

LSI 11

BASIC INSTRUCTION TESTS MD-11-DVKAA-B

EP-DVKAA-B-DL

MAR 1978

COPYRIGHT © 75-78

digital

FICHE 1 OF 1

MADE IN USA

1
2
3
4
5
6
7
8
9
10

BASIC INSTRUCTION TESTS
MD-11-DVKAA-B

EP-DVKAA-B-DL
COPYRIGHT © 75-78
FICHE 1 OF 1

MAR 1978
digital
MADE IN USA

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 1
SEQ 0001

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DVKAA-B-D
PRODUCT NAME:	LSI-11 BASIC INSTRUCTION TESTS
DATE CREATED:	FEB 1978
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	PERVEZ A. ZAKI
MODIFIED BY:	BARRY SUSSMAN 01-NOV-77

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
DEC**

**PDP
DECUS**

UNIBUS
DECTAPE

MASSBUS

[illegible]

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
3. LOADING PROCEDURE
4. STARTING PROCEDURE
 - 4.1 STARTING ADDRESS
 - 4.2 PROGRAM AND/OR OPERATOR ACTION
5. OPERATING PROCEDURE
6. ERRORS
 - 6.1 ERROR REPORTING
 - 6.2 ERROR RECOVERY
7. RESTRICTIONS
8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 PASS COUNTER
 - 8.4 TEST NUMBER
 - 8.5 POWER FAIL
9. PROGRAM DESCRIPTION

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

96
97
98
99
100
101
102
103
104
105
106
107
108
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

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 \$TESTN (I.E. LOCATION 404) AND RESTART FROM 530 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 530. IN WHICH CASE THE PROGRAM WILL HALT
AT LOCATION 544 AND WILL WAIT FOR THE OPERATOR TO PLACE THE
STARTING ADDRESS OF THE DESIRED TEST IN PC (R7) AND TYPE A P.

4.2 PROGRAM AND/OR OPERATOR ACTION

- 1) LOAD PROGRAM INTO MEMORY USING ABS LOADER.
- 2) TYPE 200G [THERE ARE NO SWITCH SETTINGS AND THIS DIAGNOSTIC DOES NOT USE SOFTWARE SWITCH LOCATION \$SWREG]
- 3) 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 \$ENVN IS HIGH
- 5) A MINIMUM OF TWO PASSES SHOULD ALWAYS BE RUN.

5. OPERATING PROCEDURE

5.1 OPERATING MODE:

AN 8 BIT BYTE \$ENVN [I.E. LOCATION 421] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE \$ENVN HIGH, IN OTHER WORDS BY PLACING A 20000 IN LOCATION 420

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 402]. 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 404] 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.

199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254

- 6.2 ERROR RECOVERY
RESTART AT 200 OR 530 (SEE SEC 4.1)
7. RESTRICTIONS
NONE
8. MISCELLANEOUS
- 8.1 EXECUTION TIME
EXECUTION TIME OF THE DIAGNOSTIC IS LESS THAN A SECOND, FIRST
"END PASS" WILL BE TYPED OUT WITHIN A SECOND AND EVERY COSECUTIVE
"END PASS" WILL BE TYPED OUT WITHIN 20 SECONDS (SEE SEC 4.2)
- 8.2 STACK POINTER
STACK IS INITIALLY SET TO 530
- 8.3 PASS COUNT
A 16 BIT LOCATION "SPASS" (I.E. LOCATION 406) IS USED TO KEEP
PASS COUNT. IT CAN BE CLEARED BY RESTARTING THE PROGRAM AT 200
- 8.4 TEST NUMBER
A 16 BIT LOCATION "STESTN" (I.E. LOCATION 404) IS USED TO KEEP TRACK
OF THE TEST NUMBER, UPPER BYTE OF THIS LOCATION GIVES THE ITERATION
NUMBER AND THE LOWER BYTE THE TEST THAT WAS BEING EXECUTED
- 8.5 POWER FAIL
THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. TO USE,
START THE DIAGNOSTIC AS USUAL AND POWER DOWN THEN UP AT ANY TIME.
THE PROGRAM SHOULD TYPE "POWER" AND RESTART AT 530 WITH TEST # 0
HOWEVER THE DIAGNOSTIC WILL NOT RECOVER IF IT IS STORED IN A
MEMORY NOT CAPABLE OF HOLDING DATA WITH POWER DOWN
9. PROGRAM DESCRIPTION
THIS PROGRAM TESTS ALL THE BASIC INSTRUCTIONS OF THE LSI-11 (EXCEPT
TRAP-TYPE) WHICH INCLUDES CONTROL CHIP, DATA CHIP, MICROMS, PLA,

. GO1

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 6
DVKAAB.P11 25-OCT-77 13:09

SEQ 0006

255
256
257
258

.ENDR

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.

HO1

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 7
DVKAAB.P11 25-OCT-77 13:09

SEQ 0007

259

101

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 8
DVKAAB.P11 25-OCT-77 13:09

SEQ 0008

260

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 9
DVKAAB.P11 25-OCT-77 13:09

SEQ 0009

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283

```
.ABS  
;; LSI-11 MACRO INSTRUCTION EXERCISER  
.NLIST MC,MD,CND  
.LIST ME  
.TITLE DVKAAB  
;*COPYRIGHT (C) 1975, 1977  
;*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.  
;*:  
$TN=1  
$SWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
```

000001
160000

K01

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 10
 DVKAAB.P11 25-OCT-77 13:09

SEQ 0010

284	000000	. = 0	
285			
286		; *****	
287		; TRAP CATCHERS OF .+2 AND HALT IN LOCATIONS 0 THRU 776 [IT IS NLISTED]	
288			
289		.SBTTL ACT11 HOOKS	
290		; *****	
291		; HOOKS REQUIRED BY ACT11	
292		\$SVPC = .	; SAVE PC
293	001000	. = 46	
294	000046	\$ENDAD	; 1) SET LOC. 46 TO ADDRESS OF \$ENDAD IN .SEOP
295	000052	. = 52	
296	000000	.WORD 0	; 2) SET LOC. 52 TO ZERO
297	001000	. = \$SVPC	; RESTORE PC
298			
299			
300			
301			
302			
303			
304			
305			
306			
307			
308			
309			
310			
311			
312			
313	000400	. = 400	
314	000000	R0	= %0
315	000001	R1	= %1
316	000002	R2	= %2
317	000003	R3	= %3
318	000004	R4	= %4
319	000005	R5	= %5
320	000006	R6	= %6
321	000006	SP	= %6
322	000007	PC	= %7
323	000254	CLNZ	= 254
324	000001	ERRNM	= 1
325	000260	NOP1	= 260
326	000263	SEVC	= 263
327	000273	SENV	= 273
328	000000	\$TN	= 0
329	000004	.TYPE	= IOT

```

330 .SBTTL APT MAILBOX-ETABLE
331
332 ;*****
333 .EVEN
334 $MAIL: APT MAILBOX
335 $MSGTY: .WORD AMSGTY ;; MESSAGE TYPE CODE
336 $FATAL: .WORD AFATAL ;; FATAL ERROR NUMBER
337 $TESTN: .WORD ATESTN ;; TEST NUMBER
338 $PASS: .WORD APASS ;; PASS COUNT
339 $DEVCT: .WORD ADEVCT ;; DEVICE COUNT
340 $UNIT: .WORD AUNIT ;; I/O UNIT NUMBER
341 $MSGAD: .WORD AMSGAD ;; MESSAGE ADDRESS
342 $MSGLG: .WORD AMSGLG ;; MESSAGE LENGTH
343 $ETABLE: APT ENVIRONMENT TABLE
344 $ENV: .BYTE AENV ENVIRONMENT BYTE
345 $ENVN: .BYTE AENVN ENVIRONMENT MODE BITS
346 $SWREG: .WORD ASWREG APT SWITCH REGISTER
347 $USWR: .WORD AUSWR USER SWITCHES
348 $CPUOP: .WORD ACPUOP CPU TYPE, OPTIONS
349 ;
350 ; BIT 15-11=CPU TYPE
351 ; 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
352 ; 11/70=06, PDQ=07, Q=10
353 ;
354 ; BIT 10=REAL TIME CLOCK
355 ; BIT 9=FLOATING POINT PROCESSOR
356 ; BIT 8=MEMORY MANAGEMENT
357 $ETEND:
358 .MEXIT
359 .SBTTL APT PARAMETER BLOCK
360 ;*****
361 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
362 ;*****
363 .SX= .SAVE CURRENT LOCATION
364 .=24 .SET POWER FAIL TO POINT TO START OF PROGRAM
365 200 .FOR APT START UP
366 .=44 .POINT TO APT INDIRECT ADDRESS PNTR.
367 $APTHDR .POINT TO APT HEADER BLOCK
368 .=.SX .RESET LOCATION COUNTER
369 ;*****
370 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
371 ;INTERFACE SPEC.
372 $APTHD:
373 $HIBTS: .WORD 0 ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
374 $MBADR: .WORD $MAIL ;; ADDRESS OF APT MAILBOX (BITS 0-15)
375 $TSTM: .WORD 1 ;; RUN TIM OF LONGEST TEST
376 $PASTM: .WORD 1 ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
377 $UNITM: .WORD 1 ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
378 .WORD $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)

```


DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 12
 DVKAAB.P11 25-OCT-77 13:09 APT PARAMETER BLOCK

SEQ 0012

379		000430				=SAPTHD	
380	000430					ADR:	
381		000432				.=ADR+2	
382	000432					ADR1:	
383		000434				.=ADR1+2	
384	000434					ADR2:	
385		000436				.=ADR2+2	
386	000436					DUMMY:	
387		000440				.=DUMMY+2	
388	000440					TEMP:	
389		000442				.=TEMP+2	
390	000442					TEMP1:	
391		000444				.=TEMP1+2	
392	000444					TEMP2:	
393		000446				.=TEMP2+2	
394	000446	177564				TPS:	177564
395	000450	177566				TPB:	177566
396	000452	006402				MARK:	2
397	000454	005015	042440	042116	MARK2:		
398	000462	050040	051501	000123	ENDPAS:	.ASCIZ	<15><12>" END PASS"
399						.EVEN	
400	000470	005015	047520	042527	POWER:	.ASCIZ	<15><12>/POWER/
401	000476	000122					
402						.EVEN	
403	000500	000023				.BLKW	19.
404							
405		000020				.=20	
406	000020	017064				TYPE	
407	000022	000000				0	
408							
409							
410							

; OUTPUT TTY STATUS REGISTER
; OUTPUT BUFFER

```

411                                     ;STARTING OF THE PROGRAM
412                                     ;-----
413
414
415
416                                     . =200
417 000200 012737 017034 000024      MOV      #PWKDN,2#24      ; SERVICE POWER DOWN ROUTINE ON ANY FUTURE POWER DOWN
418 000206 012700 000420              MOV      #SETABLE,R0
419 000212 005040                      2$:  CLR      -(R0)          ; START CLEANING THE STACK
420 000214 020027 000400              CMP      R0,#$MAIL      ; FOR INITIALIZATION
421 000220 101374                      BHI      2$
422 000222 000167 000302              JMP      START
423                                     ;-----
424
425
426                                     . =530
427
428 000530 012706 000530      START:  MOV      #START,SP      ; SET THE STACK POINTER
429 000534 012705 000404      MOV      #TESTN,R5          ; PLACE THE ADDRESS OF LOCATION $TESTN IN R5
430 000540 005715              TST      (R5)              ; CHECK THE SEQUENCE COUNTER
431 000542 001401              BEQ      NOBIT              ; IF THIS IS THE STARTING OF THE TEST THEN
432                                     ; GO TO NOBIT TEST
433 000544 000000              HALT                        ; OTHERWISE HALT AND WAIT FOR THE OPERATOR
434                                     ; TO START AT THE PROPER TEST NUMBER
435

```


DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 14
 DVKAAB.P11 25-OCT-77 13:09 TO

CHECK BRANCH INSTRUCTIONS WITH ZERO CONDITION CODES

SEQ 0014

```

436 ;*****
437 ;*TEST: 0 CHECK BRANCH INSTRUCTIONS WITH ZERO CONDITION CODES
438 ;*****
439
440 NOBIT:
441 000546 021527 000000      CMP      (R5), #0
442 000552 001017      BNE      CC0      ; IF IN WRONG SEQUENCE GO TO HALT AT END OF THE TEST
443 000554 005215      INC      (R5)      ; ZERO CONDITION CODES, NZVC=0000
444 000556 000257      CCC
445 000560 103414      BCS      CC0
446 000562 102413      BVS      CC0
447 000564 001412      BEQ      CC0
448 000566 100411      BMI      CC0
449 000570 000260      NOP1      ; CHECK NOP1 INSTRUCTION I.E. OP-CODE 260
450 000572 103407      BCS      CC0
451 000574 102406      BVS      CC0
452 000576 001405      BEQ      CC0
453 000600 100404      BMI      CC0
454 000602 002403      BLT      CC0
455 000604 003402      BLE      CC0
456 000606 101401      BLOS     CC0
457 000610 101004      BHI      ENDCC0
458 000612
459 000612 012745 000001      MOV      #1, -(R5)
460 000616 005245      INC      -(R5)
461 000620 000000      HALT
462 000622 102000      ENDCC0: BVC      NBIT      ; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE
463
464 ;*****
465 ;*TEST: 1 CHECK BRANCH INSTRUCTIONS WITH N BIT SET
466 ;*****
467
468 NBIT:
469 000624 021527 000001      CMP      (R5), #1
470 000624 001012      BNE      CC1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
471 000630 005215      INC      (R5)      ; NBIT IS SET, NZVC=1000
472 000632 000270      SEN
473 000634 000270      BPL      CC1
474 000636 100007      BEQ      CC1
475 000640 001406      BGE      CC1
476 000642 002005      BGT      CC1
477 000644 003004      BCS      CC1
478 000646 103403      BLOS     CC1
479 000650 101402      BLO      CC1
480 000652 103401      BLE      ENDCC1
481 000654 003404      CC1:
482 000656      MOV      #2, -(R5)
483 000656 012745 000002      INC      -(R5)
484 000662 005245      HALT
485 000664 000000      ENDCC1: BNE      VBIT      ; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE
486 000666 001000

```

```

487
488
489
490
491 000670
492 000670 021527 000002
493 000674 001014
494 000676 005215
495 000700 000270
496 000702 000262
497 000704 102010
498 000706 001407
499 000710 100006
500 000712 103405
501 000714 002404
502 000716 003403
503 000720 101402
504 000722 103401
505 000724 003004
506 000726
507 000726 012745 000003
508 000732 005245
509 000734 000000
510 000736 002000
511
512
513
514
515
516
517 000740
518 000740 021527 000003
519 000744 001013
520 000746 005215
521 000750 000270
522 000752 000262
523 000754 000261
524 000756 001406
525 000760 100005
526 000762 102004
527 000764 002403
528 000766 003402
529 000770 101001
530 000772 002004
531 000774
532 000774 012745 000004
533 001000 005245
534 001002 000000
535

```

```

*****
;TEST: 2      CHECK BRANCH INSTRUCTIONS WITH N&V BITS SET
*****
VBIT:
      CMP      (R5), #2
      BNE      CC2
1$:   INC      (R5)
      SEN
      SEV
      BVC      CC2
      BEQ      CC2
      BPL      CC2
      BCS      CC2
      BLT      CC2
      BLE      CC2
      BLOS     CC2
      BLO      CC2
      BGT      ENDCC2
      CC2:     MOV      #3, -(R5)
      INC      -(R5)
      HALT
ENDCC2: BGE      CBIT
; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
; V AND N BIT SET, NZVC = 1010
; ONE OF THE ABOVE BRANCHES FAILED OR WRONG SEQUENCE
*****
;TEST: 3      CHECK BRANCH INSTRUCTIONS WITH N,V&C BITS SET
*****
CBIT:
      CMP      (R5), #3
      BNE      CC3
1$:   INC      (R5)
      SEN
      SEV
      SEC
      BEQ      CC3
      BPL      CC3
      BVC      CC3
      BLT      CC3
      BLE      CC3
      BHI      CC3
      BGE      ZBIT
      CC3:     MOV      #4, -(R5)
      INC      -(R5)
      HALT
; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
; C, V, AND N BITS ARE SET, NZVC=1011
; ONE OF THE ABOVE BRANCHES FAILED
; OR WRONG SEQUENCE

```



```

586
587
588
589
590 001122
591 001122 021527 000006
592 001126 001013
593 001130 005215
594 001132 000277
595 001134 000241
596 001136 103407
597 001140 000242
598 001142 102405
599 001144 000244
600 001146 001403
601 001150 000250
602 001152 100401
603 001154 101004
604 001156
605 001156 012745 000007
606 001162 005245
607 001164 000000
608 001166 100000
609
610
611
612
613
614
615
616 001170
617 001170 021527 000007
618 001174 001404
619 001176 012745 000010
620 001202 005245
621 001204 000000
622 001206 005215
623 001210 000416
624 001212 012745 000011
625 001216 005245
626 001220 000000
627 001222 000404
628 001224 012745 000012
629 001230 005245
630 001232 000000
631 001234 000411
632 001236 012745 000013
633 001242 005245
634 001244 000000
635 001246 000765
636 001250 012745 000014
637 001254 005245
638 001256 000000
639 001260 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
      CLC
      BCS      CC5          ; NZVC=1111
                              ; NZVC=1110
      CLV
      BVS      CC5          ; NZVC=1100
      CLZ
      BEQ      CC5          ; NZVC=1000
      CLN
      BMI      CC5          ; NZVC=0000
      BHI      ENDCC5
CC5:
      MOV      #7, -(R5)
      INC      -(R5)
      HALT
ENDCC5: BPL      BRANCH          ; ONE OF THE ABOVE CLEARS FAILED OR WRONG SEQUENCE

```

```

*****
;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
2$:
      BR       3$          ; FORWARD BRANCH FAILED
      MOV      #12, -(R5)
      INC      -(R5)
      HALT
3$:
      BR       5$          ; FORWARD BRANCH FAILED
      MOV      #13, -(R5)
      INC      -(R5)
      HALT
4$:
      BR       2$          ; FORWARD BRANCH FAILED
      MOV      #14, -(R5)
      INC      -(R5)
      HALT
5$:
      BR       1$          ; BACKWARD BRANCH FAILED
      JMP1

```

```

640      ;*****
641      ;*TEST: 10      CHECK JMP INSTRUCTIONS FOR MODE 1
642      ;*****
643
644      JMP1:
645      001262      021527      000010      CMP      (R5), #10
646      001262      001033      BNE      ENDJP1      ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF THE TEST
647      001270      005215      1$:      INC      (R5)
648      001272      012700      001312      MOV      #2$, R0      ; TEST JUMP INSTRUCTION MODE 1
649      001276      000277      SCC
650      001300      000110      JMP      (R0)
651      001302      012745      000015      MOV      #15, -(R5)
652      001306      005245      INC      -(R5)
653      001310      000000      HALT      ; JUMP INSTRUCTION FAILED
654      001312
655      001312      100003      2$:      BPL      3$
656      001314      001002      BNE      3$
657      001316      102001      BVC      3$
658      001320      103404      BCS      4$
659      001322
660      001322      012745      000016      3$:      MOV      #16, -(R5)
661      001326      005245      INC      -(R5)
662      001330      000000      HALT      ; WRONG CC
663      001332      020027      001312      4$:      CMP      R0, #2$
664      001336      001404      BEQ      5$      ; CONTINUE IF R0 IS OK
665      001340      012745      000017      MOV      #17, -(R5)
666      001344      005245      INC      -(R5)
667      001346      000000      HALT
668      001350      012700      001366      5$:      MOV      #JMP2, R0      ; TEST JUMP INSTRUCTION MODE 1
669      001354      000110      JMP      (R0)
670      001356
671      001356      012745      000020      ENDJP1:      MOV      #20, -(R5)
672      001362      005245      INC      -(R5)
673      001364      000000      HALT      ; JUMP INSTRUCTION FAILED OR WRONG SEQUENCE
674
675      ;*****
676      ;*TEST: 11      CHECK JMP INSTRUCTIONS FOR MODES 2 AND 3
677      ;*****
678
679
680      JMP2:
681      001366      021527      000011      CMP      (R5), #11
682      001372      001073      BNE      ENDJP3      ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF TEST
683      001374      005215      INC      (R5)
684      001376      012700      001416      MOV      #3$, R0      ; TEST JUMP INSTRUCTION MODE 2
685      001402      000277      SCC
686      001404      000120      JMP      (R0)+
687      001406      012745      000021      MOV      #21, -(R5)
688      001412      005245      INC      -(R5)
689      001414      000000      HALT      ; JUMP INSTRUCTION FAILED
690      001416
691      001416      100003      3$:      BPL      4$
692      001420      001002      BNE      4$
693      001422      102001      BVC      4$
694      001424      103404      BCS      5$
695      001426
4$:

```


G02

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 19
 DVKAAB.P11 25-OCT-77 13:09 T11 CHECK JMP INSTRUCTIONS FOR MODES 2 AND 3

SEQ 0019

696	001426	012745	000022			MOV	#22, -(R5)	
697	001432	005245				INC	-(R5)	
698	001434	000000				HALT		; WRONG CC
699	001436	020027	001420		5\$:	CMP	RO, #3\$+2	; IS THERE AUTO INC.?
700	001442	001404				BEQ	6\$	
701	001444	012745	000023			MOV	#23, -(R5)	
702	001450	005245				INC	-(R5)	
703	001452	000000				HALT		; MODE 2 FAILED FOR JMP INSTRUCTION
704	001454	012700	001472		6\$:	MOV	#JMP3, RO	; TEST JUMP INSTRUCTION MODE 2
705	001460	000120				JMP	(RO)+	
706	001462	012745	000024			MOV	#24, -(R5)	
707	001466	005245				INC	-(R5)	
708	001470	000000				HALT		; JUMP INSTRUCTION FAILED
709								
710	001472	012767	001526	176740	JMP3:	MOV	#3\$, TEMP	; TEST JUMP INSTRUCTION MODE 3
711	001500	012767	001546	176734		MOV	#4\$, TEMP+2	
712	001506	012700	000440			MOV	#TEMP, RO	
713	001512	000277				SCC		
714	001514	000130				JMP	2(RO)+	
715	001516	012745	000025			MOV	#25, -(R5)	
716	001522	005245				INC	-(R5)	
717	001524	000000				HALT		; JUMP INSTRUCTION FAILED
718	001526	027067	000000	000012	3\$:	CMP	2(RO), 4\$	; IS THERE AUTO INC.?
719	001534	001404				BEQ	4\$	
720	001536	012745	000026			MOV	#26, -(R5)	
721	001542	005245				INC	-(R5)	
722	001544	000000				HALT		; JMP INSTRUCTION FAILED IN MODE 2
723	001546	012767	001572	176664	4\$:	MOV	#JMP4, TEMP	; TEST JUMP INSTRUCTION MODE 3
724	001554	012700	000440			MOV	#TEMP, RO	
725	001560	000130				JMP	2(RO)+	
726	001562				ENDJP3:			
727	001562	012745	000027			MOV	#27, -(R5)	
728	001566	005245				INC	-(R5)	
729	001570	000000				HALT		; JUMP ERROR OR WRONG SEQUENCE

H02

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 20
 DVKAAB.P11 25-OCT-77 13:09 T12

TEST JUMP INSTRUCTION FOR MODE 4, 5

SEQ 0020

```

730 ;*****
731 ;*TEST: 12 TEST JUMP INSTRUCTION FOR MODE 4, 5
732 ;*****
733
734 001572 JMP4: CMP (R5), #12
735 001572 021527 000012 BNE ENDJP5 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
736 001576 001075 INC (R5)
737 001600 005215 MOV #35, R0 ; TEST JUMP INSTRUCTION MODE 4
738 001602 012700 001624 SCC
739 001606 000277 JMP -(R0)
740 001610 000140 MOV #30, -(R5)
741 001612 012745 000030 INC -(R5)
742 001616 005245 HALT ; JUMP INSTRUCTION FAILED
743 001620 000000 BR 45 ; JUMP SHOULD LAND HERE
744 001622 000404
745 001624
746 001624 012745 000031 MOV #31, -(R5)
747 001630 005245 INC -(R5)
748 001632 000000 HALT ; NO AUTO DECREMENT FROM JMP4
749 001634 022700 001622 45: CMP #35-2, R0 ; CHECK R0
750 001640 001404 BEQ 55
751 001642 012745 000032 MOV #32, -(R5)
752 001646 005245 INC -(R5)
753 001650 000000 HALT
754 001652 012700 001672 55: MOV #JMP5+2, R0 ; TEST JUMP INSTRUCTION MODE 4
755 001656 000140 JMP -(R0)
756 001660 012745 000033 MOV #33, -(R5)
757 001664 005245 INC -(R5)
758 001666 000000 HALT ; JUMP INSTRUCTION FAILED
759
760 001670 012767 001722 176544 JMP5: MOV #35, TEMP1 ; TEST JUMP INSTRUCTION MODE 5
761 001676 012700 000442 MOV #TEMP1, R0
762 001702 012767 001732 176530 MOV #45, TEMP1-2 ;
763 001710 000150 JMP 2-(R0)
764 001712 012745 000034 MOV #34, -(R5)
765 001716 005245 INC -(R5)
766 001720 000000 HALT ; JUMP INSTRUCTION FAILED
767 001722
768 001722 012745 000035 35: MOV #35, -(R5)
769 001726 005245 INC -(R5)
770 001730 000000 HALT ; ERROR, NO AUTO DECREMENT
771 001732 022700 000440 45: CMP #TEMP1-2, R0 ; CHECK R0
772 001736 001404 BEQ 55
773 001740 012745 000036 MOV #36, -(R5)
774 001744 005245 INC -(R5)
775 001746 000000 HALT ; JUMP ONSTRUCTION FAILED IN MODE 5
776 001750 012767 001722 176464 55: MOV #35, TEMP1 ; TEST JUMP INSTRUCTION MODE 5
777 001756 012700 000442 MOV #TEMP1, R0
778 001762 012767 002002 176450 MOV #JMP6, TEMP1-2 ;
779 001770 000150 JMP 2-(R0)
780 001772
781 001772 012745 000037 ENDJP5: MOV #37, -(R5)
782 001776 005245 INC -(R5)
783 002000 000000 HALT ; JUMP ERROR OR WRONG SEQUENCE
784
785 ;*****

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 21
 DVKAAB.P11 25-OCT-77 13:09 T13

TEST JMP INSTRUCTION FOR MODE 6 AND 7

SEQ 0021

