2\$: HALT ; SXT INSTRUCTION FAILED  
SENVC ; SET N, V & C BITS  
SXT R2 ; SIGN EXTEND  
JSR PC,@#SCC11 ; CHECK FOR CC = 11  
CMP #-1,R2 ; REG. 2 SHOULD NOW HAVE -1  
BEQ SWAB0 ; CONTINUE IF OK  
ESXT0:  
MOV #167,-(R5)  
INC -(R5)  
HALT ; WRONG RESULT IN R2 OR WRONG SEQUENCE

1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825 006316  
1826 006316 021527 000047  
1827 006322 001031  
1828 006324 005215  
1829 006326 012703 125125  
1830 006332 000277  
1831 006334 000250  
1832 006336 000303  
1833 006340 004737 017236  
1834 006344 022703 052652  
1835 006350 001404  
1836 006352 012745 000170  
1837 006356 005245  
1838 006360 000000  
1839 006362 012703 000377  
1840 006366 000277  
1841 006370 000244  
1842 006372 000303  
1843 006374 004737 017150  
1844 006400 022703 177400  
1845 006404 001404  
1846 006406

\*\*\*\*\*  
\*TEST: 47 NEW INSTRUCTION IN THIS SECTION IS SWAB  
\*\*\*\*\*

SWAB0:  
1\$: CMP (R5),#47  
BNE ESWAB0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
MOV #125125,R3 ; LOAD BIT PATTERN INTO REGISTER  
SCC  
CLN  
SWAB R3 ; SWAP BYTES OF REGISTER  
JSR PC,@#SCC10 ; CHECK FOR CC = 10  
CMP #52652,R3 ; CHECK IT  
BEQ 1\$ ; CONTINUE IF OK  
MOV #170,-(R5)  
INC -(R5)  
HALT ; SWAB INSTRUCTION FAILED  
1\$: MOV #377,R3  
SCC  
CLZ  
SWAB R3  
JSR PC,@#SCC4 ; CHECK FOR CC = 4  
CMP #177400,R3  
BEQ XCP0  
ESWAB0:



NEW INSTRUCTION IN THIS SECTION IS SWAB

SEQ 0049

847	006406	012745	000171
1843	006412	005245	
1849	006414	000000	

MOV	#171, -(R5)
INC	-(R5)
HALT	

; WRONG RESULT IN R3 OR WRONG SEQUENCE

```

1850
1851
1852
1853
1854 006416
1855 006416 021527 000050
1856 006422 001034
1857 006424 005215
1858 006426 012704 177777
1859 006432 012703 177777
1860 006436 000277
1861 006440 074403
1862 006442 004737 017172
1863 006446 012703 077777
1864 006452 010400
1865 006454 000263
1866 006456 000244
1867 006460 074003
1868 006462 004737 017256
1869 006466 012702 125252
1870 006472 012704 052525
1871 006476 000277
1872 006500 074204
1873 006502 004737 017256
1874 006506 022704 177777
1875 006512 001404
1876 006514
1877 006514 012745 000172
1878 006520 005245
1879 006522 000000
1880
1881
1882
1883
1884
1885
1886
1887 006524
1888 006524 021527 000051
1889 006530 001055
1890 006532 005215
1891 006534 012701 021421
1892 006540 060101
1893 006542 004737 017046
1894 006546 022701 043042
1895 006552 001404
1896 006554 012745 000173
1897 006560 005245
1898 006562 000000
1899 006564 012700 156357
1900 006570 060000
1901 006572 004737 017256
1902 006576 022700 134736
1903 006602 001404
1904 006604 012745 000174
1905 006610 005245

```

```

*****
*TEST: 50 NEW INSTRUCTION IN THIS SECTION IS XOR
*****

```

```

XOR0:
  CMP      (R5),#50
  BNE     EXOR0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #-1,R4     ; LOAD REGISTERS
  MOV     #-1,R3     ;
  SCC
  XOR     R4,R3      ; SHOULD PRODUCE 0'S IN REG. 3
  JSR     PC,@#5CC5  ; CHECK FOR CC = 5
  MOV     #7777,R3
  MOV     R4,R0      ; PLACE A -1 IN R0
  SEVCL  ; SET V & C BITS
  CLZ
  XOR     R0,R3
  JSR     PC,@#5CC11 ; CHECK FOR CC = 11
  MOV     #125252,R2 ; LOAD REGISTERS
  MOV     #52525,R4
  SCC
  XOR     R2,R4      ; SHOULD PRODUCE ALL 1'S IN REG. 4
  JSR     PC,@#5CC11 ; CHECK FOR CC = 11
  CMP     #-1,R4     ; CHECK IT
  BEQ     ADD0      ; CONTINUE IF OK

EXOR0:
  MOV     #172,-(R5)
  INC     -(R5)
  HALT      ; WRONG RESULT IN R4 OR WRONG SEQUENCE

```

```

*****
*TEST: 51 NEW INSTRUCTION IN THIS SECTION IS ADD
*****

```

```

ADD0:
  CMP     (R5),#51
  BNE     EADD0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #21421,R1  ; LOAD REGISTERS
  ADD     R1,R1      ; ADD
  JSR     PC,@#5CC0  ; CHECK FOR CC = 0
  CMP     #43042,R1 ; CHECK IT
  BEQ     1$        ; CONTINUE IF OK
  MOV     #173,-(R5)
  INC     -(R5)
  HALT      ; ADD INSTRUCTION FAILED

1$:
  MOV     #-21421,R0 ; LOAD REGISTERS
  ADD     R0,R0      ; ADD
  JSR     PC,@#5CC11 ; CHECK FOR CC = 11
  CMP     #-43042,R0 ; CHECK IT
  BEQ     2$        ; CONTINUE IF OK
  MOV     #174,-(R5)
  INC     -(R5)

```

906	006612	000000		HALT			: ADD INSTRUCTION FAILED
1907	006614	012702	100000	2\$: MOV	#100000,R2		: LOAD REGISTERS
1908	006620	060202		ADD	R2,R2		: ADD SHOULD RESULT AS 0'S
1909	006622	004737	017214	JSR	PC,2*3CC7		: CHECK FOR CC = 7
1910	006626	012704	021421	MOV	#21421,R4		: LOAD REGISTERS
1911	006632	012701	156357	MOV	#-21421,R1		:
1912	006636	060401		ADD	R4,R1		: ADD SHOULD RESULT AS 0'S
1913	006640	001404		BEQ	3\$		: CONTINUE IF OK
1914	006642	012745	000175	MOV	#175,-(R5)		
1915	006646	005245		INC	-(R5)		
1916	006650	000000		3\$: HALT			: ADD INSTRUCTION FAILED
1917	006652	005404		NEG	R4		: SWITCH SOURCE AND DESTINATION
1918	006654	012701	021421	MOV	#21421,R1		:
1919	006660	060104		ADD	R1,R4		: SHOULD RESULT AS 0'S
1920	006662	001404		BEQ	SUB0		: CONTINUE IF OK
1921	006664			F ADD0:			
1922	006664	012745	000175	MOV	#175,-(R5)		
1923	006670	005245		INC	-(R5)		
1924	006672	000000		HALT			: WRONG RESULT IN R1 OR WRONG SEQUENCE

```

1925
1926
1927
1928
1929 006674
1930 006674 021527 000052
1931 006700 001404
1932 006702 012745 000177
1933 006706 005245
1934 006710 000000
1935 006712 005215
1936 006714 012702 021421
1937 006720 012703 156357
1938 006724 160203
1939 006726 004737 017236
1940 006732 022703 134736
1941 006736 001404
1942 006740 012745 000200
1943 006744 005245
1944 006746 000000
1945 006750 012703 021421
1946 006754 010204
1947 006756 160403
1948 006760 001404
1949 006762 012745 000201
1950 006766 005245
1951 006770 000000
1952 006772 012703 177777
1953 006776 012702 077777
1954 007002 160302
1955 007004 004737 017320
1956 007010 022702 100000
1957 007014 001404
1958 007016 012745 000202
1959 007022 005245
1960 007024 000000
1961 007026 012704 177777
1962 007032 160304
1963 007034 004737 017150
1964
1965
1966
1967
1968
1969
1970 007040
1971 007040 021527 000053
1972 007044 001032
1973 007046 005215
1974 007050 012701 177777
1975 007054 005000
1976 007056
1977 007056 106400
1978 007060 004737 017046
1979 007064
1980 007064 006701

```

```

:*****
:*TEST: 52 NEW INSTRUCTION IN THIS SECTION IS SUB
:*****

```

```

SUB0:
      CMP      (R5),#52
      BEQ     2$
      MOV     #177,-(R5) ; IF IN WRONG SEQUENCE GO TO HLT BELOW
      INC     -(R5)
      HALT
      2$:     INC     (R5) ; PROGRAM IS IN WRONG SEQUENCE
      MOV     #21421,R2 ; LOAD REGISTERS
      MOV     #-21421,R3
      SUB     R2,R3 ; RESULT SHOULD=-43042
      JSR     PC,#$CC10 ; CHECK FOR CC = 10
      CMP     #-43042,R3 ; CHECK IT
      BEQ     4$ ; CONTINUE IF OK
      MOV     #200,-(R5)
      INC     -(R5)
      HALT
      4$:     MOV     #21421,R3 ; SUB INSTRUCTION FAILED
      MOV     R2,R4 ; LOAD REGISTER
      SUB     R4,R3 ; NOW R4 = #21421
      BEQ     6$ ; RESULT SHOULD=0
      MOV     #201,-(R5)
      INC     -(R5)
      HALT
      6$:     MOV     #-1,R3 ; SUB INSTRUCTION FAILED
      MOV     #77777,R2 ; LOAD REGISTERS
      SUB     R3,R2 ; LOAD REGISTERS
      JSR     PC,#$CC13 ; RESULT SHOULD BE 100000 AND OVERFLOW
      CMP     #100000,R2 ; CHECK FOR CC - 13
      BEQ     8$ ; CHECK IT
      MOV     #201,-(R5) ; CONTINUE IF OK
      INC     -(R5)
      HALT
      8$:     MOV     #-1,R4 ; SUB INSTRUCTION FAILED
      SUB     R3,R4
      JSR     PC,#$CC4 ; CHECK FOR CC = 4

```

```

:*****
:*TEST: 53 NEW INSTRUCTIONS IN THIS SECTION IS MTPS & MFPS
:*****

```

```

PSW:
      1$:     CMP     (R5),#53
      BNE     EPSW ; IF IN WRONG SEQUENCE THEN GO TO HLT AT THE END OF THE
      INC     (R5)
      MOV     #177777,R1
      CLR     R0
      MTPS   R0 ; SET PSW TO 0
      .WORD 106400...C
      JSR     PC,#$CC0 ; CHECK FOR CC = 0
      MFPS   R1 ; MOVE PSW TO R1
      .WORD 106700...

```

NEW INSTRUCTIONS IN THARE SECTION IS MTPS & MFPS

SEQ 0053

981	007066	001404	
982	007070	012745	000203
983	007074	005245	
984	007076	000000	
985	007100	004737	017150
986	007104	012700	000377
987	007110		
988	007110	106400	
989	007112	004737	017340
990	007116		
991	007116	106701	
992	007120	004737	017256
993	007124	022701	177757
994	007130	001404	
995	007132		
996	007132	012745	000004
997	007136	005245	
998			

2\$:

EPC:

```

BEQ 2$
MOV #203,-(R5)
INC -(R5)
HALT
JSR PC,@$CC4
MOV #377,R0
MTPS R0
.WORD 106400!..C
JSR PC,@$CC17
MFPS R1
.WORD 106700!..C
JSR PC,@$CC11
CMP #177757,R1
BEQ MODE0
MOV #204,-(R5)
INC -(R5)
HALT

```

```

: CONTINUE IF BIT 8 OF PSW WAS EXTENDED IN R1
: MTPS OR MFPS INSTRUCTION FAILED
: CHECK FOR CC = 4
: SET PSW TO 357 SINCE MTPS DOES NOT SET T BIT
: CHECK FOR CC = 17
: MOVE PSW TO R1
: CHECK FOR CC = 11 [C BIT SHOULD NOT BE EFFECTED BY MFP
: CHECK TO SEE IF BIT 8 OF PSW WAS EXTENDED THRU R1
: MTPS OR MFPS INSTRUCTION FAILED OR WRONG SEQUENCE

```



2055 ;\*TEST: 55 CHECK MODE 2 USING THE MOV B AND MOV INSTRUCTIONS  
2056 ;:\*\*\*\*\*  
2057 ;:

2058 007326  
2059 007326 021527 000055  
2060 007332 001050  
2061 007334 005215  
2062 007336 012700 000136  
2063 007342 012701 000140  
2064 007346 012702 000142  
2065 007352 105022  
2066 007354 112710 000252  
2067 007360 112021  
2068 007362 105201  
2069 007364 111167 170546  
2070 007370 105200  
2071 007372 112021  
2072 007374 124227 000252  
2073 007400 001003  
2074 007402 105767 170530  
2075 007406 001404  
2076 007410  
2077 007410 012745 000211  
2078 007414 005245  
2079 007416 000000  
2080  
2081 007420 005741  
2082 007422 005022  
2083 007424 012740 125252  
2084 007430 012020  
2085 007432 011067 170500  
2086 007436 012121  
2087 007440 024227 125252  
2088 007444 001003  
2089 007446 005767 170464  
2090 007452 001404  
2091 007454  
2092 007454 012745 000212  
2093 007460 005245  
2094 007462 000000  
2095

MODE2:

CMP (R5),#55  
BNE EMODE2 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
MOV #TEMP,R0 ; LOAD ADDRESSES  
MOV #TEMP1,R1  
MOV #TEMP2,R2  
CLRB (R2)+ ; START CLEAN  
MOVB #252,(R0) ; LOAD THE LOCATIONS  
MOVB (R0)+,(R1)+ ; TEMP ----> TEMP1  
INCB R1 ; MAKE IT EVEN  
MOVB (R1),TEMP ; MORE 0'S INTO TEMP  
INCB R0 ; MAKE IT EVEN  
MOVB (R0)+,(R1)+ ; TEMP1 ----> TEMP2  
CMPB -(R2),#252 ; CHECK IT  
BNE 1\$ ; FAILED  
TSTB TEMP ; CHECK IT  
BEQ 2\$ ; OK, CONTINUE

1\$:

MOV #211,-(R5)  
INC -(R5)  
HALT ; INSTRUCTIONS FAILED IN MODE 2

2\$:

TST -(R1)  
CLR (R2)+ ; START CLEAN  
MOV #125252,-(R0) ; LOAD LOCATIONS  
MOV (R0)+,(R0)+ ; TEMP ----> TEMP1  
MOV (R0),TEMP ; 0 ----> TEMP  
MOV (R1)+,(R1)+ ; 125252 ----> TEMP2  
CMPB -(R2),#125252 ; CHECK IT  
BNE EMODE2 ; FAILED  
TSTB TEMP ; CHECK IT  
BEQ MODE3 ; OK, CONTINUE

EMODE2:

MOV #212,-(R5)  
INC -(R5)  
HALT ; INSTRUCTIONS FAILED IN MODE 2  
; OR WRONG SEQUENCE

2096  
2097  
2098  
2099  
:\*\*\*\*\*  
:\*TEST: 56 CHECK MODE 3 USING THE MOVB AND MOV INSTRUCTIONS  
:\*\*\*\*\*

```

MODE3:
2100 007464          021527 000056      CMP      (R5),#56
2101 007464 021527 000056      BNE     EMODE3          ; IF IN WRONG SEQUENCE GO TO HLT ABOVE
2102 007470 001066          INC     (R5)
2103 007472 005215          MOV     #TEMP,ADR      ; LOAD ADDRESSES
2104 007474 012767 000136 170424  MOV     #TEMP1,ADR1
2105 007502 012767 000140 170420  MOV     #TEMP2,ADR2
2106 007510 012767 000142 170414  MOV     #ADR,R0
2107 007516 012700 000126          MOV     #ADR1,R1      ; LOAD ADDRESSES OF ADDRESSES
2108 007522 012701 000130          CLR    TEMP2
2109 007526 105067 170410          CLRB   TEMP2          ; START CLEAN
2110 007532 112767 000125 170376  MOVB   #125,TEMP
2111 007540 113031          MOVB   @ (R0)+,@ (R1)+ ; TEMP ----> TEMP1
2112 007542 113167 170370          MOVB   @ (R1)+,TEMP    ; TEMP2 ----> TEMP
2113 007546 113030          MOVB   @ (R0)+,@ (R0)+ ; TEMP1 ----> TEMP2
2114 007550 122767 000125 170364  CMPB   #125,TEMP2
2115 007556 001003          BNE    1$              ; CHECK IT
2116 007560 105767 170352          TSTB  TEMP            ; FAILED
2117 007564 001404          BEQ    2$              ; CHECK IT
2118 007566          BEQ    2$              ; OK, CONTINUE
2119 007566 012745 000213          MOV     #213,-(R5)
2120 007572 005245          INC    -(R5)
2121 007574 000000          HALT
2122 007576 005067 170340          CLR    TEMP2          ; INSTRUCTIONS FAILED IN MODE 3
2123 007602 012767 052525 170326  MOV     #52525,TEMP    ; START CLEAN
2124 007610 012700 000126          MOV     #ADR,R0        ; LOAD LOCATIONS
2125 007614 012701 000130          MOV     #ADR1,R1       ; LOAD ADDRESSES OF ADDRESSES
2126 007620 013030          MOV     @ (R0)+,@ (R0)+ ; TEMP ----> TEMP1
2127 007622 013067 170310          MOV     @ (R0)+,TEMP    ; TEMP2 ----> TEMP
2128 007626 013131          MOV     @ (R1)+,@ (R1)+ ; TEMP1 ----> TEMP2
2129 007630 022767 052525 170304  CMP     #52525,TEMP2
2130 007636 001003          BNE    EMODE3          ; CHECK IT
2131 007640 005767 170272          TST    TEMP            ; FAILED
2132 007644 001404          BEQ    MODE4           ; CHECK IT
2133 007646          BEQ    MODE4           ; OK, CONTINUE
EMODE3:
2134 007646 012745 000214          MOV     #214,-(R5)
2135 007652 005245          INC    -(R5)
2136 007654 000000          HALT                    ; INSTRUCTIONS FAILED IN MODE 3

