

↑ 0002 .MAIN

```
01 ;
02 ; PAGE 1
03 ;
04 ;
05 ;
06 ; RC3603 CPU AND PERIPHERAL EXERCISER
07 ;
08 ;
09 ; PURPOSE:
10 ;
11 ; TO BE USED AS A MAINTENANCE PROGRAM TO FUNCTIONALLY
12 ; TEST THE RC3603 INSTRUCTION SET AND OPERATION OF THE
13 ; TELETYPE READER/PUNCH, THE HIGH SPEED READER/PUNCH,
14 ; AND THE REAL TIME CLOCK
15 ;
16 ; HARDWARE REQUIREMENTS:
17 ;
18 ; CPU UTILIZING THE RC3603 INSTRUCTION SET
19 ; 2K READ/WRITE MEMORY
20 ; OPTIONAL EQUIPMENT:
21 ; TELETYPE READER
22 ; TELETYPE PUNCH
23 ; HIGH SPEED READER
24 ; HIGH SPEED PUNCH
25 ; REAL TIME CLOCK
26 ;
27 ; OPERATING INSTRUCTIONS:
28 ;
29 ; LOAD THE PROGRAM USING THE BINARY LOADER. THE
30 ; PROGRAM WILL AUTOMATICALLY BEGIN RUNNING AND
31 ; CONTINUE TO RUN UNTIL AN ERROR IS DETECTED OR
32 ; IT IS MANUALLY STOPPED BY THE OPERATOR. SHOULD
33 ; THE CPU HALT BY ITSELF AN ERROR HAS BEEN DETECTED
34 ; AND THE OPERATOR SHOULD CONSULT THE DISPLAY
35 ; PANEL TO DETERMINE THE HALT ADDRESS. CHECKING
36 ; THIS ADDRESS IN THE EXERCISER PROGRAM LISTING
37 ; WILL TELL THE INSTRUCTION BEING TESTED WHEN
38 ; THE FAILURE OCCURED AND WHENEVER POSSIBLE THE
39 ; THE CONDITIONS CAUSING THE FAILURE.
40 ; THE NORMAL PROGRAM LOOP TESTS THE MOST BASIC
41 ; INSTRUCTIONS FIRST WITH THE LEAST ENVIRONMENTAL.
42 ; CHANGE UNTIL ENOUGH CONFIDENCE HAS BEEN
43 ; ACHIEVED TO MAKE PROGRESSIVELY MORE COMPLEX
44 ; TESTS UNTIL THE ENTIRE INSTRUCTION SET HAS
45 ; BEEN TESTED. SPECIAL MACHINE FUNCTIONS ARE
46 ; INCLUDED SUCH AS AUTO INCREMENT, AUTO
47 ; DECREMENT AND MULTIPLE (OR CHAINED)
48 ; INDIRECTS. IF ALL TESTS ARE SUCCESSFUL
49 ; THE PROGRAM LOOPS BACK TO REPEAT THE
50 ; PROCESS.
51 ;
```

```
01 ;
02 ;
03 ;
04 ; I/O AND REAL TIME CLOCK TESTING:
05 ;
06 ; ONCE PER INSTRUCTION TESTING LOOP THE DISPLAY
07 ; PANEL SWITCHES ARE INTERROGATED TO DETERMINE
08 ; WHETHER THE OPERATOR WISHES TO TEST THE TAPE
09 ; I/O DEVICES OR THE REAL TIME CLOCK. THE SWITCH
10 ; SETTINGS FOR TESTING THESE DEVICES ARE:
11 ;
12 ; SWITCH 15 - HIGH SPEED PUNCH
13 ; SWITCH 14 - HIGH SPEED READER
14 ; SWITCH 13 - TELETYPE PUNCH
15 ; SWITCH 12 - TELETYPE READER
16 ; SWITCH 11 - REAL TIME CLOCK
17 ;
18 ; THE TESTING OF THESE DEVICES ARE INTERRUPT
19 ; CONTROLLED TO ALSO TEST THE INTERRUPT SYSTEM.
20 ;
21 ; TO TEST HIGH SPEED PUNCH:
22 ;
23 ; SET SWITCH 15 AND TURN ON THE HIGH SPEED
24 ; PUNCH. WHEN THE PROGRAM SENSES THE REQUEST
25 ; IT WILL CAUSE THE HIGH SPEED PUNCH TO GENERATE
26 ; A SECTION OF LEADER FOLLOWED BY A DATA TAPE
27 ; CONTAINING ALL POSSIBLE 8 BIT DATA PATTERNS
28 ; FOLLOWED BY A SECTION OF TRAILER. AT THIS
29 ; TIME THE CPU WILL HALT ALLOWING THE
30 ; OPERATOR TIME TO RESET SWITCH 15 AND TO
31 ; TURN OFF THE HIGH SPEED PUNCH. BY THEN
32 ; PRESSING CONTINUE THE CPU WILL RESUME ITS
33 ; INSTRUCTION TESTING FUNCTION. IF THE OPERATOR
34 ; PRESSES CONTINUE WITHOUT RESETTING SWITCH
35 ; 15 OR TURNING OFF THE PUNCH THE PROGRAM
36 ; WILL IMMEDIATELY PUNCH ANOTHER TAPE UPON
37 ; SENSING SWITCH 15. IF THE OPERATOR TURNS
38 ; OFF THE PUNCH BUT FAILS TO RESET SWITCH 15
39 ; THE PROGRAM WILL REMAIN IN A "READY TO PUNCH"
40 ; STATE HOWEVER NO TAPE WILL BE PUNCHED AS NO
41 ; INTERRUPTS WILL BE RECIEVED FROM THE PUNCH.
42 ; THE REMAINDER OF THE TEST IS COMPLETED BY
43 ; RUNNING THE GENERATED TAPE IN EITHER THE
44 ; HIGH SPEED OR THE TELETYPE READER.
45 ;
```

```
01 ;
02 ;
03 ;
04 ; TO TEST THE HIGH SPEED READER:
05 ;
06 ;   LOAD AN EXERCISER GENERATED TAPE IN THE
07 ;   HIGH SPEED READER MAKING CERTAIN THAT THE
08 ;   READ HEAD IS ABOVE THE LEADER PORTION OF
09 ;   THE TAPE. SET SWITCH 14 ON THE DISPLAY PANEL
10 ;   AND TURN ON THE READER. AS THE TAPE IS BEING
11 ;   READ THE PROGRAM WILL COMPARE THE INPUT
12 ;   DATA WITH THE KNOWN GENERATED DATA FOR
13 ;   POSSIBLE ERRORS. IF A HALT OCCURS AND THE
14 ;   READ HEAD IS POSITIONED ABOVE THE FIRST
15 ;   FRAME OF TRAILER NO ERRORS WERE ENCOUNTERED.
16 ;   THIS IMPLIES THAT NOT ONLY WAS THE GENER-
17 ;   ATED DATA PUNCHED CORRECTLY BUT THE READER
18 ;   WAS ABLE TO READ IT CORRECTLY. IF HOWEVER
19 ;   THE READ HEAD WAS POSITIONED OVER THE DATA
20 ;   PORTION OF THE TAPE WHEN THE HALT OCCURRED
21 ;   A PUNCH OR READ ERROR WAS DETECTED. SINCE
22 ;   THE DATA GENERATED WAS MONOTONICALLY DECREAS-
23 ;   ING FROM 377 TO 1 IT IS A RELATIVELY EASY
24 ;   TASK TO DETECT AN INCONSISTENCY OF DATA ON
25 ;   THE TAPE WHERE THE HALT OCCURRED. IF NONE
26 ;   WAS TO BE FOUND IT IS OBVIOUS THAT A READ
27 ;   ERROR OCCURED. BY REPEATING THE TEST IT
28 ;   MAY BE ASCERTAINED WHETHER THE ERROR IS BIT
29 ;   PATTERN DEPENDENT. IN EITHER EVENT DURING
30 ;   THE HALT THE OPERATOR HAS THE OPPURTUNITY OF
31 ;   RESETTING SWITCH 14 AND TURNING OFF THE HIGH.
32 ;   SPEED READER. BY THEN PRESSING CONTINUE
33 ;   THE PROGRAM WILL RESUME ITS INSTRUCTION
34 ;   TESTING FUNCTION. IF HOWEVER THE OPERATOR PRESSES
35 ;   CONTINUE WITHOUT RESETTING SWITCH 14 OR
36 ;   TURNING OFF THE READER THE REMAINDER OF THE
37 ;   TAPE WILL BE READ AND TESTED. IF THE READER
38 ;   IS TURNED OFF BUT SWITCH 14 IS NOT RESET THE
39 ;   READER WILL REMAIN IN A "READY TO READ" COND-
40 ;   ITION BUT NO ACTION WILL OCCUR SINCE NO
41 ;   INTERRUPTS WILL BE RECIEVED FROM THE READER.
42 ;
43 ; TO TEST TELETYPE PUNCH:
44 ;
45 ;   SET SWITCH 13 AND TURN ON THE TELETYPE
46 ;   PUNCH. THE REMAINDER OF THE TEST FUNCTIONS
47 ;   EXACTLY THE SAME AS THAT OF THE HIGH SPEED
48 ;   PUNCH TEST.
49 ;
50 ;
```