```

786 ;*TEST: 13      TEST JMP INSTRUCTION FOR MODE 6 AND 7
787 ;*****
788
789 JMP6:
790 002002 021527 000013      CMP      (R5), #13
791 002006 001071      BNE      ENDJP7      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
792 002010 005215      INC      (R5)
793 002012 012703 002040      MOV      #15+6, R3
794 002016 000163 177772      JMP      -6(R3)
795 002022 012745 000040      MOV      #40, -(R5)
796 002026 005245      INC      -(R5)
797 002030 000000      HALT      ; JUMP INSTRUCTION FAILED
798 002032 020327 002040 1$:      CMP      R3, #15+6      ; CHECK R3
799 002036 001404      BEQ      2$
800 002040 012745 000041      MOV      #41, -(R5)
801 002044 005245      INC      -(R5)
802 002046 000000      HALT      ; WRONG VALUE IN REGISTER AFTER JUMP MODE 6
803                                     ; OR JUMP INSTRUCTION FAILED IN MODE 6
804 002050 000167 000010 2$:      JMP      3$-4(PC)      ; TEST JUMP INSTRUCTION MODE 6
805 002054 012745 000042      MOV      #42, -(R5)
806 002060 005245      INC      -(R5)
807 002062 000000      HALT      ; JUMP INSTRUCTION FAILED
808 002064 012703 002104 3$:      MOV      #JMP7, R3      ; JUMP SHOULD LAND HERE
809 002070 000163 000000      JMP      0(R3)
810 002074 012745 000043      MOV      #43, -(R5)
811 002100 005245      INC      -(R5)
812 002102 000000      HALT      ; JUMP INSTRUCTION FAILED
813
814 002104 012703 000440      JMP7:      MOV      #TEMP, R3
815 002110 012713 002130      MOV      #15, (R3)
816 002114 000173 000000      JMP      2(R3)
817 002120 012745 000044      MOV      #44, -(R5)
818 002124 005245      INC      -(R5)
819 002126 000000      HALT      ; JUMP INSTRUCTION FAILED
820 002130 012713 002154 1$:      MOV      #35, (R3)      ; TEST JUMP INSTRUCTION MODE 7
821 002134 012700 000434      MOV      #TEMP-4, R0
822 002140 000170 000004      JMP      24(R0)
823 002144 012745 000045      MOV      #45, -(R5)
824 002150 005245      INC      -(R5)
825 002152 000000      HALT      ; JUMP INSTRUCTION FAILED
826 002154 012767 002202 176256 3$:      MOV      #JSRTST, TEMP      ; CONTINUE
827 002162 012700 000440      MOV      #TEMP, R0
828 002166 000170 000000      JMP      20(R0)
829
830 002172 012745 000046      ENDJP7:      MOV      #46, -(R5)
831 002176 005245      INC      -(R5)
832 002200 000000      HALT      ; JUMP ERROR OR SEQUENCE ERROR

```



```

833      ;*****
834      ;*TEST: 14      CHECK JSR AND MARK INSTRUCTIONS
835      ;*****
836
837      JSRTST:
838      002202      021527      000014      CMP      (R5),#14
839      002206      001177      BNE      ENDJSR      ; IF IN WRONG SEQUENCE GO TO HALT AT THE END OF THE TEST
840      002210      005215      INC      (R5)
841      002212      012706      000530      MOV      #START,SP      ; SET UP STACK POINTER.
842      002216      000277      SCC
843      002220      004767      000026      JSR      PC,3$
844      002224
845      002224      012745      000047      1$:      MOV      #47,-(R5)
846      002230      005245      INC      -(R5)
847      002232      000000      HALT
848      002234      022706      000530      2$:      CMP      #START,SP      ; JSR INSTRUCTION FAILED
849      002240      001441      BEQ      JSRM      ; HAS SP BEEN RESTORED?
850      002242      012745      000050      MOV      #50,-(R5)
851      002246      005245      INC      -(R5)
852      002250      000000      HALT      ; SP WAS NOT RESTORED BY RTS INSTRUCTION
853      002252
854      002252      100003      3$:      BPL      4$
855      002254      001002      BNE      4$
856      002256      102001      BVC      4$
857      002260      103404      BCS      5$
858      002262
859      002262      012745      000051      4$:      MOV      #51,-(R5)
860      002266      005245      INC      -(R5)
861      002270      000000      HALT
862      002272      022706      000526      5$:      CMP      #START-2,SP      ; WRONG CC
863      002276      001404      BEQ      6$      ; WAS THE SP EFFECTED?
864      002300      012745      000052      MOV      #52,-(R5)
865      002304      005245      INC      -(R5)
866      002306      000000      HALT
867      002310      022716      002224      6$:      CMP      #1$, (SP)      ; WRONG SP AFTER EXECUTION OF JSR INSTRUCTION
868      002314      001404      BEQ      7$      ; IS THE RETURN ADDRESS =1$
869      002316      012745      000053      MOV      #53,-(R5)
870      002322      005245      INC      -(R5)
871      002324      000000      HALT
872
873      002326      012716      002234      7$:      MOV      #2$, (SP)
874      002332      000207      RTS      PC
875      002334      012745      000054      MOV      #54,-(R5)
876      002340      005245      INC      -(R5)
877      002342      000000      HALT
878      002344      010546      JSRM:      MOV      R5,-(SP)      ; RTS INSTRUCTION FAILED
879      002346      016746      176064      MOV      DUMMY,-(SP)      ; MOV R5 TO STACK
880      002352      016746      176060      MOV      DUMMY,-(SP)
881      002356      016746      176070      MOV      MARK2,-(SP)
882      002362      010503      MOV      R5,R3      ; STORE MARK 2 ON THE STACK.
883      002364      004467      000130      MOV      R4,10$      ; SAVE R5 IN R3
884      002370
885      002370      012745      000055      1$:      MOV      #55,-(R5)
886      002374      005245      INC      -(R5)
887      002376      000000      HALT      ; JSR INSTRUCTION FAILED
888      002400      2$:

```

889	002400	100003		BPL	3\$	
890	002402	001002		BNE	3\$	
891	002404	102001		BVC	3\$	
892	002406	103404		BCS	4\$	
893	002410		3\$:			
894	002410	012743	000056	MOV	#56, -(R3)	
895	002414	005243		INC	-(R3)	
896	002416	000000		HALT		; WRONG CC
897	002420	022705	000404	4\$:	CMP	#\$TESTN, R5
898	002424	001404		BEQ	5\$	
899	002426	012743	000057	MOV	#57, -(R3)	
900	002432	005243		INC	-(R3)	
901	002434	000000		HALT		; MARK INSTRUCTION FAILED
902	002436	022706	000530	5\$:	CMP	#START, SP
903	002442	001404		BEQ	6\$	
904	002444	012745	000060	MOV	#60, -(R5)	
905	002450	005245		INC	-(R5)	
906	002452	000000		HALT		; MARK INSTRUCTION FAILED
907	002454	012701	002562	6\$:	MOV	#12\$, R1
908	002460	004011		JSR	RO, (R1)	; PLACE THE ADDRESS OF 12\$ IN R1
909	002462			7\$:		; GO TO TAG 12\$
910	002462	012745	000061	MOV	#61, -(R5)	
911	002466	005245		INC	-(R5)	
912	002470	000000		HALT		; JSR INSTRUCTION FAILED
913	002472	012745	000062	MOV	#62, -(R5)	
914	002476	005245		INC	-(R5)	
915	002500	000000		HALT		; RTS BROUGHT THE PROGRAM BACK IN WRONG
916						; PLACE
917	002502	022706	000530	8\$:	CMP	#START, SP
918	002506	001443		BEQ	REGS	
919	002510	012745	000063	MOV	#63, -(R5)	
920	002514	005245		INC	-(R5)	
921	002516	000000		HALT		; STACK POINTER WAS NOT RESET
922						; IS THE RETURN ADDRESS =1\$?
923	002520	020427	002370	10\$:	CMP	R4, #1\$
924	002524	001404		BEQ	11\$	
925	002526	012745	000064	MOV	#64, -(R5)	
926	002532	005245		INC	-(R5)	
927	002534	000000		HALT		; WRONG RETURN ADDRESS IN UNKAGE REGISTER R4
928	002536	010605		11\$:	MOV	SP, R5
929	002540	005725		TST	(R5)+	; SET UP ADDRESS IN R5 AT MARK 2 INSTRUCTION
930	002542	012716	002400	MOV	#2\$, (SP)	; SET RETURN ADDRESS =2\$
931	002546	000277		SCC		
932	002550	000205		RTS	R5	; RETURN USING R5 AND IN-TURN USING MARK INSTRUCTION
933	002552	012745	000065	MOV	#65, -(R5)	
934	002556	005245		INC	-(R5)	
935	002560	000000		HALT		; RTS INSTRUCTION FAILED
936						
937	002562	020027	002462	12\$:	CMP	RO, #7\$
938						; DOES RO CONTAIN THE RETURN ADDRESS?
939	002566	001404		BEQ	13\$	
940	002570	012745	000066	MOV	#66, -(R5)	
941	002574	005245		INC	-(R5)	
942	002576	000000		HALT		; WRONG RETURN ADDRESS IN LINKAGE REGISTER RO
943	002600	012700	002502	13\$:	MOV	#8\$, RO
944	002604	000200		RTS	RO	; SET RETURN ADDRESS AT 8\$

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 24
 DVKAAB.P11 25-OCT-77 13:09 T14 CHECK JSR AND MARK INSTRUCTIONS

SEQ 0024

```

945 002606
946 002606 012745 000067
947 002612 005245
948 002614 000000
949
950
951
952
953
954 002616
955 002616 021527 000015
956 002622 001034
957 002624 005215
958 002626 010667 175606
959 002632 012700 000001
960 002636 012701 000004
961 002642 012702 000020
962 002646 012703 000100
963 002652 012704 000400
964 002656 005006
965 002660 060006
966 002662 060106
967 002664 060206
968 002666 060306
969 002670 060406
970 002672 060506
971 002674 022706 001131
972 002700 001003
973 002702 016706 175532
974 002706 000406
975 002710 016706 175524
976 002714
977 002714 012745 000070
978 002720 005245
979 002722 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


```

980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035

```

002724	021527	000016
002724	001404	
002730	012745	000071
002732	005245	
002736	000000	
002740	005215	
002742	000277	
002744	105000	
002746	004737	017230
002750	105700	
002754	004737	017230
002756	112701	000377
002762	004737	017316
002766	105701	
002772	004737	017316
003000		
003000	021527	000017
003004	001051	
003006	005215	
003010	000277	
003012	152702	000377
003016	004737	017336
003022	122702	000377
003026	001404	
003030	012745	000072
003034	005245	
003036	000000	
003040	112700	000077
003044	120002	
003046	100004	
003050	012745	000073
003054	005245	
003056	000000	
003060	120200	
003062	100404	
003064	012745	000074
003070	005245	
003072	000000	
003074	112702	000377
003100	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$
      MOV      #71,-(R5)
      INC      -(R5)
      HALT
2$:    INC      (R5)
      SCC
      CLRB     R0
      JSR      PC,#$SCC4
      TSTB     R0
      JSR      PC,#$SCC4
      MOVB     #377,R1
      JSR      PC,#$SCC10
      TSTB     R1
      JSR      PC,#$SCC10
; IF IN WRONG SEQUENCE GO TO HLT BELOW
; PROGRAM IS IN WRONG SEQUENCE
; CLEAR THE REGISTER
; CHECK FOR CC = 4
; CHECK IT
; CHECK FOR CC = 4
; LOAD THE REGISTER
; CHECK FOR CC = 10
; CHECK IT
; CHECK FOR CC = 10

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

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 26
 DVKAAB.P11 25-OCT-77 13:09 T17

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB

SEQ 0026

1036	003104	001404		BEQ	SS	; CONTINUE IF OK
1037	003106	012745	000075	MOV	#75, -(R5)	
1038	003112	005245		INC	-(R5)	
1039	003114	000000		HALT		
1040	003116	112702	000377	SS:	MOV8	#377, R2
1041	003122	120227	000377	CMPB	R2, #377	; CMPB BECAME CMP INSTRUCTION
1042	003126	001404		BEQ	BIC60	; LOAD REGISTER, SIGN EXTEND
1043	003130					; CHECK IF BYTE INSTRUCTION
1044	003130	012745	000076	ECMPB0:	MOV	#76, -(R5)
1045	003134	005245		INC	-(R5)	
1046	003136	000000		HALT		; CONTINUE IF OK
						; WRONG CC OR WRONG SEQUENCE

1047
1048
1049
1050
1051 003140
1052 003140 021527 000020
1053 003144 001404
1054 003146 012745 000077
1055 003152 005245
1056 003154 000000
1057 003156 005215
1058 003160 112703 000377
1059 003164 112700 000252
1060 003170 000277
1061 003172 140003
1062 003174 004737 017146
1063 003200 130003
1064 003202 001404
1065 003204 012745 000100
1066 003210 005245
1067 003212 000000
1068 003214 132703 000125
1069 003220 004737 017146
1070 003224 150003
1071 003226 100404
1072 003230 012745 000101
1073 003234 005245
1074 003236 000000
1075 003240 142703 000177
1076 003244 004737 017336
1077 003250 132703 000377
1078 003254 004737 017336
1079
1080
1081
1082
1083
1084
1085
1086 003260
1087 003260 021527 000021
1088 003264 001404
1089 003266 012745 000102
1090 003272 005245
1091 003274 000000
1092 003276 005215
1093 003300 112704 000177
1094 003304 000261
1095 003306 105204
1096 003310 004737 017400
1097 003314 112704 000376
1098 003320 105204
1099 003322 004737 017336
1100 003326 105204
1101 003330 004737 017252
1102 003334 105204

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

BICB0:

CMP (R5), #20
BEQ 2S ; IF IN WRONG SEQUENCE GO TO HLT BELOW
MOV #77, -(R5)
INC -(R5)
HALT ; PROGRAM IS IN WRONG SEQUENCE
2S: INC (R5)
MOVB #377, R3 ; LOAD REGISTER
MOVB #252, R0 ; PLACE #252 IN R0
SCC
BICB R0, R3 ; CLEAR EVERY OTHER BIT
JSR PC, @#SCC1 ; CHECK FOR CC = 1
BITB R0, R3 ; CHECK IT
BEQ 4S ; CONTINUE IF OK
MOV #100, -(R5)
INC -(R5)
HALT ; BICB OR BITB INSTRUCTION FAILED
4S: BITB #125, R3 ; CHECK IT
JSR PC, @#SCC1 ; CHECK FOR CC = 1
BISB R0, R3 ; SET THE BITS THAT WERE CLEARED
BMI 6S
MOV #101, -(R5)
INC -(R5)
HALT ; BISB INSTRUCTION FAILED
6S: BICB #177, R3 ; CLEAR ALL THE BITS EXCEPT FOR SIGN
JSR PC, @#SCC11 ; CHECK FOR CC = 11
BITB #377, R3
JSR PC, @#SCC11 ; CHECK FOR CC = 11

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

INCB0:

CMP (R5), #21
BEQ 1S ; IF IN WRONG SEQUENCE GO TO HLT
MOV #102, -(R5)
INC -(R5)
HALT ; PROGRAM IS IN WRONG SEQUENCE
1S: INC (R5)
MOVB #177, R4 ; R4 = 177
SEC
INCB R4 ; ADD ONES INTO REG. 4
JSR PC, @#SCC13 ; CHECK FOR CC = 13
MOVB #376, R4
INCB R4
JSR PC, @#SCC11 ; CHECK FOR CC = 11
INCB R4
JSR PC, @#SCC5 ; CHECK FOR CC = 5
INCB R4

C03

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 28
 DVKAAB.P11 25-OCT-77 13:09 T21

NEW INSTRUCTIONS USED IN THIS SECTION ARE INCB, DECB

SEQ 0028

1103	003336	004737	017146	JSR	PC, @#SCC1	; CHECK FOR CC = 1
1104	003342	122704	000001	CMPB	#1, R4	; CHECK IT
1105	003346	001404		BEQ	2S	; CONTINUE IF OK
1106	003350	012745	000103	MOV	#103, -(R5)	
1107	003354	005245		INC	-(R5)	
1108	003356	000000		HALT		; INCB INSTRUCTION FAILED
1109	003360	000261		SEC		
1110	003362	105304		DECB	R4	; SUBTRACT ONES FROM REG. 4
1111	003364	004737	017252	JSR	PC, @#SCC5	; CHECK FOR CC = 5
1112	003370	105304		DECB	R4	
1113	003372	004737	017336	JSR	PC, @#SCC11	; CHECK FOR CC = 11
1114	003376	012704	000200	MOV	#200, R4	
1115	003402	105304		DECB	R4	
1116	003404	004737	017206	JSR	PC, @#SCC3	; CHECK FOR CC = 3
1117	003410	105304		DECB	R4	
1118	003412	004737	017146	JSR	PC, @#SCC1	; CHECK FOR CC = 1

2S:

```

1119
1120
1121
1122
1123 003416 021527 000022
1124 003416 001404
1125 003422 012745 000104
1126 003424 005245
1127 003430 000000
1128 003432 000000
1129 003434 005215
1130 003436 112703 000252
1131 003442 000277
1132 003444 105103
1133 003446 004737 017146
1134 003452 122703 000125
1135 003456 001404
1136 003460 012745 000105
1137 003464 005245
1138 003466 000000
1139 003470 000277
1140 003472 105103
1141 003474 004737 017336
1142 003500 122703 000252
1143 003504 001404
1144 003506 012745 000106
1145 003512 005245
1146 003514 000000
1147 003516 012703 000377
1148 003522 000277
1149 003524 105103
1150 003526 004737 017252
1151
1152
1153
1154
1155
1156
1157
1158 003532 021527 000023
1159 003532 001025
1160 003536 005215
1161 003540 112700 000001
1162 003542 105400
1163 003546 004737 017336
1164 003550 122700 000377
1165 003554 001404
1166 003560 012745 000107
1167 003562 005245
1168 003566 000000
1169 003570 012700 000200
1170 003572 105400
1171 003576 004737 017400
1172 003600 122700 000200
1173 003604 001404
1174

```

```

*****
;TEST: 22 NEW INSTRUCTION IN THIS SECTION IS COMB
*****

```

COMBO:

```

      CMP      (R5),#22
      BEQ      1$      ; IF IN WRONG SEQUENCE GO TO HLT
      MOV      #104, -(R5)
      INC      -(R5)
      HALT
1$:    INC      (R5)      ; PROGRAM IS IN WRONG SEQUENCE
      MOVB     #252,R3    ; LOAD EVERY OTHER BIT
      SCC
      COMB     R3         ; 1'S COMPLEMENT
      JSR      PC, @#SCC1 ; CHECK FOR CC = 1
      CMPB     #125,R3    ; CHECK IT
      BEQ      2$         ; CONTINUE IF OK
      MOV      #105, -(R5)
      INC      -(R5)
      HALT
2$:    SCC
      COMB     R3         ; COMB INSTRUCTION FAILED
      JSR      PC, @#SCC1 ; COMPLEMENT BACK
      CMPB     #252,R3    ; CHECK FOR CC = 11
      BEQ      3$         ; CHECK IT
      MOV      #106, -(R5) ; CONTINUE IF OK
      INC      -(R5)
      HALT
3$:    MOV      #377,R3    ; COMB INSTRUCTION FAILED
      SCC
      COMB     R3
      JSR      PC, @#SCC5 ; CHECK FOR CC = 5

```

```

*****
;TEST: 23 NEW INSTRUCTION IN THIS SECTION IS NEGB
*****

```

NEGB0:

```

      CMP      (R5),#23
      BNE      ENEGB0    ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC      (R5)
      MOVB     #1,R0      ; LOAD THE REGISTER
      NEGB     R0         ; 2'S COMPLEMENT
      JSR      PC, @#SCC11 ; CHECK FOR CC = 11
      CMPB     #377,R0    ; CHECK IT
      BEQ      2$         ; CONTINUE IF OK
      MOV      #107, -(R5)
      INC      -(R5)
      HALT
2$:    MOV      #200,R0    ; NEGB INSTRUCTION FAILED
      NEGB     R0         ; 2'S COMPLEMENT
      JSR      PC, @#SCC13 ; CHECK FOR CC = 13
      CMPB     #200,R0    ; CHECK IT
      BEQ      ROLB0      ; CONTINUE IF OK

```

E03

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 30
DVKAAB.P11 25-OCT-77 13:09 T23 NEW INSTRUCTION IN THIS SECTION IS NEGB

SEQ 0030

1175 003612
1176 003612 012745 000110
1177 003616 005245
1178 003620 000000

ENEGBO:

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

; WRONG RESULT IN RD OR WRONG SEQUENCE

F03

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 31
 DVKAAB.P11 25-OCT-77 13:09 T24

NEW INSTRUCTION IN THIS SECTION IS ROLB

SEQ 0031

1179
 1180
 1181
 1182
 1183 003622
 1184 003622 021527 000024
 1185 003626 001026
 1186 003630 005215
 1187 003632 112701 000040
 1188 003636 000257
 1189 003640 106101
 1190 003642 106101
 1191 003644 004737 017360
 1192 003650 122701 000200
 1193 003654 001404
 1194 003656 012745 000111
 1195 003662 005245
 1196 003664 000000
 1197 003666 106101
 1198 003670 004737 017274
 1199 003674 106101
 1200 003676 122701 000001
 1201 003702 001404
 1202 003704
 1203 003704 012745 000112
 1204 003710 005245
 1205 003712 000000
 1206
 1207
 1208
 1209
 1210
 1211
 1212 003714
 1213 003714 021527 000025
 1214 003720 001026
 1215 003722 005215
 1216 003724 112702 000004
 1217 003730 000257
 1218 003732 106002
 1219 003734 106002
 1220 003736 122702 000001
 1221 003742 001404
 1222 003744 012745 000113
 1223 003750 005245
 1224 003752 000000
 1225 003754 106002
 1226 003756 004737 017274
 1227 003762 106002
 1228 003764 004737 017360
 1229 003770 122702 000200
 1230 003774 001404
 1231 003776
 1232 003776 012745 000114
 1233 004002 005245
 1234 004004 000000

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

ROLBO:

```

CMP      (R5), #24
BNE      EROLBO      ; 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
ROLB     R1
JSR      PC, @#SCC12  ; CHECK FOR CC = 12
CMPB     #200, R1     ; CHECK IT
BEQ      IS          ; CONTINUE IF OK
MOV      #111, -(R5)
INC      -(R5)
HALT
IS:      ROLB         R1      ; ROLB INSTRUCTION FAILED
        JSR          PC, @#SCC7 ; SHIFT
        ROLB         R1      ; CHECK FOR CC = 7
        CMPB         #1, R1   ; SHIFT
        BEQ          EROLBO   ; CHECK IT
        MOV          #112, -(R5) ; CONTINUE IF OK
        INC          -(R5)
        HALT
        ; WRONG RESULT IN R1 OR WRONG SEQUENCE

```

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

RORBO:

```

CMP      (R5), #25
BNE      ERORBO      ; 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
RORB     R2
CMPB     #1, R2      ; CHECK IT
BEQ      IS          ; CONTINUE IF OK
MOV      #113, -(R5)
INC      -(R5)
HALT
IS:      RORB         R2      ; RORB INSTRUCTION FAILED
        JSR          PC, @#SCC7 ; SHIFT
        RORB         R2      ; CHECK FOR CC = 7
        JSR          PC, @#SCC12 ; SHIFT
        CMPB         #200, R2   ; CHECK FOR CC = 12
        BEQ          ASLBO     ; CHECK IT
        MOV          #114, -(R5) ; CONTINUE IF OK
        INC          -(R5)
        HALT

```

```

1235
1236
1237
1238
1239 004006
1240 004006 021527 000026
1241 004012 001404
1242 004014 012745 000115
1243 004020 005245
1244 004022 000000
1245 004024 005215
1246 004026 112703 000040
1247 004032 000257
1248 004034 106303
1249 004036 106303
1250 004040 004737 017360
1251 004044 122703 000200
1252 004050 001404
1253 004052 012745 000116
1254 004056 005245
1255 004060 000000
1256 004062 106303
1257 004064 004737 017274
1258 004070 106303
1259 004072 004737 017230
1260
1261
1262
1263
1264
1265 004076
1266 004076 021527 000027
1267 004102 001034
1268 004104 005215
1269 004106 112704 000004
1270 004112 000257
1271 004114 106204
1272 004116 106204
1273 004120 122704 000001
1274 004124 001404
1275 004126 012745 000117
1276 004132 005245
1277 004134 000000
1278 004136 106204
1279 004140 004737 017274
1280 004144 106204
1281 004146 004737 017230
1282 004152 112703 000202
1283 004156 106203
1284 004160 106203
1285 004162 004737 017336
1286 004166 122703 000340
1287 004172 001404
1288 004174
1289 004174 012745 000120
1290 004200 005245

```

```

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

```

```

ASLBO:
      CMP      (R5), #26
      BEQ      2$
      MOV      #115, -(R5)
      INC      -(R5)
      HALT
      ; IF IN WRONG SEQUENCE GO TO HLT BELOW

2$:   INC      (R5)
      MOV      #40, R3
      CCC
      ASLB     R3
      ASLB     R3
      JSR      PC, @#SCC12
      CMPB     #200, R3
      BEQ      4$
      MOV      #116, -(R5)
      INC      -(R5)
      HALT
      ; PROGRAM IS IN WRONG SEQUENCE

4$:   ASLB     R3
      JSR      PC, @#SCC7
      ASLB     R3
      JSR      PC, @#SCC4
      ; LOAD REGISTER
      ; CLEAR FLAGS
      ; SHIFT
      ; CHECK FOR CC = 12
      ; CHECK IT
      ; CONTINUE IF OK
      ; ASLB INSTRUCTION FAILED
      ; SHIFT
      ; CHECK FOR CC = 7
      ; SHIFT
      ; CHECK FOR CC = 4

```

```

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

```

```

ASRBO:
      CMP      (R5), #27
      BNE      EASRBO
      INC      (R5)
      MOV      #4, R4
      CCC
      ASRB     R4
      ASRB     R4
      CMPB     #1, R4
      BEQ      2$
      MOV      #117, -(R5)
      INC      -(R5)
      HALT
      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST

2$:   ASRB     R4
      JSR      PC, @#SCC7
      ASRB     R4
      JSR      PC, @#SCC4
      MOV      #202, R3
      ASRB     R3
      ASRB     R3
      JSR      PC, @#SCC11
      CMPB     #340, R3
      BEQ      ADCBO
      ; LOAD REGISTER
      ; CLEAR FLAGS
      ; SHIFT
      ; CHECK IT
      ; CONTINUE IF OK
      ; ASRB INSTRUCTION FAILED
      ; SHIFT
      ; CHECK FOR CC = 7
      ; SHIFT
      ; CHECK FOR CC = 4
      ; LOAD REGISTER
      ; SHIFT
      ; CHECK FOR CC = 11
      ; CHECK IT
      ; CONTINUE IF OK

EASRBO:
      MOV      #120, -(R5)
      INC      -(R5)

```

H03

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 33
DVKAAB.P11 25-OCT-77 13:09 T27 NEW INSTRUCTION IN THIS SECTION IS ASRB
1291 004202 000000 HALT