```

2137  
2138  
2139  
2140  
2141  
2142  
2143  
:\*\*\*\*\*  
:\*TEST: 57 CHECK MODE 4 USING THE MOVB AND MOV INSTRUCTIONS  
:\*\*\*\*\*

```

MODE4:
2144 007656          021527 000057      CMP      (R5),#57
2145 007656 021527 000057      BNE     EMODE4          ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2146 007662 001120          INC     (R5)
2147 007664 005215          MOV     #TEMP,R0
2148 007666 105067 170244          CLRB   TEMP            ; START CLEAN
2149 007672 012700 000136          MOV     #TEMP,R0       ; LOAD ADDRESSES
2150 007676 012701 000140          MOV     #TEMP1,R1
2151 007702 012702 000142          MOV     #TEMP2,R2

```



```

2152 007706 005202      INC      R2      ; ADJUST THE POINTER
2153 007710 021267 170227  CMP      (R2),TEMP2+1
2154 007714 001404      BEQ      1$
2155 007716 012745 000215  MOV      #215,-(R5)
2156 007722 005245      INC      -(R5)
2157 007724 000000      HALT
2158 007726 112742 000252  1$:  MOVB     #252,-(R2) ; INSTRUCTIONS FAILED IN MODE 4
2159 007732 005201      INC      R1      ; LOAD TEMP2
2160 007734 005202      INC      R2      ; ADJUST THE POINTERS
2161 007736 114241      MOVB     -(R2),-(R1) ; TEMP2 ----> TEMP1
2162 007740 005200      INC      R0      ; ADJUST THE POINTERS
2163 007742 005202      INC      R2
2164 007744 114042      MOVB     -(R0),-(R2) ; TEMP ----> TEMP2
2165 007746 105200      INCB     R0      ; ADJUST THE POINTERS
2166 007750 021067 170163  CMP      (R0),TEMP+1
2167 007754 001404      BEQ      2$
2168 007756 012745 000216  MOV      #216,-(R5)
2169 007762 005245      INC      -(R5)
2170 007764 000000      HALT
2171 007766 105201      2$:  INCB     R1      ; INSTRUCTIONS FAILED IN MODE 4
2172 007770 114140      MOVB     -(R1),-(R0) ; TEMP1 ----> TEMP
2173 007772 122767 000252 170136  CMPB     #252,TEMP ; CHECK IT
2174 010000 001003      BNE      3$      ; FAILED
2175 010002 105767 170134  TSTB     TEMP2    ; CHECK IT
2176 010006 001404      BEQ      4$      ; OK, CONTINUE
2177 010010
2178 010010 012745 000217  3$:  MOV      #217,-(R5)
2179 010014 005245      INC      -(R5)
2180 010016 000000      HALT
2181 010020 005067 170112  4$:  CLR      TEMP    ; INSTRUCTIONS FAILED IN MODE 4
2182 010024 012700 000136  MOV      #TEMP,R0 ; START CLEAN
2183 010030 012701 000140  MOV      #TEMP1,R1 ; LOAD ADDRESSES
2184 010034 012702 000142  MOV      #TEMP2,R2
2185 010040 005722      TST      (R2)+   ;
2186 010042 021267 170076  CMP      (R2),TEMP2+2 ; ADJUST THE POINTER
2187 010046 001404      BEQ      5$
2188 010050 012745 000220  MOV      #220,-(R5)
2189 010054 005245      INC      -(R5)
2190 010056 000000      HALT
2191 010060 012742 125252  5$:  MOV      #125252,-(R2) ; INSTRUCTIONS FAILED IN MODE 4
2192 010064 005721      TST      (R1)+   ; LOAD TEMP2
2193 010066 005722      TST      (R2)+   ; ADJUST THE POINTERS
2194 010070 014241      MOV      -(R2),-(R1) ; TEMP2 ----> TEMP1
2195 010072 005720      TST      (R0)+   ; ADJUST POINTERS
2196 010074 005722      TST      (R2)+
2197 010076 014042      MOV      -(R0),-(R2) ; TEMP ----> TEMP2
2198 010100 005720      TST      (R0)+   ; ADJUST THE POINTERS
2199 010102 005721      TST      (R1)+
2200 010104 014140      MOV      -(R1),-(R0) ; TEMP1 ----> TEMP
2201 010106 022767 125252 170022  CMP      #125252,TEMP ; CHECK IT
2202 010114 001003      BNE      EMODE4 ; FAILED
2203 010116 005767 170020  TST      TEMP2    ; CHECK IT
2204 010122 001404      BEQ      MODE5   ; OK, CONTINUE
2205 010124
2206 010124 012745 000221  EMODE4: MOV      #221,-(R5)
2207 010130 005245      INC      -(R5)

```

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 59  
CVKAAC.P11 09-OCT-78 08:58 T57

G 5

CHECK MODE 4 USING THE MOVB AND MOV INSTRUCTIONS

SEQ 0058

2208 010132 000000  
2209

HALT

; INSTRUCTIONS FAILED IN MODE 4  
; OR WRONG SEQUENCE

```

2210
2211
2212
2213
2214 010134
2215 010134 021527 000u60
2216 010140 001105
2217 010142 005215
2218 010144 105067 167766
2219 010150 012767 000136 167750
2220 010156 012767 000140 167744
2221 010164 012767 000142 167740
2222 010172 012700 000126
2223 010176 012701 000130
2224 010202 012702 000132
2225 010206 005722
2226 010210 112752 000125
2227 010214 022122
2228 010216 115251
2229 010220 022022
2230 010222 115052
2231 010224 022022
2232 010226 125052
2233 010230 001404
2234 010232 012745 000222
2235 010236 005245
2236 010240 000000
2237 010242 022120
2238 010244 115150
2239 010246 122767 000125 167662
2240 010254 001003
2241 010256 105767 167660
2242 010262 001404
2243 010264
2244 010264 012745 000223
2245 010270 005245
2246 010272 000000
2247 010274 005067 167636
2248 010300 012700 000126
2249 010304 012701 000130
2250 010310 012702 000132
2251 010314 005722
2252 010316 012752 052525
2253 010322 022122
2254 010324 015251
2255 010326 022022
2256 010330 015052
2257 010332 022021
2258 010334 015150
2259 010336 022767 052525 167572
2260 010344 001003
2261 010346 005767 167570
2262 010352 001404
2263 010354
2264 010354 012745 000224
2265 010360 005245
    
```

MODE5:

```

CMP (R5),#60
BNE EMODE5 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
INC (R5)
CLRB TEMP ; START CLEAN
MOV #TEMP,ADR ; LOAD ADDRESSES
MOV #TEMP1,ADR1
MOV #TEMP2,ADR2
MOV #ADR,R0 ; LOAD ADDRESSES OF ADDRESSES
MOV #ADR1,R1
MOV #ADR2,R2
TST (R2)+ ; ADJUST THE POINTER
MOVB #125,@-(R2) ; LOAD TEMP2
CMP (R1)+,(R2)+ ; ADJUST THE POINTERS
MOVB @-(R2),@-(R1) ; TEMP2 ----> TEMP1
CMP (R0)+,(R2)+ ; ADJUST THE POINTERS
MOVB @-(R0),@-(R2) ; TEMP ----> TEMP2
CMP (R0)+,(R2)+ ; ADJUST THE POINTERS
CMPSB @-(R0),@-(R2) ; CHECK IT
BEQ 1$
MOV #222,-(R5)
INC -(R5)
1$:
CMP (R1)+,(R0)+ ; ADJUST THE POINTERS
MOVB @-(R1),@-(R0) ; TEMP1 ----> TEMP
CMPSB #125,TEMP ; CHECK IT
BNE 2$ ; FAILED
TSTB TEMP2 ; CHECK IT
BEQ 3$ ; OK, CONTINUE
2$:
MOV #223,-(R5)
INC -(R5)
3$:
CLR TEMP ; INSTRUCTIONS FAILED IN MODE 5
MOV #ADR,R0 ; START CLEAN
MOV #ADR1,R1 ; LOAD ADDRESSES OF ADDRESSES
MOV #ADR2,R2
TST (R2)+ ; ADJUST THE POINTER
MOVB #52525,@-(R2) ; LOAD TEMP2
CMP (R1)+,(R2)+ ; ADJUST THE POINTERS
MOVB @-(R2),@-(R1) ; TEMP2 ----> TEMP1
CMP (R0)+,(R2)+ ; ADJUST THE POINTERS
MOVB @-(R0),@-(R2) ; TEMP ----> TEMP2
CMP (R0)+,(R1)+ ; ADJUST THE POINTERS
MOVB @-(R1),@-(R0) ; TEMP1 ----> TEMP
CMP #52525,TEMP ; CHECK IT
BNE EMODE5 ; FAILED
TST TEMP2 ; CHECK IT
BEQ MODE6 ; OK, CONTINUE
    
```

EMODE5:

```

MOV #224,-(R5)
INC -(R5)
    
```

2266 010362 00CC00 HALT ; INSTRUCTIONS FAILED IN MODE 5  
 2267 ; OR WRONG SEQUENCE  
 2268  
 2269  
 2270

2271 :\*\*\*\*\*  
 2272 :\*TEST: 61 CHECK MODE 6 USING THE MOVB AND MOV INSTRUCTIONS  
 2273 :\*\*\*\*\*

2274 010364 MODE6:  
 2275 010364 021527 200067 CMP (R5),#61  
 2276 010370 0C1055 BNE EMODE6 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
 2277 010372 005215 INC (R5)  
 2278 010374 005067 167542 CLR TEMP2 ; START CLEAN  
 2279 010400 012700 000136 MOV #TEMP,R0 ; LOAD ADDRESSES  
 2280 010404 012701 000140 MOV #TEMP1,R1  
 2281 010410 012702 000142 MOV #TEMP2,R2  
 2282 010414 112760 000252 000000 MOVB #252,0(R0) ; LOAD TEMP (LOW BYTE)  
 2283 010422 112760 000252 000001 MOVB #252,1(R0) ; LOAD TEMP (HIGH BYTE)  
 2284 010430 022767 125252 167500 CMP #125252,TEMP ; CHECK IT  
 2285 010436 001012 BNE 1\$ ; FAILED  
 2286 010440 116062 000001 000000 MOVB 1(R0),0(R2) ; TEMP(H) ----> TEMP2(L)  
 2287 010446 116160 000002 000005 MOVB 2(R1),5(R0) ; TEMP2(L) ----> TEMP2(H)  
 2288 010454 022767 125252 167460 CMP #125252,TEMP2 ; CHECK IT  
 2289 010462 001404 BEQ 2\$ ; OK, CONTINUE  
 2290 010464 1\$:  
 2291 010464 012745 000225 MOV #225,-(R5)  
 2292 010470 005245 INC -(R5)  
 2293 010472 000000 HALT ; INSTRUCTIONS FAILED IN MODE 6  
 2294 010474 005067 167440 2\$: CLR TEMP1 ; START CLEAN  
 2295 010500 012760 052525 000000 MOV #52525,0(R0) ; LOAD TEMP  
 2296 010506 016260 177774 000002 MOV -4(R2),2(R0) ; TEMP ----> TEMP1  
 2297 010514 022767 052525 167416 CMP #52525,TEMP1 ; CHECK IT  
 2298 010522 001404 BEQ MODE7 ; OK, CONTINUE  
 2299 010524 E:MODE6:  
 2300 010524 012745 000226 MOV #226,-(R5)  
 2301 010530 005245 INC -(R5)  
 2302 010532 000000 HALT ; INSTRUCTIONS FAILED IN MODE 6  
 2303 ; OR WRONG SEQUENCE

```

2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
    
```

MODE 7:

```

    CMP      (R5),#62
    INC      EMODE7
    INC      (R5)          ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
    CLR      TEMP1
    MOV      #TEMP,ADR    ; START CLEAN
    MOV      #TEMP1,ADR1  ; LOAD ADDRESSES
    MOV      #TEMP2,ADR2
    MOV      #ADR,R0
    MOV      #ADR1,R1    ; LOAD ADDRESSES OF ADDRESSES
    MOV      #ADR2,R2
    MOV      #252,@0(R0) ; LOAD TEMP
    MOV      @-4(R2),@2(R0) ; TEMP ---> TEMP1
    CMP      #252,TEMP1  ; CHECK IT
    BEQ      1$          ; OK, CONTINUE
    INC      #227,-(R5)
    INC      -(R5)
    HALT
    1$: MOV      #125252,@0(R0) ; MODE 7 IS FAILING
    MOV      @-4(R2),@2(R0) ; LOAD TEMP
    CMP      #125252,TEMP1 ; TEMP ---> TEMP1
    BEQ      TSTB1      ; CHECK IT
    TSTB1 ; OK, CONTINUE

EMODE7:
    MOV      #230,-(R5)
    INC      -(R5)
    HALT
    ; INSTRUCTIONS FAILED IN MODE 7
    ; OR WRONG SEQUENCE
    
```

2335  
 2336  
 2337  
 2338  
 2339  
 2340  
 2341  
 2342  
 2343  
 2344  
 2345 010676  
 2346 010676 021527 000063  
 2347 010702 001042  
 2348 010704 005215  
 2349 010706 012700 000136  
 2350 010712 012701 000140  
 2351 010716 000277  
 2352 010720 105010  
 2353 010722 004737 017150  
 2354 010726 105710  
 2355 010730 004737 017150  
 2356 010734 112711 000377  
 2357 010740 004737 017236  
 2358 010744 105711  
 2359 010746 004737 017236  
 2360 010752 010002  
 2361 010754 112762 000200 000000  
 2362 010762 112241  
 2363 010764 026127 177777 100200  
 2364 010772 001404  
 2365 010774 012745 000231  
 2366 011000 005245  
 2367 011002 000000  
 2368 011004 020102  
 2369 011006 001404  
 2370 011010  
 2371 011010 012745 000232  
 2372 011014 005245  
 2373 011016 000000  
 2374  
 2375  
 2376

: CHECK BYTE INSTRUCTIONS, NOT DESTINATION MODE 0  
 : -----  
 : \*\*\*\*\*  
 : \*TEST: 63 NEW INSTRUCTIONS USED IN THIS SECTION ARE TSTB, CLR B, MOV B  
 : \*\*\*\*\*

TSTB1:  
 CMP (R5),#63  
 BNE ETSTB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF TEST  
 2\$: INC (R5) ;  
 MOV #TEMP,R0 ; LOAD ADDRESSES  
 MOV #TEMP1,R1 ;  
 SCC ;  
 CLR B (R0) ; CLEAR THE LOCATION  
 JSR PC,@#SCC4 ; CHECK FOR CC = 4  
 TSTB (R0) ; CHECK IT  
 JSR PC,@#SCC4 ; CHECK FOR CC = 4  
 MOV B #377,(R1) ; LOAD THE LOCATION  
 JSR PC,@#SCC10 ; CHECK FOR CC = 10  
 TSTB (R1) ; CHECK IT  
 JSR PC,@#SCC10 ; CHECK FOR CC = 10  
 MOV R0,R2 ; R2 IS NOW POINTING TO LOCATION TEMP  
 MOV B #200,0(R2) ; PLACE #200 IN LOCATION TEMP  
 MOV B (R2)+,-(R1) ; MOVE #200 TO LOCATION TEMP+1  
 CMP -1(R1),#100200 ; CHECK THE DATA IN LOCATION TEMP  
 BEQ 4\$  
 MOV #231,-(R5)  
 INC -(R5)  
 HALT ; MOV B INSTRUCTION FAILED  
 4\$: CMP R1,R2 ; CHECK THE REGISTERS FOR PROPER VALUE  
 BEQ CMPB1  
 ETSTB1:  
 MOV #232,-(R5)  
 INC -(R5)  
 HALT ; MOV B INSTRUCTION FAILED OR WRONG SEQUENCE

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB

SEQ 0063

2377  
2378  
2379  
2380  
2381 011020  
2382 011020 021527 000064  
2383 011024 001032  
2384 011026 005215  
2385 011030 012701 000142  
2386 011034 012702 000136  
2387 011040 012711 000077  
2388 011044 112704 000377  
2389 011050 150412  
2390 011052 004737 017236  
2391 011056 120412  
2392 011060 001404  
2393 011062 012745 000233  
2394 011066 005245  
2395 011070 000000  
2396 011072 121112  
2397 011074 100004  
2398 011076 012745 000234  
2399 011102 005245  
2400 011104 000000  
2401 011106 121211  
2402 011110 100404  
2403 011112  
2404 011112 012745 000235  
2405 011116 005245  
2406 011120 000000  
2407  
2408  
2409  
2410

\*\*\*\*\*  
\*TEST: 64 NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB  
\*\*\*\*\*

CMPB1:  
1\$: CMP (R5),#64  
BNE ECMPB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
MOV #TEMP2,R1  
MOV #TEMP,R2 ; LOAD ADDRESS  
MOV #77,(R1) ; PLACE 77 IN LOCATION TEMP2  
MOVB #377,R4 ; R4 SHOULD CONTAIN #177777  
BISB R4,(R2) ; LOAD LOCATION  
JSR PC,@#SCC10 ; CHECK FOR CC = 10  
CMPB R4,(R2) ; CHECK COMPARE  
BEQ 2\$ ; CONTINUE IF OK  
MOV #233,-(R5)  
INC -(R5)  
2\$: HALT ; BISB OR CMPB INSTRUCTION FAILED  
CMPB (R1),(R2) ; CHECK IT AGAIN  
BPL 3\$ ; CONTINUE IF OK  
MOV #234,-(R5)  
INC -(R5)  
3\$: HALT ; CMPB INSTRUCTION FAILED [WRONG CC]  
CMPB (R2),(R1) ; ONCE MORE  
BMI BICB1 ; CONTINUE IF OK  
ECMPB1:  
MOV #235,-(R5)  
INC -(R5)  
HALT