↑ 0005 .MAIN

PAGE 4

```
01 ;
02 ;
03 ;
04 ; TO TEST THE TELETYPE READER:
05 ;
06 ; SET SWITCH 12 AND TURN ON THE TELETYPE
07 ; READER. THE REMAINDER OF THE TEST FUNCTIONS
08 ; EXACTLY THE SAME AS THAT OF THE HIGH SPEED
09 ; READER TEST
10 ;
11 ; TO TEST THE REAL TIME CLOCK:
12 ;
13 ; SET SWITCH 11. THE PROGRAM WILL START THE
14 ; DEVICE AND ENABLE THE INTERRUPTS. IF AN
15 ; ERROR SHOULD OCCUR CONSULT THE LISTING AT
16 ; THE ADDRESS SPECIFIED ON THE DISPLAY PANEL.
17 ;
18 ; RESTRICTIONS:
19 ;
20 ; IT IS ASSUMED THAT ONLY ONE PUNCH OR ONE
21 ; READER WILL EVER BE USED AT ANY ONE TIME.
22 ; IF SWITCH 15 AND 13 OR SWITCH 14 AND 12
23 ; SHOULD EVER BECOME SET SIMULTANEOUSLY AN
24 ; ERROR CONDITION WILL OCCUR. IN THIS EVENT
25 ; MERELY RESTART THE PROGRAM AT LOCATION
26 ; 76.
27
```