SEQ 0033


```

1292      ;*****
1293      ;*TEST: 30      NEW INSTRUCTION IN THIS SECTION IS ADCB
1294      ;*****
1295
1296      004204      021527      000030      ADCB0:      CMP      (R5),#30
1297      004204      001404      BEQ      2$      ; IF IN WRONG SEQUENCE GO TO HLT BELOW
1298      004210      012745      MOV      #121,-(R5)
1299      004212      000121      INC      -(R5)
1300      004216      005245      HALT      ; PROGRAM IS IN WRONG SEQUENCE
1301      004220      000000      2$:      INC      (R5)
1302      004222      005215      CLRB     R0      ; CLEAR THE REGISTER
1303      004224      105000      CCC      ; CLEAR FLAGS
1304      004226      000257      ADCB     R0      ; ADD C BIT = 0
1305      004230      105500      JSR      PC,#$SCC4      ; CHECK FOR CC = 4
1306      004232      004737      017230      SEC      C=1
1307      004236      000261      ADCB     R0      ; ADD C BIT=1
1308      004240      105500      SEC      C=1
1309      004242      000261      ADCB     R0      ; AGAIN
1310      004244      105500      JSR      PC,#$SCC0      ; CHECK FOR CC = 0
1311      004246      004737      017126      CMPB     #2,R0      ; CHECK IT
1312      004252      122700      000002      BEQ      4$      ; CONTINUE IF OK
1313      004256      001404      MOV      #122,-(R5)
1314      004260      012745      000122      INC      -(R5)
1315      004264      005245      HALT      ; ADCB INSTRUCTION FAILED
1316      004266      000000      4$:      MOVB     #177,R0      ; LOAD LARGEST POSITIVE NUMBER
1317      004270      112700      000177      SEC      C=1
1318      004274      000261      ADCB     R0      ; ADD C BIT=1
1319      004276      105500      JSR      PC,#$SCC12      ; CHECK FOR CC = 12
1320      004300      004737      017360      CMPB     #200,R0      ; CHECK IT
1321      004304      122700      000200      BEQ      6$      ; CONTINUE IF OK
1322      004310      001404      MOV      #123,-(R5)
1323      004312      012745      000123      INC      -(R5)
1324      004316      005245      HALT      ; ADCB INSTRUCTION FAILED
1325      004320      000000      6$:      MOVB     #377,R0      ; LOAD -1
1326      004322      112700      000377      SEC      C=1
1327      004326      000261      ADCB     R0      ; ADD C BIT=1
1328      004330      105500      JSR      PC,#$SCC5      ; CHECK FOR CC = 5
1329      004332      004737      017252
1330
1331      ;*****
1332      ;*TEST: 31      NEW INSTRUCTION IN THIS SECTION IS SBCB
1333      ;*****
1334
1335      004336      021527      000031      SBCB0:      CMP      (R5),#31
1336      004336      001404      BEQ      1$      ; IF IN WRONG SEQUENCE GO TO HLT BELOW
1337      004342      012745      MOV      #124,-(R5)
1338      004344      000124      INC      -(R5)
1339      004350      005245      HALT      ; PROGRAM IS IN WRONG SEQUENCE
1340      004352      000000      1$:      INC      (R5)      ; TEST IS IN WRONG SEQUENCE
1341      004354      005215      MOVB     #3,R1      ; LOAD REGISTER
1342      004356      112701      000003      CCC      ; CLEAR FLAGS
1343      004362      000257      SBCB     R1      ; SUBTRACT C BIT=0
1344      004364      105601      JSR      PC,#$SCC0      ; CHECK FOR CC = 0
1345      004366      004737      017126

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 35
 DVKAAB.P11 25-OCT-77 13:09 T31

NEW INSTRUCTION IN THIS SECTION IS SBCB

SEQ 0035

1348	004372	122701	000003	CMPB	#3,R1	; CHECK IT
1349	004376	001404		BEQ	2\$	; CONTINUE IF OK
1350	004400	012745	000125	MOV	#125,-(R5)	
1351	004404	005245		INC	-(R5)	
1352	004406	000000		HALT		; SBCB INSTRUCTION FAILED
1353	004410	000261		SEC		C=1
1354	004412	105601		SBCB	R1	SUBTRACT C BIT=1
1355	004414	000261		SEC		C=1
1356	004416	105601		SBCB	R1	
1357	004420	004737	017126	JSR	PC,2\$SCC0	CHECK FOR CC = 0
1358	004424	122701	000001	CMPB	#1,R1	CHECK IT
1359	004430	001404		BEQ	3\$	CONTINUE IF OK
1360	004432	012745	000126	MOV	#126,-(R5)	
1361	004436	005245		INC	-(R5)	
1362	004440	000000		HALT		; SBCB INSTRUCTION FAILED
1363	004442	000261		SEC		C=1
1364	004444	105601		SBCB	R1	SUBTRACT C BIT=1
1365	004446	004737	017230	JSR	PC,2\$SCC4	CHECK FOR CC = 4
1366	004452	000261		SEC		C=1
1367	004454	105601		SBCB	R1	SUBTRACT C BIT = 1
1368	004456	004737	017336	JSR	PC,2\$SCC11	CHECK FOR CC = 11
1369	004462	122701	000377	CMPB	#377,R1	CHECK IT
1370	004466	001404		BEQ	4\$	CONTINUE IF OK
1371	004470	012745	000127	MOV	#127,-(R5)	
1372	004474	005245		INC	-(R5)	
1373	004476	000000		HALT		; SBCB INSTRUCTION FAILED
1374	004500	112701	000200	MOVB	#200,R1	LOAD R1
1375	004504	000261		SEC		C=1
1376	004506	105601		SBCB	R1	SUBTRACT C BIT = 1
1377	004510	004737	017166	JSR	PC,2\$SCC2	CHECK FOR CC = 2

```

1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390 004514
1391 004514 021527 000032
1392 004520 001404
1393 004522 012745 000130
1394 004526 005245
1395 004530 000000
1396 004532 005215
1397 004534 000277
1398 004536 005000
1399 004540 004737 017230
1400 004544 005700
1401 004546 004737 017230
1402 004552 012704 177777
1403 004556 010401
1404 004560 004737 017316
1405 004564 005701
1406 004566 004737 017316
1407 004572 020401
1408 004574 001404
1409 004576 012745 000131
1410 004602 005245
1411 004604 000000
1412 004606 000263
1413 004610 010000
1414 004612 004767 012434
1415
1416
1417
1418
1419
1420
1421
1422 004616
1423 004616 021527 000033
1424 004622 001026
1425 004624 005215
1426 004626 012700 177777
1427 004632 050002
1428 004634 004737 017316
1429 004640 020002
1430 004642 001404
1431 004644 012745 000132
1432 004650 005245
1433 004652 000000

```

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
1$:    INC      (R5)      ; TEST IS IN WRONG SEQUENCE
      SCC
      CLR      R0      ; CLEAR THE REGISTER
      JSR      PC, @#SCC4 ; CHECK FOR CC = 4
      TST      R0      ; CHECK IT
      JSR      PC, @#SCC4 ; CHECK FOR CC = 4
      MOV      #177777, R4 ; LOAD THE REGISTER
      MOV      R4, R1
      JSR      PC, @#SCC10 ; CHECK FOR CC = 10
      TST      R1      ; CHECK IT
      JSR      PC, @#SCC10 ; CHECK FOR CC = 10
      CMP      R4, R1    ; CHECK R1 TO CONTAIN PROPER DATA
      BEQ      2$
      MOV      #131, -(R5)
      INC      -(R5)
      HALT
2$:    SEVC
      MOV      R0, R0
      JSR      PC, @#SCC5 ; SET V & C BITS

```

```

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

```

```

CMPO:
      CMP      (R5), #33
      BNE      ECMP0    ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$:    INC      (R5)
      MOV      #177777, R0 ; LOAD REGISTER
      BIS      R0, R2      ; CHECK THE BIS INSTRUCTION
      JSR      PC, @#SCC10 ; CHECK FOR CC = 10
      CMP      R0, R2      ; CHECK COMPARE
      BEQ      2$          ; CONTINUE IF OK
      MOV      #132, -(R5)
      INC      -(R5)
      HALT
      ; BIS OR CMP INSTRUCTION FAILED

```


DVKAAB MACY11 30(1746) 25-OCT-77 13:11 PAGE 37
 DVKAAB.P11 25-OCT-77 13:09 T33

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMP, BIS

SEQ 0037

1434	004654	022702	000077	2\$:	CMP	#77,R2	:	CHECK IT AGAIN
1435	004660	100004			BPL	3\$	:	CONTINUE IF OK
1436	004662	012745	000133		MOV	#133, -(R5)		
1437	004666	005245			INC	-(R5)		
1438	004670	000000			HALT		:	CMP INSTRUCTION FAILED [WRONG CC]
1439	004672	020227	000077	3\$:	CMP	R2, #77	:	ONCE MORE
1440	004676	100404			BMI	BICC	:	CONTINUE IF OK
1441	004700			ECMPO:				
1442	004700	012745	000134		MOV	#134, -(R5)		
1443	004704	005245			INC	-(R5)		
1444	004706	000000			HALT		:	WRONG CC OR WRONG SEQUENCE

```

1445
1446
1447
1448
1449 004710
1450 004710 021527 000034
1451 004714 001053
1452 004716 005215
1453 004720 012703 177777
1454 004724 012700 000440
1455 004730 012710 125252
1456 004734 000277
1457 004736 041003
1458 004740 004737 017146
1459 004744 031003
1460 004746 001404
1461 004750 012745 000135
1462 004754 005245
1463 004756 000000
1464 004760 032703 052525
1465 004764 004737 017146
1466 004770 052703 125252
1467 004774 100404
1468 004776 012745 000136
1469 005002 005245
1470 005004 000000
1471 005006 042703 077777
1472 005012 004737 017336
1473 005016 012700 177777
1474 005022 030003
1475 005024 004737 017336
1476 005030 000263
1477 005032 040000
1478 005034 004737 017252
1479 005040 005700
1480 005042 001404
1481 005044
1482 005044 012745 000137
1483 005050 005245
1484 005052 000000
1485
1486

```

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

BICO:

```

    CMP      (R5), #34
    BNE      EBICO
    INC      (R5)
    MOV      #177777, R3
    MOV      #TEMP, R0
    MOV      #125252, (R0)
    SCC
    BIC      (R0), R3
    JSR      PC, @#SCC1
    BIT      (R0), R3
    BEQ      1$
    MOV      #135, -(R5)
    INC      -(R5)
    HALT
1$:
    BIT      #52525, R3
    JSR      PC, @#SCC1
    BIS      #125252, R3
    BMI      2$
    MOV      #136, -(R5)
    INC      -(R5)
    HALT
2$:
    BIC      #77777, R3
    JSR      PC, @#SCC1
    MOV      #177777, R0
    BIT      R0, R3
    JSR      PC, @#SCC11
    SEVC
    BIC      R0, R0
    JSR      PC, @#SCC5
    TST      R0
    BEQ      INCO
EBICO:
    MOV      #137, -(R5)
    INC      -(R5)
    HALT

```

; IF IN WRONG SEQUENCE GO TO HLT ABOVE
 ; LOAD REGISTER
 ; PLACE THE ADDRESS OF LOCATION TEMP IN R0
 ; SET (R0)
 ; CLEAR EVERY OTHER BIT
 ; CHECK FOR CC = 1
 ; CHECK IT
 ; CONTINUE IF OK
 ; BIC OR BIT INSTRUCTION FAILED
 ; CHECK IT
 ; CHECK FOR CC = 1
 ; SET THE BITS THAT WERE CLEARED
 ; CONTINUE IF OK
 ; BIT OR BIS INSTRUCTION FAILED
 ; CLEAR ALL THE BITS EXCEPT FOR SIGN
 ; CHECK FOR CC = 11
 ; CHECK IT
 ; CHECK FOR CC = 11
 ; SET V & C BITS
 ; CHECK CC = 5
 ; CHECK R0 TO CONTAIN 0
 ; BIC FAILED OR SEQUENCE ERROR

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 39
 DVKAAB.P11 25-OCT-77 13:09 T34

NEW INSTRUCTIONS USED IN THIS SECTION ARE BIC, BIT

SEQ 0039

1487			
1488			
1489			
1490			
1491			
1492	005054		
1493	005054	021527	000035
1494	005060	001404	
1495	005062	012745	000140
1496	005066	005245	
1497	005070	000000	
1498	005072	005215	
1499	005074	012704	077777

```

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

INCO:

```

CMP      (R5),#35
BEQ      2$
MOV      #140,-(R5)
INC      -(R5)
HALT
2$: INC   (R5)
MOV      #77777,R4
    
```

; IF IN WRONG SEQUENCE GO TO HLT BELOW

; PROGRAM IS IN WRONG SEQUENCE

; R4=77777

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 40
 DVKAAB.P11 25-OCT-77 13:09 T35 NEW INSTRUCTIONS USED IN THIS SECTION ARE INC, DEC

SEQ 0040

1500	005100	000261		SEC		
1501	005102	005204		INC	R4	; ADD ONES INTO REG. 4
1502	005104	004737	017400	JSR	PC, @#SCC13	; CHECK FOR CC = 13
1503	005110	012704	177776	MOV	#177776, R4	
1504	005114	005204		INC	R4	
1505	005116	004737	017336	JSR	PC, @#SCC11	; CHECK FOR CC = 11
1506	005122	005204		INC	R4	
1507	005124	004737	017252	JSR	PC, @#SCC5	; CHECK FOR CC = 5
1508	005130	005204		INC	R4	
1509	005132	004737	017146	JSR	PC, @#SCC1	; CHECK FOR CC = 1
1510	005136	022704	000001	CMP	#1, R4	; CHECK IT
1511	005142	001404		BEG	4S	; FAILED
1512	005144	012745	000141	MOV	#141, -(R5)	
1513	005150	005245		INC	-(R5)	
1514	005152	000000		HALT		; INC INSTRUCTION FAILED
1515	005154	000261		SEC		
1516	005156	005304		DEC	R4	; SUBTRACT ONES FROM REG. 4
1517	005160	004737	017252	JSR	PC, @#SCC5	; CHECK FOR CC = 5
1518	005164	005304		DEC	R4	
1519	005166	004737	017336	JSR	PC, @#SCC11	; CHECK FOR CC = 11
1520	005172	012704	100000	MOV	#100000, R4	
1521	005176	005304		DEC	R4	
1522	0J5200	004737	017206	JSR	PC, @#SCC3	; CHECK FOR CC = 3
1523	005204	005304		DEC	R4	
1524	005206	004737	017146	JSR	PC, @#SCC1	; CHECK FOR CC = 1

4S:

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 41
 DVKAAB.P11 25-OCT-77 13:09

T36 NEW INSTRUCTION IN THIS SECTION IS COM

SEQ 0041

1525
 1526
 1527
 1528
 1529 005212
 1530 005212 021527 000036
 1531 005216 001404
 1532 005220 012745 000142
 1533 005224 005245
 1534 005226 000000
 1535 005230 005215
 1536 005232 012703 125252
 1537 005236 000277
 1538 005240 005103
 1539 005242 004737 017146
 1540 005246 022703 052525
 1541 005252 001404
 1542 005254 012745 000143
 1543 005260 005245
 1544 005262 000000
 1545 005264 000277
 1546 005266 005103
 1547 005270 004737 017336
 1548 005274 022703 125252
 1549 005300 001404
 1550 005302 012745 000144
 1551 005306 005245
 1552 005310 000000
 1553 005312 012703 177777
 1554 005316 000277
 1555 005320 005103
 1556 005322 004737 017252
 1557
 1558
 1559
 1560
 1561
 1562
 1563
 1564 005326
 1565 005326 021527 000037
 1566 005332 001025
 1567 005334 005215
 1568 005336 012700 000001
 1569 005342 005400
 1570 005344 004737 017336
 1571 005350 022700 177777
 1572 005354 001404
 1573 005356 012745 000145
 1574 005362 005245
 1575 005364 000000
 1576 005366 012700 100000
 1577 005372 005400
 1578 005374 004737 017400
 1579 005400 022700 100000
 1580 005404 001404

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

COMO:

1\$:
 2\$:
 3\$:

```

CMP      (R5), #36
BEQ      1$      ; IF IN WRONG SEQUENCE GO TO HLT BELOW
MOV      #142, -(R5)
INC      -(R5)
HALT
INC      (R5)      ; TEST IS IN WRONG SEQUENCE
MOV      #125252, R3      ; LOAD EVERY OTHER BIT
SCC
COM      R3      ; 1'S COMPLEMENT
JSR      PC, @#SCC1      ; CHECK FOR CC = 1
CMP      #52525, R3      ; CHECK IT
BEQ      2$      ; CONTINUE IF OK
MOV      #143, -(R5)
INC      -(R5)
HALT      ; COM INSTRUCTION FAILED
SCC
COM      R3      ; COMPLEMENT BACK
JSR      PC, @#SCC11      ; CHECK FOR CC = 11
CMP      #125252, R3      ; CHECK IT
BEQ      3$      ; CONTINUE IF OK
MOV      #144, -(R5)
INC      -(R5)
HALT      ; COM INSTRUCTION FAILED
MOV      #177777, R3
SCC
COM      R3
JSR      PC, @#SCC5      ; CHECK FOR CC = 5

```

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

NEGO:

1\$:
 2\$:

```

CMP      (R5), #37
BNE      ENEG0      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
INC      (R5)
MOV      #1, R0      ; LOAD THE REGISTER
NEG      R0      ; 2'S COMPLEMENT
JSR      PC, @#SCC11      ; CHECK FOR CC = 11
CMP      #177777, R0      ; CHECK IT
BEQ      2$      ; CONTINUE IF OK
MOV      #145, -(R5)
INC      -(R5)
HALT      ; NEG INSTRUCTION FAILED
MOV      #100000, R0
NEG      R0      ; 2'S COMPLEMENT
JSR      PC, @#SCC13      ; CHECK FOR CC = 13
CMP      #100000, R0      ; CHECK IT
BEQ      ROLO      ; CONTINUE IF OK

```

004

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 42
DVKAAB.P11 25-OCT-77 13:09 T37 NEW INSTRUCTION IN THIS SECTION IS NEG

SEQ 0042

1581 005406
1582 005406 012745 000146
1583 005412 005245
1584 005414 000000

ENEGO:

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

; WRONG RESULT IN RD OR WRONG SEQUENCE


```

1585
1586
1587
1588
1589 005416
1590 005416 021527 000040
1591 005422 001026
1592 005424 005215
1593 005426 012701 020000
1594 005432 000257
1595 005434 006101
1596 005436 006101
1597 005440 004737 017360
1598 005444 022701 100000
1599 005450 001404
1600 005452 012745 000147
1601 005456 005245
1602 005460 000000
1603 005462 006101
1604 005464 004737 017274
1605 005470 006101
1606 005472 022701 000001
1607 005476 001404
1608 005500
1609 005500 012745 000150
1610 005504 005245
1611 005506 000000
1612
1613
1614
1615
1616
1617
1618
1619 005510
1620 005510 021527 000041
1621 005514 001026
1622 005516 005215
1623 005520 012702 000004
1624 005524 000257
1625 005526 006002
1626 005530 006002
1627 005532 022702 000001
1628 005536 001404
1629 005540 012745 000151
1630 005544 005245
1631 005546 000000
1632 005550 006002
1633 005552 004737 017274
1634 005556 006002
1635 005560 004737 017360
1636 005564 022702 100000
1637 005570 001404
1638 005572
1639 005572 012745 000152
1640 005576 005245

```

 ;TEST: 40 NEW INSTRUCTION IN THIS SECTION IS ROL

ROL0:
 CMP (R5), #40
 BNE EROL0 ; 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, @#SCC12 ; CHECK FOR CC = 12
 CMP #100000, R1 ; CHECK IT
 BEQ 1\$; CONTINUE IF OK
 1\$:
 ROL R1 ; ROL INSTRUCTION FAILED
 JSR PC, @#SCC7 ; SHIFT
 ROL R1 ; CHECK FOR CC = 7
 CMP #1, R1 ; SHIFT
 BEQ ROR0 ; CHECK IT
 ; CONTINUE IF OK
 EROL0:
 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
 1\$:
 ROR R2 ; ROR INSTRUCTION FAILED
 JSR PC, @#SCC7 ; SHIFT
 ROR R2 ; CHECK FOR CC = 7
 JSR PC, @#SCC12 ; SHIFT
 CMP #100000, R2 ; CHECK FOR CC = 12
 BEQ ASLO ; CHECK IT
 ; CONTINUE IF OK
 EROR0:
 MOV #152, -(R5)
 INC -(R5)

F04

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 44
DVKAAB.P11 25-OCT-77 13:09 T41 NEW INSTRUCTION IN THIS SECTION IS ROR

SEQ 0044

1641 005600 000000

HALT

; WRONG RESULT IN R2 OR WRONG SEQUENCE

```

1642
1643
1644
1645
1646 005602
1647 005602 021527 000042
1648 005606 001404
1649 005610 012745 000153
1650 005614 005245
1651 005616 000000
1652 005620 005215
1653 005622 012703 020000
1654 005626 000257
1655 005630 006303
1656 005632 006303
1657 005634 004737 017360
1658 005640 022703 100000
1659 005644 001404
1660 005646 012745 000154
1661 005652 005245
1662 005654 000000
1663 005656 006303
1664 005660 004737 017274
1665 005664 006303
1666 005666 004737 017230
1667
1668
1669
1670
1671
1672 005672
1673 005672 021527 000043
1674 005676 001034
1675 005700 005215
1676 005702 012704 000004
1677 005706 000257
1678 005710 006204
1679 005712 006204
1680 005714 022704 000001
1681 005720 001404
1682 005722 012745 000155
1683 005726 005245
1684 005730 000000
1685 005732 006204
1686 005734 004737 017274
1687 005740 006204
1688 005742 004737 017230
1689 005746 012703 100002
1690 005752 006203
1691 005754 006203
1692 005756 004737 017336
1693 005762 022703 160000
1694 005766 001404
1695 005770
1696 005770 012745 000156
1697 005774 005245

```

```

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

```

ASLO:

```

      CMP      (R5), #42
      BEQ      2$ ; IF IN WRONG SEQUENCE GO TO HLT BELOW
      MOV      #153, -(R5)
      INC      -(R5)
      HALT
2$:    INC      (R5) ; PROGRAM IS IN WRONG SEQUENCE
      MOV      #20000, R3 ; LOAD REGISTER
      CCC      ; CLEAR FLAGS
      ASL      R3 ; SHIFT
      ASL      R3
      JSR      PC, @#SCC12 ; CHECK FOR CC = 12
      CMP      #100000, R3 ; CHECK IT
      BEQ      4$ ; CONTINUE IF OK
      MOV      #154, -(R5)
      INC      -(R5)
      HALT
4$:    ASL      R3 ; ASL INSTRUCTION FAILED
      JSR      PC, @#SCC7 ; SHIFT
      ASL      R3 ; CHECK FOR CC = 7
      JSR      PC, @#SCC4 ; SHIFT
      ; CHECK FOR CC = 4

```

```

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

```

ASRO:

```

      CMP      (R5), #43
      BNE      EASRO ; 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
2$:    ASR      R4 ; ASR INSTRUCTION FAILED
      JSR      PC, @#SCC7 ; SHIFT
      ASR      R4 ; CHECK FOR CC = 7
      JSR      PC, @#SCC4 ; SHIFT
      MOV      #100002, R3 ; CHECK FOR CC = 4
      ASR      R3 ; LOAD REGISTER
      ASR      R3 ; SHIFT
      JSR      PC, @#SCC11 ; CHECK FOR CC = 11
      CMP      #160000, R3 ; CHECK IT
      BEQ      EASRO ; CONTINUE IF OK
EASRO: MOV      #156, -(R5)
      INC      -(R5)

```


H04

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 46
DVKAAB.P11 25-OCT-77 13:09 T43 NEW INSTRUCTION IN THIS SECTION IS ASR

SEQ 0046

1698 005776 000000

HALT

; WRONG RESULT IN R3 OR WRONG SEQUENCE

```

1699
1700
1701
1702
1703 006000
1704 006000 021527 000044
1705 006004 001404
1706 006006 012745 000157
1707 006012 005245
1708 006014 000000
1709 006016 005215
1710 006020 005000
1711 006022 000257
1712 006024 005500
1713 006026 004737 017230
1714 006032 000261
1715 006034 005500
1716 006036 000261
1717 006040 005500
1718 006042 004737 017126
1719 006046 022700 000002
1720 006052 001404
1721 006054 012745 000160
1722 006060 005245
1723 006062 000000
1724 006064 012700 077777
1725 006070 000261
1726 006072 005500
1727 006074 004737 017360
1728 006100 022700 100000
1729 006104 001404
1730 006106 012745 000161
1731 006112 005245
1732 006114 000000
1733 006116 012700 177777
1734 006122 000261
1735 006124 005500
1736 006126 004737 017252
1737
1738
1739
1740
1741
1742
1743
1744 006132
1745 006132 021527 000045
1746 006136 001404
1747 006140 012745 000162
1748 006144 005245
1749 006146 000000
1750 006150 005215
1751 006152 012701 000003
1752 006156 000257
1753 006160 005601
1754 006162 004737 017126

```

```

*****
;TEST: 44 NEW INSTRUCTION IN THIS SECTION IS ADC
*****

```

```

ADCO:  CMP      (R5),#44
      BEQ      2$
      MOV      #157,-(R5)
      INC      -(R5)
      HALT
2$:    INC      (R5)
      CLR      R0
      CCC
      ADC      R0
      JSR      PC,#$SCC4
      SEC
      ADC      R0
      SEC
      ADC      R0
      JSR      PC,#$SCC0
      CMP      #2,R0
      BEQ      4$
      MOV      #160,-(R5)
      INC      -(R5)
      HALT
4$:    MOV      #77777,R0
      SEC
      ADC      R0
      JSR      PC,#$SCC12
      CMP      #100000,R0
      BEQ      6$
      MOV      #161,-(R5)
      INC      -(R5)
      HALT
6$:    MOV      #-1,R0
      SEC
      ADC      R0
      JSR      PC,#$SCC5

```

; IF IN WRONG SEQUENCE GO TO HLT BELOW

; PROGRAM IS IN WRONG SEQUENCE

; CLEAR THE REGISTER

; CLEAR FLAGS

; ADD C BIT = 0

; CHECK FOR CC = 4

; C=1

; ADD C BIT=1

; C=1

; AGAIN

; CHECK FOR CC = 0

; CHECK IT

; CONTINUE IF OK

; ADC INSTRUCTION FAILED

; LOAD LARGEST POSITIVE NUMBER

; C=1

; ADD C BIT=1

; CHECK FOR CC = 12

; CHECK IT

; FAILED

; ADC INSTRUCTION FAILED

; LOAD -1

; C=1

; ADD C BIT=1

; CHECK FOR CC = 5