2411  
2412  
2413  
2414 011122  
2415 011122 021527 000065  
2416 011126 001404  
2417 011130 012745 000236  
2418 011134 005245  
2419 011136 000000  
2420 011140 005215  
2421 011142 012703 000136  
2422 011146 112713 000377  
2423 011152 012700 000140  
2424 011156 010001  
2425 011160 112721 000252  
2426 011164 000277  
2427 011166 146013 000000  
2428 011172 004737 017066  
2429 011176 136113 177777  
2430 011202 001404  
2431 011204 012745 000237  
2432 011210 005245

\*\*\*\*\*  
\*TEST: 65 NEW INSTRUCTIONS USED IN THIS SECTION ARE BICB, BITB  
\*\*\*\*\*

BICB1:  
2\$: CMP (R5),#65  
BEQ 2\$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
MOV #236,-(R5)  
INC -(R5)  
2\$: HALT ; PROGRAM IS IN WRONG SEQUENCE  
INC (R5)  
MOV #TEMP,R3 ; LOAD ADDRESS  
MOVB #377,(R3) ; LOAD LOCATION  
MOV #TEMP1,R0 ; PLACE THE ADDRESS OF LOCATION TEMP1 IN R0  
MOV R0,R1 ; AND R1  
MOVB #252,(R1)+ ; PLACE #252 IN TEMP1  
SCC  
BICB 0(R0),(R3) ; CLEAR EVERY OTHER BIT  
JSR PC,@#SCC1 ; CHECK FOR CC = 1  
BITB -1(R1),(R3) ; CHECK IT  
BEQ 4\$ ; CONTINUE IF OK  
MOV #237,-(R5)  
INC -(R5)

2433	011212	000000		HALT						
2434	011214	132713	000125	4\$: BITB	#125,(R3)				: BICB OR BITB INSTRUCTION FAILED	
2435	011220	004737	017066	JSR	PC,@\$CC1				: CHECK IT	
2436	011224	154113		BISB	-(R1),(R3)				: CHECK FOR CC = 1	
2437	011226	100404		BMI	6\$				: SET THE BITS THAT WERE CLEARED	
2438	011230	012745	000240	MOV	#240,-(R5)				: CONTINUE IF OK	
2439	011234	005245		INC	-(R5)					
2440	011236	000000		HALT					: BITB OR BISB INSTRUCTION FAILED	
2441	011240	012746	000177	6\$: MOV	#177,-(SP)				: STORE #177 ON THE STACK	
2442	011244	142613		BICB	(SP)+,(R3)				: CLEAR ALL THE BITS EXCEPT SIGN BIT	
2443	011246	004737	017256	JSR	PC,@\$CC11				: CHECK FOR CC = 11	
2444	011252	132713	000377	BITB	#377,(R3)				: CHECK IT	
2445	011256	004737	017256	JSR	PC,@\$CC11				: CHECK FOR CC = 11	
2446	011262	010300		MOV	R3,R0				: PLACE THE ADDRESS OF LOCATION TEMP IN R0	
2447	011264	012710	000140	MOV	#TEMP1,(R0)				: PLACE THE ADDRESS OF LOCATION TEMP1 IN TEMP	
2448	011270	012730	000377	MOV	#377,@(R0)+				: WRITE A 377 IN LOCATION TEMP1	
2449	011274	000263		SEVC					: SET V & C BITS	
2450	011276	145070	000000	BICB	@-(R0),@(R0)				: BIT CLEAR THE CONTENTS	
2451									: OF TEMP1 TO THE CONTENTS OF TEMP1	
2452	011302	004737	017172	JSR	PC,@\$CC5				: CHECK FOR CC = 5	
2453	011306	022027	000140	CMP	(R0)+,#TEMP1				: MAKE SURE THAT (R0) IS POINTING TO LOCATION TEMP1	
2454	011312	001404		BEQ	8\$					
2455	011314	012745	000241	MOV	#241,-(R5)					
2456	011320	005245		INC	-(R5)					
2457	011322	000000		HALT					: BICB OR CMP INSTRUCTION FAILED IN THE SPECIFIC MODE	
2458	011324	005750		8\$: TST	@-(R0)				: TEST LOCATION TEMP1	
2459	011326	001404		BEQ	10\$					
2460	011330	012745	000242	MOV	#242,-(R5)					
2461	011334	005245		INC	-(R5)					
2462	011336	000000		HALT					: BICB INSTRUCTION FAILED	
2463	011340	000257		10\$: CCC						
2464	011342	141010		BICB	(R0),(R0)				: CLEAR THE LOCATION TEMP	
2465	011344	004737	017150	JSR	PC,@\$CC4				: CHECK FOR CC = 4	



```
2466 :*****  
2467 :*TEST: 66 NEW INSTRUCTIONS USED IN THIS SECTION ARE INCB, DECB  
2468 :*****  
2469  
2470 011350 INCB1:  
2471 011350 021527 000066 CMP (R5),#66  
2472 011354 001067 BNE EINCB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
2473 011356 005215 INC (R5)  
2474 011360 012704 000136 MOV #TEMP,R4 ; LOAD ADDRESS  
2475 011364 112714 000177 MOVB #177,(R4) ; TEMP LOCATION=177  
2476 011370 000261 SEC  
2477 011372 105214 INCB (R4) ; ADD ONES INTO LOCATION  
2478 011374 004737 017320 JSR PC,@#SCC13 ; CHECK FOR CC = 13  
2479 011400 012714 000376 MOV #376,(R4)  
2480 011404 012700 017256 MOV #SCC11,R0 ; MAKE R0 POINT TO CHECKING ROUTINE FOR CC = 11  
2481 011410 105224 INCB (R4)+  
2482 011412 004720 JSR PC,(R0)+ ; CHECK FOR CC = 11  
2483 011414 105744 TSTB -(R4) ; DECREMENT R4 BY 1  
2484 011416 005746 TST -(SP) ; AND SP BY 2  
2485 011420 010426 MOV R4,(SP)+ ; PLACE THE ADDRESS OF TEMP ON THE STACK  
2486 011422 000241 CLC ; CLEAR C BIT  
2487 011424 105256 INCB @-(SP) ; INCREMENT THE CONTENTS OF LOCATION TEMP  
2488 011426 004737 017150 JSR PC,@#SCC4 ; CHECK FOR CC = 4  
2489 011432 123634 CMPB @-(SP)+,@(R4)+ ; RESTORE STACK POINTER  
2490 011434 000261 SEC ; SET C BIT  
2491 011436 105264 177777 INCB -1(R4)  
2492 011442 004737 017066 JSR PC,@#SCC1 ; CHECK FOR CC = 1  
2493 011446 124427 000001 CMPB -(R4),#1 ; CHECK IT  
2494 011452 001404 BEQ 2$ ; CONTINUE IF OK  
2495 011454 012745 000243 MOV #243,-(R5)  
2496 011460 005245 INC -(R5)  
2497 011462 000000 HALT ; INCB INSTRUCTION FAILED  
2498 011464 000261 2$: SEC  
2499 011466 105314 DECB (R4) ; SUBTRACT ONES FROM LOCATION  
2500 011470 004737 017172 JSR PC,@#SCC5 ; CHECK FOR CC = 5  
2501 011474 105324 DECB (R4)+  
2502 011476 004740 JSR PC,-(R0) ; CHECK FOR CC = 11  
2503 011500 112764 000200 177777 MOVB #200,-1(R4)  
2504 011506 105344 DECB -(R4)  
2505 011510 004760 177650 JSR PC,#SCC3-SCC11(R0) ; CHECK FOR CC = 3  
2506 011514 105364 000000 DECB 0(R4)  
2507 011520 004737 017066 JSR PC,@#SCC1 ; CHECK FOR CC = 1  
2508 011524 126427 000000 000176 CMPB 0(R4),#176  
2509 011532 001404 BEQ COMB1  
2510 011534 EINCB1:  
2511 011534 012745 000244 MOV #244,-(R5)  
2512 011540 005245 INC -(R5)  
2513 011542 000000 HALT ; DECB INSTRUCTION FAILED OR SEQUENCE ERROR  
2514  
2515  
2516  
2517  
2518  
2519  
2520
```

2522  
2523  
2524  
2525  
2526  
2527 011544  
2528 011544 021527 000067  
2529 011550 001404  
2530 011552 012745 000245  
2531 011556 005245  
2532 011560 000000  
2533 011562 005215  
2534 011564 012703 000136  
2535 011570 012704 000140  
2536 011574 012714 000252  
2537 011600 112413  
2538 011602 000277  
2539 011604 105113  
2540 011606 004737 017066  
2541 011612 122713 000125  
2542 011616 001404  
2543 011620 012745 000246  
2544 011624 005245  
2545 011626 000000  
2546 011630 000277  
2547 011632 105113  
2548 011634 004737 017256  
2549 011640 010400  
2550 011642 126013 177777  
2551 011646 001404  
2552 011650 012745 000247  
2553 011654 005245  
2554 011656 000000  
2555 011660 112724 000377  
2556 011664 114413  
2557 011666 000277  
2558 011670 105113  
2559 011672 004737 017172

\*\*\*\*\*  
: \*TEST: 67 NEW INSTRUCTION IN THIS SECTION IS COMB  
: \*\*\*\*\*

COMB1:  
CMP (R5),#67  
BEQ 1\$ ; IF IN WRONG SEQUENC GO TO HLT  
MOV #245,-(R5)  
INC -(R5)  
HALT ; TEST IS IN WRONG SEQUENCE  
1\$: INC (R5)  
MOV #TEMP,R3 ; LOAD ADDRESS  
MOV #TEMP1,R4  
MOV #252,(R4)  
MOVB (R4)+,(R3) ; LOAD EVERY OTHER BIT  
SCC  
COMB (R3) ; 1'S COMPLEMENT  
JSR PC,@#SCC1 ; CHECK FOR C = 1  
CMPB #125,(R3) ; CHECK IT  
BEQ 2\$ ; CONTINUE IF OK  
MOV #246,-(R5)  
INC -(R5)  
HALT ; COMB INSTRUCTION FAILED  
2\$: SCC  
COMB (R3) ; COMPLEMENT BACK  
JSR PC,@#SCC11 ; CHECK FOR CC 11  
MOV R4,R0  
CMPB -1(R0),(R3) ; CHECK IT  
BEQ 3\$ ; CONTINUE IF OK  
MOV #247,-(R5)  
INC -(R5)  
HALT ; COMB INSTRUCTION FAILED  
3\$: MOVB #377,(R4)+  
MOVB -(R4),(R3) ; PLACE #377 IN (R3)  
SCC  
COMB (R3)  
JSR PC,@#SCC5 ; CHECK FOR CC - 5

T70 NEW INSTRUCTION IN THIS SECTION IS NEGB

SEQ 0067

2560  
2561  
2562  
2563  
2564 011676  
2565 011676 021527 00007C  
2566 011702 001027  
2567 011704 005215  
2568 011706 012700 000136  
2569 011712 112710 000001  
2570 011716 105410  
2571 011720 004737 017256  
2572 011724 122710 000377  
2573 011730 001404  
2574 011732 012745 000250  
2575 011736 005245  
2576 011740 000000  
2577 011742 012710 000200  
2578 011746 105410  
2579 011750 004737 017320  
2580 011754 122710 000200  
2581 011760 001404  
2582 011762  
2583 011762 012745 000251  
2584 011766 005245  
2585 011770 000000  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593 011772  
2594 011772 021527 000071  
2595 011776 001030  
2596 012000 005215  
2597 012002 012701 000140  
2598 012006 112711 000040  
2599 012012 000257  
2600 012014 106111  
2601 012016 106111  
2602 012020 004737 017300  
2603 012024 122711 000200  
2604 012030 001404  
2605 012032 012745 000252  
2606 012036 005245  
2607 012040 000000  
2608 012042 106111  
2609 012044 004737 017274  
2610 012050 106111  
2611 012052 122711 000001  
2612 012056 001404  
2613 012060  
2614 012060 012745 000253  
2615 012064 005245

\*\*\*\*\*  
: \*TEST: 70 NEW INSTRUCTION IN THIS SECTION IS NEGB  
: \*\*\*\*\*

NEGB1:  
CMP (R5),#70  
BNE ENEGB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
1\$: INC (R5)  
MOV #TEMP,R0 ; LOAD ADDRESS  
MOVB #1,(R0) ; LOAD THE LOCATION  
NEGB (R0) ; 2'S COMPLEMENT  
JSR PC,@#SCC11 ; CHECK FOR CC - 11  
CMPB #377,(R0) ; CHECK IT  
BEQ 2\$ ; CONTINUE IF OK  
MOV #250,-(R5)  
INC -(R5)  
HALT ; NEGB INSTRUCTION FAILED  
2\$: MOV #200,(R0)  
NEGB (R0) ; 2'S COMPLEMENT  
JSR PC,@#SCC13 ; CHECK FOR CC - 13  
CMPB #200,(R0) ; CHECK IT  
BEQ ROLB1 ; CONTINUE IF OK  
ENEGB1:  
MOV #251,-(R5)  
INC -(R5)  
HALT ; WRONG RESULT AT TEMP OR WRONG SEQUENCE

\*\*\*\*\*  
: \*TEST: 71 NEW INSTRUCTION IN THIS SECTION IS ROLB  
: \*\*\*\*\*

ROLB1:  
CMP (R5),#71  
BNE EROLB1 ; IF IN WRONG SEQUENCE GO TO HLT ABOVE  
INC (R5)  
MOV #TEMP1,R1 ; LOAD ADDRESS  
MOVB #40,(R1) ; LOAD LOCATION  
CCC ; CLEAR FLAGS  
ROIB (R1) ; SHIFT  
ROLB (R1)  
JSR PC,@#SCC12 ; CHECK FOR CC - 12  
CMPB #200,(R1) ; CHECK IT  
BEQ 1\$ ; CONTINUE IF OK  
MOV #252,-(R5)  
INC -(R5)  
HALT ; ROLB INSTRUCTION FAILED  
1\$: ROLB (R1) ; SHIFT  
JSR PC,@#SCC7 ; CHECK FOR CC = 7  
ROLB (R1) ; SHIFT  
CMPB #1,(R1) ; CHECK IT  
BEQ RORB1 ; CONTINUE IF OK  
EROLB1:  
MOV #253,-(R5)  
INC -(R5)

CVKAAC MACV11 30A(1052) 09-OCT-78 08:59 PAGE 69  
CVKAAC.P11 09-OCT-78 08:58 T71

NEW INSTRUCTION IN THIS SECTION IS ROLB

SEQ 0068

2616 012066 000000

HALT

: WRONG RESULT AT TEMP1 OR WRONG SEQUENCE

```

2617
2618
2619
2620
2621 012070
2622 012070 021527 000072
2623 012074 001030
2624 012076 005215
2625 012100 012702 000140
2626 012104 112712 000004
2627 012110 000257
2628 012112 106012
2629 012114 106012
2630 012116 122712 000001
2631 012122 001404
2632 012124 012745 00254
2633 012130 005245
2634 012132 000000
2635 012134 106012
2636 012136 004737 017214
2637 012142 106012
2638 012144 004737 017300
2639 012150 122712 000200
2640 012154 001404
2641 012156
2642 012156 012745 000255
2643 012162 005245
2644 012164 000000

```

```

:*****
:*TEST: 72 NEW INSTRUCTION IN THIS SECTION IS RORB
:*****

```

```

RORB1:
  CMP      (R5),#72
  BNE     ERORB1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #TEMP1,R2   ; LOAD ADDRESS
  MOV     #4,(R2)     ; LOAD LOCATION
  CCC
  RORB    (R2)        ; CLEAR FLAGS
  RORB    (R2)        ; SHIFT
  CMP     #1,(R2)     ;
  BEQ     1$          ; CHECK IT
  MOV     #254,-(R5)  ; CONTINUE IF OK
  INC     -(R5)
  HALT
1$:
  RCRB    (R2)        ; RORB INSTRUCTION FAILED
  JSR     PC,#$CC7    ; SHIFT
  YORB    (R2)        ; CHECK FOR CC = 7
  JSR     PC,#$CC12   ; SHIFT
  CMP     #200,(R2)   ; CHECK FOR CC = 12
  BEQ     ASLB1       ; CHECK IT
  ERORB1:
  MOV     #255,-(R5)  ; CONTINUE IF OK
  INC     -(R5)
  HALT
  ; WRONG RESULT AT TEMP1 OR WRONG SEQUENCE

```

```

2645
2646
2647
2648
2649
2650
2651
2652 012166
2653 012166 021527 000073
2654 012172 001404
2655 012174 012745 000256
2656 012200 005245
2657 012202 000000
2658 012204 005215
2659 012206 012703 000140
2660 012212 112713 000040
2661 012216 000257
2662 012220 106313
2663 012222 106313
2664 012224 004737 017300
2665 012230 122713 000200
2666 012234 001404
2667 012236 012745 000257
2668 012242 005245
2669 012244 000000
2670 012246 106313
2671 012250 004737 017214
2672 012254 106313