↑ 0006 .MAIN

PAGE 5

```
01 ;
02      000040      .LOC      40
03 00040 000351 IPROC: INPRC
04 00041 001502 BUF:  BUFF
05 00042 001702 BUFND: BFEND
06 00043 000177 DIF:  BFEND-BUFF-1
07 00044 000400 ARLOG: ALU
08 00045 000000 DATA:  0
09 00046 000000 LEAD:   0
10 00047 000000 TRAIL:  0
11 00050 000000 STRT:   0
12 00051 000003 C3:     3
13 00052 000010 C10:    10
14 00053 000015 C15:    15
15 00054 000017 C17:    17
16 00055 000037 C37:    37
17 00056 000400 C400:   400
18 00057 000000 PFLAG:  0
19 00060 000000 SREG0:  0
20 00061 000000 SREG1:  0
21 00062 000000 SREG2:  0
22 00063 000000 SREG3:  0
23 00064 000000 SAVC:   0
24 00065 000000 ENTR1:  0
25 00066 000000 ENTR2:  0
26 00067 000000 ENTR3:  0
27 00070 000000 ENTR4:  0
28 00071 000301 IVECT:  INPUT
29 00072 000215      PUNCH
30 00073 000301      INPUT
31 00074 000215      PUNCH
32 00075 000210      RTCLK
33
```

↑ 0007 .MAIN

PAGE 6

```
01
02 00076 024040 PERIP: LDA 1,IPROC
03 00077 044001 STA 1,1
04 00100 060277 INTDS
05 00101 064477 READS 1 ;READ SWITCHES
06 00102 020055 LDA 0,C37
07 00103 107400 AND 0,1
08 00104 131222 MOVZR 1,2,SZC ;CHECK IF HIGH SPEED PUNCH REQUESTED
09 00105 004140 JSR HSPON ;YES
10 00106 060213 NIOC PTP ;NO
11 00107 102400 SUB 0,0
12 00110 040065 STA 0,ENTR1
13 00111 151222 MOVZR 2,2,SZC ;CHECK IF HIGH SPEED READER REQUESTED
14 00112 004160 JSR HSRON ;YES
15 00113 060212 NIOC PTR ;NO
16 00114 102400 SUB 0,0
17 00115 040066 STA 0,ENTR2
18 00116 151222 MOVZR 2,2,SZC ;CHECK IF TTY PUNCH REQUESTED
19 00117 004171 JSR TIOON ;YES
20 00120 060211 NIOC TTO ;NO
21 00121 102400 SUB 0,0
22 00122 040067 STA 0,ENTR3
23 00123 151222 MOVZR 2,2,SZC ;CHECK IF TTY READER REQUESTED
24 00124 004201 JSR TTION ;YES
25 00125 060210 NIOC TTI ;NO
26 00126 102400 SUB 0,0
27 00127 040070 STA 0,ENTR4
28 00130 151222 MOVZR 2,2,SZC ;CHECK IF REAL TIME CLOCK REQUESTED
29 00131 004135 JSR RTCON ;YES
30 00132 125004 MOV 1,1,SZR ;WAS ANYTHING REQUESTED
31 00133 060177 NIOS CPU ;YES, ENABLE INTERRUPTS
32 00134 002044 JMP @ARLOG ;JUMP TO CPU EXERCIZER
33 00135 020051 RTCON: LDA 0,C3
34 00136 061114 DOAS 0,RTC ;TURN ON REAL TIME CLOCK
35 00137 001400 JMP 0,3
36 00140 020065 HSPON: LDA 0,ENTR1
37 00141 101004 MOV 0,0,SZR
38 00142 001403 JMP 3,3
39 00143 102520 SUBZL 0,0
40 00144 040065 STA 0,ENTR1
41 00145 102400 SUB 0,0
42 00146 061113 DOAS 0,PTP
43
```

↑ 0008 .MAIN

```
01 ;
02 00147 040057 HSP1: STA 0,PFLAG ;SET PFLAG TO INDICATE HIGH SPEED
03 00150 024056 LDA 1,C400
04 00151 044046 STA 1,LEAD ;NUMBER OF LEADER FRAMES
05 00152 044045 STA 1,DATA ;INITIALIZE FOR DATA GENERATION
06 00153 125220 MOVZR 1,1
07 00154 044047 STA 1,TRAIL ;NUMBER OF TRAILER FRAMES
08 00155 126400 SUB 1,1
09 00156 044050 STA 1,STRT
10 00157 001403 JMP 3,3
11 00160 020066 HSRON: LDA 0,ENTR2
12 00161 101004 MOV 0,0,SZR
13 00162 001403 JMP 3,3
14 00163 102520 SUBZL 0,0
15 00164 040066 STA 0,ENTR2
16 00165 060112 NIOS PTR
17 00166 102400 SUB 0,0
18 00167 040050 STA 0,STRT ;NO DATA READ YET FLAG
19 00170 000147 JMP HSP1
20 00171 020067 TTOON: LDA 0,ENTR3
21 00172 101004 MOV 0,0,SZR
22 00173 001403 JMP 3,3
23 00174 126400 SUB 1,1
24 00175 065111 DOAS 1,TTO
25 00176 102520 SUBZL 0,0
26 00177 040067 STA 0,ENTR3
27 00200 000147 JMP HSP1
28 00201 020070 TTION: LDA 0,ENTR4
29 00202 101004 MOV 0,0,SZR
30 00203 001403 JMP 3,3
31 00204 060110 NIOS TTI
32 00205 102520 SUBZL 0,0
33 00206 040070 STA 0,ENTR4
34 00207 000147 JMP HSP1
35 00210 063414 RTCLK: SKPBN RTC
36 00211 063614 SKPDN RTC
37 00212 063077 HALT ;BUSY OR DONE FLAG INCORRECT
38 00213 060114 NIOS RTC ;START CLOCK AGAIN
39 00214 000370 JMP QUIT
```