```

*****
;TEST: 45 NEW INSTRUCTION IN THIS SECTION IS SBC
*****

```

```

SBCO:  CMP      (R5),#45
      BEQ      1$
      MOV      #162,-(R5)
      INC      -(R5)
      HALT
1$:    INC      (R5)
      MOV      #3,R1
      CCC
      SBC      R1
      JSR      PC,#$SCC0

```

; IF IN WRONG SEQUENCE GO TO HLT

; TEST IS IN WRONG SEQUENCE

; LOAD REGISTER

; CLEAR FLAGS

; SUBTRACT C BIT=0

; CHECK FOR CC = 0

J04

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 48
 DVKAAB.P11 25-OCT-77 13:09 T45 NEW INSTRUCTION IN THIS SECTION IS SBC

SEQ 0048

1755	006166	022701	000003	CMP	#3,R1	; CHECK IT
1756	006172	001404		BEQ	2\$	; CONTINUE IF OK
1757	006174	012745	000163	MOV	#163,-(R5)	
1758	006200	005245		INC	-(R5)	
1759	006202	000000		HALT		; SBC INSTRUCTION FAILED
1760	006204	000261		SEC		C=1
1761	006206	005601		SBC	R1	SUBTRACT C BIT=1
1762	006210	000261		SEC		C=1
1763	006212	005601		SBC	R1	
1764	006214	004737	017126	JSR	PC,2\$SCC0	CHECK FOR CC = 0
1765	006220	022701	000001	CMP	#1,R1	CHECK IT
1766	006224	001404		BEQ	3\$	CONTINUE IF OK
1767	006226	012745	000164	MOV	#164,-(R5)	
1768	006232	005245		INC	-(R5)	
1769	006234	000000		HALT		; SBC INSTRUCTION FAILED
1770	006236	000261		SEC		C=1
1771	006240	005601		SBC	R1	SUBTRACT C BIT=1
1772	006242	004737	017230	JSR	PC,2\$SCC4	CHECK FOR CC = 4
1773	006246	000261		SEC		C=1
1774	006250	005601		SBC	R1	SUBTRACT C BIT = 1
1775	006252	004737	017336	JSR	PC,2\$SCC11	CHECK FOR CC = 11
1776	006256	022701	177777	CMP	#-1,R1	CHECK IT
1777	006262	001404		BEQ	4\$	CONTINUE IF F OK
1778	006264	012745	000165	MOV	#165,-(R5)	
1779	006270	005245		INC	-(R5)	
1780	006272	000000		HALT		; SBC INSTRUCTION FAILED
1781	006274	012701	100000	MOV	#100000,R1	LOAD R1
1782	006300	000261		SEC		C=1
1783	006302	005601		SBC	R1	SUBTRACT C BIT = 1
1784	006304	004737	017166	JSR	PC,2\$SCC2	CHECK FOR CC = 2

K04

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 49
 DVKAAB.P11 25-OCT-77 13:09 746 NEW INSTRUCTION IN THIS SECTION IS SXT

SEQ 0049

1785
 1786
 1787
 1788
 1789 006310
 1790 006310 021527 000046
 1791 006314 001024
 1792 006316 005215
 1793 006320 005002
 1794 006322 000277
 1795 006324 000254
 1796 006326 006702
 1797 006330 004737 017252
 1798 006334 005702
 1799 006336 001404
 1800 006340 012745 000166
 1801 006344 005245
 1802 006346 000000
 1803 006350 000273
 1804 006352 006702
 1805 006354 004737 017336
 1806 006360 022702 177777
 1807 006364 001404
 1808 006366
 1809 006366 012745 000167
 1810 006372 005245
 1811 006374 000000
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819 006376
 1820 006376 021527 000047
 1821 006402 001031
 1822 006404 005215
 1823 006406 012703 125125
 1824 006412 000277
 1825 006414 000250
 1826 006416 000303
 1827 006420 004737 017316
 1828 006424 022703 052652
 1829 006430 001404
 1830 006432 012745 000170
 1831 006436 005245
 1832 006440 000000
 1833 006442 012703 000377
 1834 006446 000277
 1835 006450 000244
 1836 006452 000303
 1837 006454 004737 017230
 1838 006460 022703 177400
 1839 006464 001404
 1840 006466

 ;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
 CLNZ
 SXT R2 ; SIGN EXTEND
 JSR PC, @#SCC5 ; CHECK FOR CC = 5
 TST R2 ; REG. 2 SHOULD STILL BE 0
 BEQ 2\$; CONTINUE IF OK
 2\$: MOV #166, -(R5)
 INC -(R5)
 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

 ;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 XOR0
 ESWAB0:

L04

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 50
DVKAAB.P11 25-OCT-77 13:09 T47 NEW INSTRUCTION IN THIS SECTION IS SWAB

SEQ 0050

1841 006466 012745 000171
1842 006472 005245
1843 006474 000000

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

; WRONG RESULT IN R3 OR WRONG SEQUENCE

```

1844
1845
1846
1847
1848 006476
1849 006476 021527 000050
1850 006502 001034
1851 006504 005215
1852 006506 012704 177777
1853 006512 012703 177777
1854 006516 000277
1855 006520 074403
1856 006522 004737 017252
1857 006526 012703 017777
1858 006532 010400
1859 006534 000263
1860 006536 000244
1861 006540 074003
1862 006542 004737 017336
1863 006546 012702 125252
1864 006552 012704 052525
1865 006556 000277
1866 006560 074204
1867 006562 004737 017336
1868 006566 022704 177777
1869 006572 001404
1870 006574
1871 006574 012745 000172
1872 006600 005245
1873 006602 000000
1874
1875
1876
1877
1878
1879
1880
1881 006604
1882 006604 021527 000051
1883 006610 001055
1884 006612 005215
1885 006614 012701 021421
1886 006620 060101
1887 006622 004737 017126
1888 006626 022701 043042
1889 006632 001404
1890 006634 012745 000173
1891 006640 005245
1892 006642 000000
1893 006644 012700 156357
1894 006650 060000
1895 006652 004737 017336
1896 006656 022700 134736
1897 006662 001404
1898 006664 012745 000174
1899 006670 005245

```

```

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

```

XOR0:

```

CMP      (R5), #50
BNE      EXOR0
INC      (R5)
MOV      #-1, R4
MOV      #-1, R3
SCC
XOR      R4, R3
JSR      PC, @#SCC5
MOV      #77777, R3
MOV      R4, R0
SEVC
CLZ
XOR      R0, R3
JSR      PC, @#SCC11
MOV      #125252, R2
MOV      #52525, R4
SCC
XOR      R2, R4
JSR      PC, @#SCC11
CMP      #-1, R4
BEQ      ADD0
EXOR0:   MOV      #172, -(R5)
INC      -(R5)
HALT
; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
; LOAD REGISTERS
; SHOULD PRODUCE 0'S IN REG. 3
; CHECK FOR CC = 5
; PLACE A -1 IN R0
; SET V & C BITS
; CHECK FOR CC = 11
; LOAD REGISTERS
; SHOULD PRODUCE ALL 1'S IN REG. 4
; CHECK FOR CC = 11
; CHECK IT
; CONTINUE IF OK
; WRONG RESULT IN R4 OR WRONG SEQUENCE

```

```

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

```

ADD0:

```

CMP      (R5), #51
BNE      EADD0
INC      (R5)
MOV      #21421, R1
ADD      R1, R1
JSR      PC, @#SCC0
CMP      #43042, R1
BEQ      1$
MOV      #173, -(R5)
INC      -(R5)
HALT
1$:      MOV      #-21421, R0
ADD      R0, R0
JSR      PC, @#SCC11
CMP      #-43042, R0
BEQ      2$
MOV      #174, -(R5)
INC      -(R5)
; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
; LOAD REGISTERS
; ADD
; CHECK FOR CC = 0
; CHECK IT
; CONTINUE IF OK
; ADD INSTRUCTION FAILED
; LOAD REGISTERS
; ADD
; CHECK FOR CC = 11
; CHECK IT
; CONTINUE IF OK

```


DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 52
 DVKAAB.P11 25-OCT-77 13:09 T51

NEW INSTRUCTION IN THIS SECTION IS ADD

SEQ 0052

1900	006672	000000			HALT				
1901	006674	012702	100000	2\$:	MOV	#100000,R2			: ADD INSTRUCTION FAILED
1902	006700	060202			ADD	R2,R2			: LOAD REGISTERS
1903	006702	004737	017274		JSR	PC,2#SCC7			: ADD SHOULD RESULT AS 0'S
1904	006706	012704	021421		MOV	#21421,R4			: CHECK FOR CC = 7
1905	006712	012701	156357		MOV	#-21421,R1			: LOAD REGISTERS
1906	006716	060401			ADD	R4,R1			: ADD SHOULD RESULT AS 0'S
1907	006720	001404			BEQ	3\$			: CONTINUE IF OK
1908	006722	012745	000175		MOV	#175,-(R5)			
1909	006726	005245			INC	-(R5)			
1910	006730	000000			HALT				: ADD INSTRUCTION FAILED
1911	006732	005404		3\$:	NEG	R4			: SWITCH SOURCE AND DESTINATION
1912	006734	012701	021421		MOV	#21421,R1			
1913	006740	060104			ADD	R1,R4			: SHOULD RESULT AS 0'S
1914	006742	001404			BEQ	SUB0			: CONTINUE IF OK
1915	006744			EADD0:					
1916	006744	012745	000176		MOV	#176,-(R5)			
1917	006750	005245			INC	-(R5)			
1918	006752	000000			HALT				: WRONG RESULT IN R1 OR WRONG SEQUENCE

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 53
 DVKAAB.P11 25-OCT-77 13:09 TS2

NEW INSTRUCTION IN THIS SECTION IS SUB

SEQ 0053

1919
 1920
 1921
 1922
 1923 006754
 1924 006754 021527 000052
 1925 006760 001404
 1926 006762 012745 000177
 1927 006766 005245
 1928 006770 000000
 1929 006772 005215
 1930 006774 012702 021421
 1931 007000 012703 156357
 1932 007004 160203
 1933 007006 004737 017316
 1934 007012 022703 134736
 1935 007016 001404
 1936 007020 012745 000200
 1937 007024 005245
 1938 007026 000000
 1939 007030 012703 021421
 1940 007034 010204
 1941 007036 160403
 1942 007040 001404
 1943 007042 012745 000201
 1944 007046 005245
 1945 007050 000000
 1946 007052 012703 177777
 1947 007056 012702 077777
 1948 007062 160302
 1949 007064 004737 017400
 1950 007070 022702 100000
 1951 007074 001404
 1952 007076 012745 000202
 1953 007102 005245
 1954 007104 000000
 1955 007106 012704 177777
 1956 007112 160304
 1957 007114 004737 017230
 1958
 1959
 1960
 1961
 1962
 1963
 1964 007120
 1965 007120 021527 000053
 1966 007124 001032
 1967 007126 005215
 1968 007130 012701 177777
 1969 007134 005000
 1970 007136
 1971 007136 106400
 1972 007140 004737 017126
 1973 007144
 1974 007144 106701

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

SUB0:

CMP (R5), #52
 BEQ 2S ; IF IN WRONG SEQUENCE GO TO HLT BELOW
 MOV #177, -(R5)
 INC -(R5)

2S:

HALT ; PROGRAM IS IN WRONG SEQUENCE
 INC (R5)
 MOV #21421, R2 ; LOAD REGISTERS
 MOV #21421, R3
 SUB R2, R3 ; RESULT SHOULD=-43042
 JSR PC, @#SCC10 ; CHECK FOR CC = 10
 CMP #43042, R3 ; CHECK IT
 BEQ 4S ; CONTINUE IF OK

4S:

HALT ; SUB INSTRUCTION FAILED
 MOV #21421, R3 ; LOAD REGISTER
 MOV R2, R4 ; NOW R4 = #21421
 SUB R4, R3 ; RESULT SHOULD=0
 BEQ 6S

6S:

HALT ; SUB INSTRUCTION FAILED
 MOV #-1, R3 ; LOAD REGISTERS
 MOV #77777, R2 ; LOAD REGISTERS
 SUB R3, R2 ; RESULT SHOULD BE 100000 AND OVERFLOW
 JSR PC, @#SCC13 ; CHECK FOR CC = 13
 CMP #100000, R2 ; CHECK IT
 BEQ 8S ; CONTINUE IF OK

8S:

HALT ; SUB INSTRUCTION FAILED
 MOV #-1, R4
 SUB R3, R4
 JSR PC, @#SCC4 ; CHECK FOR CC = 4

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

PSW:

CMP (R5), #53
 BNE EPSW ; IF IN WRONG SEQUENCE THEN GO TO HLT AT THE END OF THE

1S:

INC (R5)
 MOV #177777, R1
 CLR R0
 MTPS R0 ; SET PSW TO 0
 .WORD 106400!...C
 JSR PC, @#SCC0 ; CHECK FOR CC = 0
 MFPS R1 ; MOVE PSW TO R1
 .WORD 106700!...C

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 54
 DVKAAB.P11 25-OCT-77 13:09 T53

NEW INSTRUCTIONS IN THARE SECTION IS MTPS & MFPS

SEQ 0054

1975	007146	001404		BEQ	25	; CONTINUE IF BIT 8 OF PSW WAS EXTENDED IN R1
1976	007150	012745	000203	MOV	#203, -(R5)	
1977	007154	005245		INC	-(R5)	
1978	007156	000000		HALT		; MTPS OR MFPS INSTRUCTION FAILED
1979	007160	004737	017230	JSR	PC, @#SCC4	; CHECK FOR CC = 4
1980	007164	012700	000377	MOV	#377, R0	
1981	007170			MTPS	R0	; SET PSW TO 357 SINCE MTPS DOES NOT SET T BIT
1982	007170	106400		.WORD	106400!..C	
1983	007172	004737	017420	JSR	PC, @#SCC17	; CHECK FOR CC = 17
1984	007176			MFPS	R1	; MOVE PSW TO R1
1985	007176	106701		.WORD	106700!..C	
1986	007200	004737	017336	JSR	PC, @#SCC11	; CHECK FOR CC = 11 [C BIT SHOULD NOT BE EFFECTED BY MFP
1987	007204	022701	177757	CMP	#177757, R1	; CHECK TO SEE IF BIT 8 OF PSW WAS EXTENDED THRU R1
1988	007210	001404		BEQ	MODE0	
1989	007212					
1990	007212	012745	000204	MOV	#204, -(R5)	
1991	007216	005245		INC	-(R5)	
1992	007220	000000		HALT		; MTPS OR MFPS INSTRUCTION FAILED OR WRONG SEQUENCE

25:
 EPSW:


```

1993
1994
1995
1996
1997
1998
1999
2000
2001
2002 007222 021527 000054
2003 007222 001063
2004 007226 005215
2005 007230 112700 000252
2006 007232 110001
2007 007236 110102
2008 007240 122702 000252
2009 007242 001404
2010 007246 012745 000205
2011 007250 005245
2012 007254 000000
2013 007256 012700 125252
2014 007260 010001
2015 007264 010102
2016 007266 022702 125252
2017 007270 001404
2018 007274 012745 000206
2019 007276 005245
2020 007302 000000
2021 007304
2022
2023 007306 012700 000440
2024 007312 012701 000442
2025 007316 012702 000444
2026 007322 005067 171116
2027 007326 112710 000125
2028 007332 111011
2029 007334 111112
2030 007336 122767 000125 171100
2031 007344 001404
2032 007346 012745 000207
2033 007352 005245
2034 007354 000000
2035 007356 012710 052525
2036 007362 011011
2037 007364 011112
2038 007366 022767 052525 171050
2039 007374 001404
2040 007376
2041 007376 012745 000210
2042 007402 005245
2043 007404 000000
2044
2045
2046
2047
2048

```

LSI-11 INSTRUCTIONS NOT MODE 0

```

;*****
;TEST: 54 CHECK MODES 0 & 1 USING THE MOVB AND MOV INSTRUCTIONS
;*****
MODE0:
      CMP      (R5), #54
      BNE      EMODE0 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC      (R5)
      MOVB     #252, R0 ; LOAD REGISTERS
      MOVB     R0, R1
      MOVB     R1, R2
      CMPB     #252, R2 ; CHECK IT
      BEQ      1$ ; OK, CONTINUE
      MOV      #205, -(R5)
      INC      -(R5)
      HALT ; MOV INSTRUCTION FAILED IN MODE 0
1$:   MOV      #125252, R0 ; LOAD REGISTERS
      MOV      R0, R1
      MOV      R1, R2
      CMP      #125252, R2 ; CHECK IT
      BEQ      MODE1 ; OK, CONTINUE
      MOV      #206, -(R5)
      INC      -(R5)
      HALT ; MOV INSTRUCTION FAILED IN MODE 0
MODE1:
      MOV      #TEMP, R0 ; LOAD ADDRESSES INTO REGS.
      MOV      #TEMP1, R1
      MOV      #TEMP2, R2
      CLR      TEMP2
      MOVB     #125, (R0)
      MOVB     (R0), (R1)
      MOVB     (R1), (R2)
      CMPB     #125, TEMP2
      BEQ      1$ ; START CLEAN
      MOV      #207, -(R5) ; LOAD THE LOCATIONS
      INC      -(R5) ; TEMP ----> TEMP1
      HALT ; TEMP1 ----> TEMP2
1$:   MOV      #52525, (R0) ; CHECK IT
      MOV      (R0), (R1) ; OK, CONTINUE
      MOV      (R1), (R2)
      CMP      #52525, TEMP2
      BEQ      MODE2
      MOV      #210, -(R5) ; MOV INSTRUCTION FAILED IN MODE 1
      INC      -(R5) ; LOAD THE LOCATIONS
      HALT ; TEMP ----> TEMP1
      ; TEMP1 ----> TEMP2
      ; CHECK IT
      ; OK, CONTINUE
EMODE0:
      MOV      #210, -(R5)
      INC      -(R5)
      HALT ; MOV INSTRUCTION FAILED IN MODE 1
      ; OR WRONG SEQUENCE
;*****

```

E05

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 56
 DVKAAB.P11 25-OCT-77 13:09 T55

CHECK MODE 2 USING THE MOVB AND MOV INSTRUCTIONS

SEQ 0056

```

2049 ;*TEST: 55 CHECK MODE 2 USING THE MOVB AND MOV INSTRUCTIONS
2050 ;*****
2051
2052 MODE2:
2053 CMP (R5),#55
2054 BNE EMODE2 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2055 INC (R5)
2056 MOV #TEMP,R0 ; LOAD ADDRESSES
2057 MOV #TEMP1,R1
2058 MOV #TEMP2,R2
2059 CLRB (R2)+
2060 MOVB #252,(R0) ; START CLEAN
2061 MOVB (R0)+,(R1)+ ; LOAD THE LOCATIONS
2062 INCB R1 ; TEMP ----> TEMP1
2063 MOVB (R1),TEMP ; MAKE IT EVEN
2064 INCB R0 ; MORE 0'S INTO TEMP
2065 MOVB (R0)+,(R1)+ ; MAKE IT EVEN
2066 CMPB -(R2),#252 ; TEMP1 ----> TEMP2
2067 BNE 1$ ; CHECK IT
2068 TSTB TEMP ; FAILED
2069 BEQ 2$ ; CHECK IT
2070 ; OK, CONTINUE
2071 1$: MOV #211,-(R5)
2072 INC -(R5)
2073 HALT ; INSTRUCTIONS FAILED IN MODE 2
2074
2075 2$: TST -(R1)
2076 CLR (R2)+ ; START CLEAN
2077 MOV #125252,-(R0) ; LOAD LOCATIONS
2078 MOV (R0)+,(R0)+ ; TEMP ----> TEMP1
2079 MOV (R0),TEMP ; 0 ----> TEMP
2080 MOV (R1)+,(R1)+ ; 125252 ----> TEMP2
2081 CMPB -(R2),#125252 ; CHECK IT
2082 BNE EMODE2 ; FAILED
2083 TST TEMP ; CHECK IT
2084 BEQ MODE3 ; OK, CONTINUE
2085 EMODE2:
2086 MOV #212,-(R5)
2087 INC -(R5)
2088 HALT ; INSTRUCTIONS FAILED IN MODE 2
2089 ; OR WRONG SEQUENCE

```

```

2090
2091
2092
2093
2094 007544
2095 007544 021527 000056
2096 007550 001066
2097 007552 005215
2098 007554 012767 000440 170646
2099 007562 012767 000442 170642
2100 007570 012767 000444 170636
2101 007576 012700 000430
2102 007602 012701 000432
2103 007606 105067 170632
2104 007612 112767 000125 170620
2105 007620 113031
2106 007622 113167 170612
2107 007626 113030
2108 007630 122767 000125 170606
2109 007636 001003
2110 007640 105767 170574
2111 007644 001404
2112 007646
2113 007646 012745 000213
2114 007652 005245
2115 007654 000000
2116 007656 005067 170562
2117 007662 012767 052525 170550
2118 007670 012700 000430
2119 007674 012701 000432
2120 007700 013030
2121 007702 013067 170532
2122 007706 013131
2123 007710 022767 052525 170526
2124 007716 001003
2125 007720 005767 170514
2126 007724 001404
2127 007726
2128 007726 012745 000214
2129 007732 005245
2130 007734 000000
2131
2132
2133
2134
2135
2136
2137
2138 007736
2139 007736 021527 000057
2140 007742 001120
2141 007744 005215
2142 007746 105067 170466
2143 007752 012700 000440
2144 007756 012701 000442
2145 007762 012702 000444

;*****
;TEST: 56 CHECK MODE 3 USING THE MOV B AND MOV INSTRUCTIONS
;*****

MODE3:
CMP (R5),#56
BNE EMODE3 ; IF IN WRONG SEQUENCE GO TO HLT ABOVE
INC (R5)
MOV #TEMP,ADR ; LOAD ADDRESSES
MOV #TEMP1,ADR1
MOV #TEMP2,ADR2
MOV #ADR,R0 ; LOAD ADDRESSES OF ADDRESSES
MOV #ADR1,R1
CLR R2 ; START CLEAN
MOV #125,TEMP
MOV #2(R0)+,2(R1)+ ; TEMP ---> TEMP1
MOV #2(R1)+,TEMP ; TEMP2 ---> TEMP
MOV #2(R0)+,2(R0)+ ; TEMP1 ---> TEMP2
CMP #125,TEMP2 ; CHECK IT
BNE 1$ ; FAILED
TSTB TEMP ; CHECK IT
BEQ 2$ ; OK, CONTINUE

1$:
MOV #213,-(R5)
INC -(R5)
HALT

2$:
CLR TEMP2 ; INSTRUCTIONS FAILED IN MODE 3
MOV #52525,TEMP ; START CLEAN
MOV #ADR,R0 ; LOAD LOCATIONS
MOV #ADR1,R1 ; LOAD ADDRESSES OF ADDRESSES
MOV #2(R0)+,2(R0)+ ; TEMP ---> TEMP1
MOV #2(R0)+,TEMP ; TEMP2 ---> TEMP
MOV #2(R1)+,2(R1)+ ; TEMP1 ---> TEMP2
CMP #52525,TEMP2 ; CHECK IT
BNE EMODE3 ; FAILED
TST TEMP ; CHECK IT
BEQ MODE4 ; OK, CONTINUE

EMODE3:
MOV #214,-(R5)
INC -(R5)
HALT ; INSTRUCTIONS FAILED IN MODE 3

;*****
;TEST: 57 CHECK MODE 4 USING THE MOV B AND MOV INSTRUCTIONS
;*****

MODE4:
CMP (R5),#57
BNE EMODE4 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
INC (R5)
CLRB TEMP ; START CLEAN
MOV #TEMP,R0 ; LOAD ADDRESSES
MOV #TEMP1,R1
MOV #TEMP2,R2

```


2146	007766	005202		INC	R2	; ADJUST THE POINTER
2147	007770	021267	170451	CMP	(R2), TEMP2+1	
2148	007774	001404		BEQ	1\$	
2149	007776	012745	000215	MOV	#215, -(R5)	
2150	010002	005245		INC	-(R5)	
2151	010004	000000		HALT		: INSTRUCTIONS FAILED IN MODE 4
2152	010006	112742	000252	MOVB	#252, -(R2)	LOAD TEMP2
2153	010012	005201		INC	R1	ADJUST THE POINTERS
2154	010014	005202		INC	R2	
2155	010016	114241		MOVB	-(R2), -(R1)	TEMP2 ---> TEMP1
2156	010020	005200		INC	R0	ADJUST THE POINTERS
2157	010022	005202		INC	R2	
2158	010024	114042		MOVB	-(R0), -(R2)	TEMP ---> TEMP2
2159	010026	105200		INCB	R0	ADJUST THE POINTERS
2160	010030	021067	170405	CMP	(R0), TEMP+1	
2161	010034	001404		BEQ	2\$	
2162	010036	012745	000216	MOV	#216, -(R5)	
2163	010042	005245		INC	-(R5)	
2164	010044	000000		HALT		: INSTRUCTIONS FAILED IN MODE 4
2165	010046	105201		INCB	R1	
2166	010050	114140		MOVB	-(R1), -(R0)	TEMP1 ---> TEMP
2167	010052	122767	000252 170360	CMPB	#252, TEMP	CHECK IT
2168	010060	001003		BNE	3\$	FAILED
2169	010062	105767	170356	TSTB	TEMP2	CHECK IT
2170	010066	001404		BEQ	4\$	OK, CONTINUE
2171	010070					
2172	010070	012745	000217	MOV	#217, -(R5)	
2173	010074	005245		INC	-(R5)	
2174	010076	000000		HALT		: INSTRUCTIONS FAILED IN MODE 4
2175	010100	005067	170334	CLR	TEMP	START CLEAN
2176	010104	012700	000440	MOV	#TEMP, R0	LOAD ADDRESSES
2177	010110	012701	000442	MOV	#TEMP1, R1	
2178	010114	012702	000444	MOV	#TEMP2, R2	
2179	010120	005722		TST	(R2)+	ADJUST THE POINTER
2180	010122	021267	170320	CMP	(R2), TEMP2+2	
2181	010126	001404		BEQ	5\$	
2182	010130	012745	000220	MOV	#220, -(R5)	
2183	010134	005245		INC	-(R5)	
2184	010136	000000		HALT		: INSTRUCTIONS FAILED IN MODE 4
2185	010140	012742	125252	MOV	#125252, -(R2)	LOAD TEMP2
2186	010144	005721		TST	(R1)+	ADJUST THE POINTERS
2187	010146	005722		TST	(R2)+	
2188	010150	014241		MOV	-(R2), -(R1)	TEMP2 ---> TEMP1
2189	010152	005720		TST	(R0)+	ADJUST POINTERS
2190	010154	005722		TST	(R2)+	
2191	010156	014042		MOV	-(R0), -(R2)	TEMP ---> TEMP2
2192	010160	005720		TST	(R0)+	ADJUST THE POINTERS
2193	010162	005721		TST	(R1)+	
2194	010164	014140		MOV	-(R1), -(R0)	TEMP1 ---> TEMP
2195	010166	022767	125252 170244	CMP	#125252, TEMP	CHECK IT
2196	010174	001003		BNE	EMODE4	FAILED
2197	010176	005767	170242	TST	TEMP2	CHECK IT
2198	010202	001404		BEQ	MODE5	OK, CONTINUE
2199	010204					
2200	010204	012745	000221	MOV	#221, -(R5)	
2201	010210	005245		INC	-(R5)	

EMODE4:

H05

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 59

DVKAAB.P11 25-OCT-77 13:09

T57

CHECK MODE 4 USING THE MOVB AND MOV INSTRUCTIONS

SEQ 0059

2202 010212 000000
2203

HALT

: INSTRUCTIONS FAILED IN MODE 4
: OR WRONG SEQUENCE

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 60
 DVKAAB.P11 25-OCT-77 13:09 T60

CHECK MODE 5 USING THE MOVB AND MOV INSTRUCTIONS

SEQ 0060