```

```

:*****
:*TEST: 73 NEW INSTRUCTION IN THIS SECTION IS ASLB
:*****

```

```

ASLB1:
  CMP     (R5),#73
  BEQ     2$          ; IF IN WRONG SEQUENCE GO TO HLT BELOW
  MOV     #256,-(R5)
  INC     -(R5)
  HALT
2$:
  INC     (R5)        ; PROGRAM IS IN WRONG SEQUENCE
  MOV     #TEMP1,R3   ; LOAD ADDRESS
  MOV     #40,(R3)    ; LOAD LOCATION
  CCC
  ASLB    (R3)        ; CLEAR FLAGS
  ASLB    (R3)        ; SHIFT
  JSR     PC,#$CC12   ; CHECK FOR CC = 12
  CMP     #200,(R3)   ; CHECK IT
  BEQ     4$          ; CONTINUE IF OK
  MOV     #257,-(R5)
  INC     -(R5)
  HALT
4$:
  ASLB    (R3)        ; ASLB INSTRUCTION FAILED
  JSR     PC,#$CC7    ; SHIFT
  ASLB    (R3)        ; CHECK FOR CC = 7
  ASLB    (R3)        ; SHIFT

```

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 71  
CVKAAC.P11 09-OCT-78 08:58 T73

NEW INSTRUCTION IN THIS SECTION IS ASLB

SEQ 0070

2673 012256 004737 017150

JSR PC,@#SCC4 ; CHECK FOR CC = 4

2674  
 2675  
 2676  
 2677  
 2678 012262  
 2679 012262 021527 000074  
 2680 012266 001040  
 2681 012270 005215  
 2682 012272 012704 000140  
 2683 012276 012703 000142  
 2684 012302 112714 000004  
 2685 012306 000257  
 2686 012310 106214  
 2687 012312 106214  
 2688 012314 122714 000001  
 2689 012320 001404  
 2690 012322 012745 000260  
 2691 012326 005245  
 2692 012330 000000  
 2693 012332 106214  
 2694 012334 004737 017214  
 2695 012340 106214  
 2696 012342 004737 017150  
 2697 012346 112713 000202  
 2698 012352 106213  
 2699 012354 106213  
 2700 012356 004737 017256  
 2701 012362 122713 000340  
 2702 012366 001404  
 2703 012370  
 2704 012370 012745 000261  
 2705 012374 005245  
 2706 012376 000000

\*\*\*\*\*  
 : \*TEST: 74 NEW INSTRUCTION IN THIS SECTION IS ASRB  
 : \*\*\*\*\*

ASRB1:  
 CMP (R5),#74  
 BNE EASRB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
 1\$: INC (R5)  
 MOV #TEMP1,R4 ; LOAD ADDRESSES  
 MOV #TEMP2,R3 ;  
 MOVB #4,(R4) ; LOAD LOCATION  
 CCC ; CLEAR FLAGS  
 ASRB (R4) ; SHIFT  
 ASRB (R4) ;  
 CMPB #1,(R4) ; CHECK IT  
 BEQ 2\$ ; CONTINUE IF OK  
 MOV #260,-(R5)  
 INC -(R5)  
 2\$: HALT ; ASRB INSTRUCTION FAILED  
 ASRB (R4) ; SHIFT  
 JSR PC,@#SCC7 ; CHECK FOR CC - 7  
 ASRB (R4) ; SHIFT  
 JSR PC,@#SCC4 ; CHECK FOR CC - 4  
 MOVB #202,(R3) ; LOAD LOCATION  
 ASRB (R3) ; SHIFT  
 ASRB (R3) ;  
 JSR PC,@#SCC11 ; CHECK FOR CC - 11  
 CMPB #340,(R3) ; CHECK IT  
 BEQ ADCB1 ; CONTINUE IF OK  
 EASRB1:  
 MOV #261,-(R5)  
 INC -(R5)  
 HALT ; WRONG RESULT AT TEMP2 OR WRONG SEQUENCE

2707  
 2708  
 2709  
 2710  
 2711  
 2712  
 2713  
 2714 012400  
 2715 012400 021527 000075  
 2716 012404 001404  
 2717 012406 012745 000262  
 2718 012412 005245  
 2719 012414 000000  
 2720 012416 005215  
 2721 012420 012700 000142  
 2722 012424 105010  
 2723 012426 000257  
 2724 012430 105510  
 2725 012432 004737 017150  
 2726 012436 000261  
 2727 012440 105510  
 2728 012442 000261  
 2729 012444 105510

\*\*\*\*\*  
 : \*TEST: 75 NEW INSTRUCTION IN THIS SECTION IS ADCB  
 : \*\*\*\*\*

ADCB1:  
 CMP (R5),#75  
 BEQ 2\$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
 MOV #262,-(R5)  
 INC -(R5)  
 2\$: HALT ; PROGRAM IS IN WRONG SEQUENCE  
 INC (R5)  
 MOV #TEMP2,R0 ; LOAD ADDRESS  
 CLRB (R0) ; CLEAR THE LOCATION  
 CCC ; CLEAR FLAGS  
 ADCB (R0) ; ADD C BIT = 0  
 JSR PC,@#SCC4 ; CHECK FOR CC = 4  
 SEC ; C-1  
 ADCB (R0) ; ADD C BIT=1  
 SEC ; C-1  
 ADCB (R0) ; AGAIN

NEW INSTRUCTION IN THIS SECTION IS ADCB

SEQ 0072

2730	012446	004737	017046		JSR	PC,@#SCC0	:	CHECK FOR CC = 0
2731	012452	122710	000002		CMPB	#2,(R0)	:	CHECK IT
2732	012456	001404			BEQ	4\$	:	CONTINUE IF OK
2733	012460	012745	000263		MOV	#263,-(R5)		
2734	012464	005245			INC	-(R5)		
2735	012466	000000			HALT		:	ADCB INSTRUCTION FAILED
2736	012470	112710	000177	4\$:	MOVB	#177,(R0)	:	LOAD LARGEST POSITIVE BYTE
2737	012474	000261			SEC		:	C=1
2738	012476	105510			ADCB	(R0)	:	ADD C BIT=1
2739	012500	004737	017300		JSR	PC,@#SCC12	:	CHECK FOR CC = 12
2740	012504	122710	000200		CMPB	#200,(R0)	:	CHECK IT
2741	012510	001404			BEQ	6\$	:	CONTINUE IF OK
2742	012512	012745	000264		MOV	#264,-(R5)		
2743	012516	005245			INC	-(R5)		
2744	012520	000000			HALT		:	ADCB INSTRUCTION FAILED
2745	012522	112710	000377	6\$:	MOVB	#377,(R0)	:	_OAD -1
2746	012526	000261			SEC		:	C=1
2747	012530	105510			ADCB	(R0)	:	ADD C BIT=1
2748	012532	004737	017172		JSR	PC,@#SCC5	:	CHECK FOR CC = 5



```

2749
2750
2751
2752
2753 012536
2754 012536 021527 000076
2755 012542 001404
2756 012544 012745 000265
2757 012550 005245
2758 012552 000000
2759 012554 005215
2760 012556 012701 000142
2761 012562 112711 000003
2762 012566 000257
2763 012570 105611
2764 012572 004737 017046
2765 012576 122711 000003
2766 012602 001404
2767 012604 012745 000266
2768 012610 005245
2769 012612 000000
2770 012614 000261
2771 012616 105611
2772 012620 000261
2773 012622 105611
2774 012624 004737 017046
2775 012630 122711 000001
2776 012634 001404
2777 012636 012745 000267
2778 012642 005245
2779 012644 000000
2780 012646 000261
2781 012650 105611
2782 012652 004737 017150
2783 012656 000261
2784 012660 105611
2785 012662 004737 017256
2786 012666 122711 000377
2787 012672 001404
2788 012674 012745 000270
2789 012700 005245
2790 012702 000000
2791 012704 112711 000200
  
```

```

*****
*TEST: 76      NEW INSTRUCTION IN THIS SECTION IS SBCB
*****
SBCB1:
      CMP      (R5),#76
      BEQ      1$      ; IF IN WRONG SEQUENCE GO TO HLT
      MOV      #265,-(R5)
      INC      -(R5)
1$:   HALT
      INC      (R5)      ; TEST IS IN WRONG SEQUENCE
      MOV      #TEMP2,R1 ; LOAD ADDRESS
      MOVB     #3,(R1)   ; LOAD LOCATION
      LCC
      SBCB     (R1)      ; SUBTRACT C BIT=0
      JSR     PC,@#SCC0 ; CHECK FOR CC = 0
      CMPB    #3,(R1)   ; CHECK IT
      BEQ     2$      ; CONTINUE IF OK
      MOV     #266,-(R5)
      INC     -(R5)
2$:   HALT      ; SBCB INSTRUCTION FAILED
      SEC
      SBCB     (R1)      ; SUBTRACT C BIT=1
      SEC
      SBCB     (R1)      ; C-1
      JSR     PC,@#SCC0 ; CHECK FOR CC = 0
      CMPB    #1,(R1)   ; CHECK IT
      BEQ     3$      ; CONTINUE IF OK
      MOV     #267,-(R5)
      INC     -(R5)
3$:   HALT      ; SBCB INSTRUCTION FAILED
      SEC
      SBCB     (R1)      ; SUBTRACT C BIT=1
      JSR     PC,@#SCC4 ; CHECK FOR CC = 4
      SEC
      SBCB     (R1)      ; C=1
      JSR     PC,@#SCC11 ; SUBTRACT C BIT = 1
      CMPB    #377,(R1) ; CHECK FOR CC = 11
      BEQ     4$      ; CHECK IT
      MOV     #270,-(R5) ; CONTINUE IF OK
      INC     -(R5)
4$:   HALT      ; SBCB INSTRUCTION FAILED
      MOVB    #200,(R1) ; LOAD R1
  
```

NEW INSTRUCTION IN THIS SECTION IS SBCB

SEQ 0074

2792 012710 000261  
2793 012712 005611  
2794 012714 004737 017106

SEC  
SBCB (R1) : C=1  
JSR PC, @S0C2 : SUBTRACT C BIT = 1  
 : CHECK FOR CC = 2

-----  
 CHECK WORD INSTRUCTIONS, NOT DESTINATION MODE 0  
 -----

2795  
 2796  
 2797  
 2798  
 2799  
 2800  
 2801  
 2802  
 2803  
 2804  
 2805  
 2806  
 2807  
 2808 012720  
 2809 012720 021527 000077  
 2810 012724 001404  
 2811 012726 012745 000271  
 2812 012732 005245  
 2813 012734 000000  
 2814 012736 005215  
 2815 012740 012701 000136  
 2816 012744 012700 000140  
 2817 012750 000277  
 2818 012752 005010  
 2819 012754 004737 017150  
 2820 012760 005720  
 2821 012762 004737 017150  
 2822 012766 010040  
 2823 012770 012730 177777  
 2824 012774 017011 177776  
 2825 013000 004737 017236  
 2826 013004 005711  
 2827 013006 004737 017236  
 2828  
 2829  
 2830

\*\*\*\*\*  
 \*TEST: 77 NEW INSTRUCTIONS USED IN THIS SECTION ARE TST, CLR, MOV  
 \*\*\*\*\*

```

TST1:
      CMP      (R5),#77
      BEQ     1$          ; IF IN WRONG SEQUENCE GO TO HLT
      MOV     #271,-(R5)
      INC     -(R5)
      HALT
1$:   INC     (R5)          ; TEST IS IN A WRONG SEQUENCE
      MOV     #TEMP,R1    ; LOAD ADDRESSES
      MOV     #TEMP1,R0
      CLR     (R0)        ; CLEAR THE LOCATION
      JSR     PC,@#CC4    ; CHECK FOR CC = 4
      TST     (R0)+       ; CHECK IT
      JSR     PC,@#CC4    ; CHECK FOR CC = 4
      MOV     R0,-(R0)
      MOV     #177777,@(R0)+
      MOV     @-2(R0),(R1) ; LOAD THE LOCATION
      JSR     PC,@#CC10   ; CHECK FOR CC = 10
      TST     (R1)        ; CHECK IT
      JSR     PC,@#CC10   ; CHECK FOR CC = 10
  
```

```

2831
2832
2833
2834
2835 013012
2836 013012 021527 000100
2837 013016 001113
2838 013020 005215
2839 013022 012702 000140
2840 013026 012700 000136
2841 013032 012720 177777
2842 013036 054012
2843 013040 004737 017236
2844 013044 022227 177777
2845 013050 001404
2846 013052 012745 000272
2847 013056 005245
2848 013060 000000
2849 013062 020227 000142
2850 013066 001404
2851 013070 012745 000273
2852 013074 005245
2853 013076 000000
2854 013100 022742 000077
2855 013104 004737 017066
2856 013110 022722 077777
2857 013114 004737 017320
2858 013120 024227 077777
2859 013124 004737 017236
2860 013130 012767 052525 165004
2861 013136 012767 000142 164774
2862 013144 012704 000126
2863 013150 012714 000130
2864 013154 012734 125252
2865 013160 057432 177776
2866
2867 013164 010200
2868 013166 025027 177777
2869 013172 001404
2870 013174 012745 000274
2871 013200 005245
2872 013202 000000
2873 013204 020227 000142
2874
2875 013210 001404
2876 013212 012745 000275
2877 013216 005245
2878 013220 000000
2879 013222 005040
2880 013224 010067 164712
2881 013230 022020
2882 013232 055070 000002
2883 013236 022767 000136 164672
2884 013244 001404
2885 013246
2886 013246 012745 000276
    
```

\*\*\*\*\*  
 : \*TEST: 100 NEW INSTRUCTIONS USED IN THIS SECTION ARE CMP, BIS  
 : \*\*\*\*\*

```

CMP1:
    CMP      (R5),#100
    BNE     ECMP1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE EST
1$:
    INC     (R5)
    MOV     #TEMP1,R2  ; LOAD ADDRESS
    MOV     #TEMP,R0   ; PLACE THE ADDRESS OF TEMP IN R0
    MOV     #177777,(R0)+ ; PLACE #177777 IN LOCATION TEMP AND INC. R0 BY 2
    BIS     -(R0),(R2)  ; LOAD LOCATION
    JSR     PC,@#SCC10 ; CHECK FOR CC = 10
    CMP     (R2)+,#177777 ; CHECK COMPARE
    BEQ     2$         ; CONTINUE IF OK
    MOV     #272,-(R5)
    INC     -(R5)
    HALT
2$:
    CMP     R2,#TEMP1+2 ; CMP OR BIS INSTRUCTION FAILED
    BEQ     3$         ; CHECK R2 TO CONTAIN ADDRESS OF TEMP1+2
    MOV     #273,-(R5)
    INC     -(R5)
    HALT
3$:
    CMP     #77,-(R2)   ; NO AUTO INCREMENT
    JSR     PC,@#SCC1  ; CHECK IT AGAIN
    CMP     #77777,(R2)+ ; CHECK FOR CC = 1
    JSR     PC,@#SCC13 ; CHECK FOR CC = 13
    CMP     -(R2),#77777 ; ONCE MORE
    JSR     PC,@#SCC10 ; CHECK FOR CC = 10
    MOV     #52525,TEMP2 ; SET EVERY OTHER BIT IN TEMP2
    MOV     #TEMP2,TEMP1 ; PLACE THE ADDRESS OF TEMP2 IN LOCATION TEMP1
    MOV     #ADR,R4
    MOV     #ADR1,(R4)  ; PLACE THE ADDRESS OF ADR1 IN ADR POINTED BY R4
    MOV     #125252,@(R4)+ ; PLACE THE #125252 IN LOCATION ADR1
    BIS     @-2(R4),@(R2)+ ; SET EVERY OTHER BIT AT LOCATION TEMP2
    AND     AND INCREMENT R2 BY 2
    MOV     R2,R0       ; PLACE ADDRESS OF TEMP2 IN R0
    CMP     @-(R0),#177777 ; TEMP2 SHOULD CONTAIN ALL 1'S
    BEQ     4$
    MOV     #274,-(R5)
    INC     -(R5)
    HALT
4$:
    CMP     R2,#TEMP1+2 ; CMP OR BIS INSTUCTIONS FAILED IN MODES OTHER THAN 0
    BEQ     5$         ; R2 SHOULD CONTAIN THE ADDRESS FOR TEMP2
    MOV     #275,-(R5)
    INC     -(R5)
    HALT
5$:
    CLR     -(R0)       ; MODE 5 IS FAILING
    MOV     R0,TEMP2   ; PLACE A 0 IN LOCATION TEMP
    CMP     (R0)+,(R0)+ ; PLACE ADDRESS OF TEMP IN LOCATION TEMP2
    BUMP    R0 BY 4
    BIS     @-(R0),@2(R0) ; BUMP R0 BY 4
    CMP     #TEMP,TEMP ; PLACE THE CONTENTS OF LOCATION TEMP2 AT TEMP
    BEQ     BIC1       ; LOCATION TEMP SHOULD CONTAIN ITS OWN ADDRESS
    BIC1
ECMP1:
    MOV     #276,-(R5)
    
```

2887 013252 005245  
 2888 013254 000000  
 2889  
 2890  
 2891  
 2892  
 2893  
 2894  
 2895  
 2896  
 2897 013256  
 2898 013256 021527 000101  
 2899 013262 001122  
 2900 013264 005215  
 2901 013266 012703 000136  
 2902 013272 012713 177777  
 2903 013276 012704 000126  
 2904 013302 012714 000130  
 2905 013306 011334  
 2906 013310 012700 000140  
 2907 013314 012710 125252  
 2908 013320 000277  
 2909 013322 042013  
 2910 013324 004737 017066  
 2911 013330 034013  
 2912 013332 001404  
 2913 013334 012745 000277  
 2914 013340 005245  
 2915 013342 000000  
 2916 013344 032713 052525  
 2917 013350 004737 017066  
 2918 013354 056013 000000  
 2919 013360 100404  
 2920 013362 012745 000300  
 2921 013366 005245  
 2922 013370 000000  
 2923 013372 012720 077777  
 2924 013376 010002  
 2925 013400 046213 177776  
 2926 013404 004737 017256  
 2927 013410 020027 000142  
 2928 013414 001404  
 2929 013416 012745 000301  
 2930 013422 005245  
 2931 013424 000000  
 2932 013426 010020  
 2933 013430 000263  
 2934 013432 045000  
 2935 013434 004737 017177  
 2936 013440 037413 177776  
 2937 013444 004737 017256  
 2938 013450 012746 125252  
 2939 013454 017423 177776  
 2940 013460 046643 000000  
 2941 013464 022327 052525  
 2942 013470 001404