40

PAGE 7

↑ 0009 .MAIN

PAGE 8

```
01 ;
02 00215 014046 PUNCH: DSZ LEAD
03 00216 000220 JMP .+2
04 00217 000223 JMP PUN1 ;JUMP IF LEADER ALREADY PUNCHED
05 00220 102400 SUB 0,0
06 00221 004253 JSR POUT ;PUNCH ZERO FRAME
07 00222 000370 JMP QUIT ;RETURN TO INTERRUPTED PROGRAM
08 00223 010046 PUN1: ISZ LEAD
09 00224 014045 DSZ DATA
10 00225 000227 JMP .+2
11 00226 000232 JMP PUN2 ;JUMP IF DATA ALREADY PUNCHED
12 00227 020045 LDA 0,DATA
13 00230 004253 JSR POUT ;PUNCH ONE FRAME OF DATA
14 00231 000370 JMP QUIT ;RETURN TO INTERRUPTED PROGRAM
15 00232 010045 PUN2: ISZ DATA
16 00233 014047 DSZ TRAIL
17 00234 000250 JMP PUN3
18 00235 102400 SUB 0,0
19 00236 024057 LDA 1,PFLAG
20 00237 125004 MOV 1,1,SZR
21 00240 000244 JMP .+4
22 00241 040065 STA 0,ENTR1
23 00242 060213 NIOC PTP
24 00243 000246 JMP .+3
25 00244 040067 STA 0,ENTR3
26 00245 060211 NIOC TTO
27 00246 063077 HALT ;GIVE OPERATOR TIME TO RESET SWITCHES
28 00247 000370 JMP QUIT
29 00250 102400 PUN3: SUB 0,0
30 00251 004253 JSR POUT ;PUNCH ZERO FRAME
31 00252 000370 JMP QUIT ;RETURN TO INTERRUPTED PROGRAM
32 00253 024057 POUT: LDA 1,PFLAG
33 00254 125004 MOV 1,1,SZR ;CHECK IF HIGH SPEED OR TTYOUTPUT
34 00255 000267 JMP .+12 ;JUMP FOR TELETYPE PUNCH
35 00256 063413 SKPBN PTP
36 00257 063613 SKPDN PTP
37 00260 063077 HALT ;BUSY OR DONE FLAG INCORRECT
38 00261 111000 MOV 0,2
39 00262 071113 DOAS 2,PTP ;PUNCH ONE FRAME FROM HIGH SPEED PUNCH
40 00263 063513 SKPBZ PTP
41 00264 063713 SKPDZ PTP
42 00265 063077 HALT ;BUSY OR DONE FLAG INCORRECT
43 00266 001400 JMP 0,3
44 00267 063411 SKPBN TTO
45 00270 063611 SKPDN TTO
46 00271 063077 HALT ;BUSY OR DONE FLAG INCORRECT
47 00272 171000 MOV 3,2
48 00273 115000 MOV 0,3
49 00274 075111 DOAS 3,TTO ;PUNCH ONE FRAME FROM TELETYPE
50 00275 063511 SKPBZ TTO
51 00276 063711 SKPDZ TTO
52 00277 063077 HALT ;BUSY OR DONE FLAG INCORRECT
53 00300 001000 JMP 0,2
54
```

↑ 0010 .MAIN

```
01 ; PAGE 9
02 00301 004340 INPUT: JSR READ ;INPUT ONE FRAME
03 00302 101004 MOV 0,0,SZR ;CHECK IF NULL INPUT
04 00303 000331 JMP INP1 ;NO
05 00304 024050 LDA 1,STRT
06 00305 125005 MOV 1,1,SNR
07 00306 000325 JMP INP
08 00307 024045 LDA 1,DATA
09 00310 125224 MOVZR 1,1,SZR
10 00311 063077 HALT ;NON ZERO DATA WAS READ AS ZERO
11 00312 102400 SUB 0,0
12 00313 024057 LDA 1,PFLAG
13 00314 125004 MOV 1,1,SZR
14 00315 000321 JMP .+4
15 00316 040066 STA 0,ENTR2
16 00317 060212 NIOC PTR
17 00320 000323 JMP .+3
18 00321 040070 STA 0,ENTR4
19 00322 060210 NIOC TTI
20 00323 063077 HALT ;GIVE OPERATOR TIME TO RESET SWITCHES
21 00324 000370 JMP QUIT
22 00325 014046 INP: DSZ LEAD
23 00326 000370 JMP QUIT
24 00327 063077 HALT ;TOO MUCH LEADER
25 00330 002377 JMP 0377
26 00331 126520 INP1: SUBZL 1,1
27 00332 044050 STA 1,STRT
28 00333 014045 DSZ DATA
29 00334 024045 LDA 1,DATA
30 00335 106414 SUB# 0,1,SZR
31 00336 063077 HALT ;DATA INCORRECT
32 00337 000370 JMP QUIT
33 00340 024057 READ: LDA 1,PFLAG
34 00341 125004 MOV 1,1,SZR
35 00342 000346 JMP .+4
36 00343 064512 DIAS 1,PTR ;READ ONE FRAME FROM HIGH SPEED READER
37 00344 121000 MOV 1,0
38 00345 001400 JMP 0,3
39 00346 070510 DIAS 2,TTI ;READ ONE FRAME FROM TELETYPE
40 00347 141000 MOV 2,0
41 00350 001400 JMP 0,3
42 00351 040060 INPRC: STA 0,SREG0 ;SAVE REGISTERS
43 00352 044061 STA 1,SREG1
44 00353 050062 STA 2,SREG2
45 00354 054063 STA 3,SREG3
46 00355 175200 MOVR 3,3
47 00356 054064 STA 3,SAVC ;SAVE CARRY FLAG
48 00357 061477 INTA 0
49 00360 034054 LDA 3,C17
50 00361 117400 AND 0,3
51 00362 024052 LDA 1,C10
52 00363 030053 LDA 2,C15
53 00364 156433 SUBZ# 2,3,SNC
54 00365 136423 SUBZ 1,3,SNC
55 00366 063077 HALT ;UNKNOWN INTERRUPT
56 00367 003471 JMP 0IVECT,3
57
```