```

2204      ;*****
2205      ;*TEST: 60      CHECK MODE 5 USING THE MOVB AND MOV INSTRUCTIONS
2206      ;*****
2207
2208      010214      021527      000060      MODE5:      CMP      (R5), #60
2209      010214      001105      170172      BNE      EMODE5      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2210      010220      005215      170166      INC      (R5)
2211      010222      105067      170162      CLRB     TEMP      ; START CLEAN
2212      010230      012767      000440      MOV      #TEMP, ADR      ; LOAD ADDRESSES
2213      010236      012767      000442      MOV      #TEMP1, ADR1
2214      010244      012767      000444      MOV      #TEMP2, ADR2
2215      010252      012700      000430      MOV      #ADR, R0      ; LOAD ADDRESSES OF ADDRESSES
2216      010256      012701      000432      MOV      #ADR1, R1
2217      010262      012702      000434      MOV      #ADR2, R2
2218      010266      005722      000125      TST      (R2)+      ; ADJUST THE POINTER
2219      010270      022122      000125      MOVB     #125, 2-(R2)      ; LOAD TEMP2
2220      010274      022122      000125      CMP      (R1)+, (R2)+      ; ADJUST THE POINTERS
2221      010276      115251      000125      MOVB     2-(R2), 2-(R1)      ; TEMP2 ---> TEMP1
2222      010300      022022      000125      CMP      (R0)+, (R2)+      ; ADJUST THE POINTERS
2223      010302      115052      000125      MOVB     2-(R0), 2-(R2)      ; TEMP ---> TEMP2
2224      010304      022022      000125      CMP      (R0)+, (R2)+      ; ADJUST THE POINTERS
2225      010306      125052      000125      CMPB     2-(R0), 2-(R2)      ; CHECK IT
2226      010310      001404      000222      BEQ      1$
2227      010312      012745      000222      MOV      #222, -(R5)
2228      010316      005245      000222      INC      -(R5)
2229      010320      000000      000222      HALT
2230      010322      022120      000222      1$:      CMP      (R1)+, (R0)+      ; ADJUST THE POINTERS
2231      010324      115150      000222      MOVB     2-(R1), 2-(R0)      ; TEMP1 ---> TEMP
2232      010326      122767      000125      CMPB     #125, TEMP      ; CHECK IT
2233      010334      001003      000125      BNE      2$      ; FAILED
2234      010336      105767      170102      TSTB     TEMP2      ; CHECK IT
2235      010342      001404      000125      BEQ      3$      ; OK, CONTINUE
2236      010344      012745      000223      2$:      MOV      #223, -(R5)
2237      010346      005245      000223      INC      -(R5)
2238      010350      000000      000223      HALT
2239      010352      005067      170060      3$:      CLR      TEMP      ; INSTRUCTIONS FAILED IN MODE 5
2240      010354      012700      000430      MOV      #ADR, R0      ; START CLEAN
2241      010356      012701      000432      MOV      #ADR1, R1      ; LOAD ADDRESSES OF ADDRESSES
2242      010358      012702      000434      MOV      #ADR2, R2
2243      010360      005722      052525      TST      (R2)+      ; ADJUST THE POINTER
2244      010362      012752      052525      MOVB     #52525, 2-(R2)      ; LOAD TEMP2
2245      010364      022122      052525      CMP      (R1)+, (R2)+      ; ADJUST THE POINTERS
2246      010366      015251      052525      MOVB     2-(R2), 2-(R1)      ; TEMP2 ---> TEMP1
2247      010368      022022      052525      CMP      (R0)+, (R2)+      ; ADJUST THE POINTERS
2248      010370      015052      052525      MOV      2-(R0), 2-(R2)      ; TEMP ---> TEMP2
2249      010372      022021      052525      CMP      (R0)+, (R1)+      ; ADJUST THE POINTERS
2250      010374      015150      052525      MOVB     2-(R1), 2-(R0)      ; TEMP1 ---> TEMP
2251      010376      022767      052525      CMPB     #52525, TEMP      ; CHECK IT
2252      010378      001003      170014      BNE      EMODE5      ; FAILED
2253      010380      005767      170012      TST      TEMP2      ; CHECK IT
2254      010382      001404      000224      BEQ      MODE6      ; OK, CONTINUE
2255      010384      012745      000224      EMODE5:      MOV      #224, -(R5)
2256      010386      005245      000224      INC      -(R5)
2257
2258
2259

```



```

2260 010442 000000 HALT ; INSTRUCTIONS FAILED IN MODE 5
2261 ; OR WRONG SEQUENCE
2262
2263
2264 ;*****
2265 ;*TEST: 61 CHECK MODE 6 USING THE MOVB AND MOV INSTRUCTIONS
2266 ;*****
2267
2268 010444 MODE6:
2269 010444 021527 000061 CMP (R5),#61
2270 010450 001055 BNE EMODE6 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2271 010452 005215 INC (R5)
2272 010454 005067 167764 CLR TEMP2 ; START CLEAN
2273 010460 012700 000440 MOV #TEMP,R0 ; LOAD ADDRESSES
2274 010464 012701 000442 MOV #TEMP1,R1
2275 010470 012702 000444 MOV #TEMP2,R2
2276 010474 112760 000252 000000 MOVB #252,0(R0) ; LOAD TEMP (LOW BYTE)
2277 010502 112760 000252 000001 MOVB #252,1(R0) ; LOAD TEMP (HIGH BYTE)
2278 010510 022767 125252 167722 CMP #125252,TEMP ; CHECK IT
2279 010516 001012 BNE 1$ ; FAILED
2280 010520 116062 000001 000000 MOVB 1(R0),0(R2) ; TEMP(H) ----> TEMP2(L)
2281 010526 116160 000002 000005 MOVB 2(R1),5(R0) ; TEMP2(L) ----> TEMP2(H)
2282 010534 022767 125252 167702 CMP #125252,TEMP2 ; CHECK IT
2283 010542 001404 BEQ 2$ ; OK, CONTINUE
2284 010544
2285 010544 012745 000225 1$: MOV #225,-(R5)
2286 010550 005245 INC -(R5)
2287 010552 000000 HALT ; INSTRUCTIONS FAILED IN MODE 6
2288 010554 005067 167662 2$: CLR TEMP1 ; START CLEAN
2289 010560 012760 052525 000000 MOV #52525,0(R0) ; LOAD TEMP
2290 010566 016260 177774 000002 MOV -4(R2),2(R0) ; TEMP ----> TEMP1
2291 010574 022767 052525 167640 CMP #52525,TEMP1 ; CHECK IT
2292 010602 001404 BEQ MODE7 ; OK, CONTINUE
2293 010604
2294 010604 012745 000226 EMODE6: MOV #226,-(R5)
2295 010610 005245 INC -(R5)
2296 010612 000000 HALT ; INSTRUCTIONS FAILED IN MODE 6
2297 ; OR WRONG SEQUENCE

```

K05

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 62
 DVKAAB.P11 25-OCT-77 13:09 T62

CHECK MODE 7 USING THE MOVB AND MOV INSTRUCTIONS

SEQ 0062

```

2298 ;*****
2299 ;TEST: 62 CHECK MODE 7 USING THE MOVB AND MOV INSTRUCTIONS
2300 ;*****
2301
2302 MODE7:
2303 010614 021527 000062 CMP (R5), #62
2304 010614 001052 BNE EMODE7 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2305 010622 005215 INC (R5)
2306 010624 005067 167612 CLR TEMP1 ; START CLEAN
2307 010630 012767 000440 167572 MOV #TEMP, ADR ; LOAD ADDRESSES
2308 010636 012767 000442 167566 MOV #TEMP1, ADR1
2309 010644 012767 000444 167562 MOV #TEMP2, ADR2
2310 010652 012700 000430 MOV #ADR, R0 ; LOAD ADDRESSES OF ADDRESSES
2311 010656 012701 000432 MOV #ADR1, R1
2312 010662 012702 000434 MOV #ADR2, R2
2313 010666 112770 000252 000000 MOVB #252, 30(R0) ; LOAD TEMP
2314 010674 117270 177774 000002 MOVB 3-4(R2), 32(R0) ; TEMP ----> TEMP1
2315 010702 122767 000252 167532 CMPB #252, TEMP1 ; CHECK IT
2316 010710 001404 BEQ 15 ; OK, CONTINUE
2317 010712 012745 000227 MOV #227, -(R5)
2318 010716 005245 INC -(R5)
2319 010720 000000 HALT ; MODE 7 IS FAILING
2320 010722 012770 125252 000000 15: MOV #125252, 30(R0) ; LOAD TEMP
2321 010730 017270 177774 000002 MOV 3-4(R2), 32(R0) ; TEMP ----> TEMP1
2322 010736 022767 125252 167476 CMPB #125252, TEMP1 ; CHECK IT
2323 010744 001404 BEQ TSTB1 ; OK, CONTINUE
2324 010746
2325 EMODE7: MOV #230, -(R5)
2326 010746 012745 000230 INC -(R5)
2327 010752 005245
2328 010754 000000 HALT ; INSTRUCTIONS FAILED IN MODE 7
; OR WRONG SEQUENCE

```

CHECK BYTE INSTRUCTIONS, NOT DESTINATION MODE 0

 : TEST: 63 NEW INSTRUCTIONS USED IN THIS SECTION ARE TSTB, CLRB, MOVB
 : *****

TSTB1:

```

2339 010756 021527 000063
2340 010756 001042
2341 010762 005215
2342 010764 012700 000440
2343 010766 012701 000442
2344 010772 000277
2345 010776 105010
2346 011000 004737 017230
2347 011002 105710
2348 011006 004737 017230
2349 011010 112711 000377
2350 011014 004737 017316
2351 011020 105711
2352 011024 004737 017316
2353 011026 010002
2354 011032 112762 000200 000000
2355 011034 112241
2356 011042 026127 177777 100200
2357 011044 001404
2358 011052 012745 000231
2359 011054 005245
2360 011060 000000
2361 011062 020102
2362 011064 001404
2363 011070
2364 011070 012745 000232
2365 011074 005245
2366 011076 000000
2367
2368
2369
2370
  
```

```

CMP      (R5), #63
BNE      ETSTB1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF TEST
23:      INC      (R5)
MOV      #TEMP, R0    ; LOAD ADDRESSES
MOV      #TEMP1, R1
SCC
CLRB     (R0)          ; CLEAR THE LOCATION
JSR      PC, @#SCC4    ; CHECK FOR CC = 4
TSTB     (R0)          ; CHECK IT
JSR      PC, @#SCC4    ; CHECK FOR CC = 4
MOVB     #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
MOVB     #200, 0(R2)    ; PLACE #200 IN LOCATION TEMP
MOVB     (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)
4$:      HALT
CMP      R1, R2
BEQ      CMPB1
ETSTB1:  MOV      #232, -(R5)
INC      -(R5)
          ; MOVB INSTRUCTION FAILED OR WRONG SEQUENCE
  
```


M05

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 64

DVKAAB.P11 25-OCT-77 13:09

T64

NEW INSTRUCTIONS USED IN THIS SECTION ARE CMPB, BISB

SEQ 0064

```

2371
2372
2373
2374
2375 011100
2376 011100 021527 000064
2377 011104 001032
2378 011106 005215
2379 011110 012701 000444
2380 011114 012702 000440
2381 011120 012711 000077
2382 011124 112704 000377
2383 011130 150412
2384 011132 004737 017316
2385 011136 120412
2386 011140 001404
2387 011142 012745 000233
2388 011146 005245
2389 011150 000000
2390 011152 121112
2391 011154 100004
2392 011156 012745 000234
2393 011162 005245
2394 011164 000000
2395 011166 121211
2396 011170 100404
2397 011172
2398 011172 012745 000235
2399 011176 005245
2400 011200 000000
2401
2402
2403
2404
2405
2406
2407
2408 011202
2409 011202 021527 000065
2410 011206 001404
2411 011210 012745 000236
2412 011214 005245
2413 011216 000000
2414 011220 005215
2415 011222 012703 000440
2416 011226 112713 000377
2417 011232 012700 000442
2418 011236 010001
2419 011240 112721 000252
2420 011244 000277
2421 011246 146013 000000
2422 011252 004737 017146
2423 011256 136113 177777
2424 011262 001404
2425 011264 012745 000237
2426 011270 005245

```

```

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

```

```

CMPB1:
      CMP      (R5), #64
      BNE      ECMPB1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
1$:   INC      (R5)
      MOV      #TEMP2, R1
      MOV      #TEMP, R2
      MOV      #77, (R1)
      MOV      #377, R4
      BISB     R4, (R2)
      JSR      PC, @#SCC10
      CMPB     R4, (R2)
      BEQ      2$
      MOV      #233, -(R5)
      INC      -(R5)
      HALT
2$:   CMPB     (R1), (R2)
      BPL      3$
      MOV      #234, -(R5)
      INC      -(R5)
      HALT
3$:   CMPB     (R2), (R1)
      BMI      BICB1
      ECMPB1:  MOV      #235, -(R5)
      INC      -(R5)
      HALT

```

```

; LOAD ADDRESS
; PLACE 77 IN LOCATION TEMP2
; R4 SHOULD CONTAIN #177777
; LOAD LOCATION
; CHECK FOR CC = 10
; CHECK COMPARE
; CONTINUE IF OK

; BISB OR CMPB INSTRUCTION FAILED
; CHECK IT AGAIN
; CONTINUE IF OK

; CMPB INSTRUCTION FAILED (WRONG CC)
; ONCE MORE
; CONTINUE IF OK

```

```

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

```

```

BICB1:
      CMP      (R5), #65
      BEQ      2$
      MOV      #236, -(R5)
      INC      -(R5)
      HALT
2$:   INC      (R5)
      MOV      #TEMP, R3
      MOV      #377, (R3)
      MOV      #TEMP1, R0
      MOV      R0, R1
      MOV      #252, (R1)+
      SCC
      BICB     0(R0), (R3)
      JSR      PC, @#SCC1
      BITB     -1(R1), (R3)
      BEQ      4$
      MOV      #237, -(R5)
      INC      -(R5)

```

```

; IF IN WRONG SEQUENCE GO TO HLT BELOW

; PROGRAM IS IN WRONG SEQUENCE

; LOAD ADDRESS
; LOAD LOCATION
; PLACE THE ADDRESS OF LOCATION TEMP1 IN R0
; AND R1
; PLACE #252 IN TEMP1

; CLEAR EVERY OTHER BIT
; CHECK FOR CC = 1
; CHECK IT
; CONTINUE IF OK

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 65
 DVKAAB.P11 25-OCT-77 13:09 T65

NEW INSTRUCTIONS USED IN THIS SECTION ARE BICB, BITB

SEQ 0065

2427	011272	000000		HALT		; BICB OR BITB INSTRUCTION FAILED
2428	011274	132713	000125	4S: BITB	#125,(R3)	; CHECK IT
2429	011300	004737	017146	JSR	PC,2#SCC1	; CHECK FOR CC = 1
2430	011304	154113		BISB	-(R1),(R3)	; SET THE BITS THAT WERE CLEARED
2431	011306	100404		BMI	6S	; CONTINUE IF OK
2432	011310	012745	000240	MOV	#240,-(R5)	
2433	011314	005245		INC	-(R5)	
2434	011316	000000		HALT		; BITB OR BISB INSTRUCTION FAILED
2435	011320	012746	000177	6S: MOV	#177,-(SP)	; STORE #177 ON THE STACK
2436	011324	142613		BICB	(SP)+,(R3)	; CLEAR ALL THE BITS EXCEPT SIGN BIT
2437	011326	004737	017336	JSR	PC,2#SCC11	; CHECK FOR CC = 11
2438	011332	132713	000377	BITB	#377,(R3)	; CHECK IT
2439	011336	004737	017336	JSR	PC,2#SCC11	; CHECK FOR CC = 11
2440	011342	010300		MOV	R3,R0	; PLACE THE ADDRESS OF LOCATION TEMP IN R0
2441	011344	012710	000442	MOV	#TEMP1,(R0)	; PLACE THE ADDRESS OF LOCATION TEMP1 IN TEMP
2442	011350	012730	000377	MOV	#377,2(R0)+	; WRITE A 377 IN LOCATION TEMP1
2443	011354	000263		SEVC		; SET V & C BITS
2444	011356	145070	000000	BICB	2-(R0),2(R0)	; BIT CLEAR THE CONTENTS
2445						; OF TEMP1 TO THE CONTENTS OF TEMP1
2446	011362	004737	017252	JSR	PC,2#SCC5	; CHECK FOR CC = 5
2447	011366	022027	000442	CMP	(R0)+,#TEMP1	; MAKE SURE THAT (R0) IS POINTING TO LOCATION TEMP1
2448	011372	001404		BEQ	8S	
2449	011374	012745	000241	MOV	#241,-(R5)	
2450	011400	005245		INC	-(R5)	
2451	011402	000000		HALT		; BICB OR CMP INSTRUCTION FAILED IN THE SPECIFIC MODE
2452	011404	005750		8S: TST	2-(R0)	; TEST LOCATION TEMP1
2453	011406	001404		BEQ	10S	
2454	011410	012745	000242	MOV	#242,-(R5)	
2455	011414	005245		INC	-(R5)	
2456	011416	000000		HALT		; BICB INSTRUCTION FAILED
2457	011420	000257		10S: CCC		
2458	011422	141010		BICB	(R0),(R0)	; CLEAR THE LOCATION TEMP
2459	011424	004737	017230	JSR	PC,2#SCC4	; CHECK FOR CC = 4


```

2460      ;*****
2461      ;*TEST: 66      NEW INSTRUCTIONS USED IN THIS SECTION ARE INCB, DECB
2462      ;*****
2463
2464      INCB1:
2465      011430      021527      000066      CMP      (R5), #66
2466      011434      001067      BNE      EINCBI      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
2467      011436      005215      INC      (R5)
2468      011440      012704      000440      MOV      #TEMP, R4      ; LOAD ADDRESS
2469      011444      112714      000177      MOV      #177, (R4)      ; TEMP LOCATION=177
2470      011450      000261      SEC
2471      011452      105214      INCB      (R4)      ; ADD ONES INTO LOCATION
2472      011454      004737      017400      JSR      PC, @#SCC13      ; CHECK FOR CC = 13
2473      011460      012714      000376      MOV      #376, (R4)
2474      011464      012700      017336      MOV      #SCC11, R0      ; MAKE R0 POINT TO CHECKING ROUTINE FOR CC = 11
2475      011470      105224      INCB      (R4)+
2476      011472      004720      JSR      PC, (R0)+      ; CHECK FOR CC = 11
2477      011474      105744      TST      -(R4)      ; DECREMENT R4 BY 1
2478      011476      005746      TST      -(SP)      ; AND SP BY 2
2479      011500      010426      MOV      R4, (SP)+      ; PLACE THE ADDRESS OF TEMP ON THE STACK
2480      011502      000241      CLC      ; CLEAR C BIT
2481      011504      105256      INCB      @-(SP)      ; INCREMENT THE CONTENTS OF LOCATION TEMP
2482      011506      004737      017230      JSR      PC, @#SCC4      ; CHECK FOR CC = 4
2483      011512      123634      CMPB     @-(SP)+, @-(R4)+      ; RESTORE STACK POINTER
2484      011514      000261      SEC      ; SET C BIT
2485      011516      105264      177777      INCB      -1(R4)
2486      011522      004737      017146      JSR      PC, @#SCC1      ; CHECK FOR CC = 1
2487      011526      124427      000001      CMPB     -(R4), #1      ; CHECK IT
2488      011532      001404      BEQ      2$      ; CONTINUE IF OK
2489      011534      012745      000243      MOV      #243, -(R5)
2490      011540      005245      INC      -(R5)
2491      011542      000000      HALT
2492      011544      000261      2$:      SEC
2493      011546      105314      DECB     (R4)      ; SUBTRACT ONES FROM LOCATION
2494      011550      004737      017252      JSR      PC, @#SCC5      ; CHECK FOR CC = 5
2495      011554      105324      DECB     (R4)+
2496      011556      004740      JSR      PC, -(R0)      ; CHECK FOR CC = 11
2497      011560      112764      000200      177777      MOV      #200, -1(R4)
2498      011566      105344      DECB     -(R4)
2499      011570      004760      177650      JSR      PC, SCC3-SCC11(R0)      ; CHECK FOR CC = 3
2500      011574      105364      000000      DECB     0(R4)
2501      011600      004737      017146      JSR      PC, @#SCC1      ; CHECK FOR CC = 1
2502      011604      126427      000000      000176      CMPB     0(R4), #176
2503      011612      001404      BEQ      COMBI
2504      011614      EINCBI:
2505      011614      012745      000244      MOV      #244, -(R5)
2506      011620      005245      INC      -(R5)
2507      011622      000000      HALT      ; DECB INSTRUCTION FAILED OR SEQUENCE ERROR
2508
2509
2510
2511
2512
2513
2514
2515

```



```

2516
2517
2518
2519
2520
2521 011624
2522 011624 021527 000067
2523 011630 001404
2524 011632 012745 000245
2525 011636 005245
2526 011640 000000
2527 011642 005215
2528 011644 012703 000440
2529 011650 012704 000442
2530 011654 012714 000252
2531 011660 112413
2532 011662 000277
2533 011664 105113
2534 011666 004737 017146
2535 011672 122713 000125
2536 011676 001404
2537 011700 012745 000246
2538 011704 005245
2539 011706 000000
2540 011710 000277
2541 011712 105113
2542 011714 004737 017336
2543 011720 010400
2544 011722 126013 177777
2545 011726 001404
2546 011730 012745 000247
2547 011734 005245
2548 011736 000000
2549 011740 112724 000377
2550 011744 114413
2551 011746 000277
2552 011750 105113
2553 011752 004737 017252

```

```

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

```

COMB1:

```

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

```

```

2554
2555
2556
2557
2558 011756
2559 011756 021527 000070
2560 011756 001027
2561 011756 005215
2562 011756 012700 000440
2563 011772 112710 000001
2564 011776 105410
2565 012000 004737 017336
2566 012004 122710 000377
2567 012010 001404
2568 012012 012745 000250
2569 012016 005245
2570 012020 000000
2571 012022 012710 000200
2572 012026 105410
2573 012030 004737 017400
2574 012034 122710 000200
2575 012040 001404
2576 012042
2577 012042 012745 000251
2578 012046 005245
2579 012050 000000
2580
2581
2582
2583
2584
2585
2586
2587 012052
2588 012052 021527 000071
2589 012056 001030
2590 012060 005215
2591 012062 012701 000442
2592 012066 112711 000040
2593 012072 000257
2594 012074 106111
2595 012076 106111
2596 012100 004737 017360
2597 012104 122711 000200
2598 012110 001404
2599 012112 012745 000252
2600 012116 005245
2601 012120 000000
2602 012122 106111
2603 012124 004737 017274
2604 012130 106111
2605 012132 122711 000001
2606 012136 001404
2607 012140
2608 012140 012745 000253
2609 012144 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$:   INC      (R5)
      MOV      #TEMP,R0    ; LOAD ADDRESS
      MOVB     #1,(R0)     ; LOAD THE LOCATION
      NEGB     (R0)         ; 2'S COMPLEMENT
      JSR      PC,#$CC11   ; CHECK FOR CC = 11
      CMPB     #377,(R0)   ; CHECK IT
      BEQ      2$          ; CONTINUE IF OK
      MOV      #250,-(R5)
      INC      -(R5)
      HALT
2$:   MOV      #200,(R0)
      NEGB     (R0)         ; NEGB INSTRUCTION FAILED
      JSR      PC,#$CC13   ; 2'S COMPLEMENT
      CMPB     #200,(R0)   ; CHECK FOR CC = 13
      BEQ      ROLB1       ; CHECK IT
      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
      ROLB     (R1)         ; CLEAR FLAGS
      ROLB     (R1)         ; SHIFT
      JSR      PC,#$CC12   ; CHECK FOR CC = 12
      CMPB     #200,(R1)   ; CHECK IT
      BEQ      1$          ; CONTINUE IF OK
      MOV      #252,-(R5)
      INC      -(R5)
      HALT
1$:   ROLB     (R1)
      JSR      PC,#$CC7    ; ROLB INSTRUCTION FAILED
      ROLB     (R1)         ; SHIFT
      CMPB     #1,(R1)     ; CHECK FOR CC = 7
      BEQ      RORB1       ; CHECK IT
      MOV      #253,-(R5)
      INC      -(R5)
      EROLB1:

```

E06

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 69
DVKAAB.P11 25-OCT-77 13:09 T71 NEW INSTRUCTION IN THIS SECTION IS ROLB

SEQ 0069

2610 012146 000000

HALT

; WRONG RESULT AT TEMP1 OR WRONG SEQUENCE


```

2611
2612
2613
2614
2615 012150
2616 012150 021527 000072
2617 012154 001030
2618 012156 005215
2619 012160 012702 000442
2620 012164 112712 000004
2621 012170 000257
2622 012172 106012
2623 012174 106012
2624 012176 122712 000001
2625 012202 001404
2626 012204 012745 000254
2627 012210 005245
2628 012212 000000
2629 012214 106012
2630 012216 004737 017274
2631 012222 106012
2632 012224 004737 017360
2633 012230 122712 000200
2634 012234 001404
2635 012236
2636 012236 012745 000255
2637 012242 005245
2638 012244 000000
2639
2640
2641
2642
2643
2644
2645
2646 012246
2647 012246 021527 000073
2648 012252 001404
2649 012254 012745 000256
2650 012260 005245
2651 012262 000000
2652 012264 005215
2653 012266 012703 000442
2654 012272 112713 000040
2655 012276 000257
2656 012300 106313
2657 012302 106313
2658 012304 004737 017360
2659 012310 122713 000200
2660 012314 001404
2661 012316 012745 000257
2662 012322 005245
2663 012324 000000
2664 012326 106313
2665 012330 004737 017274
2666 012334 106313