INC -(R5)  
 HALT ; CMP OR BIS INSTRUCTIONS FAILED OR WRONG ; SEQUENCE COUNTER

\*\*\*\*\*  
 ; \*TEST: 101 NEW INSTRUCTIONS USED IN THIS SECTION ARE BIC, BIT  
 ; \*\*\*\*\*

BIC1:  
 CMP (R5),#101  
 BNE EBIC1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
 INC (R5)  
 MOV #TEMP,R3 ; LOAD ADDRESS  
 MOV #177777,(R3) ; LOAD LOCATION  
 MOV #ADR,R4 ; PLACE THE ADDRESS OF ADR IN R4  
 MOV #ADR1,(R4) ; PLACE THE ADDRESS OF ADR1 IN ADR  
 MOV (R3),@ (R4)+ ; LOAD LOCATION ADR1 WITH #177777  
 MOV #TEMP1,R0 ; PLACE THE ADDRESS OF TEMP1 IN R0  
 MOV #125252,(R0) ; SET EVERY OTHER BIT AT LOCATION TEMP1  
 SCC  
 BIC (R0)+,(R3) ; CLEAR EVERY OTHER BIT  
 JSR PC,@#SCC1 ; CHECK FOR CC = 1  
 BIT -(R0),(R3) ; CHECK IT  
 BEQ 1\$ ; CONTINUE IF OK  
 MOV #27,-(R5)  
 INC -(R5)  
 HALT ; BIC OR BIT INSTRUCTION FAILED  
 1\$: BIT #52525,(R3) ; CHECK IT  
 JSR PC,@#SCC1 ; CHECK FOR CC = 1  
 BIS 0(R0),(R3) ; SET THE BITS THAT WERE CLEARED  
 BMI 2\$ ; CONTINUE IF OK  
 MOV #300,-(R5)  
 INC -(R5)  
 HALT ; BIT OR BIS INSTRUCTION FAILED  
 2\$: MOV #77777,(R0)+ ; SET ALL THE BITS AT LOCATION TEMP1 EXCEPT SIGN BIT  
 MOV R0,R2  
 BIC -2(R2),(R3) ; TRY CLEARING THE OTHER BITS  
 JSR PC,@#SCC1 ; CHECK FOR CC = 11  
 CMP R0,#TEMP1+2 ; R0 SHOULD CONTAIN THE ADDRESS OF TEMP1+2  
 BEQ 3\$  
 MOV #301,-(R5)  
 INC -(R5)  
 HALT  
 3\$: MOV R0,(R0)+ ; PLACE THE ADDRESS OF LOCATION TEMP2 IN TEMP2  
 SEVC ; SET V & C BITS  
 BIC @-(R0),R0 ; CLEAR R0  
 JSR PC,@#SCC5 ; CHECK FOR CC = 5  
 BIT @-2(R4),(R3) ; CHECK IT  
 JSR PC,@#SCC11 ; CHECK FOR CC = 11  
 MOV #125252,-(SP) ; SET EVERY OTHER BIT ON THE STACK  
 MOV @-2(R4),(R3)+ ; SET ALL THE BITS AT LOCATION TEMP  
 BIC 0(SP),-(R3) ; CLEAR EVERY OTHER BIT AT LOCATION TEMP  
 CMP (R2)+,#52525 ; TEMP SHOULD CONTAIN # 52525  
 BEQ 4\$

2943	013472	012745	000302		MOV	#302,-(R5)	
2944	013476	005245			INC	-(R5)	
2945	013500	000000			HALT		
2946	013502	012700	000144	4\$:	MOV	#TEMP2+2,R0	: BIC FAILED IN MODE 6
2947	013506	010340			MOV	R3,-(R0)	: PLACE THE ADDRESS OF TEMP2+2 IN R0
2948	013510	014330			MOV	-(R3),@ (R0)+	: PLACE THE ADDRESS OF TEMP1 IN TEMP2
2949	013512	000263			SEVC		: MOVE # 52525 IN LOCATION TEMP1
2950	013514	035026			BIT	@-(R0),(SP)+	: SET V & C BITS
2951	013516	004737	017172		JSR	PC,@#CC5	: BIT TEST TEMP1 WITH STACK AND RESTORE STACK POINTER
2952	013522	020627	000450		CMP	SP,#START	: CHECK FOR CC = 5
2953	013526	001404			BEQ	INC1	: MAKE SURE THAT THE SP IS OK
2954	013530			FBIC1:			
2955	013530	012745	000303		MOV	#303,-(R5)	
2956	013534	005245			INC	-(R5)	
2957	013536	000000			HALT		: STACK POINTER FOULED UP OR SEQUENCE ERROR

2958  
2959  
2960  
2961  
2962 013540  
2963 013540 021527 000102  
2964 013544 001404  
2965 013546 012745 000304  
2966 013552 005245  
2967 013554 000000  
2968 013556 005215  
2969 013560 012704 000140  
2970 013564 012714 077777  
2971 013570 000261  
2972 013572 005214  
2973 013574 004737 017320  
2974 013600 012714 177776  
2975 013604 012700 000136  
2976 013610 012710 017256  
2977  
2978 013614 005214  
2979 013616 004730  
2980 013620 005214  
2981 013622 004737 017172  
2982 013626 005214  
2983 013630 004737 017066  
2984 013634 026427 000000 000001  
2985 013642 001404  
2986 013644 012745 000305  
2987 013650 005245  
2988 013652 000000  
2989 013654 000261  
2990 013656 005314  
2991 013660 004737 017172  
2992 013664 005314  
2993 013666 004770 177776  
2994 013672 012714 100000  
2995 013676 005314  
2996 013700 004737 017126  
2997 013704 005314  
2998 013706 004737 017066  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006 013712  
3007 013712 021527 000103  
3008 013716 001404  
3009 013720 012745 000306  
3010 013724 005245  
3011 013726 000000  
3012 013730 005215  
3013 013732 012703 000140

::\*\*\*\*\*  
:\*TEST: 102 NEW INSTRUCTIONS USED IN THIS SECTION ARE INC, DEC  
:\*\*\*\*\*

INC1:  
CMP (R5),#102  
BEQ 2\$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW  
MOV #304,-(R5)  
INC -(R5)  
HALT ; PROGRAM IS IN WRONG SEQUENCE  
2\$:  
INC (R5)  
MOV #TEMP1,R4 ; LOAD ADDRESS  
MOV #77777,(R4) ; TEMP1 = 77777  
SEC  
INC (R4) ; ADD ONES INTO LOCATION  
JSR PC,@#SCC13 ; CHECK FOR CC = 13  
MOV #177776,(R4)  
MOV #TEMP,R0 ; R0 IS POINTING TO LOCATION TEMP  
MOV #SCC11,(R0) ; PLACE THE ADDRESS OF SUBROUTINE TO CHECK CC - 11  
; IN LOCATION TEMP  
; CHECK FOR CC = 11  
; CHECK FOR CC = 5  
; CHECK FOR CC = 1  
; CHECK IT  
; CONTINUE IF OK  
4\$:  
SEC ; INC INSTRUCTION FAILED  
DEC (R4) ; SUBTRACT ONES FROM LOCATION  
JSR PC,@#SCC5 ; CHECK FOR CC = 5  
DEC (R4)  
JSR PC,@-2(R0) ; CHECK FOR CC = 11  
MOV #100000,(R4)  
DEC (R4)  
JSR PC,@#SCC3 ; CHECK FOR CC = 3  
DEC (R4)  
JSR PC,@#SCC1 ; CHECK FOR CC = 1

::\*\*\*\*\*  
:\*TEST: 103 NEW INSTRUCTION IN THIS SECTION IS COM  
:\*\*\*\*\*

COM1:  
CMP (R5),#103  
BEQ 1\$ ; IF IN WRONG SEQUENCE GO TO HLT  
MOV #306,-(R5)  
INC -(R5)  
HALT ; TEST IS IN WRONG SEQUENCE  
1\$:  
INC (R5)  
MOV #TEMP1,R3 ; LOAD ADDRESS

3014	013736	012713	125252	MOV	#125252,(R3)	: LOAD EVERY OTHER BIT
3015	013742	000277		SCC		
3016	013744	005163	000000	COM	0(R3)	: 1'S COMPLEMENT
3017	013750	004737	017066	JSR	PC,@#SCC1	: CHECK FOR CC = 1
3018	013754	022713	052525	CMP	#52525,(R3)	: CHECK IT
3019	013760	001404		BEQ	2\$	: CONTINUE IF OK
3020	013762	012745	000307	MOV	#307,-(R5)	
3021	013766	005245		INC	-(R5)	
3022	013770	000000		HALT		: COM INSTRUCTION FAILED
3023	013772	000277		2\$: SCC		
3024	013774	005123		COM	(R3)+	: COMPLEMENT BACK
3025	013776	004737	017256	JSR	PC,@#SCC11	: CHECK FOR CC = 11
3026	014002	022743	125252	CMP	#125252,-(R3)	: CHECK IT
3027	014006	001404		BEQ	3\$	: CONTINUE IF OK
3028	014010	012745	000310	MOV	#310,-(R5)	
3029	014014	005245		INC	-(R5)	
3030	014016	000000		HALT		: COM INSTRUCTION FAILED
3031	014020	010300		3\$: MOV	R3,R0	: R0 IS NOW POINTING TO LOCATION TEMP1
3032	014022	012710	177777	MOV	#177777,(R0)	
3033	014026	000277		SCC		
3034	014030	005110		COM	(R0)	
3035	014032	004737	017172	JSR	PC,@#SCC5	: CHECK FOR CC - 5



3036  
 3037  
 3038  
 3039  
 3040 014036  
 3041 014036 021527 000104  
 3042 014042 001033  
 3043 014044 005215  
 3044 014046 012704 000140  
 3045 014052 012724 000001  
 3046 014056 010402  
 3047 014060 012762 100000 000000  
 3048 014066 005444  
 3049 014070 004737 017256  
 3050 014074 022724 177777  
 3051 014100 001404  
 3052 014102 012745 000311  
 3053 014106 005245  
 3054 014110 000000  
 3055 014112 016444 000000  
 3056 014116 005414  
 3057 014120 004737 017320  
 3058 014124 026214 000000  
 3059 014130 001404  
 3060 014132  
 3061 014132 012745 000312  
 3062 014136 005245  
 3063 014140 000000  
 3064  
 3065  
 3066  
 3067  
 3068  
 3069  
 3070

```

:*****
:*TEST: 104 NEW INSTRUCTION IN THIS SECTION IS NEG
:*****
NEG1:
  CMP (R5),#104
  BNE ENEG1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$:
  INC (R5)
  MOV #TEMP1,R4 ; LOAD ADDRESS
  MOV #1,(R4)+ ; LOAD THE LOCATION
  MOV R4,R2
  MOV #100000,0(R2)
  NEG -(R4) ; 2'S COMPLEMENT
  JSR PC,@#CC11 ; CHECK FOR CC - 11
  CMP #177777,(R4)+ ; CHECK IT
  BEQ 2$ ; CONTINUE IF OK
  MOV #311,-(R5)
  INC -(R5)
2$:
  MOV 0(R4),-(R4) ; NEG INSTRUCTION FAILED
  NEG (R4) ; TEMP1 CONTAINS THE LARGEST NEGATIVE NUMBER
  JSR PC,@#CC13 ; 2'S COMPLEMENT
  CMP 0(R2),(R4) ; CHECK FOR CC = 13
  BEQ ROL1 ; CHECK IT
  BEQ ROL1 ; CONTINUE IF OK
ENEG1:
  MOV #312,-(R5)
  INC -(R5)
  HALT ; WRONG RESULT IN TEMP2 OR WRONG SEQUENCE

```

3071 014142  
 3072 014142 021527 000105  
 3073 014146 001032  
 3074 014150 005215  
 3075 014152 012701 000142  
 3076 014156 012711 020000  
 3077 014162 000257  
 3078 014164 006121  
 3079 014166 006141  
 3080 014170 004737 017300  
 3081 014174 022711 100000  
 3082 014200 001404  
 3083 014202 012745 000313  
 3084 014206 005245  
 3085 014210 000000  
 3086 014212 006161 000000  
 3087 014216 004737 017214  
 3088 014222 010102  
 3089 014224 006112  
 3090 014226 022711 000001  
 3091 014232 001404

```

:*****
:*TEST: 105 NEW INSTRUCTION IN THIS SECTION IS ROL
:*****
ROL1:
  CMP (R5),#105
  BNE EROL1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC (R5)
  MOV #TEMP2,R1 ; LOAD ADDRESS
  MOV #20000,(R1) ; LOAD LOCATION
  CCC ; CLEAR FLAGS
  ROL (R1)+ ; SHIFT
  ROL -(R1)
  JSR PC,@#CC12 ; CHECK FOR CC - 12
  CMP #100000,(R1) ; CHECK IT
  BEQ 1$ ; CONTINUE IF OK
  MOV #313,-(R5)
  INC -(R5)
1$:
  ROL 0(R1) ; ROL INSTRUCTION FAILED
  ROL (R1) ; SHIFT
  JSR PC,@#CC7 ; CHECK FOR CC - 7
  MOV R1,R2 ; R2 IS NOW POINTING TO LOCATION TEMP2
  ROL (R2) ; SHIFT
  CMP #,(R1) ; CHECK IT
  BEQ ROR1 ; CONTINUE IF OK

```

3092	014234		
3093	014234	012745	000314
3094	01424C	005245	
3095	014242	000000	

EROL1:

MOV	#314,-(R5)
INC	-(R5)
HALT	

; WRONG RESULT AT TEMP2 OR WRONG SEQUENCE

3096  
 3097  
 3098  
 3099  
 3100 014244  
 3101 014244 021527 000106  
 3102 014250 001030  
 3103 014252 005215  
 3104 014254 012702 000142  
 3105 014260 012712 000004  
 3106 014264 000257  
 3107 014266 006012  
 3108 014270 006012  
 3109 014272 022712 000001  
 3110 014276 001404  
 3111 014300 012745 000315  
 3112 014304 005245  
 3113 014306 000000  
 3114 014310 006012  
 3115 014312 004737 017214  
 3116 014316 006012  
 3117 014320 004737 017300  
 3118 014324 022712 100000  
 3119 014330 001404  
 3120 014332  
 3121 014332 012745 000316  
 3122 014336 005245  
 3123 014340 000000  
 3124  
 3125  
 3126  
 3127  
 3128  
 3129  
 3130  
 3131 014342  
 3132 014342 021527 000107  
 3133 014346 001404  
 3134 014350 012745 000317  
 3135 014354 005245  
 3136 014356 000000  
 3137 014360 005215  
 3138 014362 012703 000142  
 3139 014366 012713 020000  
 3140 014372 000257  
 3141 014374 006313  
 3142 014376 006313  
 3143 014400 004737 017300  
 3144 014404 022713 100000  
 3145 014410 001404  
 3146 014412 012745 000320  
 3147 014416 005245  
 3148 014420 000000  
 3149 014422 006313  
 3150 014424 004737 017214  
 3151 014430 006313

```

:*****
:*TEST: 106 NEW INSTRUCTION IN THIS SECTION IS ROR
:*****
ROR1:
  CMP      (R5),#106
  BNE     EROR1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #TEMP2,R2  ; LOAD ADDRESS
  MOV     #4,(R2)    ; LOAD LOCATION
  CCC     (R2)       ; CLEAR FLAGS
  ROR     (R2)       ; SHIFT
  ROR     (R2)
  CMP     #1,(R2)    ; CHECK IT
  BEQ     1$         ; CONTINUE IF OK
  MOV     #315,-(R5)
  INC     -(R5)
  HALT
1$:
  ROR     (R2)       ; ROR INSTRUCTION FAILED
  JSR     PC,@#SCC7  ; SHIFT
  ROR     (R2)       ; CHECK FOR CC - 7
  JSR     PC,@#SCC12 ; SHIFT
  CMP     #10000,(R2) ; CHECK FOR CC - 12
  BEQ     ASL1       ; CHECK IT
  ASL1
  EROR1:
  MOV     #316,-(R5)
  INC     -(R5)
  HALT
  ; WRONG RESULT AT TEMP2 OR WRONG SEQUENCE

```

```

:*****
:*TEST: 107 NEW INSTRUCTION IN THIS SECTION IS ASL
:*****

```

```

ASL1:
  CMP     (R5),#107
  BEQ     2$         ; IF IN WRONG SEQUENCE GO TO HLT BELOW
  MOV     #317,-(R5)
  INC     -(R5)
  HALT
2$:
  INC     (R5)       ; PROGRAM IS IN WRONG SEQUENCE
  MOV     #TEMP2,R3  ; LOAD ADDRESS
  MOV     #20000,(R3) ; LOAD LOCATION
  CCC     (R3)       ; CLEAR FLAGS
  ASL     (R3)       ; SHIFT
  ASL     (R3)
  JSR     PC,@#SCC12 ; CHECK FOR CC - 12
  CMP     #10000,(R3) ; CHECK IT
  BEQ     4$         ; CONTINUE IF OK
  MOV     #320,-(R5)
  INC     -(R5)
  HALT
4$:
  ASL     (R3)       ; ASL INSTRUCTION FAILED
  JSR     PC,@#SCC7  ; SHIFT
  ASL     (R3)       ; CHECK FOR CC = 7