↑ 0011 ,MAIN

PAGE 10

```
01 ;
02 00370 020064 QUIT: LDA 0,SAVC ;RESTORE CARRY
03 00371 101100 MOVL 0,0
04 00372 020060 LDA 0,SREG0 ;RESTORE REGISTERS
05 00373 024061 LDA 1,SREG1
06 00374 030062 LDA 2,SREG2
07 00375 034063 LDA 3,SREG3
08 00376 060177 INTEN
09 00377 002000 JMP 00
10 00400 102421 ALU: SUBZ 0,0,SKP
11 00401 063077 HALT ;UNCONDITIONAL SKIP FAILED
12 00402 101101 MOVL 0,0,SKP
13 00403 063077 HALT ;UNCONDITIONAL SKIP FAILED
14 00404 101041 MOVO 0,0,SKP
15 00405 063077 HALT ;UNCONDITIONAL SKIP FAILED
16 00406 102441 SUBO 0,0,SKP
17 00407 063077 HALT ;UNCONDITIONAL SKIP FAILED
18 ;
19 ; TEST SZR AND SNR FUNCTION
20 ;
21 00410 103405 AND 0,0,SNR
22 00411 101004 MOV 0,0,SZR
23 00412 063077 HALT ;SZR OR SNR FAILED
24 00413 101404 INC 0,0,SZR
25 00414 103405 AND 0,0,SNR
26 00415 063077 HALT ;SZR OR SNR FAILED
27 00416 103044 ADDO 0,0,SZR
28 00417 101005 MOV 0,0,SNR
29 00420 063077 HALT ;SZR OR SZR FAILED
30 00421 102425 SUBZ 0,0,SNR
31 00422 101004 MOV 0,0,SZR
32 00423 063077 HALT ;SZR OR SNR FAILED
33 ;
34 ; TEST SZC AND SNC FUNCTION
35 ;
36 00424 102422 SUBZ 0,0,SZC
37 00425 101003 MOV 0,0,SNC
38 00426 063077 HALT ;SZC OR SNC FAILED
39 00427 101402 INC 0,0,SZC
40 00430 101003 MOV 0,0,SNC
41 00431 063077 HALT ;SZC OR SNC FAILED
42 00432 101023 MOVZ 0,0,SNC
43 00433 101002 MOV 0,0,SZC
44 00434 063077 HALT ;SZC OR SNC FAILED
45 00435 102443 SUBO 0,0,SNC
46 00436 101002 MOV 0,0,SZC
47 00437 063077 HALT ;SZC OR SNC FAILED
48
```

↑ 0012 .MAIN

PAGE 11

```
01 ;
02 ;
03 ; TEST SEZ AND SBN FUNCTION
04 ;
05 00440 102447 SUBO 0,0,SBN
06 00441 101006 MOV 0,0,SEZ
07 00442 063077 HALT ;SEZ OR SBN FAILED
08 00443 101047 MOVO 0,0,SBN
09 00444 101006 MOV 0,0,SEZ
10 00445 063077 HALT ;SEZ OR SBN FAILED
11 00446 101406 INC 0,0,SEZ
12 00447 101007 MOV 0,0,SBN
13 00450 063077 HALT ;SEZ OR SBN FAILED
14 00451 101067 MOVC 0,0,SBN
15 00452 101006 MOV 0,0,SEZ
16 00453 063077 HALT ;SEZ OR SBN FAILED
17 ;
18 ; TEST CARRY
19 ;
20 00454 102525 SUBZL 0,0,SNR
21 00455 063077 HALT ;CARRY DID NOT SHIFT INTO ACO
22 00456 101002 MOV 0,0,SZC
23 00457 063077 HALT ;CARRY DID NOT GET UPDATED
24 00460 101203 MOVR 0,0,SNC
25 00461 063077 HALT ;ONE BIT DID NOT SHIFT INTO CARRY
26 00462 101062 MOVC 0,0,SZC
27 00463 063077 HALT ;CARRY BIT DID NOT COMPLEMENT
28 00464 101043 MOVO 0,0,SNC
29 00465 063077 HALT ;CARRY BIT WAS NOT FORCED TO ONE
30 00466 101022 MOVZ 0,0,SZC
31 00467 063077 HALT ;CARRY BIT WAS NOT FORCED TO ZERO
32 00470 102620 SUBZR 0,0 ;(ACO) = 100000
33 00471 126520 SUBZL 1,1 ;(AC1) = 000001
34 00472 122402 SUB 1,0,SZC
35 00473 101003 MOV 0,0,SNC
36 00474 063077 HALT ;SUBTRACTION DID NOT GENERATE CARRY
37 ;
38 ; TEST REGISTER TO REGISTER MOVE
39 ;
40 00475 102400 SUB 0,0
41 00476 100005 COM 0,0,SNR
42 00477 063077 HALT ;COM FAILED, CHECK PANEL
43 00500 105000 MOV 0,1
44 00501 135000 MOV 1,3
45 00502 171000 MOV 3,2
46 00503 112404 SUB 0,2,SZR
47 00504 063077 HALT ;RESULT OF SUB SHOULD HAVE BEEN ZERO
48 00505 136404 SUB 1,3,SZR
49 00506 063077 HALT ;RESULT OF SUB SHOULD HAVE BEEN ZERO
50
```