```

```

*****
;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)      ; CHECK IT
      BEQ      1$          ; CONTINUE IF OK
      MOV      #254, -(R5)
      INC      -(R5)
      HALT
1$:   RORB     (R2)      ; RORB INSTRUCTION FAILED
      JSR      PC, @#SCC7    ; SHIFT
      RORB     (R2)      ; CHECK FOR CC = 7
      JSR      PC, @#SCC12   ; SHIFT
      CMP      #200, (R2)    ; CHECK FOR CC = 12
      BEQ      ASLB1        ; CHECK IT
      ASLB1
ERORB1:
      MOV      #255, -(R5)
      INC      -(R5)
      HALT
      ; WRONG RESULT AT TEMP1 OR WRONG SEQUENCE

```

```

*****
;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$:   INC      (R5)      ; PROGRAM IS IN WRONG SEQUENCE
      MOV      #TEMP1, R3
      MOV      #40, (R3)
      CCC
      ASLB     (R3)
      ASLB     (R3)
      JSR      PC, @#SCC12   ; LOAD ADDRESS
      CMP      #200, (R3)    ; LOAD LOCATION
      BEQ      4$          ; CLEAR FLAGS
      MOV      #257, -(R5)   ; SHIFT
      INC      -(R5)
      HALT
4$:   ASLB     (R3)      ; ASLB INSTRUCTION FAILED
      JSR      PC, @#SCC7    ; SHIFT
      ASLB     (R3)      ; CHECK FOR CC = 7

```

G06

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 71
DVKAAB.P11 25-OCT-77 13:09 173

NEW INSTRUCTION IN THIS SECTION IS ASLB

SEQ 0071

2667 012336 004737 017230

JSR PC,3#SCC4 ; CHECK FOR CC = 4

```

2668
2669
2670
2671
2672 012342
2673 012342 021527 000074
2674 012346 001040
2675 012350 005215
2676 012352 012704 000442
2677 012356 012703 000444
2678 012362 112714 000004
2679 012366 000257
2680 012370 106214
2681 012372 106214
2682 012374 122714 000001
2683 012400 001404
2684 012402 012745 000260
2685 012406 005245
2686 012410 000000
2687 012412 106214
2688 012414 004737 017274
2689 012420 106214
2690 012422 004737 017230
2691 012426 112713 000202
2692 012432 106213
2693 012434 106213
2694 012436 004737 017336
2695 012442 122713 000340
2696 012446 001404
2697 012450
2698 012450 012745 000261
2699 012454 005245
2700 012456 000000
2701
2702
2703
2704
2705
2706
2707
2708 012460
2709 012460 021527 000075
2710 012464 001404
2711 012466 012745 000262
2712 012472 005245
2713 012474 000000
2714 012476 005215
2715 012500 012700 000444
2716 012504 105010
2717 012506 000257
2718 012510 105510
2719 012512 004737 017230
2720 012516 000261
2721 012520 105510
2722 012522 000261
2723 012524 105510

```

```

*****
; *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$:     INC     (R5)
        MOV     #TEMP1, R4   ; LOAD ADDRESSES
        MOV     #TEMP2, R3   ; LOAD LOCATION
        MOV     #4, (R4)     ; CLEAR FLAGS
        CCC
        ASRB    (R4)         ; SHIFT
        ASRB    (R4)
        CMP     #1, (R4)     ; CHECK IT
        BEQ     2$           ; CONTINUE IF OK
        MOV     #260, -(R5)
        INC     -(R5)
        HALT
2$:     ASRB    (R4)         ; ASRB INSTRUCTION FAILED
        JSR     PC, @#SCC7   ; SHIFT
        ASRB    (R4)         ; CHECK FOR CC = 7
        JSR     PC, @#SCC4   ; SHIFT
        MOV     #202, (R3)   ; CHECK FOR CC = 4
        ASRB    (R3)         ; LOAD LOCATION
        ASRB    (R3)         ; SHIFT
        JSR     PC, @#SCC11  ; CHECK FOR CC = 11
        CMP     #340, (R3)   ; CHECK IT
        BEQ     ADCB1        ; CONTINUE IF OK
EASRB1: MOV     #261, -(R5)
        INC     -(R5)
        HALT
        ; WRONG RESULT AT TEMP2 OR WRONG SEQUENCE

```

```

*****
; *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)
        HALT
2$:     INC     (R5)         ; PROGRAM IS IN WRONG SEQUENCE
        MOV     #TEMP2, R0   ; LOAD ADDRESS
        CLRB    (R0)         ; CLEAR THE LOCATION
        CCC
        ADCB    (R0)         ; CLEAR FLAGS
        JSR     PC, @#SCC4   ; ADD C BIT = 0
        SEC     C=1          ; CHECK FOR CC = 4
        ADCB    (R0)         ; C=1
        SEC     C=1          ; ADD C BIT=1
        ADCB    (R0)         ; C=1
        ; AGAIN

```


DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 73
 DVKAAB.P11 25-OCT-77 13:09 T75

NEW INSTRUCTION IN THIS SECTION IS ADCB

SEQ 0073

2724	012526	004737	017126		JSR	PC,2#SCC0	:	CHECK FOR CC = 0
2725	012532	122710	000002		CMPB	#2,(R0)	:	CHECK IT
2726	012536	001404			BEQ	4\$	:	CONTINUE IF OK
2727	012540	012745	000263		MOV	#263,-(R5)		
2728	012544	005245			INC	-(R5)		
2729	012546	000000			HALT		:	ADCB INSTRUCTION FAILED
2730	012550	112710	000177	4\$:	MOVB	#177,(R0)	:	LOAD LARGEST POSITIVE BYTE
2731	012554	000261			SEC		:	C=1
2732	012556	105510			ADCB	(R0)	:	ADD C BIT=1
2733	012560	004737	017360		JSR	PC,2#SCC12	:	CHECK FOR CC = 12
2734	012564	122710	000200		CMPB	#200,(R0)	:	CHECK IT
2735	012570	001404			BEQ	6\$	:	CONTINUE IF OK
2736	012572	012745	000264		MOV	#264,-(R5)		
2737	012576	005245			INC	-(R5)		
2738	012600	000000			HALT		:	ADCB INSTRUCTION FAILED
2739	012602	112710	000377	6\$:	MOVB	#377,(R0)	:	LOAD -1
2740	012606	000261			SEC		:	C=1
2741	012610	105510			ADCB	(R0)	:	ADD C BIT=1
2742	012612	004737	017252		JSR	PC,2#SCC5	:	CHECK FOR CC = 5

```

2743
2744
2745
2746
2747 012616 021527 000076
2748 012616 001404
2749 012622 012745 000265
2750 012624 005245
2751 012630 000000
2752 012632 005215
2753 012634 012701 000444
2754 012636 112711 000003
2755 012642 000257
2756 012646 105611
2757 012650 004737 017126
2758 012652 122711 000003
2759 012656 001404
2760 012662 012745 000266
2761 012664 005245
2762 012670 000000
2763 012672 000261
2764 012674 105611
2765 012676 000261
2766 012700 105611
2767 012702 004737 017126
2768 012704 122711 000001
2769 012710 001404
2770 012714 012745 000267
2771 012716 005245
2772 012722 000000
2773 012724 000261
2774 012730 105611
2775 012732 004737 017230
2776 012736 000261
2777 012740 105611
2778 012742 004737 017336
2779 012746 122711 000377
2780 012752 001404
2781 012754 012745 000270
2782 012760 005245
2783 012762 000000
2784 012764 112711 000200
2785

```

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

K06

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 75
DVKAAB.P11 25-OCT-77 13:09 T76 NEW INSTRUCTION IN THIS SECTION IS SBCB

SEQ 0075

2786 012770 000261
2787 012772 105611
2788 012774 004737 017166

SEC
SBCB (R1)
JSR PC,0#SCC2 ; C=1
; SUBTRACT C BIT = 1
; CHECK FOR CC = 2

CHECK WORD INSTRUCTIONS, NOT DESTINATION MODE 0

```

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

```

```

TST1:
      CMP      (R5), #77
      BEQ      IS
      MOV      #271, -(R5)
      INC      -(R5)
      HALT
IS:   INC      (R5)
      MOV      #TEMP, R1
      MOV      #TEMP1, R0
      SCC
      CLR      (R0)
      JSR      PC, @#SCC4
      TST      (R0)+
      JSR      PC, @#SCC4
      MOV      R0, -(R0)
      MOV      #177777, @ (R0)+
      MOV      @-2(R0), (R1)
      JSR      PC, @#SCC10
      TST      (R1)
      JSR      PC, @#SCC10
; IF IN WRONG SEQUENCE GO TO HLT
; TEST IS IN A WRONG SEQUENCE
; LOAD ADDRESSES
; CLEAR THE LOCATION
; CHECK FOR CC = 4
; CHECK IT
; CHECK FOR CC = 4
; LOAD THE LOCATION
; CHECK FOR CC = 10
; CHECK IT
; CHECK FOR CC = 10

```

```

2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802 013000 021527 000077
2803 013000 001404
2804 013004 001404
2805 013006 012745 000271
2806 013012 005245
2807 013014 000000
2808 013016 005215
2809 013020 012701 000440
2810 013024 012700 000442
2811 013030 000277
2812 013032 005010
2813 013034 004737 017230
2814 013040 005720
2815 013042 004737 017230
2816 013046 010040
2817 013050 012730 177777
2818 013054 017011 177776
2819 013060 004737 017316
2820 013064 005711
2821 013066 004737 017316
2822
2823
2824

```

```

2825 013072 021527 000100
2826 013072 001113
2827 013100 005215
2828 013102 012702 000442
2829 013106 012700 000440
2830 013112 012720 177777
2831 013116 054012
2832 013120 004737 017316
2833 013124 022227 177777
2834 013130 001404
2835 013132 012745 000272
2836 013136 005245
2837 013140 000000
2838 013142 020227 000444
2839 013146 001404
2840 013150 012745 000273
2841 013154 005245
2842 013156 000000
2843 013160 022742 000077
2844 013164 004737 017146
2845 013170 022722 077777
2846 013174 004737 017400
2847 013200 024227 077777
2848 013204 004737 017316
2849 013210 012767 052525 165226
2850 013216 012767 000444 165216
2851 013224 012704 000430
2852 013230 012714 000432
2853 013234 012734 125252
2854 013240 057432 177776
2855 013244 010200
2856 013246 025027 177777
2857 013252 001404
2858 013254 012745 000274
2859 013260 005245
2860 013262 000000
2861 013264 020227 000444
2862 013270 001404
2863 013272 012745 000275
2864 013276 005245
2865 013300 000000
2866 013302 005040
2867 013304 010067 165134
2868 013310 022020
2869 013312 055070 000002
2870 013316 022767 000440 165114
2871 013324 001404
2872 013326
2873 013326
2874 013326
2875 013326
2876 013326
2877 013326
2878 013326
2879 013326
2880 013326 012745 000276

CMP1:
CMP (R5), #100
BNE ECMP1
1S: INC (R5)
MOV #TEMP1, R2
MOV #TEMP, R0
MOV #177777, (R0)+
BIS -(R0), (R2)
JSR PC, #SCC10
CMP (R2)+, #177777
BEQ 2S
MOV #272, -(R5)
INC -(R5)
HALT
2S: CMP R2, #TEMP1+2
BEQ 3S
MOV #273, -(R5)
INC -(R5)
HALT
3S: CMP #77, -(R2)
JSR PC, #SCC1
CMP #77777, (R2)+
JSR PC, #SCC13
CMP -(R2), #77777
JSR PC, #SCC10
MOV #52525, TEMP2
MOV #TEMP2, TEMP1
MOV #ADR, R4
MOV #ADR1, (R4)
MOV #125252, #2(R4)+
BIS #2-2(R4), #2(R2)+
MOV R2, R0
CMP #-(R0), #177777
BEQ 4S
MOV #274, -(R5)
INC -(R5)
HALT
4S: CMP R2, #TEMP1+2
BEQ 5S
MOV #275, -(R5)
INC -(R5)
HALT
5S: CLR -(R0)
MOV R0, TEMP2
CMP (R0)+, (R0)+
BIS #-(R0), #2(R0)
CMP #TEMP, TEMP
BEQ ECMP1
ECMP1: MOV #276, -(R5)

; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE EST
; LOAD ADDRESS
; PLACE THE ADDRESS OF TEMP IN R0
; PLACE #177777 IN LOCATION TEMP AND INC. R0 BY 2
; LOAD LOCATION
; CHECK FOR CC = 10
; CHECK COMPARE
; CONTINUE IF OK
; CMP OR BIS INSTRUCTION FAILED
; CHECK R2 TO CONTAIN ADDRESS OF TEMP1+2
; NO AUTO INCREMENT
; CHECK IT AGAIN
; CHECK FOR CC = 1
; CHECK FOR CC = 13
; ONCE MORE
; CHECK FOR CC = 10
; SET EVERY OTHER BIT IN TEMP2
; PLACE THE ADDRESS OF TEMP2 IN LOCATION TEMP1
; PLACE THE ADDRESS OF ADR1 IN ADR POINTED BY R4
; PLACE THE #125252 IN LOCATION ADR1
; SET EVERY OTHER BIT AT LOCATION TEMP2
; AND INCREMENT R2 BY 2
; PLACE ADDRESS OF TEMP2 IN R0
; TEMP2 SHOULD CONTAIN ALL 1'S
; CMP OR BIS INSTRUCTIONS FAILED IN MODES OTHER THAN 0
; R2 SHOULD CONTAIN THE ADDRESS FOR TEMP2
; I.E. TEMP1+2
; MODE 5 IS FAILING
; PLACE A 0 IN LOCATION TEMP
; PLACE ADDRESS OF TEMP IN LOCATION TEMP2
; BUMP R0 BY 4
; PLACE THE CONTENTS OF LOCATION TEMP2 AT TEMP
; LOCATION TEMP SHOULD CONTAIN ITS OWN ADDRESS

```



```

2881 013332 005245
2882 013334 000000
2883
2884
2885
2886
2887
2888
2889
2890
2891 013336
2892 013336 021527 000101
2893 013342 001122
2894 013344 005215
2895 013346 012703 000440
2896 013352 012713 177777
2897 013356 012704 000430
2898 013362 012714 000432
2899 013366 011334
2900 013370 012700 000442
2901 013374 012710 125252
2902 013400 000277
2903 013402 042013
2904 013404 004737 017146
2905 013410 034013
2906 013412 001404
2907 013414 012745 000277
2908 013420 005245
2909 013422 000000
2910 013424 032713 052525
2911 013430 004737 017146
2912 013434 056013 000000
2913 013440 100404
2914 013442 012745 000300
2915 013446 005245
2916 013450 000000
2917 013452 012720 077777
2918 013456 010002
2919 013460 046213 177776
2920 013464 004737 017336
2921 013470 020027 000444
2922 013474 001404
2923 013476 012745 000301
2924 013502 005245
2925 013504 000000
2926 013506 010020
2927 013510 000263
2928 013512 045000
2929 013514 004737 017252
2930 013520 037413 177776
2931 013524 004737 017336
2932 013530 012746 125252
2933 013534 017423 177776
2934 013540 046643 000000
2935 013544 022327 052525
2936 013550 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
INC (R5)
MOV #TEMP, R3
MOV #177777, (R3)
MOV #ADR, R4
MOV #ADR1, (R4)
MOV (R3), (R4)+
MOV #TEMP1, R0
MOV #125252, (R0)
SCC
BIC (R0)+, (R3)
JSR PC, @#SCC1
BIT -(R0), (R3)
BEQ 1$
MOV #277, -(R5)
INC -(R5)
HALT
1$: BIT #52525, (R3)
JSR PC, @#SCC1
BIS 0(R0), (R3)
BMI 2$
MOV #300, -(R5)
INC -(R5)
HALT
2$: MOV #77777, (R0)+
MOV R0, R2
BIC -2(R2), (R3)
JSR PC, @#SCC11
CMP R0, #TEMP1+2
BEQ 3$
MOV #301, -(R5)
INC -(R5)
HALT
3$: MOV R0, (R0)+
SEVC
BIC 2-(R0), R0
JSR PC, @#SCC5
BIT 2-(R4), (R3)
JSR PC, @#SCC11
MOV #125252, -(SP)
MOV 2-(R4), (R3)+
BIC 0(SP), -(R3)
CMP (R3)+, #52525
BEQ 4$

```

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

```

; LOAD ADDRESS
; LOAD LOCATION
; PLACE THE ADDRESS OF ADR IN R4
; PLACE THE ADDRESS OF ADR1 IN ADR
; LOAD LOCATION ADR1 WITH #177777
; PLACE THE ADDRESS OF TEMP1 IN R0
; SET EVERY OTHER BIT AT LOCATION TEMP1

```

```

; CLEAR EVERY OTHER BIT
; CHECK FOR CC = 1
; CHECK IT
; CONTINUE IF OK

```

```

; BIC OR BIT INSTRUCTION FAILED
; CHECK IT
; CHECK FOR CC = 1
; SET THE BITS THAT WERE CLEARED
; CONTINUE IF OK

```

```

; BIT OR BIS INSTRUCTION FAILED
; SET ALL THE BITS AT LOCATION TEMP1 EXCEPT SIGN BIT
; TRY CLEARING THE OTHER BITS
; CHECK FOR CC = 11
; R0 SHOULD CONTAIN THE ADDRESS OF TEMP1+2

```

```

; PLACE THE ADDRESS OF LOCATION TEMP2 IN TEMP2
; SET V & C BITS
; CLEAR R0
; CHECK FOR CC = 5
; CHECK IT
; CHECK FOR CC = 11
; SET EVERY OTHER BIT ON THE STACK
; SET ALL THE BITS AT LOCATION TEMP
; CLEAR EVERY OTHER BIT AT LOCATION TEMP
; TEMP SHOULD CONTAIN # 52525

```


DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 79
 DVKAAB.P11 25-OCT-77 13:09 T101

NEW INSTRUCTIONS USED IN THIS SECTION ARE BIC, BIT

SEQ 0079

2937	013552	012745	000302		MOV	#302, -(R5)	
2938	013556	005245			INC	-(R5)	
2939	013560	000000			HALT		
2940	013562	012700	000446	4S:	MOV	#TEMP2+2, R0	; BIC FAILED IN MODE 6
2941	013566	010340			MOV	R3, -(R0)	; PLACE THE ADDRESS OF TEMP2+2 IN R0
2942	013570	014330			MOV	-(R3), 2(R0)+	; PLACE THE ADDRESS OF TEMP1 IN TEMP2
2943	013572	000263			SEVC		; MOVE # 52525 IN LOCATION TEMP1
2944	013574	035026			BIT	2-(R0), (SP)+	; SET V & C BITS
2945	013576	004737	017252		JSR	PC, 2#SCC5	; BIT TEST TEMP1 WITH STACK AND RESTORE STACK POINTER
2946	013602	020627	000530		CMP	SP, #START	; CHECK FOR CC = 5
2947	013606	001404			BEG	INC1	; MAKE SURE THAT THE SP IS OK
2948	013610			EBIC1:			
2949	013610	012745	000303		MOV	#303, -(R5)	
2950	013614	005245			INC	-(R5)	
2951	013616	000000			HALT		; STACK POINTER FOULED UP OR SEQUENCE ERROR

```

2952          013620 021527 000102
2953          013620 001404
2954          013624 012745 000304
2955          013632 005245
2956          013634 000000
2957          013636 005215
2958          013640 012704 000442
2959          013644 012714 077777
2960          013650 000261
2961          013652 005214
2962          013654 004737 017400
2963          013656 012714 177776
2964          013660 012700 000440
2965          013670 012710 017336
2966          013674 005214
2967          013676 004730
2968          013700 005214
2969          013702 004737 017252
2970          013706 005214
2971          013710 004737 017146
2972          013714 026427 000000 000001
2973          013722 001404
2974          013724 012745 000305
2975          013730 005245
2976          013732 000000
2977          013734 000261
2978          013736 005314
2979          013740 004737 017252
2980          013744 005314
2981          013746 004770 177776
2982          013752 012714 100000
2983          013756 005314
2984          013760 004737 017206
2985          013764 005314
2986          013766 004737 017146
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000          013772 021527 000103
3001          013776 001404
3002          014000 012745 000306
3003          014004 005245
3004          014006 000000
3005          014010 005215
3006          014012 012703 000442
3007

```

 ;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
 INC (R4)
 JSR PC, @ (R0)+ ; CHECK FOR CC = 11
 INC (R4)
 JSR PC, @#SCC5 ; CHECK FOR CC = 5
 INC (R4)
 JSR PC, @#SCC1 ; CHECK FOR CC = 1
 CMP 0(R4), #1 ; CHECK IT
 BEQ 4\$; CONTINUE IF OK
 MOV #305, -(R5)
 INC -(R5)
 HALT ; INC INSTRUCTION FAILED
 4\$: SEC
 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

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 81

DVKAAB.P11 25-OCT-77 13:09 T103

NEW INSTRUCTION IN THIS SECTION IS COM

SEQ 0081

3008	014016	012713	125252	MOV	#125252, (R3)	; LOAD EVERY OTHER BIT
3009	014022	000277		SCC		
3010	014024	005163	000000	COM	0(R3)	; 1'S COMPLEMENT
3011	014030	004737	017146	JSR	PC, 2#SCC1	; CHECK FOR CC = 1
3012	014034	022713	052525	CMP	#52525, (R3)	; CHECK IT
3013	014040	001404		BEQ	2\$	; CONTINUE IF OK
3014	014042	012745	000307	MOV	#307, -(R5)	
3015	014046	005245		INC	-(R5)	
3016	014050	000000		HALT		; COM INSTRUCTION FAILED
3017	014052	000277		SCC		
3018	014054	005123		COM	(R3)+	; COMPLEMENT BACK
3019	014056	004737	017336	JSR	PC, 2#SCC11	; CHECK FOR CC = 11
3020	014062	022743	125252	CMP	#125252, -(R3)	; CHECK IT
3021	014066	001404		BEQ	3\$	; CONTINUE IF OK
3022	014070	012745	000310	MOV	#310, -(R5)	
3023	014074	005245		INC	-(R5)	
3024	014076	000000		HALT		; COM INSTRUCTION FAILED
3025	014100	010300		MOV	R3, R0	; R0 IS NOW POINTING TO LOCATION TEMP1
3026	014102	012710	177777	MOV	#177777, (R0)	
3027	014106	000277		SCC		
3028	014110	005110		COM	(R0)	
3029	014112	004737	017252	JSR	PC, 2#SCC5	; CHECK FOR CC = 5

3030				::*****		
3031				::*TEST: 104	NEW INSTRUCTION IN THIS SECTION IS NEG	
3032				::*****		
3033						
3034	014116			NEG1:	CMP	(R5), #104
3035	014116	021527	000104		BNE	ENEG1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
3036	014122	001033		1\$:	INC	(R5)
3037	014124	005215			MOV	#TEMP1, R4 ; LOAD ADDRESS
3038	014126	012704	000442		MOV	#1, (R4)+ ; LOAD THE LOCATION
3039	014132	012724	000001		MOV	R4, R2
3040	014136	010402			MOV	#100000, 0(R2)
3041	014140	012762	100000 000000		NEG	-(R4) ; 2'S COMPLEMENT
3042	014146	005444			JSR	PC, @#SCC11 ; CHECK FOR CC = 11
3043	014150	004737	017336		CMP	#177777, (R4)+ ; CHECK IT
3044	014154	022724	177777		BEQ	2\$; CONTINUE IF OK
3045	014160	001404			MOV	#311, -(R5)
3046	014162	012745	000311		INC	-(R5)
3047	014166	005245			HALT	
3048	014170	000000		2\$:	MOV	0(R4), -(R4) ; NEG INSTRUCTION FAILED
3049	014172	016444	000000		NEG	(R4) ; TEMP1 CONTAINS THE LARGEST NEGATIVE NUMBER
3050	014176	005414			JSR	PC, @#SCC13 ; 2'S COMPLEMENT
3051	014200	004737	017400		CMP	0(R2), (R4) ; CHECK FOR CC = 13
3052	014204	026214	000000		BEQ	ROL1 ; CHECK IT
3053	014210	001404				
3054	014212			ENEG1:	MOV	#312, -(R5) ; CONTINUE IF OK
3055	014212	012745	000312		INC	-(R5)
3056	014216	005245			HALT	
3057	014220	000000				; WRONG RESULT IN TEMP2 OR WRONG SEQUENCE
3058						
3059						
3060						
3061				::*****		
3062				::*TEST: 105	NEW INSTRUCTION IN THIS SECTION IS ROL	
3063				::*****		
3064						
3065	014222			ROL1:	CMP	(R5), #105
3066	014222	021527	000105		BNE	EROL1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
3067	014226	001032			INC	(R5)
3068	014230	005215			MOV	#TEMP2, R1 ; LOAD ADDRESS
3069	014232	012701	000444		MOV	#20000, (R1) ; LOAD LOCATION
3070	014236	012711	020000		CCC	; CLEAR FLAGS
3071	014242	000257			ROL	(R1)+ ; SHIFT
3072	014244	006121			ROL	-(R1)
3073	014246	006141			JSR	PC, @#SCC12 ; CHECK FOR CC = 12
3074	014250	004737	017360		CMP	#100000, (R1) ; CHECK IT
3075	014254	022711	100000		BEQ	1\$; CONTINUE IF OK
3076	014260	001404			MOV	#313, -(R5)
3077	014262	012745	000313		INC	-(R5)
3078	014266	005245			HALT	
3079	014270	000000				
3080	014272	006161	000000	1\$:	ROL	0(R1) ; ROL INSTRUCTION FAILED
3081	014276	004737	017274		JSR	PC, @#SCC7 ; CHECK FOR CC = 7
3082	014302	010102			MOV	R1, R2 ; R2 IS NOW POINTING TO LOCATION TEMP2
3083	014304	006112			ROL	(R2) ; SHIFT
3084	014306	022711	000001		CMP	#1, (R1) ; CHECK IT
3085	014312	001404			BEQ	ROL1 ; CONTINUE IF OK

F07

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 83
DVKAAB.P11 25-OCT-77 13:09 T105 NEW INSTRUCTION IN THIS SECTION IS ROL

SEQ 0083

3086 014314
3087 014314 012745 000314
3088 014320 005245
3089 014322 000000

EROL1:

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

; WRONG RESULT AT TEMP2 OR WRONG SEQUENCE

```

3090
3091
3092
3093
3094 014324
3095 014324 021527 000106
3096 014330 001030
3097 014332 005215
3098 014334 012702 000444
3099 014340 012712 000004
3100 014344 000257
3101 014346 006012
3102 014350 006012
3103 014352 022712 000001
3104 014356 001404
3105 014360 012745 000315
3106 014364 005245
3107 014366 000000
3108 014370 006012
3109 014372 004737 017274
3110 014376 006012
3111 014400 004737 017360
3112 014404 022712 100000
3113 014410 001404
3114 014412
3115 014412 012745 000316
3116 014416 005245
3117 014420 000000
3118
3119
3120
3121
3122
3123
3124
3125 014422
3126 014422 021527 000107
3127 014426 001404
3128 014430 012745 000317
3129 014434 005245
3130 014436 000000
3131 014440 005215
3132 014442 012703 000444
3133 014446 012713 020000
3134 014452 000257
3135 014454 006313
3136 014456 006313
3137 014460 004737 017360
3138 014464 022713 100000
3139 014470 001404
3140 014472 012745 000320
3141 014476 005245
3142 014500 000000
3143 014502 006313
3144 014504 004737 017274
3145 014510 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      ; 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      #100000, (R2)      ; CHECK FOR CC = 12
      BEQ      ASL1      ; CHECK IT
      ; CONTINUE IF OK
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$:
      INC      (R5)      ; PROGRAM IS IN WRONG SEQUENCE
      MOV      #TEMP2, R3
      MOV      #20000, (R3)      ; LOAD ADDRESS
      CCC      ; LOAD LOCATION
      ASL      (R3)      ; CLEAR FLAGS
      ASL      (R3)      ; SHIFT
      JSR      PC, @#SCC12      ; CHECK FOR CC = 12
      CMP      #100000, (R3)      ; CHECK IT
      BEQ      4$      ; 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
      ; SHIFT

```


H07

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 85
DVKAAB.P11 25-OCT-77 13:09 T107

NEW INSTRUCTION IN THIS SECTION IS ASL

SEQ 0085

3146 014512 004737 017230

JSR PC,2#SCC4 ; CHECK FOR CC = 4

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 86
 DVKAAB.P11 25-OCT-77 13:09 T110 NEW INSTRUCTION IN THIS SECTION IS ASR

SEQ 0086

```

3147
3148
3149
3150
3151 014516 021527 000110
3152 014516 001040
3153 014522 005215
3154 014524 012704 000444
3155 014526 012703 000440
3156 014532 012714 000004
3157 014536 000257
3158 014542 006214
3159 014544 006214
3160 014546 006214
3161 014550 022714 000001
3162 014554 001404
3163 014556 012745 000321
3164 014562 005245
3165 014564 000000
3166 014566 006214
3167 014570 004737 017274
3168 014574 006214
3169 014576 004737 017230
3170 014602 012713 100002
3171 014606 006213
3172 014610 006213
3173 014612 004737 017336
3174 014616 022713 160000
3175 014622 001404
3176 014624
3177 014624 012745 000322
3178 014630 005245
3179 014632 000000
3180
3181
3182
3183
3184
3185
3186
3187 014634 021527 000111
3188 014634 001404
3189 014640 012745 000323
3190 014642 005245
3191 014646 000000
3192 014650 005215
3193 014654 012700 000440
3194 014660 005010
3195 014662 000257
3196 014664 005510
3197 014666 004737 017230
3198 014672 000261
3200 014674 005510
3201 014676 000261
3202 014700 005510

```

```

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

```

```

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

```

```

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

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 87
 DVKAAB.P11 25-OCT-77 13:09 T111

NEW INSTRUCTION IN THIS SECTION IS ADC

SEQ 0087

3203	014702	004737	017126		JSR	PC, @SCC0	:	CHECK FOR CC = 0
3204	014706	022710	000002		CMP	#2, (R0)	:	CHECK IT
3205	014712	001404			BEQ	4\$	:	CONTINUE IF OK
3206	014714	012745	000324		MOV	#324, -(R5)		
3207	014720	005245			INC	-(R5)		
3208	014722	000000			HALT		:	ADC INSTRUCTION FAILED
3209	014724	012710	077777	4\$:	MOV	#77777, (R0)	:	LOAD LARGEST POSITIVE NUMBER
3210	014730	000261			SEC		:	C=1
3211	014732	005510			ADC	(R0)	:	ADD C BIT=1
3212	014734	004737	017360		JSR	PC, @SCC12	:	CHECK FOR CC = 12
3213	014740	022710	100000		CMP	#100000, (R0)	:	CHECK IT
3214	014744	001404			BEQ	6\$	:	CONTINUE IF OK
3215	014746	012745	000325		MOV	#325, -(R5)		
3216	014752	005245			INC	-(R5)		
3217	014754	000000			HALT		:	ADC INSTRUCTION AILED
3218	014756	012710	177777	6\$:	MOV	#-1, (R0)	:	LOAD -1
3219	014762	000261			SEC		:	C=1
3220	014764	005510			ADC	(R0)	:	ADD C BIT=1
3221	014766	004737	017252		JSR	PC, @SCC5	:	CHECK FOR CC = 5


```

3222 014772 021527 000112
3223 014772 001404
3224 015000 012745 000326
3225 015004 005245
3226 015006 000000
3227 015010 005215
3228 015012 012701 000440
3229 015016 012711 000003
3230 015022 000257
3231 015024 005611
3232 015026 004737 017126
3233 015032 022711 000003
3234 015036 001404
3235 015040 012745 000327
3236 015044 005245
3237 015046 000000
3238 015050 000261
3239 015052 005611
3240 015054 000261
3241 015056 005611
3242 015060 004737 017126
3243 015064 022711 000001
3244 015070 001404
3245 015072 012745 000330
3246 015076 005245
3247 015100 000000
3248 015102 000261
3249 015104 005611
3250 015106 004737 017230
3251 015112 000261
3252 015114 005611
3253 015116 004737 017336
3254 015122 022711 177777
3255 015126 001404
3256 015130 012745 000331
3257 015134 005245
3258 015136 000000
3259 015140 012711 100000
3260 015144 000261
3261 015146 005611
3262 015150 004737 017166
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275 015154 021527 000113
3276 015154 001026
3277 015160

```

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

SBC1:

```

    CMP (R5),#112
    BEQ 1S ; IF IN WRONG SEQUENCE GO TO HLT
    MOV #326,-(R5)
    INC -(R5)
    HALT ; TEST IS IN WRONG SEQUENCE
1S: INC (R5)
    MOV #TEMP,R1 ; LOAD ADDRESS
    MOV #3,(R1) ; LOAD LOCATION
    CCC ; CLEAR FLAGS
    SBC (R1) ; SUBTRACT C BIT=0
    JSR PC,#$SCC0 ; CHECK FOR CC = 0
    CMP #3,(R1) ; CHECK IT
    BEQ 2S ; CONTINUE IF OK
    MOV #327,-(R5)
    INC -(R5)
    HALT ; SBC INSTRUCTION FAILED
2S: SEC ; C=1
    SBC (R1) ; SUBTRACT C BIT=1
    SEC ; C=1
    SBC (R1)
    JSR PC,#$SCC0 ; CHECK FOR CC = 0
    CMP #1,(R1) ; CHECK IT
    BEQ 3S
    MOV #330,-(R5)
    INC -(R5)
    HALT ; SBC INSTRUCTION FAILED
3S: SEC ; C=1
    SBC (R1) ; SUBTRACT C BIT=1
    JSR PC,#$SCC4 ; CHECK FOR CC = 4
    SEC ; C=1
    SBC (R1) ; SUBTRACT C BIT = 1
    JSR PC,#$SCC11 ; CHECK FOR CC = 11
    CMP #-1,(R1) ; CHECK IT
    BEQ 4S ; CONTINUE IF OK
    MOV #331,-(R5)
    INC -(R5)
    HALT ; SBC INSTRUCTION FAILED
4S: MOV #100000,(R1) ; LOAD R1
    SEC ; C=1
    SBC (R1) ; SUBTRACT C BIT = 1
    JSR PC,#$SCC2 ; CHECK FOR CC = 2

```

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

SXT1:

```

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

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 89
 DVKAAB.P11 25-OCT-77 13:09 T113

NEW INSTRUCTION IN THIS SECTION IS SXT

SEQ 0089

3278	015162	005215		1S:	INC	(R5)		
3279	015164	012702	000442		MOV	#TEMP1,R2	:	LOAD ADDRESS
3280	015170	005012			CLR	(R2)	:	CLEAR LOCATIONS
3281	015172	000277			SCC			
3282	015174	000254			CLNZ			
3283	015176	006712			SXT	(R2)	:	SIGN EXTEND
3284	015200	004737	017252		JSR	PC, #SCC5	:	CHECK FOR CC = 5
3285	015204	005712			TST	(R2)	:	LOCATION SHOULD STILL BE 0
3286	015206	001404			BEQ	2S	:	CONTINUE IF OK
3287	015210	012745	000332		MOV	#332, -(R5)		
3288	015214	005245			INC	-(R5)		
3289	015216	000000			HALT		:	SXT INSTRUCTION FAILED
3290	015220	000273		2S:	SENV		:	SET N, V & C BITS
3291	015222	006712			SXT	(R2)	:	SIGN EXTEND
3292	015224	004737	017336		JSR	PC, #SCC11	:	CHECK FOR CC = 11
3293	015230	022712	177777		CMP	#-1, (R2)	:	LOCATION SHOULD NOW HAVE -1
3294	015234	001404			BEQ	SWAB1	:	CONTINUE IF OK
3295	015236			ESXT1:				
3296	015236	012745	000333		MOV	#333, -(R5)		
3297	015242	005245			INC	-(R5)		
3298	015244	000000			HALT		:	WRONG RESULT IN TEMP1 OR WRONG SEQUENCE


```

3299
3300
3301
3302
3303 015246
3304 015246 021527 000114
3305 015252 001034
3306 015254 005215
3307 015256 012703 000444
3308 015262 012713 125125
3309 015266 000277
3310 015270 000250
3311 015272 000313
3312 015274 004727 017316
3313 015300 022713 052652
3314 015304 001404
3315 015306 012745 000334
3316 015312 005245
3317 015314 000000
3318 015316 012713 000377
3319 015322 000277
3320 015324 000244
3321 015326 000363 000000
3322 015332 004737 017230
3323 015336 022713 177400
3324 015342 001404
3325 015344
3326 015344 012745 000335
3327 015350 005245
3328 015352 000000
3329
3330
3331
3332
3333
3334
3335
3336 015354
3337 015354 021527 000115
3338 015360 001041
3339 015362 005215
3340 015364 012704 177777
3341 015370 012767 177777 163044
3342 015376 000277
3343 015400 074467 163036
3344 015404 004737 017252
3345 015410 012767 077777 163024
3346 015416 012700 000442
3347 015422 000263
3348 015424 000244
3349 015426 074410
3350 015430 004737 017336
3351 015434 012701 125252
3352 015440 012720 052525
3353 015444 000277
3354 015446 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
1$:      MOV      #377, (R3)    ; SWAB INSTRUCTION FAILED
        SCC
        CLZ
        SWAB      0(R3)
        JSR      PC, @#SCC4    ; CHECK FOR CC = 4
        CMP      #177400, (R3)
        BEQ
ESWAB1: MOV      #335, -(R5)
        INC      -(R5)
        HALT                  ; WRONG RESULT IN: TEMP2 OR WRONG SEQUENCE

```

```

*****
;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
        MOV      #-1, TEMP1 ; LOAD LOCATIONS
        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
        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

```


N07

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 91
 DVKAAB.P11 25-OCT-77 13:09 T115

NEW INSTRUCTION IN THIS SECTION IS XOR

SEQ 0091

3355 015450 004737 017336
 3356 015454 022737 177777 000442
 3357 015462 001404
 3358 015464
 3359 015464 012745 000336
 3360 015470 005245
 3361 015472 000000

EXOR1:

JSR PC, @SCC11
 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

```

3362      ;*****
3363      ;*TEST: 116      NEW INSTRUCTION IN THIS SECTION IS ADD
3364      ;*****
3365
3366      ADD1:
3367      015474 021527 000116      CMP      (R5), #116
3368      015500 001133      BNE      EADD1      ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
3369      015502 005215      INC      (R5)
3370      015504 012700 000444      MOV      #TEMP2, R0      ; LOAD ADDRESSES
3371      015510 012701 000440      MOV      #TEMP, R1
3372      015514 012767 021421 162722      MOV      #21421, TEMP2      ; LOAD LOCATIONS
3373      015522 011011      MOV      (R0), (R1)
3374      015524 061011      ADD      (R0), (R1)      ; ADD
3375      015526 004737 017126      JSR      PC, #SCC0      ; CHECK FOR CC = 0
3376      015532 022767 043042 162700      CMP      #43042, TEMP      ; CHECK IT
3377      015540 001404      BEQ      1$      ; CONTINUE IF OK
3378      015542 012745 000337      MOV      #337, -(R5)
3379      015546 005245      INC      -(R5)
3380      015550 000000      HALT
3381      015552 005010      1$:      CLR      (R0)      ; ADD INSTRUCTION FAILED
3382      015554 060020      ADD      R0, (R0)+      ; CLEAR LOCATION TEMP2
3383      015556 024027 000444      CMP      -(R0), #TEMP2      ; PLACE THE ADDRESS OF TEMP2 IN LOCATION TEMP2
3384      015562 001404      BEQ      2$      ; CHECK IT
3385      015564 012745 000340      MOV      #340, -(R5)
3386      015570 005245      INC      -(R5)
3387      015572 000000      HALT
3388      015574 012767 156357 162642 2$:      MOV      #-21421, TEMP2      ; ADD INSTRUCTION FAILED IN MODE 2
3389      015602 012011      MOV      (R0)+, (R1)      ; LOAD LOCATIONS
3390      015604 064011      ADD      -(R0), (R1)
3391      015606 004737 017336      JSR      PC, #SCC11      ; ADD
3392      015612 022767 134736 162620      CMP      #-43042, TEMP      ; CHECK FOR CC = 11
3393      015620 001404      BEQ      3$      ; CHECK IT
3394      015622 012745 000341      MOV      #341, -(R5)      ; CONTINUE IF OK
3395      015626 005245      INC      -(R5)
3396      015630 000000      HALT
3397      015632 012767 100000 162604 3$:      MOV      #100000, TEMP2      ; ADD INSTRUCTION FAILED
3398      015640 011061 000000      MOV      (R0), 0(R1)      ; LOAD LOCATIONS
3399      015644 066011 000000      ADD      0(R0), (R1)      ; ADD SHOULD RESULT AS 0'S
3400      015650 004737 017274      JSR      PC, #SCC7      ; CHECK FOR CC=7
3401      015654 012767 021421 162560      MOV      #21421, TEMP1      ; LOAD LOCATION TEMP1
3402      015662 012760 000442 000000      MOV      #TEMP1, 0(R0)      ; PLACE THE ADDRESS OF LOCATION TEMP1 IN TEMP2
3403      015670 012711 156357      MOV      #-21421, (R1)      ; LOAD LOCATION TEMP
3404      015674 010004      MOV      R0, R4      ; MAKE R4 POINT TO LOCATION TEMP2
3405      015676 067411 000000      ADD      30(R4), (R1)      ; ADD SHOULD RESULT AS 0'S
3406      015702 004737 017252      JSR      PC, #SCC5      ; CHECK FOR CC=5
3407      015706 005430      NEG      2(R0)+      ; NEGATE THE CONTENTS OF TEMP1
3408      015710 012746 021421      MOV      #21421, -(SP)      ; PLACE # 21421 ON THE STACK
3409      015714 065066 000000      ADD      2-(R0), 0(SP)      ; ADD, SHOULD=0'S
3410      015720 004737 017252      JSR      PC, #SCC5      ; CHECK FOR CC=5
3411      015724 005726      TST      (SP)+      ; CHECK THE STACK TO CONTAIN 0, ALSO
3412
3413      015726 001404      BEQ      4$      ; RESTORE THE STACK POINTER
3414      015730 012745 000342      MOV      #342, -(R5)
3415      015734 005245      INC      -(R5)
3416      015736 000000      HALT
3417      015740 012767 137777 162476 4$:      MOV      #137777, TEMP2      ; ADD INSTRUCTION FAILED IN MODE 5

```



```

3418 015746 062767 137777 162470 ADD #137777,TEMP2
3419 015754 004737 017206 JSR PC,2#SCC3 ; CHECK CC=3
3420 015760 022767 077776 162456 CMP #77776,TEMP2
3421 015766 001404 BEQ SUB1
3422 015770 EADD1: MOV #343,-(R5)
3423 015770 012745 000343 INC -(R5)
3424 015774 005245 HALT ; WRONG RESULT AT TEMP OR WRONG SEQUENCE
3425 015776 000000

```

```

*****
;TEST: 117 NEW INSTRUCTION IN THIS SECTION IS SUB
*****

```

```

3433 016000 SUB1: CMP (R5),#117
3434 016000 021527 000117 BNE ESUB1 ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
3435 016004 001100 INC (R5)
3436 016006 005215 MOV #TEMP,R2 ; LOAD ADDRESSES
3437 016010 012702 000440 MOV #TEMP1,R3 ; LOAD LOCATIONS
3438 016014 012703 000442 MOV #21421,TEMP
3439 016020 012767 021421 162412 MOV #-21421,TEMP1
3440 016026 012767 156357 162406 SUB (R2),(R3) ; RESULT SHOULD=-43042
3441 016034 161213 JSR PC,2#SCC10 ; CHECK FOR CC = 10
3442 016036 004737 017316 CMP #-43042,TEMP1 ; CHECK IT
3443 016042 022767 134736 162372 BEQ 1$ ; CONTINUE IF OK
3444 016050 001404 MOV #344,-(R5)
3445 016052 012745 000344 INC -(R5)
3446 016056 005245 HALT ; SUB INSTRUCTION FAILED
3447 016060 000000 1$: MOV #21421,TEMP1 ; LOAD LOCATION
3448 016062 012767 021421 162352 SUB (R2),(R3) ; RESULT SHOULD=0
3449 016070 161213 BEQ 2$
3450 016072 001404 MOV #345,-(R5)
3451 016074 012745 000345 INC -(R5)
3452 016100 005245 HALT ; SUB INSTRUCTION FAILED
3453 016102 000000 2$: MOV #-1,TEMP1 ; LOAD LOCATIONS
3454 016104 012767 177777 162330 MOV #77777,TEMP ; LOAD LOCATIONS
3455 016112 012767 077777 162320 SUB (R3),(R2) ; RESULT SHOULD GIVE 100000 AND OVERFLOW
3456 016120 161312 JSR PC,2#SCC13 ; CHECK FOR CC = 13
3457 016122 004737 017400 CMP #100000,TEMP ; CHECK IT
3458 016126 022767 100000 162304 BEQ 3$ ; CONTINUE IF OK
3459 016134 001404 MOV #346,-(R5)
3460 016136 012745 000346 INC -(R5)
3461 016142 005245 HALT ; SUB INSTRUCTION FAILED
3462 016144 000000 3$: MOV #-1,(R2)
3463 016146 012712 177777 SUB (R3),(R2)
3464 016152 161312 JSR PC,2#SCC4 ; CHECK FOR CC = 4
3465 016154 004737 017230 MOV #77777,TEMP
3466 016160 012767 077777 162252 SUB #77777,TEMP
3467 016166 162767 077777 162244 JSR PC,2#SCC4 ; CHECK FOR CC=4
3468 016174 004737 017230 TST TEMP ; TEMP SHOULD BE =0
3469 016200 005767 162234 BEQ SOB
3470 016204 001404 ESUB1: MOV #347,-(R5)
3471 016206 012745 000347 INC -(R5)
3472 016206 005245
3473 016212

```


D08

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 94
DVKAAB.P11 25-OCT-77 13:09 T117

NEW INSTRUCTION IN THIS SECTION IS SUB

SEQ 0094

3474 016214 000000

HALT

; SUB INSTRUCTION FAILED OR SEQUENCE ERROR

```

3475
3476
3477
3478
3479 016216 021527 000120
3480 016216 001042
3481 016222 005215
3482 016224 012700 000012
3483 016226 005001
3484 016232 005201
3485 016234 020127 000012
3486 016236 003404 000350
3487 016242 012745
3488 016244 005245
3489 016250 000000
3490 016252 000277
3491 016254 077012
3492 016256 004737 017420
3493 016260 005700
3494 016264 001404
3495 016266 012745 000351
3496 016270 005245
3497 016274 000000
3498 016276 022701 000012
3499 016300 001404
3500 016304 012745 000352
3501 016306 005245
3502 016312 000000
3503 016314 012704 000010
3504 016316 077401
3505 016322 005704
3506 016324 001404
3507 016326 016330
3508 016330 012745 000353
3509 016330 005245
3510 016334 000000
3511 016336
3512
3513
3514
3515
3516
3517
3518
3519
3520 016340 021527 000121
3521 016340 001042
3522 016344 005215
3523 016346 012700 000440
3524 016350 012701 000442
3525 016354 012711 177777
3526 016360 005010
3527 016364 106410
3528 016366 004737 017126
3529 016370
3530

```

```

*****
;TEST: 120 NEW INSTRUCTION IN THIS SECTION IS SOB
*****
SOB:
      CMP      (R5),#120
      BNE      ES06          ; IF IN WRONG SEQUENCE GO TO HLT AT THE END OF THE TEST
      INC      (R5)
      MOV      #10.,R0      ; LOAD REGISTERS
      CLR      R1
      INC      R1          ; KEEP COUNT
      CMP      R1,#10.
      BLE      2$
      MOV      #350,-(R5)
      INC      -(R5)
      HALT
2$:   SCC
      SOB      R0,1$        ; SOB INSTRUCTION FAILED
      JSR      PC,0$SCC17   ; SUB. 1 FROM REG. 0, GO BACK TO 1$
      TST      R0          ; CHECK FOR CC = 17
      BEQ      3$          ; REG. 0 = 0 ?
      BEQ      3$          ; NO, FAILED
      MOV      #351,-(R5)
      INC      -(R5)
      HALT
3$:   CMP      #10.,R1      ; SOB INSTRUCTION FAILED
      BEQ      4$          ; DID IT GO THRU 10 TIMES ?
      MOV      #352,-(R5)
      INC      -(R5)
      HALT
4$:   MOV      #10,R4      ; SOB INSTRUCTION FAILED
      SOB      R4,5$       ; PLACE #10 IN R4
      TST      R4          ; STAY HERE UNTILL R4 = 0
      BEQ      PSWNO      ; CONTINUE IF OK
      MOV      #353,-(R5)
      INC      -(R5)
      HALT
      ; SOB FAILED OR WRONG SEQUENCE

```

```

*****
;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,0$SCC0   ; CHECK FOR CC = 0

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 96
 DVKAAB.P11 25-OCT-77 13:09 T121

NEW INSTRUCTIONS IN THIS SECTION ARE MTPS & MFPS

SEQ 0096

3531	016374			MFPS	(R1)	; MOVE PSW TO TEMP1
3532	016374	106711		.WORD	106700!...C	
3533	016376	004737	017230	JSR	PC, @#SCC4	; CHECK FOR CC = 4
3534	016402	022711	177400	CMP	#177400, (R1)	; CHECK TEMP1 TO MAKE SURE THAT ONLY
3535						; THE LOWER BYTE WAS AFFECTED BY MFPS
3536	016406	001404		BEQ	1\$	
3537	016410	012745	000354	MOV	#354, -(R5)	
3538	016414	005245		INC	-(R5)	
3539	016416	000000		HALT		; MTPS OR MFPS INSTRUCTION FAILED
3540	016420	005011		CLR	(R1)	
3541	016422			MTPS	#377	; SET PSW = 357 SINCE T BIT CAN NOT BE SET BY MTPS
3542	016422	106427		.WORD	106400!...C	
3543	016426	004737	017420	JSR	PC, @#SCC17	; CHECK FOR CC = 17
3544	016432			MFPS	TEMP1	; MOVE PSW TO TEMP1
3545	016432	106767		.WORD	106700!...C	
3546	016436	004737	017336	JSR	PC, @#SCC11	; CHECK FOR CC = 11 [C BIT SHOULD NOT BE EFFECTED BY MFP
3547	016442	022767	000357 161772	CMP	#357, TEMP1	
3548	016450	001404		BEQ	BTWRD	
3549	016452			EPSONO:		
3550	016452	012745	000355	MOV	#355, -(R5)	
3551	016456	005245		INC	-(R5)	
3552	016460	000000		HALT		; MFPS INSTRUCTION FAILED IN MODE 6
3553						; OR SEQUENCE ERROR


```

3554
3555
3556
3557
3558 016462
3559 016462 021527 000122
3560 016466 001124
3561 016470 005215
3562 016472 005000
3563 016474 000277
3564 016476 112700 000200
3565
3566 016502 004737 017336
3567 016506 022700 177600
3568 016512 001404
3569 016514 012745 000356
3570 016520 005245
3571 016522 000000
3572 016524 000277
3573 016526 012700 177777
3574 016532 112700 000000
3575 016536 004737 017252
3576 016542 005700
3577 016544 001404
3578 016546 012745 000357
3579 016552 005245
3580 016554 000000
3581 016556 012704 000444
3582 016562 012714 000377
3583 016566 012706 000526
3584 016572 116426 000000
3585 016576 022706 000530
3586 016602 001404
3587 016604 012745 000360
3588 016610 005245
3589 016612 000000
3590
3591 016614 124627 000377
3592
3593 016620 001404
3594 016622 012745 000361
3595 016626 005245
3596 016630 000000
3597 016632 022706 000526
3598
3599 016636 001404
3600 016640 012745 000362
3601 016644 005245
3602 016646 000000
3603 016650 016467 000000 161562
3604 016656 005726
3605 016660 000277
3606 016662 114667 161553
3607 016666 004737 017336
3608 016672 022767 177777 161540
3609 016700 001404

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$: MC, #TEMP2, R4 ; R4 IS POINTING TO TEMP2
MOV #377, (R4) ; PLACE #377 IN LOCATION TEMP2
MOV #START-2, R6
MOVB 0(R4), (R6)+ ; PUSH # 377 ON STACK
CMP #START, R6
BEQ 3$
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$: MOV 0(R4), TEMP ; SET THE LOWER BYTE OF LOCATION TEMP
TST (R6)+ ; RESTORE STACK POINTER
SCC
MOVB -(SP), TEMP+1 ; SET THE HIGHER BYTE OF LOCATION TEMP
JSR PC, @#SCC11 ; CHECK FOR CC=11
CMP #177777, TEMP ; CHECK TEMP FOR THE CORRECT VALUE
BEQ 6$

```

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 98

DVKAAB.P11 25-OCT-77 13:09 T122

BYTE INSTRUCTIONS REQUIRING WORD INST. TO CHECK

SEQ 0098

3610	016702	012745	000363		MOV	#363, -(R5)	
3611	016706	005245			INC	-(R5)	
3612	016710	000000			HALT		; TEMP FOULED UP
3613	016712	005067	161522	6S:	CLR	TEMP	
3614	016716	000241			CLC		
3615	016720	105167	161515		COMB	TEMP+1	; WRITE 1'S IN THE HIGHER BYTE OF TEMP
3616	016724	004737	017336		JSR	PC, @#SCC11	; CHECK FOR CC=11
3617	016730	022767	177400	161502	CMP	#177400, TEMP	
3618	016736	001404			BEG	NEXT	
3619	016740			EBTWRD:			
3620	016740	012745	000364		MOV	#364, -(R5)	
3621	016744	005245			INC	-(R5)	
3622	016746	000000			HALT		; WRONG VALUE IN TEMP OR WRONG SEQUENCE