```

CVKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 85  
CVKAAC.P11 09-OCT-78 08:58 T107

G 7

NEW INSTRUCTION IN THIS SECTION IS ASL

SEQ 008-

3152 014432 004737 017150

JSR PC.#\$CC4 ; CHECK FOR CC 4

```

3153
3154
3155
3156
3157 014436
3158 014436 021527 000110
3159 014442 001040
3160 014444 005215
3161 014446 012704 000142
3162 014452 012703 000136
3163 014456 012714 000004
3164 014462 000257
3165 014464 006214
3166 014466 006214
3167 014470 022714 000001
3168 014474 001404
3169 014476 012745 00032
3170 014502 005245
3171 014504 000000
3172 014506 006214
3173 014510 004737 017214
3174 014514 006214
3175 014516 004737 017150
3176 014522 012713 100002
3177 014526 006213
3178 014530 006213
3179 014532 004737 017256
3180 014536 022713 160000
3181 014542 001404
3182 014544
3183 014544 012745 000322
3184 014550 005245
3185 014552 000000
  
```

```

*****
: *TEST: 110 NEW INSTRUCTION IN THIS SECTION IS ASR
*****
  
```

```

ASR1:
      CMP      (R5),#110
      BNE     EASR1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1>:   INC      (R5)
      MOV     #TEMP2,R4  ; LOAD ADDRESSES
      MOV     #TEMP,R3
      MOV     #4,(R4)    ; LOAD LOCATION
      CCC     ; CLEAR FLAGS
      ASR     (R4)       ; SHIFT
      ASR     (R4)
      CMP     #1,(R4)    ; CHECK IT
      BEQ     2$         ; CONTINUE IF OK
      MOV     #321,-(R5)
      INC     -(R5)
      HALT    ; ASR INSTRUCTION FAILED
2$:   ASR     (R4)       ; SHIFT
      JSR     PC,@#5CC7  ; CHECK FOR CC = 7
      ASR     (R4)       ; SHIFT
      JSR     PC,@#5CC4  ; CHECK FOR CC = 4
      MOV     #100002,(R3) ; LOAD LOCATION
      ASR     (R3)       ; SHIFT
      ASR     (R3)
      JSR     PC,@#5CC11 ; CHECK FOR CC = 11
      CMP     #160000,(R3) ; CHECK IT
      BEQ     ADC1       ; CONTINUE IF OK
EASR1:
      MOV     #322,-(R5)
      INC     -(R5)
      HALT    ; WRONG RESULT IN TEMP OR WRONG SEQUENCE
  
```

```

3186
3187
3188
3189
3190
3191
3192
3193 014554
3194 014554 021527 000111
3195 014560 001404
3196 014562 012745 000323
3197 014566 005245
3198 014570 000000
3199 014572 005215
3200 014574 012700 000136
3201 014600 005010
3202 014602 000257
3203 014604 005510
3204 014606 004737 017150
3205 014612 000261
3206 014614 005510
3207 014616 000261
3208 014620 005510
  
```

```

*****
: *TEST: 111 NEW INSTRUCTION IN THIS SECTION IS ADC
*****
  
```

```

ADC1:
      CMP     (R5),#111
      BEQ     2$         ; IF IN WRONG SEQUENCE GO TO HLT BELOW
      MOV     #323,-(R5)
      INC     -(R5)
      HALT    ; PROGRAM IS IN WRONG SEQUENCE
2$:   INC     (R5)
      MOV     #TEMP,R0  ; LOAD ADDRESS
      CLR     (R0)      ; CLEAR THE LOCATION
      CCC     ; CLEAR FLAGS
      ADC     (R0)       ; ADD C BIT = 0
      JSR     PC,@#5CC4  ; CHECK FOR CC = 4
      SEC     ; C = 1
      ADC     (R0)       ; ADD C BIT = 1
      SEC     ; C = 1
      ADC     (R0)       ; AGAIN
  
```

NEW INSTRUCTION IN THIS SECTION IS ADC

SEU 008'

3209	014622	004737	017046		JSR	PC,@#SCC0	:	CHECK FOR CC = 0
3210	014626	022710	000002		CMP	#2,(R0)	:	CHECK IT
3211	014632	001404			BEQ	4\$	:	CONTINUE IF OK
3212	014634	012745	000324		MOV	#324,-(R5)		
3213	014640	005245			INC	-(R5)		
3214	014642	000000			HALT		:	ADC INSTRUCTION FAILED
3215	014644	012710	077777	4\$:	MOV	#77777,(R0)	:	LOAD LARGEST POSITIVE NUMBER
3216	014650	000261			SEC		:	C=1
3217	014652	005510			ADC	(R0)	:	ADD C BIT=1
3218	014654	004737	017300		JSR	PC,@#SCC12	:	CHECK FOR CC = 12
3219	014660	022710	100000		CMP	#100000,(R0)	:	CHECK IT
3220	014664	001404			BEQ	6\$	:	CONTINUE IF OK
3221	014666	012745	000325		MOV	#325,-(R5)		
3222	014672	005245			INC	-(R5)		
3223	014674	000000			HALT		:	ADC INSTRUCTION AILED
3224	014676	012710	177777	6\$:	MOV	#-1,(R0)	:	LOAD -1
3225	014702	000261			SEC		:	C=1
3226	014704	005510			ADC	(R0)	:	ADD C BIT=1
3227	014706	004737	017172		JSR	PC,@#SCC5	:	CHECK FOR CC = 5

3228  
 3229  
 3230  
 3231  
 3232 014712  
 3233 014712 021527 000112  
 3234 014716 001404  
 3235 014720 012745 000326  
 3236 014724 005245  
 3237 014726 000000  
 3238 014730 005215  
 3239 014732 012701 000136  
 3240 014736 012711 000003  
 3241 014742 000257  
 3242 014744 005611  
 3243 014746 004737 017046  
 3244 014752 022711 000003  
 3245 014756 001404  
 3246 014760 012745 000327  
 3247 014764 005245  
 3248 014766 000000  
 3249 014770 000261  
 3250 014772 005611  
 3251 014774 000261  
 3252 014776 005611  
 3253 015000 004737 017046  
 3254 015004 022711 000001  
 3255 015010 001404  
 3256 015012 012745 000330  
 3257 015016 005245  
 3258 015020 000000  
 3259 015022 000261  
 3260 015024 005611  
 3261 015026 004737 017150  
 3262 015032 000261  
 3263 015034 005611  
 3264 015036 004737 017256  
 3265 015042 022711 177777  
 3266 015046 001404  
 3267 015050 012745 000331  
 3268 015054 005245  
 3269 015056 000000  
 3270 015060 012711 100000  
 3271 015064 000261  
 3272 015066 005611  
 3273 015070 004737 017106  
 3274  
 3275  
 3276  
 3277  
 3278  
 3279  
 3280  
 3281 015074  
 3282 015074 021527 000113  
 3283 015100 001026

\*\*\*\*\*  
 \*TEST: 112 NEW INSTRUCTION IN THIS SECTION IS SBC  
 \*\*\*\*\*

SBC1:  
 CMP (R5),#112  
 BEQ 1\$ ; IF IN WRONG SEQUENCE GO TO HLT  
 MOV #326,-(R5)  
 INC -(R5)  
 HALT ; TEST IS IN WRONG SEQUENCE  
 1\$: INC (R5)  
 MOV #TEMP,R1 ; LOAD ADDRESS  
 MOV #3,(R1) ; LOAD LOCATION  
 CCC ; CLEAR FLAGS  
 SBC (R1) ; SUBTRACT C BIT=0  
 JSR PC,@#S\$CC0 ; CHECK FOR CC = 0  
 CMP #3,(R1) ; CHECK IT  
 BEQ 2\$ ; CONTINUE IF OK  
 MOV #327,-(R5)  
 INC -(R5)  
 HALT ; SBC INSTRUCTION FAILED  
 2\$: SEC ; C=1  
 SBC (R1) ; SUBTRACT C BIT=1  
 SEC ; C-1  
 SBC (R1)  
 JSR PC,@#S\$CC0 ; CHECK FOR CC = 0  
 CMP #1,(R1) ; CHECK IT  
 BEQ 3\$  
 MOV #330,-(R5)  
 INC -(R5)  
 HALT ; SBC INSTRUCTION FAILED  
 3\$: SEC ; C-1  
 SBC (R1) ; SUBTRACT C BIT=1  
 JSR PC,@#S\$CC4 ; CHECK FOR CC = 4  
 SEC ; C=1  
 SBC (R1) ; SUBTRACT C BIT = 1  
 JSR PC,@#S\$CC11 ; CHECK FOR CC = 11  
 CMP #-1,(R1) ; CHECK IT  
 BEQ 4\$ ; CONTINUE IF OK  
 MOV #331,-(R5)  
 INC -(R5)  
 HALT ; SBC INSTRUCTION FAILED  
 4\$: MOV #100000,(R1) ; LOAD R1  
 SEC ; C=1  
 SBC (R1) ; SUBTRACT C BIT = 1  
 JSR PC,@#S\$CC2 ; CHECK FOR CC = 2

\*\*\*\*\*  
 \*TEST: 113 NEW INSTRUCTION IN THIS SECTION IS SXT  
 \*\*\*\*\*

SXT1:  
 CMP (R5),#113  
 BNF ESXT1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST

3284	015102	005215		1\$:	INC	(R5)		
3285	015104	012702	000140		MOV	#TEMP1,R2	:	LOAD ADDRESS
3286	015110	005012			CLR	(R2)	:	CLEAR LOCATIONS
3287	015112	000277			SCC			
3288	015114	000254			CLNZ			
3289	015116	006712			SXT	(R2)	:	SIGN EXTEND
3290	015120	004737	017172		JSR	PC,@#SCC5	:	CHECK FOR CC = 5
3291	015124	005712			TST	(R2)	:	LOCATION SHOULD STILL BE 0
3292	015126	001404			BEQ	2\$	:	CONTINUE IF OK
3293	015130	012745	000332		MOV	#332,-(R5)		
3294	015134	005245			INC	-(R5)		
3295	015136	000000			HALT		:	SXT INSTRUCTION FAILED
3296	015140	000273		2\$:	SENV		:	SET N, V & C BITS
3297	015142	006712			SXT	(R2)	:	SIGN EXTEND
3298	015144	004737	017256		JSR	PC,@#SCC11	:	CHECK FOR CC = 11
3299	015150	022712	177777		CMP	#-1,(R2)	:	LOCATION SHOULD NOW HAVE -1
3300	015154	001404			BEQ	SWAB1	:	CONTINUE IF OK
3301	015156			ESX*1:				
3302	015156	012745	000333		MOV	#333,-(R5)		
3303	015162	005245			INC	-(R5)		
3304	015164	000000			HALT		:	WRONG RESULT IN TEMP1 OR WRONG SEQUENCE



3305  
3306  
3307  
3308  
3309 015166  
3310 015166 021527 000114  
3311 015172 001034  
3312 015174 005215  
3313 015176 012703 000142  
3314 015202 012713 125125  
3315 015206 000277  
3316 015210 000250  
3317 015212 000313  
3318 015214 004737 017236  
3319 015220 022713 052652  
3320 015224 001404  
3321 015226 012745 000334  
3322 015232 005245  
3323 015234 000000  
3324 015236 012713 000377  
3325 015242 000277  
3326 015244 000244  
3327 015246 000363 000000  
3328 015252 004737 017150  
3329 015256 022713 177400  
3330 015262 001404  
3331 015264  
3332 015264 012745 000335  
3333 015270 005245  
3334 015272 000000  
3335  
3336  
3337  
3338  
3339  
3340  
3341  
3342 015274  
3343 015274 021527 000115  
3344 015300 001041  
3345 015302 005215  
3346 015304 012704 177777  
3347 015310 012767 177777 162622  
3348 015316 000277  
3349 015320 074467 162614  
3350 015324 004737 017172  
3351 015330 012767 077777 162602  
3352 015336 012700 000140  
3353 015342 000263  
3354 015344 000244  
3355 015346 074410  
3356 015350 004737 017256  
3357 015354 012701 125252  
3358 015360 012720 052525  
3359 015364 000277  
3360 015366 074140

```
*****  
: *TEST: 114 NEW INSTRUCTION IN THIS SECTION IS SWAB  
: *****  
SWAB1:  
CMP (R5),#114  
BNE ESWAB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
MOV #TEMP2,R3 ; LOAD ADDRESS  
MOV #125125,(R3) ; LOAD BIT PATTERN INTO LOCATION  
SCC  
CLN  
SWAB (R3) ; SWAP BYTES OF LOCATIONS  
JSR PC,@#SCC10 ; CHECK FOR CC = 10  
CMP #52652,(R3) ; CHECK IT  
BEQ 1$ ; CONTINUE IF OK  
MOV #334,-(R5)  
INC -(R5)  
HALT ; SWAB INSTRUCTION FAILED  
1$: MOV #377,(R3)  
SCC  
CLZ  
SWAB 0(R3)  
JSR PC,@#SCC4 ; CHECK FOR CC = 4  
CMP #177400,(R3)  
BEQ XOR1  
ESWAB1:  
MOV #335,-(R5)  
INC -(R5)  
HALT ; WRONG RESULT IN: TEMP2 OR WRONG SEQUENCE
```

3338  
3339  
3340  
3341  
3342 015274  
3343 015274 021527 000115  
3344 015300 001041  
3345 015302 005215  
3346 015304 012704 177777  
3347 015310 012767 177777 162622  
3348 015316 000277  
3349 015320 074467 162614  
3350 015324 004737 017172  
3351 015330 012767 077777 162602  
3352 015336 012700 000140  
3353 015342 000263  
3354 015344 000244  
3355 015346 074410  
3356 015350 004737 017256  
3357 015354 012701 125252  
3358 015360 012720 052525  
3359 015364 000277  
3360 015366 074140

```
*****  
: *TEST: 115 NEW INSTRUCTION IN THIS SECTION IS XOR  
: *****  
XOR1:  
CMP (R5),#115  
BNE EXOR1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST  
INC (R5)  
MOV #-1,R4 ; LOAD LOCATIONS  
MOV #-1,TEMP1 ;  
SCC  
XOR R4,TEMP1 ; SHOULD PRODUCE 0'S IN TEMP1  
JSR PC,@#SCC5 ; CHECK FOR CC = 5  
MOV #77777,TEMP1  
MOV #TEMP1,R0 ; PLACE THE ADDRESS OF TEMP1 IN R0  
SEVC ; SET V & C BITS  
CLZ  
XOR R4,(R0)  
JSR PC,@#SCC11 ; CHECK FOR CC = 11  
MOV #125252,R1 ; LOAD LOCATIONS  
MOV #52525,(R0) ;  
SCC  
XOR R1,-(R0) ; SHOULD PRODUCE ALL 1'S IN TEMP1
```

CJKAAC MACY11 30A(1052) 09-OCT-78 08:59 PAGE 91  
CJKAAC.P11 09-OCT-78 08:58 T115

NEW INSTRUCTION IN THIS SECTION IS XOR

SEQ 0090

3361 015370 004737 017256  
3362 015374 022737 177777 000140  
3363 015402 001404  
3364 015404  
3365 015404 012745 000336  
3366 015410 005245  
3367 015412 000000

EXOR1:

JSR PC, @R5CC11  
CMP #-1, @TEMP1  
BEQ ADD1  
MOV #336, -(R5)  
INC -(R5)  
HALT

: CHECK FOR CC = 11  
: CHECK IT  
: CONTINUE IF OK  
: WRONG RESULT IN TEMP1 OR WRONG SEQUENCE

```

3368
3369
3370
3371
3372 015414
3373 015414 021527 000116
3374 015420 001133
3375 015422 005215
3376 015424 012700 000142
3377 015430 012701 000136
3378 015434 012767 021421 162500
3379 015442 011011
3380 015444 061011
3381 015446 004737 017046
3382 015452 022767 043042 162456
3383 015460 001404
3384 015462 012745 000337
3385 015466 005245
3386 015470 000000
3387 015472 005010 1$:
3388 015474 060020
3389 015476 024027 000142
3390 015502 001404
3391 015504 012745 000340
3392 015510 005245
3393 015512 000000
3394 015514 012767 156357 162420 2$:
3395 015522 012011
3396 015524 064011
3397 015526 004737 017256
3398 015532 022767 134736 162376
3399 015540 001404
3400 015542 012745 000341
3401 015546 005245
3402 015550 000000
3403 015552 012767 100000 162362 3$:
3404 015560 011061 000000
3405 015564 066011 000000
3406 015570 004737 017214
3407 015574 012767 021421 162336
3408 015602 012760 000140 000700
3409 015610 012711 156357
3410 015614 010004
3411 015616 067411 000000
3412 015622 004737 017172
3413 015626 005430
3414 015630 012746 021421
3415 015634 065066 000000
3416 015640 004737 017172
3417 015644 005726
3418
3419 015646 001404
3420 015650 012745 000342
3421 015654 005245
3422 015656 000000
3423 015660 012767 137777 162254 4$:

```

```

:*****
:*TEST: 116 NEW INSTRUCTION IN THIS SECTION IS ADD
:*****

```

```

ADD1:
CMP (R5),#116
BNE EADD1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
INC (R5)
MOV #TEMP2,R0 ; LOAD ADDRESSES
MOV #TEMP,R1
MOV #21421,TEMP2 ; LOAD LOCATIONS
MOV (R0),(R1)
ADD (R0),(R1) ; ADD
JSR PC,@#SCC0 ; CHECK FOR CC 0
CMP #43042,TEMP ; CHECK IT
BEQ 1$ ; CONTINUE IF OK
MOV #337,-(R5)
INC -(R5)
HALT ; ADD INSTRUCTION FAILED
1$: CLR (R0) ; CLEAR LOCATION TEMP2
ADD R0,(R0)+ ; PLACE THE ADDRESS OF TEMP2 IN LOCATION TEMP2
CMP -(R0),#TEMP2 ; CHECK IT
BEQ 2$
MOV #340,-(R5)
INC -(R5)
HALT ; ADD INSTRUCTION FAILED IN MODE 2
2$: MOV #-21421,TEMP2 ; LOAD LOCATIONS
MOV (R0),(R1)
ADD -(R0),(R1) ; ADD
JSR PC,@#SCC11 ; CHECK FOR CC - 11
CMP #-43042,TEMP ; CHECK IT
BEQ 3$ ; CONTINUE IF OK
MOV #341,-(R5)
INC -(R5)
HALT ; ADD INSTRUCTION FAILED
3$: MOV #100000,TEMP2 ; LOAD LOCATIONS
MOV (R0),0(R1)
ADD 0(R0),(R1) ; ADD SHOULD RESULT AS 0'S
JSR PC,@#SCC7 ; CHECK FOR CC-7
MOV #21421,TEMP1 ; LOAD LOCATION TEMP1
MOV #TEMP1,0(RU) ; PLACE THE ADDRESS OF LOCATION TEMP1 IN TEMP2
MOV #-21421,(R1) ; LOAD LOCATION TEMP
MOV R0,R4 ; MAKE R4 POINT TO LOCATION TEMP2
ADD @0(R4),(R1) ; ADD SHOULD RESULT AS 0'S
JSR PC,@#SCC5 ; CHECK FOR CC=5
NEG @0(R0)+ ; NEGATE THE CONTENTS OF TEMP1
MOV #21421,-(SP) ; PLACE # 21421 ON THE STACK
ADD @-(R0),0(SP) ; ADD, SHOULD=0'S
JSR PC,@#SCC5 ; CHECK FOR CC=5
TST (SP)+ ; CHECK THE STACK TO CONTAIN 0, ALSO
; RESTORE THE STACK POINTER
BEQ 4$
MOV #342,-(R5)
INC -(R5)
HALT ; ADD INSTRUCTION FAILED IN MODE 5
4$: MOV #137777,TEMP2

```

```

3424 015666 062767 137777 162246      ADD    #137777,TEMP2
3425 015674 004737 017126                JSR    PC,#$CC3      ; CHECK CC=3
3426 015700 022767 077776 162234      CMP    #77776,TEMP2
3427 015706 001404                BEQ    SUB1
3428 015710                EADD1:
3429 015710 012745 000343      MOV    #343,-(R5)
3430 015714 005245                INC    -(R5)
3431 015715 000000                HALT    ; WRONG RESULT AT TEMP OR WRONG SEQUENCE
3432
3433
3434
3435
3436
3437
3438
3439 015720                SUB1:
3440 015720 021527 000117      CMP    (R5),#117
3441 015724 001100      BNE    ESUB1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
3442 015726 005215                INC    (R5)
3443 015730 012702 000136      MOV    #TEMP,R2      ; LOAD ADDRESSES
3444 015734 012703 000140      MOV    #TEMP1,R3      ;
3445 015740 012767 021421 162170      MOV    #21421,TEMP      ; LOAD LOCATIONS
3446 015746 012767 156357 162164      MOV    #-21421,TEMP1      ;
3447 015754 161213      SUB    (R2),(R3)      ; RESULT SHOULD=-43042
3448 015756 004737 017236      JSR    PC,#$CC10      ; CHECK FOR CC = 10
3449 015762 022767 134736 162150      CMP    #-43042,TEMP1      ; CHECK IT
3450 015770 001404                BEQ    1$      ; CONTINUE IF OK
3451 015772 012745 000344      MOV    #344,-(R5)
3452 015776 005245                INC    -(R5)
3453 016000 000000                HALT    ; SUB INSTRUCTION FAILED
3454 016002 012767 021421 162130 1$: MOV    #21421,TEMP1      ; LOAD LOCATION
3455 016010 161213      SUB    (R2),(R3)      ; RESULT SHOULD=0
3456 016012 001404                BEQ    2$
3457 016014 012745 000345      MOV    #345,-(R5)
3458 016020 005245                INC    -(R5)
3459 016022 000000                HALT    ; SUB INSTRUCTION FAILED
3460 016024 012767 177777 162106 2$: MOV    #-1,TEMP1      ; LOAD LOCATIONS
3461 016032 012767 077777 162076      MOV    #77777,TEMP      ; LOAD LOCATIONS
3462 016040 161312      SUB    (R3),(R2)      ; RESULT SHOULD GIVE 100000 AND OVERFLOW
3463 016042 004737 017320      JSR    PC,#$CC13      ; CHECK FOR CC = 13
3464 016046 022767 100000 162062      CMP    #100000,TEMP      ; CHECK IT
3465 016054 001404                BEQ    3$      ; CONTINUE IF OK
3466 016056 012745 000346      MOV    #346,-(R5)
3467 016062 005245                INC    -(R5)
3468 016064 000000                HALT    ; SUB INSTRUCTION FAILED
3469 016066 012712 177777      3$: MOV    #-1,(R2)
3470 016072 161312      SUB    (R3),(R2)
3471 016074 004737 017150      JSR    PC,#$CC4      ; CHECK FOR CC = 4
3472 016100 012767 077777 162030      MOV    #77777,TEMP
3473 016106 162767 077777 162022      SUB    #77777,TEMP
3474 016114 004737 017150      JSR    PC,#$CC4      ; CHECK FOR CC=4
3475 016120 005767 162012      TST    TEMP
3476 016124 001404                BEQ    SOB      ; TEMP SHOULD BE 0
3477 016126                ESUB1:
3478 016126 012745 000347      MOV    #717,-(R5)
3479 016132 005245                INC    -(R5)

```

CVKAAC MACY11 304(1052) 09-OCT-78 08:59 PAGE 94  
CVKAAC.P11 09-OCT-78 08:58 T117

C 8

NEW INSTRUCTION IN THIS SECTION IS SUB

SEQ 0093

3480 016134 000000

HALT

; SUB INSTRUCTION FAILED OR SEQUENCE ERROR

3481  
 3482  
 3483  
 3484  
 3485 016136  
 3486 016136 021527 000120  
 3487 016142 001042  
 3488 016144 005215  
 3489 016146 012700 000012  
 3490 016152 005001  
 3491 016154 005201  
 3492 016156 020127 000012  
 3493 016162 003404  
 3494 016164 012745 000350  
 3495 016170 005245  
 3496 016172 000000  
 3497 016174 000277  
 3498 016176 077012  
 3499 016200 004757 017340  
 3500 016204 005700  
 3501 016206 001404  
 3502 016210 012745 000351  
 3503 016214 005245  
 3504 016216 000000  
 3505 016220 022701 000012  
 3506 016224 001404  
 3507 016226 012745 000352  
 3508 016232 005245  
 3509 016234 000000  
 3510 016236 012704 000010  
 3511 016242 077401  
 3512 016244 005704  
 3513 016246 001404  
 3514 016250  
 3515 016250 012745 000353  
 3516 016254 005245  
 3517 016256 000000

```

:*****
:*TEST: 120 NEW INSTRUCTION IN THIS SECTION IS SOB
:*****
SOB:
  CMP      (R5),#120
  BNE     ESOB      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #10.,R0   ; LOAD REGISTERS
  CLR     R1
  1$:    INC     R1   ; KEEP COUNT
        CMP     R1,#10.
        BLE     2$
        MOV     #350,-(R5)
        INC     -(R5)
        HALT    ; SOB INSTRUCTION FAILED
  2$:    SCC
        SOB     R0,1$ ; SUB. 1 FROM REG. 0, GO BACK TO 1$
        JSR     PC,@#CC17 ; CHECK FOR CC = 17
        TST     R0   ; REG. 0 = 0 ?
        BEQ     3$   ; NO, FAILED
        MOV     #351,-(R5)
        INC     -(R5)
        HALT    ; SOB INSTRUCTION FAILED
  3$:    CMP     #10.,R1 ; DID IT GO THRU 10 TIMES ?
        BEQ     4$   ; CONTINUE IF OK
        MOV     #352,-(R5)
        INC     -(R5)
        HALT    ; SOB INSTRUCTION FAILED
  4$:    MOV     #10,R4 ; PLACE #10 IN R4
  5$:    SOB     R4,5$ ; STAY HERE UNTILL R4 = 0
        TST     R4
        BEQ     PSWNO ; CONTINUE IF OK
ESOB:
  MOV     #353,-(R5)
  INC     -(R5)
  HALT    ; SOB FAILED OR WRONG SEQUENCE

```

3518  
 3519  
 3520  
 3521  
 3522  
 3523  
 3524  
 3525  
 3526 016260  
 3527 016260 021527 000121  
 3528 016264 001042  
 3529 016266 005215  
 3530 016270 012700 000176  
 3531 016274 012701 000140  
 3532 016300 012711 177777  
 3533 016304 005010  
 3534 016306  
 3535 016306 106410  
 3536 016310 004737 017046

```

:*****
:*TEST: 121 NEW INSTRUCTIONS IN THIS SECTION ARE MTPS & MFPS
:*****
PSWNO:
  CMP     (R5),#121
  BNE     EPSWNO   ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
  INC     (R5)
  MOV     #TEMP,R0 ; PUT THE ADDRESS OF TEMP IN R0
  MOV     #TEMP1,R1 ; PUT THE ADDRESS OF TEMP1 IN R1
  MOV     #177777,(R1) ; TEMP1 = 177777
  CLR     (R0)     ; TEMP = 0
  MTPS   (R0)     ; PSW = 0
  .WORD  106400...C
  JSR     PC,@#CC0 ; CHECK FOR CC = 0

```

```

3537 016314 MFPS (R1) ; MOVE PSW TO TEMP1
3538 016314 106711 .WORD 106700!...C
3539 016316 004737 017150 JSR PC, @PSCC4 ; CHECK FOR CC = 4
3540 016322 022711 177400 CMP #177400, (R1) ; CHECK TEMP1 TO MAKE SURE THAT ONLY
; THE LOWER BYTE WAS AFFECTED BY MFPS
3541
3542 016326 001404 BEQ 1$
3543 016330 012745 000354 MOV #354, -(R5)
3544 016334 005245 INC -(R5)
3545 016336 000000 HALT
3546 016340 005011 i$: CLR (R1) ; MTPS OR MFPS INSTRUCTION FAILED
3547 016342 MTPS #377 ; SET PSW = 357 SINCE T BIT CAN NOT BE SET BY MTPS
3548 016342 106427 .WORD 106400...C
3549 016346 004737 017340 JSR PC, @PSCC17 ; CHECK FOR CC = 17
3550 016352 MFPS TEMP1 ; MOVE PSW TO TEMP1
3551 016352 106767 .WORD 106700...C
3552 016356 004737 017256 JSR PC, @PSCC11 ; CHECK FOR CC = 11 [C BIT SHOULD NOT BE EFFECTED BY MFP
3553 016362 022767 000357 161550 CMP #357, TEMP1
3554 016370 001404 BEQ BTWRD
3555 016372 EPSWNO:
3556 016372 012745 000355 MOV #355, -(R5)
3557 016376 005245 INC -(R5)
3558 016400 000000 HALT ; MFPS INSTRUCTION FAILED IN MODE 6
3559 ; OR SEQUENCE ERROR

```

3560  
3561  
3562  
3563  
3564 016402  
3565 016402 021527 000122  
3566 016406 001124  
3567 016410 005215  
3568 016412 005000  
3569 016414 000277  
3570 016416 112700 000200  
3571  
3572 016422 004737 017256  
3573 016426 022700 177600  
3574 016432 001404  
3575 016434 012745 000356  
3576 016440 005245  
3577 016442 000000  
3578 016444 000277  
3579 016446 012700 177777  
3580 016452 112700 000000  
3581 016456 004737 017172  
3582 016462 005700  
3583 016464 001404  
3584 016466 012745 000357  
3585 016472 005245  
3586 016474 000000  
3587 016476 012704 000142  
3588 016502 012714 000377  
3589 016506 012706 000446  
3590 016512 116426 000000  
3591 016516 022706 000450  
3592 016522 001404  
3593 016524 012745 000360  
3594 016530 005245  
3595 016532 000000  
3596  
3597 016534 124627 000377  
3598  
3599 016540 001404  
3600 016542 012745 000361  
3601 016546 005245  
3602 016550 000000  
3603 016552 022706 000446  
3604  
3605 016556 001404  
3606 016560 012745 000362  
3607 016564 005245  
3608 016566 000000  
3609 016570 016467 000000 161340  
3610 016576 005726  
3611 016600 000277  
3612 016602 114667 161331  
3613 016606 004737 017256  
3614 016612 022767 177777 161316  
3615 016620 001404

\*\*\*\*\*  
: \*TEST: 122 BYTE INSTRUCTIONS REQUIRING WORD INST. TO CHECK  
: \*\*\*\*\*

BTWRD:

CMP (R5),#122  
BNE EBTWRD ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF THE TEST  
INC (R5)  
CLR R0  
SCC  
MOVB #200,R0 ; SET THE HIGHEST BIT OF THE  
; LOWER BYTE  
JSR PC,@#SCC11 ; CHECK FOR CC=11  
CMP #177600,R0 ; CHECK FOR SIGN EXTENSION IN R0  
BEQ 1\$  
MOV #356,-(R5)  
INC -(R5)  
HALT ; SIGN WAS NOT EXTENDED IN R0  
1\$: SCC  
MOV #177777,R0  
MOVB #0,R0 ; CLEAR THE LOWER BYTE OF R0.  
JSR PC,@#SCC5 ; CHECK FOR CC=5  
TST R0 ; CHECK R0 FOR SIGN EXTENSION  
BEQ 2\$  
MOV #357,-(R5)  
INC -(R5)  
HALT ; SIGN WAS NOT EXTENDED IN R0.  
2\$: MOV #TEMP2,R4 ; R4 IS POINTING TO TEMP2  
MOV #377,(R4) ; PLACE #377 IN LOCATION TEMP2  
MOV #START-2,R6  
MOVB 0(R4),(R6)+ ; PUSH # 377 ON STACK  
CMP #START,R6  
BEQ 3\$  
MOV #360,-(R5)  
INC -(R5)  
HALT ; R6 DID NOT GET INCREMENTED  
; BY 2 BY A BYTE INSTRUCTION  
3\$: CMPB -(R6),#377 ; CHECK LOCATION START-2 TO  
; CONTAIN PROPER DATA  
BEQ 4\$  
MOV #361,-(R5)  
INC -(R5)  
HALT ; BYTE INSTRUCTION IS FAILING WITH R6  
4\$: CMP #START-2,R6 ; CHECK THAT R6 WAS DECREMENTED  
; BY 2 BY A BYTE INSTRUCTION  
BEQ 5\$  
MOV #362,-(R5)  
INC -(R5)  
HALT ; R6 WAS NOT DECREMENTED  
5\$: MOV 0(R4),TEMP ; SET THE LOWER BYTE OF LOCATION TEMP  
TST (R6)+ ; RESTORE STACK POINTER  
SCC  
MOVB -(SP),TEMP+1 ; SET THE HIGHER BYTE OF LOCATION TEMP  
JSR PC,@#SCC11 ; CHECK FOR CC=11  
CMP #177777,TEMP ; CHECK TEMP FOR THE CORRECT VALUE  
BEQ 6\$



3616	016622	012745	000363		MOV	#363,-(R5)	
3617	016626	005245			INC	-(R5)	
3618	016630	000000			HALT		; TEMP FOULED UP
3619	016632	005067	161300	6\$:	CLR	TEMP	
3620	016636	000241			CLC		
3621	016640	105167	161273		COMB	TEMP+1	; WRITE 1'S IN THE HIGHER BYTE OF TEMP
3622	016644	004737	017256		JSR	PC,@#SCC11	; CHECK FOR CC=11
3623	016650	022767	177400	161260	CMP	#177400,TEMP	
3624	016656	001404			BEQ	NEXT	
3625	016660				EBTWRD:		
3626	016660	012745	000364		MOV	#364,-(R5)	
3627	016664	005245			INC	-(R5)	
3628	016666	000000			HALT		; WRONG VALUE IN TEMP OR WRONG SEQUENCE

```
3629
3630
3631
3632 : END OF PASS
3633 : *****
3634
3635
3636
3637 016670 NEXT:
3638 016670 021527 000123 CMP (R5),#123
3639 016674 001404 BEQ 2$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW
3640 016676 012745 000365 MOV #365,-(R5)
3641 016702 005245 INC -(R5)
3642 016704 000000 HALT ; PROGRAM IS IN WRONG SEQUENCE
3643 016706 005267 161172 2$: INC $PASS
3644 016712 126727 161166 000001 CMPB $PASS,#1 ; ALLOW THE TYPE OUT OF END OF
; PASS EVERY 377 PASSES
3645
3646 016720 001011 BNE DOAGN
3647 016722 000004 000152 .TYPE ,ENDPAS ; TYPE END OF PASS MESSAGE
3648 016726 013700 000042 GFT42: MOV @#42,R0
3649 016732 001404 BEQ DOAGN
3650 016734 004710 $ENDAD: JSR PC,(R0)
3651 016736 000240 NOP
3652 016740 000240 NOP
3653 016742 000240 NOP
3654 016744 005067 161132 DOAGN: CLR $TESTN ; PREPARE TO START FROM TEST 0
3655 016750 000167 161474 RETURN: JMP START ; START TEST OVER AT BEGINNING
3656 :
3657 :
3658 :;*****
3659 :
3660 .SBTTL POWER FAIL ROUTINE
3661
3662
3663 016754 012737 016764 000024 PWRDN: MOV #PWRUP,@#24 ; GO TO POWER UP ROUTINE AFTER THE POWER COMES BACK
3664 016762 000000 HALT
3665
3666 016764 012706 000450 PWRUP: MOV #START,$P
3667 016770 012737 016754 000024 MOV #PWRDN,@#24
3668 016776 000004 000166 .TYPE ,POWER
3669 017002 000760 BR DOAGN
```

```
3670  
3671  
3672          .SBTTL  TYPE ROUTINE  
3673  
3674  
3675 017004 132737 000040 000117 TYPE:  BITB  #40,@#SENVN  : HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?  
3676 017012 001012          BNE   4$          : IF SO THEN GO TO 4$  
3677 0170  + 017603 000000          MOV   @(SP),R3      : GET ADDRESS OF MESSAGE  
3678  
3679 017020 105713          '$:  TSTB  (R3)          : END OF MESSAGE :  
3680 017022 001406          BEQ   4$          : YES, GO WRAP IT UP  
3681  
3682 017024 105777 161114          3$:  TSTB  @TPS          : READY FOR NEXT CHARACTER ?  
3683 017030 100375          BPL   3$          : NO, WAIT  
3684 017032 112377 161110          MOVB  (R3)+,@T+3    : LOAD AND TYPE THE CHARACTER  
3685 017036 000770          BR    1$          : YES, GET THE NEXT CHARACTER  
3686  
3687 017040 062716 000002          4$:  ADD   #2,(SP)      : ADJUST THE RETURN PC  
3688 017044 000006          RT   : RETURN  
3689
```