↑ 0013 .MAIN

PAGE 12

```
01 ;
02 ;
03 ; TEST LEFT AND RIGHT SHIFT
04 ;
05 00507 102525 SUBZL 0,0,SNR
06 00510 063077 HALT ;BIT DID NOT LEFT SHIFT
07 00511 101203 MOVR 0,0,SNC
08 00512 063077 HALT ;BIT DID NOT RIGHT SHIFT INTO CARRY
09 00513 102625 SUBZR 0,0,SNR
10 00514 063077 HALT ;BIT DID NOT RIGHT SHIFT
11 00515 101103 MOVL 0,0,SNC
12 00516 063077 HALT ;BIT DID NOT LEFT SHIFT INTO CARRY
13 ;
14 ; TEST INC AGAINST ADD THROUGH ENTIRE RANGE
15 ;
16 00517 102520 SUBZL 0,0 ;(AC0) = 1
17 00520 111000 MOV 0,2 ;(AC2) = 1
18 00521 145000 MOV 2,1 ;(AC1) = 1
19 00522 101400 INC 0,0
20 00523 147004 ADD 2,1,SZR
21 00524 000776 JMP ,-2
22 00525 101004 MOV 0,0,SZR
23 00526 063077 HALT ;INC OR ADD FAILED
24 ;
25 ; TEST COMPLEMENT
26 ;
27 00527 102400 SUB 0,0 ;(AC0) = 0
28 00530 126520 SUBZL 1,1 ;(AC1) = 1
29 00531 100000 COM 0,0 ;(AC0) SHOULD = 177777
30 00532 123004 ADD 1,0,SZR
31 00533 063077 HALT ;COMPLEMENT FAILED
32 ;
33 ; TEST LOAD INHIBIT
34 ;
35Q00534 100010 COM# 0,0
36 00535 101004 MOV 0,0,SZR
37 00536 063077 HALT ;LOAD INHIBIT FAILED
38 00537 101400 INC 0,0
39Q00540 102410 SUB# 0,0
40 00541 101005 MOV 0,0,SNR
41 00542 063077 HALT ;LOAD INHIBIT FAILED
42 00543 126400 SUB 1,1
43Q00544 107010 ADD# 0,1
44 00545 125004 MOV 1,1,SZR
45 00546 063077 HALT ;LOAD INHIBIT FAILED
46Q00547 105010 MOV# 0,1
47 00550 125004 MOV 1,1,SZR
48 00551 063077 HALT ;LOAD INHIBIT FAILED
49
```

↑ 0014 ,MAIN

PAGE 13

```
01 ;
02 ;
03 ; TEST ABILITY TO LEFT SHIFT A BIT THROUGH THE
04 ; ENTIRE REGISTER AND INTO THE CARRY
05 ;
06 00552 102520 SUBZL 0,0 ;(AC0) = 1
07 00553 105000 MOV 0,1 ;(AC1) = 1
08 00554 101120 MOVZL 0,0 ;(AC0) = 2
09 00555 127000 ADD 1,1 ;(AC1) = 2
10 00556 106414 SUB# 0,1,SZR
11 00557 063077 HALT ;BIT 15 DID NOT LEFT SHIFT
12 00560 101120 MOVZL 0,0 ;(AC0) = 4
13 00561 127000 ADD 1,1 ;(AC1) = 4
14 00562 106414 SUB# 0,1,SZR
15 00563 063077 HALT ;BIT 14 DID NOT LEFT SHIFT
16 00564 101120 MOVZL 0,0 ;(AC0) = 10
17 00565 127000 ADD 1,1 ;(AC1) = 10
18 00566 106414 SUB# 0,1,SZR
19 00567 063077 HALT ;BIT 13 DID NOT LEFT SHIFT
20 00570 101120 MOVZL 0,0 ;(AC0) = 20
21 00571 127000 ADD 1,1 ;(AC1) = 20
22 00572 106414 SUB# 0,1,SZR
23 00573 063077 HALT ;BIT 12 DID NOT LEFT SHIFT
24 00574 101120 MOVZL 0,0 ;(AC0) = 40
25 00575 127000 ADD 1,1 ;(AC1) = 40
26 00576 106414 SUB# 0,1,SZR
27 00577 063077 HALT ;BIT 11 DID NOT LEFT SHIFT
28 00600 101120 MOVZL 0,0 ;(AC0) = 100
29 00601 127000 ADD 1,1 ;(AC1) = 100
30 00602 106414 SUB# 0,1,SZR
31 00603 063077 HALT ;BIT 10 DID NOT LEFT SHIFT
32 00604 101120 MOVZL 0,0 ;(AC0) = 200
33 00605 127000 ADD 1,1 ;(AC1) = 200
34 00606 106414 SUB# 0,1,SZR
35 00607 063077 HALT ;BIT 9 DID NOT LEFT SHIFT
36 00610 101120 MOVZL 0,0 ;(AC0) = 400
37 00611 127000 ADD 1,1 ;(AC1) = 400
38 00612 106414 SUB# 0,1,SZR
39 00613 063077 HALT ;BIT 8 DID NOT LEFT SHIFT
40 00614 101120 MOVZL 0,0 ;(AC0) = 1000
41 00615 127000 ADD 1,1 ;(AC1) = 1000
42 00616 106414 SUB# 0,1,SZR
43 00617 063077 HALT ;BIT 7 DID NOT LEFT SHIFT
44 00620 101120 MOVZL 0,0 ;(AC0) = 2000
45 00621 127000 ADD 1,1 ;(AC1) = 2000
46 00622 106414 SUB# 0,1,SZR
47 00623 063077 HALT ;BIT 6 DID NOT LEFT SHIFT
48
```