```

3623
3624
3625
3626          :          END OF PASS
3627          :          *****
3628
3629
3630
3631          016750          NEXT:
3632          016750          021527 000123          CMP      (R5),#123
3633          016754          001404          BEQ      2$          ; IF IN WRONG SEQUENCE GO TO HLT BELOW
3634          016756          012745 000365          MOV      #365,-(R5)
3635          016762          005245          INC      -(R5)
3636          016764          000000          HALT
3637          016766          005267 161414          2$:      INC      $PASS          ; PROGRAM IS IN WRONG SEQUENCE
3638          016772          126727 161410 000001      CMPB     $PASS,#1          ; ALLOW THE TYPE OUT OF END OF
3639
3640          017000          001002          BNE      GET42          ; PASS EVERY 377 PASSES
3641          017002          000004 000454          .TYPE     GET42:      ENDPAS          ; TYPE END OF PASS MESSAGE
3642          017006          013700 000042          MOV      #42,R0
3643          017012          001404          SENDAD:      BEQ      DOAGN
3644          017014          004710          JSR      PC,(R0)
3645          017016          000240          NOP
3646          017020          000240          NOP
3647          017022          000240          NOP
3648          017024          005067 161354          DOAGN:      CLR      $TESTN          ; PREPARE TO START FROM TEST 0
3649          017030          000167 161474          RETURN:      JMP      START          ; START TEST OVER AT BEGINNING
3650
3651          ;
3652          ;*****
3653
3654          .SBTTL  POWER FAIL ROUTINE
3655
3656
3657          017034          012737 017044 000024      PWRDN:      MOV      #PWRUP,#24          ; GO TO POWER UP ROUTINE AFTER THE POWER COMES BACK
3658          017042          000000          HALT
3659
3660          017044          012706 000530          PWRUP:      MOV      #START,SP
3661          017050          012737 017034 000024      MOV      #PWRDN,#24
3662          017056          000004 000470          .TYPE     POWER
3663          017062          000760          BR      DOAGN

```



```

TYPE:    BITB      #40, @#SENVN      ; HAS THE CONSOLE OUTPUTS BEEN SUPPRESSED?
        BNE       4$                ; IF SO THEN GO TO 4$
        MOV       @ (SP), R3         ; GET ADDRESS OF MESSAGE

1$:      TSTB      (R3)               ; END OF MESSAGE ?
        BEQ       4$                ; YES, GO WRAP IT UP

3$:      TSTB      @TPS               ; READY FOR NEXT CHARACTER ?
        BPL       3$                ; NO, WAIT
        MOVB      (R3)+, @TPB        ; LOAD AND TYPE THE CHARACTER
        BR        1$                ; YES, GET THE NEXT CHARACTER

4$:      ADD       #2, (SP)           ; ADJUST THE RETURN PC
        RTT                          ; RETURN

```

3684	017126	003402		\$CC0:	BLE	1\$	
3685	017130	100401			BMI	1\$	
3686	017132	103004			BCC	2\$	
3687	017134			1\$:			
3688	017134	012745	000366		MOV	#366, -(R5)	
3689	017140	005245			INC	-(R5)	
3690	017142	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 0
3691	017144	000207		2\$:	RTS	PC	
3692							
3693	017146	003402		\$CC1:	BLE	1\$	
3694	017150	100401			BMI	1\$	
3695	017152	103404			BCS	2\$	
3696	017154			1\$:			
3697	017154	012745	000367		MOV	#367, -(R5)	
3698	017160	005245			INC	-(R5)	
3699	017162	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 1
3700	017164	000207		2\$:	RTS	PC	
3701							
3702	017166	100402		\$CC2:	BMI	1\$	
3703	017170	101401			BLOS	1\$	
3704	017172	102404			BVS	2\$	
3705	017174			1\$:			
3706	017174	012745	000370		MOV	#370, -(R5)	
3707	017200	005245			INC	-(R5)	
3708	017202	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 2
3709	017204	000207		2\$:	RTS	PC	
3710							
3711	017206	100403		\$CC3:	BMI	1\$	
3712	017210	001402			BEQ	1\$	
3713	017212	102001			BVC	1\$	
3714	017214	103404			BCS	2\$	
3715	017216			1\$:			
3716	017216	012745	000371		MOV	#371, -(R5)	
3717	017222	005245			INC	-(R5)	
3718	017224	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 3
3719	017226	000207		2\$:	RTS	PC	
3720							
3721	017230	100403		\$CC4:	BMI	1\$	
3722	017232	001002			BNE	1\$	
3723	017234	102401			BVS	1\$	
3724	017236	103004			BCC	2\$	
3725	017240			1\$:			
3726	017240	012745	000372		MOV	#372, -(R5)	
3727	017244	005245			INC	-(R5)	
3728	017246	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 4
3729	017250	000207		2\$:	RTS	PC	
3730							
3731	017252	100403		\$CC5:	BMI	1\$	
3732	017254	001002			BNE	1\$	
3733	017256	102401			BVS	1\$	
3734	017260	103404			BCS	2\$	
3735	017262			1\$:			
3736	017262	012745	000373		MOV	#373, -(R5)	
3737	017266	005245			INC	-(R5)	
3738	017270	000000			HALT		;WRONG CC, IT SHOULD HAVE BEEN = 5
3739	017272	000207		2\$:	RTS	PC	

3740					
3741					
3742	017274	100403			
3743	017276	001002			
3744	017300	102001			
3745	017302	103404			
3746	017304				
3747	017304	012745	000374		
3748	017310	005245			
3749	017312	000000			
3750	017314	000207			
3751					
3752	017316	100002			
3753	017320	101401			
3754	017322	102004			
3755	017324				
3756	017324	012745	000375		
3757	017330	005245			
3758	017332	000000			
3759	017334	000207			
3760					
3761	017336	100003			
3762	017340	001402			
3763	017342	102401			
3764	017344	103404			
3765	017346				
3766	017346	012745	000376		
3767	017352	005245			
3768	017354	000000			
3769	017356	000207			
3770					
3771	017360	100002			
3772	017362	101401			
3773	017364	102404			
3774	017366				
3775	017366	012745	000377		
3776	017372	005245			
3777	017374	000000			
3778	017376	000207			
3779					
3780	017400	100002			
3781	017402	003401			
3782	017404	103404			
3783	017406				
3784	017406	012745	000400		
3785	017412	005245			
3786	017414	000000			
3787	017416	000207			
3788					
3789	017420	100003			
3790	017422	001002			
3791	017424	102001			
3792	017426	103404			
3793	017430				
3794	017430	012745	000401		
3795	017434	005245			

SCC7:	BMI	1\$	
	BNE	1\$	
	BVC	1\$	
	BCS	2\$	
1\$:	MOV	#374, -(R5)	
	INC	-(R5)	
	HALT		;WRONG CC, IT SHOULD HAVE BEEN = 7
2\$:	RTS	PC	
SCC10:	BPL	1\$	
	BLOS	1\$	
	BVC	2\$	
1\$:	MOV	#375, -(R5)	
	INC	-(R5)	
	HALT		;WRONG CC, IT SHOULD HAVE BEEN = 10
2\$:	RTS	PC	
SCC11:	BPL	1\$	
	BEQ	1\$	
	BVS	1\$	
	BCS	2\$	
1\$:	MOV	#376, -(R5)	
	INC	-(R5)	
	HALT		;WRONG CC, IT SHOULD HAVE BEEN = 11
2\$:	RTS	PC	
SCC12:	BPL	1\$	
	BLOS	1\$	
	BVS	2\$	
1\$:	MOV	#377, -(R5)	
	INC	-(R5)	
	HALT		;WRONG CC, IT SHOULD HAVE BEEN = 12
2\$:	RTS	PC	
SCC13:	BPL	1\$	
	BLE	1\$	
	BCS	2\$	
1\$:	MOV	#400, -(R5)	
	INC	-(R5)	
	HALT		;WRONG CC, IT SHOULD HAVE BEEN = 13
2\$:	RTS	PC	
SCC17:	BPL	1\$	
	BNE	1\$	
	BVC	1\$	
	BCS	2\$	
1\$:	MOV	#401, -(R5)	
	INC	-(R5)	

M08

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 103

DVKAAB.P11 25-OCT-77 13:09

ROUTINES TO CHECK CONDITION CODES

SEQ 0103

3796 017436 000000
3797 017440 000207
3798
3799 000001

2S: HALT
RTS PC
.END

;WRONG CC, IT SHOULD HAVE BEEN = 17

ASRB1	012342	2672#																
ASRO	005672	1672#																
ASR1	014516	3151#																
ASWREG=	000000	333	346															
ATESTN=	000000	333	337															
AUNIT =	000000	333	340															
AUSWR =	000000	333	347															
AVECT1=	000000	333																
AVECT2=	000000	333																
BICB0	003140	1042	1051#															
BICB1	011202	2396	2408#															
BICO	004710	1440	1449#															
BIC1	013336	2878	2891#															
BRANCH	001170	608	616#															
BTWRD	016462	3548	3558#															
CBIT	000740	510	517#															
CC0	000612	442	445	446	447	448	450	451	452	453	454	455	456	458#				
CC1	000656	471	474	475	476	477	478	479	480	482#								
CC2	000726	493	497	498	499	500	501	502	503	504	506#							
CC3	000774	519	524	525	526	527	528	529	531#									
CC4	001044	542	548	549	550	551	552	553	554	556#								
CC5	001156	592	596	598	600	602	604#											
CC6	001112	569	572	573	574	575	577	578	579	580	582#							
CLNZ =	000254	323#	1795	3282														
CMPB0	003000	1011#																
CMPB1	011100	2363	2375#															
CMP0	004616	1422#																
CMP1	013072	2829#																
COMB0	003416	1123#																
COMB1	011624	2503	2521#															
COM0	005212	1529#																
COM1	013772	3000#																
DOAGN	017024	3643	3648#	3663														
DUMMY	000436	386#	387	879	880													
EAD00	006744	1883	1915#															
EAD01	015770	3368	3422#															
EASRB0	004174	1267	1288#															
EASRB1	012450	2674	2697#															
EASRO	005770	1674	1695#															
EASR1	014624	3153	3176#															
EBICO	005044	1451	1481#															
EBIC1	013610	2893	2948#															
EBTWRD	016740	3560	3619#															
ECMPB0	003130	1013	1043#															
ECMPB1	011172	2377	2397#															
ECMP0	004700	1424	1441#															
ECMP1	013326	2831	2879#															
EINCB1	011614	2466	2504#															
EMODE0	007376	2004	2040#															
EMODE2	007534	2054	2082	2085#														
EMODE3	007726	2096	2124	2127#														
EMODE4	010204	2140	2196	2199#														
EMODE5	010434	2210	2254	2257#														
EMODE6	010604	2270	2293#															
EMODE7	010746	2304	2324#															
ENDCC0	000622	457	462#															

ESOB	016330	3163	3166#	3177	3180#	3190	3193#	3206	3209#	3215	3218#	3229	3232#	3240
ESUB1	016206	3243#	3250	3253#	3261	3264#	3287	3290#	3296	3299#	3315	3318#	3326	3329#
ESWAB0	006466	3359	3362#	3378	3381#	3385	3388#	3394	3397#	3414	3417#	3423	3426#	3445
ESWAB1	015344	3448#	3451	3454#	3460	3463#	3472	3475#	3488	3491#	3496	3499#	3501	3504#
ESXT0	006366	3509	3512#	3537	3540#	3550	3553#	3569	3572#	3578	3581#	3587	3590#	3594
ESXT1	015236	3597#	3600	3603#	3610	3613#	3620	3623#	3634	3637#	3688	3691#	3697	3700#
ETST81	011070	3706	3709#	3716	3719#	3726	3729#	3736	3739#	3747	3750#	3756	3759#	3766
EXOR0	006574	3769#	3775	3778#	3784	3787#	3794	3797#						
EXOR1	015464	3481	3508#											
GET42	017006	3435	3471#											
INCB0	003260	1821	1840#											
INCB1	011430	3305	3325#											
INCO	005054	1791	1808#											
INC1	013620	3277	3295#											
JMP1	001262	2341	2364#											
JMP2	001366	1850	1870#											
JMP3	001472	3338	3358#											
JMP4	001572	3640	3642#											
JMP5	001670	1086#												
JMP6	002002	2464#												
JMP7	002104	1480	1492#											
JSRM	002344	2947	2956#											
JSRTST	002202	639	644#											
MARK2	000452	668	680#											
MODE0	007222	704	710#											
MODE1	007306	723	734#											
MODE2	007406	754	760#											
MODE3	007544	778	789#											
MODE4	007736	808	814#											
MODE5	010214	849	878#											
MODE6	010444	826	837#											
MODE7	010614	396#	881											
NBIT	000624	1988	2002#											
NEGB0	003532	2018	2023#											
NEGB1	011756	2039	2052#											
NEGO	005326	2084	2094#											
NEG1	014116	2126	2138#											
NEXT	016750	2198	2208#											
NOBIT	000546	2256	2268#											
NOP1	= 000260	2292	2302#											
NOTCC	001122	462	469#											
POWER	000470	1158#												
PSW	007120	2558#												
PSWNO	016340	1564#												
PWRDN	017034	3034#												
PWRUP	017044	3618	3631#											
REGS	002616	431	440#											
RETURN	017030	325#	449											
		581	590#											
		400#	3662											
		1964#												
		3507	3520#											
		417	3657#											
		3657	3660#											
		918	954#											
		3649#												

3661

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 110
 DVKAAB.P11 25-OCT-77 13:09

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0109

SCC11	017336	1017	1076	1078	1099	1113	1141	1164	1285	1368	1472	1475	1505	1519
		1547	1570	1692	1775	1805	1862	1867	1895	1986	2437	2439	2474	2499
		2542	2565	2694	2779	2920	2931	2970	3019	3043	3173	3258	3292	3350
SCC12	017360	3355	3391	3546	3566	3607	3616	3761#						
		1191	1228	1250	1320	1597	1635	1657	1727	2596	2632	2658	2733	3074
		3111	3137	3212	3771#									
SCC13	017400	1096	1172	1502	1578	1949	2472	2573	2851	2967	3051	3457	3780#	
SCC17	017420	1983	3493	3543	3789#									
SCC2	017166	1377	1784	2788	3267	3702#								
SCC3	017206	1116	1522	2499	2990	3419	3711#							
SCC4	017230	997	999	1259	1281	1306	1365	1399	1401	1666	1688	1713	1772	1837
		1957	1979	2347	2349	2459	2482	2667	2690	2719	2776	2813	2815	3146
		3169	3198	3255	3322	3465	3468	3533	3721#					
SCC5	017252	1101	1111	1157	1329	1414	1478	1507	1517	1556	1736	1797	1856	2446
		2494	2553	2742	2929	2945	2975	2985	3029	3221	3284	3344	3406	3410
		3575	3731#											
SCC7	017274	1198	1226	1257	1279	1604	1633	1664	1686	1903	2603	2630	2665	2688
		3081	3109	3144	3167	3400	3742#							
SCPUOP	000426	348#												
SDEVCT	000410	339#												
SENDAD	017014	296	3644#											
SENV	000420	344#												
SENVN	000421	345#	3669											
SETABL	000420	343#	418											
SETEND	000430	355#	378											
SFATAL	000402	336#												
SHO =	000003	276	277											
SHIBTS	000430	373#												
SMAIL	000400	334#	374	378	420									
SMBADR	000432	374#												
SMSGAD	000414	341#												
SMSGLG	000416	342#												
SMSGTY	000400	335#												
SPASS	000406	338#	3637#	3638										
SPASTH	000436	376#												
SSVPC =	001000	294#	299											
SSWR =	160000	276	277#											
SSWREG	000422	346#												
STESTN	000404	337#	429	897	971	3648#								
STN =	000124	276#	328#	436	441	442#	465	470	471#	487	492	493#	513	518
		519#	536	541	542#	563	568	569#	586	591	592#	612	617	618#
		640	645	646#	676	681	682#	730	735	736#	785	790	791#	833
		838	839#	950	955	956#	984	989	990#	1007	1012	1013#	1047	1052
		1053#	1082	1087	1088#	1119	1124	1125#	1154	1159	1160#	1179	1184	1185#
		1208	1213	1214#	1235	1240	1241#	1261	1266	1267#	1292	1297	1298#	1333
		1338	1339#	1386	1391	1392#	1418	1423	1424#	1445	1450	1451#	1488	1493
		1494#	1525	1530	1531#	1560	1565	1566#	1585	1590	1591#	1615	1620	1621#
		1642	1647	1648#	1668	1673	1674#	1699	1704	1705#	1740	1745	1746#	1785
		1790	1791#	1815	1820	1821#	1844	1849	1850#	1877	1882	1883#	1919	1924
		1925#	1960	1965	1966#	1998	2003	2004#	2048	2053	2054#	2090	2095	2096#
		2134	2139	2140#	2204	2209	2210#	2264	2269	2270#	2298	2303	2304#	2335
		2340	2341#	2371	2376	2377#	2404	2409	2410#	2460	2465	2466#	2517	2522
		2523#	2554	2559	2560#	2583	2588	2589#	2611	2616	2617#	2642	2647	2648#
		2668	2673	2674#	2704	2709	2710#	2743	2748	2749#	2798	2803	2804#	2825
		2830	2831#	2887	2892	2893#	2952	2957	2958#	2996	3001	3002#	3030	3035
		3036#	3061	3066	3067#	3090	3095	3096#	3121	3126	3127#	3147	3152	3153#

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 111
 DVKAAB.P11 25-OCT-77 13:09

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0110

STSTM	000434	3183	3188	3189#	3222	3227	3228#	3271	3276	3277#	3299	3304	3305#	3332
SUNIT	000412	3337	3338#	3362	3367	3368#	3429	3434	3435#	3475	3480	3481#	3515	3521
SUNITM	000440	3522#	3554	3559	3560#	3632	3633#							
SUSWR	000424	375#												
.	= 017442	340#												
		377#												
		347#												
		284#	290	294	295#	297#	299#	313#	362	363#	365#	367#	379#	381#
		383#	385#	387#	389#	391#	393#	403#	405#	416#	426#	804	1971#	1972#
		1974#	1975#	1982#	1983#	1985#	1986#	3529#	3530#	3532#	3533#	3542#	3543#	3545#
		3546#												
.TYPE	= 000004	329#	3641	3662										
.SX	= 000430	362#	367											
..A	= 016432	1971#	1974#	1982#	1985#	3529#	3532#	3542#	3545#					
..B	= 016436	1971#	1972	1974#	1975	1982#	1983	1985#	1986	3529#	3530	3532#	3533	3542#
		3543	3545#	3546										
..C	= 000067	1971#	1974#	1982#	1985#	3529#	3532#	3542#	3545#					

ERROR	307#	459	483	507	532	557	583	605	619	624	628	632	636	651	660
	665	671	687	696	701	706	715	720	727	741	746	751	756	764	768
	773	781	795	800	805	810	817	823	830	845	850	859	864	869	875
	885	894	899	904	910	913	919	925	933	940	946	977	991	1020	1026
	1031	1037	1044	1054	1065	1072	1089	1106	1126	1136	1144	1167	1176	1194	1203
	1222	1232	1242	1253	1275	1289	1299	1314	1323	1340	1350	1360	1371	1393	1409
	1431	1436	1442	1461	1468	1482	1495	1512	1532	1542	1550	1573	1582	1600	1609
	1629	1639	1649	1660	1682	1696	1706	1721	1730	1747	1757	1767	1778	1800	1809
	1830	1841	1871	1890	1898	1908	1916	1926	1936	1943	1952	1976	1990	2011	2019
	2032	2041	2071	2086	2113	2128	2149	2162	2172	2182	2200	2228	2238	2258	2285
	2294	2317	2325	2359	2365	2387	2392	2398	2411	2425	2432	2449	2454	2489	2505
	2524	2537	2546	2568	2577	2599	2608	2626	2636	2649	2661	2684	2698	2711	2727
	2736	2750	2761	2771	2782	2805	2840	2845	2864	2870	2880	2907	2914	2923	2937
	2949	2959	2980	3003	3014	3022	3046	3055	3077	3087	3105	3115	3128	3140	3163
	3177	3190	3206	3215	3229	3240	3250	3261	3287	3296	3315	3326	3359	3378	3385
	3394	3414	3423	3445	3451	3460	3472	3488	3496	3501	3509	3537	3550	3569	3578
	3587	3594	3600	3610	3620	3634	3688	3697	3706	3716	3726	3736	3747	3756	3766
	3775	3784	3794												
HLT	304#	458	482	506	531	556	582	604	619	624	628	632	636	651	659
	665	670	687	695	701	706	715	720	726	741	745	751	756	764	767
	773	780	795	800	805	810	817	823	829	844	850	858	864	869	875
	884	904	909	913	919	925	933	940	945	976	991	1020	1026	1031	1037
	1043	1054	1065	1072	1089	1106	1126	1136	1144	1167	1175	1194	1202	1222	1231
	1242	1253	1275	1288	1299	1314	1323	1340	1350	1360	1371	1393	1409	1431	1436
	1441	1461	1468	1481	1495	1512	1532	1542	1550	1573	1581	1600	1608	1629	1638
	1649	1660	1682	1695	1706	1721	1730	1747	1757	1767	1778	1800	1808	1830	1840
	1870	1890	1898	1908	1915	1926	1936	1943	1952	1976	1989	2011	2019	2032	2040
	2070	2085	2112	2127	2149	2162	2171	2182	2199	2228	2237	2257	2284	2293	2317
	2324	2359	2364	2387	2392	2397	2411	2425	2432	2449	2454	2489	2504	2524	2537
	2546	2568	2576	2599	2607	2626	2635	2649	2661	2684	2697	2711	2727	2736	2750
	2761	2771	2782	2805	2840	2845	2864	2870	2879	2907	2914	2923	2937	2948	2959
	2980	3003	3014	3022	3046	3054	3077	3086	3105	3114	3128	3140	3163	3176	3190
	3206	3215	3229	3240	3250	3261	3287	3295	3315	3325	3358	3378	3385	3394	3414
	3422	3445	3451	3460	3471	3488	3496	3501	3508	3537	3549	3569	3578	3587	3594
	3600	3610	3619	3634	3687	3696	3705	3715	3725	3735	3746	3755	3765	3774	3783
	3793														
HLT1	306#	893	899												
MFPS	304#	1973	1984	3531	3544										
MTPS	302#	1970	1981	3528	3541										
NWTEST	308#	436	465	487	513	536	563	586	612	640	676	730	785	833	950
	984	1007	1047	1082	1119	1154	1179	1208	1235	1261	1292	1333	1386	1418	1445
	1488	1525	1560	1585	1615	1642	1668	1699	1740	1785	1815	1844	1877	1919	1960
	1998	2048	2090	2134	2204	2264	2298	2335	2371	2404	2460	2517	2554	2583	2611
	2642	2668	2704	2743	2798	2825	2887	2952	2996	3030	3061	3090	3121	3147	3183
	3222	3271	3299	3332	3362	3429	3475	3515	3554						
SEQCHK	311#	440	469	491	517	540	567	590	616	644	680	734	789	837	954
	988	1011	1051	1086	1123	1158	1183	1212	1239	1265	1296	1337	1390	1422	1449
	1492	1529	1564	1589	1619	1646	1672	1703	1744	1789	1819	1848	1881	1923	1964
	2002	2052	2094	2138	2208	2268	2302	2339	2375	2408	2464	2521	2558	2587	2615
	2646	2672	2708	2747	2802	2829	2891	2956	3000	3034	3065	3094	3125	3151	3187
	3226	3275	3303	3336	3366	3433	3479	3520	3558	3631					
SEQUEN	310#	441	470	492	518	541	568	591	617	645	681	735	790	838	955
	989	1012	1052	1087	1124	1159	1184	1213	1240	1266	1297	1338	1391	1423	1450
	1493	1530	1565	1590	1620	1647	1673	1704	1745	1790	1820	1849	1882	1924	1965
	2003	2053	2095	2139	2209	2269	2303	2340	2376	2409	2465	2522	2559	2588	2616
	2647	2673	2709	2748	2803	2830	2892	2957	3001	3035	3066	3095	3126	3152	3188

DVKAAB MACY11 30(1046) 25-OCT-77 13:11 PAGE 114
 DVKAAB.P11 25-OCT-77 13:09

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0112

STARS	3227	3276	3304	3337	3367	3434	3480	3521	3559	3632					
	266#	286	292	310#	332	359	361	368	436	438	465	467	487	489	513
	515	536	538	563	565	586	588	612	614	640	642	676	678	730	732
	785	787	833	835	950	952	984	986	1007	1009	1047	1049	1082	1084	1119
	1121	1154	1156	1179	1181	1208	1210	1235	1237	1261	1263	1292	1294	1333	1335
	1386	1388	1418	1420	1445	1447	1488	1490	1525	1527	1560	1562	1585	1587	1615
	1617	1642	1644	1668	1670	1699	1701	1740	1742	1785	1787	1815	1817	1844	1846
	1877	1879	1919	1921	1960	1962	1998	2000	2048	2050	2090	2092	2134	2136	2204
	2206	2264	2266	2298	2300	2335	2337	2371	2373	2404	2406	2460	2462	2517	2519
	2554	2556	2583	2585	2611	2613	2642	2644	2668	2670	2704	2706	2743	2745	2798
	2800	2825	2827	2887	2889	2952	2954	2996	2998	3030	3032	3061	3063	3090	3092
	3121	3123	3147	3149	3183	3185	3222	3224	3271	3273	3299	3301	3332	3334	3362
	3364	3429	3431	3475	3477	3515	3517	3554	3556	3652					
SC17	301#	654	690	853	888										
SNWTST	309#	436	465	487	513	536	563	586	612	640	676	730	785	833	950
	984	1007	1047	1082	1119	1154	1179	1208	1235	1261	1292	1333	1386	1418	1445
	1488	1525	1560	1585	1615	1642	1668	1699	1740	1785	1815	1844	1877	1919	1960
	1998	2048	2090	2134	2204	2264	2298	2335	2371	2404	2460	2517	2554	2583	2611
	2642	2668	2704	2743	2798	2825	2887	2952	2996	3030	3061	3090	3121	3147	3183
	3222	3271	3299	3332	3362	3429	3475	3515	3554						
.HEADE	266#														
.SACT1	266#	290													
.SAPT8	266#	330													
.SAPTH	266#	357													

. ABS. 017442 000

ERRORS DETECTED: 0

DVKAAB.BIN DVKAAB.LST/CRF/SOL/NL:TOC=DVKAAB.P11
 RUN-TIME: 9 10 1 SECONDS
 RUN-TIME RATIO: 359/21=16.8
 CORE USED: 9K (17 PAGES)

J09