3690	017046	003402		\$CC0:	BLE	1\$	
3691	017050	100401			BMI	1\$	
3692	017052	103004			BCC	2\$	
3693	017054			1\$:			
3694	017054	012745	000366		MOV	#366, -(R5)	
3695	017060	005245			INC	-(R5)	
3696	017062	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN - 0
3697	017064	000207		2\$:	RTS	PC	
3698							
3699	017066	003402		\$CC1:	BLE	1\$	
3700	017070	100401			BMI	1\$	
3701	017072	103404			BCS	2\$	
3702	017074			1\$:			
3703	017074	012745	000367		MOV	#367, -(R5)	
3704	017100	005245			INC	-(R5)	
3705	017102	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN - 1
3706	017104	000207		2\$:	RTS	PC	
3707							
3708	017106	100402		\$CC2:	BMI	1\$	
3709	017110	101401			BLOS	1\$	
3710	017112	102404			BVS	2\$	
3711	017114			1\$:			
3712	017114	012745	000370		MOV	#370, -(R5)	
3713	017120	005245			INC	-(R5)	
3714	017122	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN = 2
3715	017124	000207		2\$:	RTS	PC	
3716							
3717	017126	100403		\$CC3:	BMI	1\$	
3718	017130	001402			BEQ	1\$	
3719	017132	102001			BVC	1\$	
3720	017134	103404			BCS	2\$	
3721	017136			1\$:			
3722	017136	012745	000371		MOV	#371, -(R5)	
3723	017142	005245			INC	-(R5)	
3724	017144	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN 3
3725	017146	000207		2\$:	RTS	PC	
3726							
3727	017150	100403		\$CC4:	BMI	1\$	
3728	017152	001002			BNE	1\$	
3729	017154	102401			BVS	1\$	
3730	017156	103004			BCC	2\$	
3731	017160			1\$:			
3732	017160	012745	000372		MOV	#372, -(R5)	
3733	017164	005245			INC	-(R5)	
3734	017166	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN 4
3735	017170	000207		2\$:	RTS	PC	
3736							
3737	017172	100403		\$CC5:	BMI	1\$	
3738	017174	001002			BNE	1\$	
3739	017176	102401			BVS	1\$	
3740	017200	103404			BCS	2\$	
3741	017202			1\$:			
3742	017202	012745	000373		MOV	#373, -(R5)	
3743	017206	005245			INC	-(R5)	
3744	017210	000000			HALT		:WRONG CC, IT SHOULD HAVE BEEN 5
3745	017212	000207		2\$:	RTS	PC	

3744					
3748	017214	100403		\$CC7:	BMI 1\$
3749	017216	001002			BNE 1\$
3750	01 220	102001			BVC 1\$
3751	017222	103404			BCS 2\$
3752	017224			1\$:	
3753	017224	012745	000374		MOV #374, -(R5)
3754	017230	005245			INC -(R5)
3755	017232	000000			HALT ;WRONG CC, IT SHOULD HAVE BEEN = 7
3756	017234	000207		2\$:	RTS PC
3757					
3758	017236	100002		\$CC10:	BPL 1\$
3759	017240	101401			BLOS 1\$
3760	017242	102004			BVC 2\$
3761	017244			1\$:	
3762	017244	012745	000375		MOV #375, -(R5)
3763	017250	005245			INC -(R5)
3764	017252	000000			HALT ;WRONG CC, IT SHOULD HAVE BEEN - 10
3765	017254	000207		2\$:	RTS PC
3766					
3767	017256	100003		\$CC11:	BPL 1\$
3768	017260	001402			BEQ 1\$
3769	017262	102401			BVS 1\$
3770	017264	103404			BCS 2\$
3771	017266			1\$:	
3772	017266	012745	000376		MOV #376, -(R5)
3773	017272	005245			INC -(R5)
3774	017274	000000			HALT ;WRONG CC, IT SHOULD HAVE BEEN - 11
3775	017276	000207		2\$:	RTS PC
3776					
3777	017300	100002		\$CC12:	BPL 1\$
3778	017302	101401			BLOS 1\$
3779	017304	102404			BVS 2\$
3780	017306			1\$:	
3781	017306	012745	000377		MOV #377, -(R5)
3782	017312	005245			INC -(R5)
3783	017314	000000			HALT ;WRONG CC, IT SHOULD HAVE BEEN = 12
3784	017316	000207		2\$:	RTS PC
3785					
3786	017320	100002		\$CC13:	BPL 1\$
3787	017322	003401			BLE 1\$
3788	017324	103404			BCS 2\$
3789	017326			1\$:	
3790	017326	012745	000400		MOV #400, -(R5)
3791	017332	005245			INC -(R5)
3792	017334	000000			HALT ;WRONG CC, IT SHOULD HAVE BEEN - 13
3793	017336	000207		2\$:	RTS PC
3794					
3795	017340	100003		\$CC17:	BPL 1\$
3796	017342	001002			BNE 1\$
3797	017344	102001			BVC 1\$
3798	017346	103404			BCS 2\$
3799	017350			\$:	
3800	017350	012745	000401		MOV #401, -(R5)
3801	017354	005245			INC -(R5)

CVKAAAC MACV11 30A(1052) 09-OCT-78 08:59 PAGE 103  
CVKAAAC.P11 09-OCT-78 08:58

ROUTINES TO CHECK CONDITION CODES

SEQ 0102

3802 017356 70000  
3803 017360 70007  
3804  
3805

28: HALT  
RTS PC  
.END

;WRONG CC, IT SHOULD HAVE BEEN 17













CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0108

SCC11	017256	1023	1082	1084	1105	1119	1147	1170	129*	1374	1478	1481	1511	1525
		1553	1576	1698	1781	1811	1868	1873	1901	1992	2443	2445	2480	2505
		2548	2571	2700	2785	2926	2937	2976	3025	3049	3179	3264	3298	3356
		3361	3397	3552	3572	3613	3622	3767#						
SCC12	017300	1197	1234	1256	1326	1603	1641	1663	1733	2602	2638	2664	2739	3080
		3117	3143	3218	3777#									
SCC13	017320	1102	1178	1508	1584	1955	2478	2579	2857	2973	3057	3463	3786#	
SCC17	017340	1989	3499	3549	3795#									
SCC2	017106	1383	1790	2794	3273	3708#								
SCC3	017126	1122	1528	2505	2996	3425	3717#							
SCC4	017150	1003	1005	1265	1287	1312	1371	1405	1407	1672	1694	1719	1778	1843
		1963	1985	2353	2355	2465	2488	2673	2696	2725	2782	2819	2821	3152
		3175	3204	3261	3328	3471	3474	3539	3727#					
SCC5	017172	1107	1117	1156	1335	1420	1484	1513	1523	1562	1742	1803	1862	2452
		2500	2559	2748	2935	2951	2981	2991	3035	3227	3290	3350	3412	3416
		3581	3737#											
SCC7	017214	1204	1232	1263	1285	1610	1639	1670	1692	1909	2609	2636	2671	2694
		3087	3115	3150	3173	3406	3748#							
\$CPUOP	000124	354#												
\$DEVCT	000106	345#												
\$ENDAD	016734	302	3650#											
\$ENV	000116	350#												
\$ENVM	000117	351#	3675											
\$ETABL	000116	349#	424											
\$ETEND	000126	361#	384											
\$FATAL	000100	342#												
\$HD =	000003	282	283											
\$HIBTS	000126	379#												
\$MAIL	000076	340#	380	384	426									
\$MBADR	000130	380#												
\$MSGAD	000112	347#												
\$MSGLG	000114	348#												
\$MSGTY	000076	341#												
\$PASS	000104	344#	3643*	3644										
\$PASTM	000134	382#												
\$SVPK =	001000	300#	305											
\$SWR =	160000	282	283#											
\$SWREG	000120	352#												
\$TESTN	000102	343#	435	903	977	3654*								
\$TN =	000124	282#	334#	442	447	448#	471	476	477#	493	498	499#	519	524
		525#	542	547	548#	569	574	575#	592	597	598#	618	623	624#
		646	651	652#	682	687	688#	736	741	742#	791	796	797#	839
		844	845#	956	961	962#	990	995	996#	1013	1018	1019#	1053	1058
		1059#	1088	1093	1094#	1125	1130	1131#	1160	1165	1166#	1185	1190	1191#
		1214	1219	1220#	1241	1246	1247#	1267	1272	1273#	1298	1303	1304#	1339
		1344	1345#	1392	1397	1398#	1424	1429	1430#	1451	1456	1457#	1494	1499
		1500#	1531	1536	1537#	1566	1571	1572#	1591	1596	1597#	1621	1626	1627#
		1648	1653	1654#	1674	1679	1680#	1705	1710	1711#	1746	1751	1752#	1791
		1796	1797#	1821	1826	1827#	1850	1855	1856#	1883	1888	1889#	1925	1930
		1931#	1966	1971	1972#	2004	2009	2010#	2054	2059	2060#	2096	2101	2102#
		2140	2145	2146#	2210	2215	2216#	2270	2275	2276#	2304	2309	2310#	2341
		2346	2347#	2377	2382	2383#	2410	2415	2416#	2466	2471	2472#	2523	2528
		2529#	2560	2565	2566#	2589	2594	2595#	2617	2622	2623#	2648	2653	2654#
		2674	2679	2680#	2710	2715	2716#	2749	2754	2755#	2804	2809	2810#	2831
		2836	2837#	2893	2898	2899#	2958	2963	2964#	3002	3007	3008#	3036	3041
		3042#	3067	3072	3073#	3096	3101	3102#	3127	3132	3133#	3153	3158	3159#

CROSS REFERENCE TABLE -- USER SYMBOLS

		3189	3194	3195#	3228	3233	3234#	3277	3282	3283#	3305	3310	3311#	3338
		3343	3344#	3368	3373	3374#	3435	3440	3441#	3481	3486	3487#	3521	3527
		3528#	3560	3565	3566#	3638	3639#							
\$TSTM	000132	381#												
\$UNIT	000110	346#												
\$UNITM	000136	383#												
\$USWR	000122	353#												
.	= 017362	290#	296	300	301#	303#	305#	319#	368	369#	371#	373#	385#	387#
		389#	391#	393#	395#	397#	399#	409#	411#	422#	432#	81U	1977#	1978#
		1980#	1981#	1988#	1989#	1991#	1992#	3535#	3536#	3538#	3539#	3548#	3549#	3551#
		3552#												
.TYPE	= 000004	335#	3647	3668										
..X	= 000126	368#	373											
..A	= 016352	1977#	1980#	1988#	1991#	3535#	3538#	3548#	3551#					
..B	= 016356	1977#	1978	1980#	1981	1988#	1989	1991#	1992	3535#	3536	3538#	3539	3548#
		3549	3551#	3552										
..C	= 000067	1977#	1980#	1988#	1991#	3535#	3538#	3548#	3551#					

COMMEN	1#														
ENDCOM	1#														
ERROR	313#	465	499	513	538	563	589	611	625	630	634	638	642	657	666
	671	677	693	702	707	712	721	726	733	747	752	757	762	770	774
	779	787	801	806	811	816	823	829	836	851	856	865	870	875	881
	891	900	905	910	916	919	925	931	939	946	952	983	997	1026	1032
	1037	1043	1050	1060	1071	1078	1095	1112	1132	1142	1150	1173	1182	1200	1209
	1228	1238	1248	1259	1281	1295	1305	1320	1329	1346	1356	1366	1377	1399	1415
	1437	1442	1448	1467	1474	1488	1501	1518	1538	1548	1556	1579	1588	1606	1615
	1635	1645	1655	1666	1688	1702	1712	1727	1736	1753	1763	1773	1784	1806	1815
	1836	1847	1877	1896	1904	1914	1922	1932	1942	1949	1958	1982	1996	2017	2025
	2038	2047	2077	2092	2119	2134	2155	2168	2178	2188	2206	2234	2244	2264	2291
	2300	2323	2331	2365	2371	2393	2398	2404	2417	2431	2438	2455	2460	2495	2511
	2530	2543	2552	2574	2583	2605	2614	2632	2642	2655	2667	2690	2704	2717	2733
	2742	2756	2767	2777	2788	2811	2846	2851	2870	2876	2886	2913	2920	2929	2943
	2955	2965	2986	3009	3020	3028	3052	3061	3083	3093	3111	3121	3134	3146	3169
	3183	3196	3212	3221	3235	3246	3256	3267	3293	3302	3321	3332	3365	3384	3391
	3400	3420	3429	3451	3457	3466	3478	3494	3502	3507	3515	3543	3556	3575	3584
	3593	3600	3606	3616	3626	3640	3694	3703	3712	3722	3732	3742	3753	3762	3772
	3781	3790	3800												
ESCAPE	1#														
GETPRI	1#														
GETSWR	1#														
HLT	310#	464	488	512	537	562	588	610	625	630	634	638	642	657	665
	671	676	693	701	707	712	721	726	732	747	751	757	762	770	773
	779	786	801	806	811	816	823	829	835	850	856	864	870	875	881
	890	910	915	919	925	931	939	946	951	982	997	1026	1032	1037	1043
	1049	1060	1071	1078	1095	1112	1132	1142	1150	1173	1181	1200	1208	1228	1237
	1248	1259	1281	1294	1305	1320	1329	1346	1356	1366	1377	1399	1415	1437	1442
	1447	1467	1474	1487	1501	1518	1538	1548	1556	1579	1587	1606	1614	1635	1644
	1655	1666	1688	1701	1712	1727	1736	1753	1763	1773	1784	1806	1814	1836	1846
	1876	1896	1904	1914	1921	1932	1942	1949	1958	1982	1995	2017	2025	2038	2046
	2076	2091	2118	2133	2155	2168	2177	2188	2205	2234	2243	2263	2290	2299	2323
	2330	2365	2370	2393	2398	2403	2417	2431	2438	2455	2460	2495	2510	2530	2543
	2552	2574	2582	2605	2613	2632	2641	2655	2667	2690	2703	2717	2733	2742	2756
	2767	2777	2788	2811	2846	2851	2870	2876	2885	2913	2920	2929	2943	2954	2965
	2986	3009	3020	3028	3052	3060	3083	3092	3111	3120	3134	3146	3169	3182	3196
	3212	3221	3235	3246	3256	3267	3293	3301	3321	3331	3364	3384	3391	3400	3420
	3428	3451	3457	3466	3477	3494	3502	3507	3514	3543	3555	3575	3584	3593	3600
	3606	3616	3625	3640	3693	3702	3711	3721	3731	3741	3752	3761	3771	3780	3789
	3799														
HLT1	312#	899	905												
MFPS	310#	1979	1990	3537	3550										
MTPS	308#	1976	1987	3534	3547										
MULT	1#														
NEWST	1#														
NEWTEST	314#	442	471	497	519	542	569	592	618	646	682	736	791	839	956
	990	1013	1053	1088	1125	1160	1185	1214	1241	1267	1298	1339	1392	1424	1451
	1494	1531	1566	1591	1621	1648	1674	1705	1746	1791	1821	1850	1883	1925	1966
	2004	2054	2096	2140	2210	2270	2304	2341	2377	2410	2466	2523	2560	2589	2617
	2648	2674	2710	2749	2804	2831	2893	2958	3002	3036	3067	3096	3127	3153	3189
	3228	3277	3305	3338	3368	3435	3481	3521	3560						
POP	1#														
PUSH	1#														
REPORT	1#														
SEQCHK	317#	446	475	497	523	546	573	596	622	650	686	740	795	843	960



.\$CMTA	1#
.\$DB20	1#
.\$DB20	1#
.\$DIV	1#
.\$EOP	1#
.\$ERRO	1#
.\$ERRT	1#
.\$MULT	1#
.\$POWE	1#
.\$RAND	1#
.\$RDDE	1#
.\$RDOC	1#
.\$RFAD	1#
.\$R2AZ	1#
.\$SAVE	1#
.\$SB2D	1#
.\$SB20	1#
.\$SCOP	1#
.\$SIZE	1#
.\$SUPR	1#
.\$TRAP	1#
.\$TYPB	1#
.\$TYPD	1#
.\$TYPE	1#
.\$TYPO	1#
.\$4OCA	1#
.1170	1#

. ABS. 017362 000

ERRORS DETECTED: 0

CVKAAC.BIN, CVKAAC.LST/CRF/SOL/NL:TOC=CVKAAC.SML, CVKAAC.P11  
 RUN-TIME: 13 18 1 SECONDS  
 RUN-TIME RATIO: 103/33=3.0  
 CORE USED: 32K (63 PAGES)