↑ 0015 .MAIN

PAGE 14

```
01
02 00624 101120      MOVZL  0,0      ;(AC0) = 4000
03 00625 127000      ADD     1,1      ;(AC1) = 4000
04 00626 106414      SUB#    0,1,SZR
05 00627 063077      HALT                    ;BIT 5 DID NOT LEFT SHIFT
06 00630 101120      MOVZL  0,0      ;(AC0) = 10000
07 00631 127000      ADD     1,1      ;(AC1) = 10000
08 00632 106414      SUB#    0,1,SZR
09 00633 063077      HALT                    ;BIT 4 DID NOT LEFT SHIFT
10 00634 101120      MOVZL  0,0      ;(AC0) = 20000
11 00635 127000      ADD     1,1      ;(AC1) = 20000
12 00636 106414      SUB#    0,1,SZR
13 00637 063077      HALT                    ;BIT 3 DID NOT LEFT SHIFT
14 00640 101120      MOVZL  0,0      ;(AC0) = 40000
15 00641 127000      ADD     1,1      ;(AC1) = 40000
16 00642 106414      SUB#    0,1,SZR
17 00643 063077      HALT                    ;BIT 2 DID NOT LEFT SHIFT
18 00644 101120      MOVZL  0,0      ;(AC0) = 100000
19 00645 127000      ADD     1,1      ;(AC1) = 100000
20 00646 106414      SUB#    0,1,SZR
21 00647 063077      HALT                    ;BIT 1 DID NOT LEFT SHIFT
22 00650 101123      MOVZL  0,0,SNC
23 00651 063077      HALT                    ;BIT DID NOT LEFT SHIFT INTO CARRY
24 00652 125124      MOVZL  1,1,SZR
25 00653 063077      HALT                    ;(AC1) SHOULD HAVE ENDED UP ZERO
26
27 ; TEST ABILITY TO RIGHT SHIFT A BIT THROUGH
28 ; AN ENTIRE WORD INTO THE CARRY
29 ;
30 00654 152620      SUBZR  2,2      ;SET AC2 TO 100000
31 00655 153600      ANDR   2,2
32 00656 153600      ANDR   2,2
33 00657 153600      ANDR   2,2
34 00660 153600      ANDR   2,2
35 00661 153600      ANDR   2,2
36 00662 153600      ANDR   2,2
37 00663 153600      ANDR   2,2
38 00664 153600      ANDR   2,2
39 00665 153600      ANDR   2,2
40 00666 153600      ANDR   2,2
41 00667 153600      ANDR   2,2
42 00670 153600      ANDR   2,2
43 00671 153600      ANDR   2,2
44 00672 153600      ANDR   2,2
45 00673 153600      ANDR   2,2
46 00674 153603      ANDR   2,2,SNC
47 00675 063077      HALT                    ;BIT DID NOT ROTATE INTO CARRY
48
```

↑ 0016 .MAIN

PAGE 15

```
01 ;
02 00676 000402 JMP ,+2
03 00677 177400 LFHAF: 177400
04 ;
05 ; TEST ABILITY TO SWAP HALVES OF REGISTER
06 ;
07 00700 020777 LDA 0,LFHAF ;(AC0) = 177400
08 00701 024776 LDA 1,LFHAF ;(AC1) = 177400
09 00702 030775 LDA 2,LFHAF ;(AC2) = 177400
10 00703 115300 MOVS 0,3 ;(AC3) = 377
11 00704 117000 ADD 0,3 ;(AC3) = 177777
12 00705 174004 COM 3,3,SZR ;(AC3) SHOULD NOW BE ZERO
13 00706 063077 HALT ;SWAP FAILED
14 00707 135300 MOVS 1,3
15 00710 137000 ADD 1,3
16 00711 174004 COM 3,3,SZR
17 00712 063077 HALT ;SWAP FAILED
18 00713 155300 MOVS 2,3
19 00714 157000 ADD 2,3
20 00715 174004 COM 3,3,SZR
21 00716 063077 HALT ;SWAP FAILED
22 00717 034760 LDA 3,LFHAF
23 00720 165300 MOVS 3,1
24 00721 167000 ADD 3,1
25 00722 124004 COM 1,1,SZR
26 00723 063077 HALT ;SWAP FAILED
27 ;
28 ; TEST ADD COMPLEMENT
29 ;
30 00724 176420 SUBZ 3,3 ;INITIALIZE AC0 TO ZERO
31 00725 164000 COM 3,1 ;PLACE COMPLEMENT IN AC1
32 00726 161000 MOV 3,0
33 00727 102000 ADC 0,0
34 00730 122404 SUB 1,0,SZR
35 00731 063077 HALT ;ADC FAILED
36 00732 175404 INC 3,3,SZR
37 00733 000773 JMP ,-5
38 ;
39 ; TEST NEGATE
40 ;
41 00734 102420 SUBZ 0,0
42 00735 152520 SUBZL 2,2
43 00736 104000 COM 0,1
44 00737 147000 ADD 2,1 ;FORM NEGATIVE OF AC0
45 00740 114400 NEG 0,3
46 00741 136404 SUB 1,3,SZR ;COMPARE SEPARATELY DERIVED VALUES
47 00742 063077 HALT ;NEGATE FAILED
48 00743 101404 INC 0,0,SZR
49 00744 000772 JMP ,-6
50
```



```

01 ;
02 ;
03 ; TEST AND FUNCTION
04 ;
05 00745 152520 SUBZL 2,2
06 00746 126520 SUBZL 1,1 ;FLOATING MASK
07 00747 141000 MOV 2,0
08 00750 123440 ANDO 1,0
09 00751 142415 SUB# 2,0,SNR
10 00752 000404 JMP ,+4
11 00753 125405 INC 1,1,SNR
12 00754 000404 JMP ,+4
13 00755 000772 JMP , -6
14 00756 101002 MOV 0,0,SZC
15 00757 122404 SUB 1,0,SZR
16 00760 063077 HALT ;AND FAILED, CHECK IF CARRY STILL 1
17 00761 153004 ADD 2,2,SZR
18 00762 000764 JMP , -14
19 ;
20 ; TEST I/O COMMANDS WITH DEVICE 0 AND 77
21 ;
22 00763 060100 NIOS 0
23 00764 063400 SKPBN 0
24 00765 063500 SKPBZ 0
25 00766 063077 HALT ;BUSY SHOULD ALWAYS BE ZERO
26 00767 063600 SKPDN 0
27 00770 063700 SKPDZ 0
28 00771 063077 HALT ;DONE SHOULD ALWAYS BE ZERO
29 00772 060177 NIOS CPU ;ENABLE INTERRUPT
30 00773 060077 NIO CPU ;THESE INSTRUCTIONS SHOULD
31 00774 060377 NIOP CPU ; NOT AFFECT INTERRUPT ON
32 00775 060200 NIOC 0
33 00776 063577 SKPBZ CPU
34 00777 063477 SKPBN CPU
35 01000 063077 HALT ;INTERRUPT ON NOT SET
36 01001 060277 NIOC CPU ;DISABLE INTERRUPT
37 01002 060077 NIO CPU
38 01003 060377 NIOP CPU
39 01004 060100 NIOS 0
40 01005 063477 SKPBN CPU
41 01006 063577 SKPBZ CPU
42 01007 063077 HALT ;INTERRUPT ON SHOULD BE CLEAR
43 01010 126000 ADC 1,1
44 01011 064400 DIA 1,0
45 01012 125004 MOV 1,1,SZR
46 01013 063077 HALT ;UNDEFINED DEVICE, SHOULD CLEAR AC
47 01014 126000 ADC 1,1
48 01015 065400 DIB 1,0
49 01016 125004 MOV 1,1,SZR
50 01017 063077 HALT ;UNDEFINED DEVICE, SHOULD CLEAR AC
51 01020 126000 ADC 1,1
52 01021 066400 DIC 1,0
53 01022 125004 MOV 1,1,SZR
54 01023 063077 HALT ;UNDEFINED DEVICE, SHOULD CLEAR AC
55 01024 060177 NIOS CPU
56

```

```

01 ;
02 ;
03 ; TEST ABILITY OF PC COUNTER TO INCREMENT BY
04 ; FILLING A BUFFER WITH A DSZ 0,2 INSTRUCTION.
05 ; IF PC COUNTS PROPERLY AND DSZ WORKS PROPERLY
06 ; A NORMAL RETURN IS TAKEN. THE LDA AND STA
07 ; INSTRUCTION ARE ALSO TESTED WHILE LOADING THE
08 ; BUFFER
09 ;
10 01025 024043 PC: LDA 1,DIF ; ITERATION COUNT
11 01026 044436 STA 1,PC3 ; INITIALIZE ADDR. TO BE DECREMENTED
12 01027 124400 NEG 1,1 ; NEGATE FOR STORE COUNTER
13 01030 030432 LDA 2,PC1 ; DSZ 0,2 INSTRUCTION
14 01031 034041 LDA 3,BUF ; BUFFER START ADDR.
15 01032 051400 STA 2,0,3 ; STORE INSTR. IN BUFFER
16 01033 175400 INC 3,3 ; INCREMENT BUFFER ADDR.
17 01034 125400 INC 1,1 ; INCREMENT STORE COUNT
18 01035 031777 LDA 2,-1,3 ; LOAD LAST VALUE STORED
19 01036 051400 STA 2,0,3 ; STORE IN CURRENT LOCATION
20 01037 175400 INC 3,3 ; INCREMENT BUFFER ADDR.
21 01040 125404 INC 1,1,SZR ; INCREMENT COUNT, SKIP IF DONE
22 01041 000774 JMP , -4 ; CONTINUE STORING
23 01042 024420 LDA 1,PC1 ; LOAD ORIGINAL VALUE STORED
24 01043 132414 SUB# 1,2,SZR ; SKIP IF FIRST SAME AS LAST
25 01044 063077 HALT ; LOAD OR STORE ERROR SINCE
26 ; EACH VALUE IN THE BUFFER
27 ; DEPENDS ON THE ONE BEFORE IT
28 01045 030416 LDA 2,PC2 ; JMP 0,3 INSTRUCTION
29 01046 051400 STA 2,0,3 ; NEXT TO LAST IN BUFFER
30 01047 175400 INC 3,3 ; INCREMENT BUFFER ADDR.
31 01050 151400 INC 2,2 ; CREATE JMP 1,3 INSTRUCTION
32 01051 051400 STA 2,0,3 ; LAST IN BUFFER
33 01052 030413 LDA 2,PC4 ; ADDR. TO BE DECREMENTED
34 01053 034041 LDA 3,BUF ; BUFFER START ADR.
35 01054 005400 JSR 0,3 ; GO TO BUFFER
36 01055 063077 HALT ; (PC3)=0 IMPLIES DSZ DID NOT SKIP
37 ; ON 0. (PC3)≠0 IMPLIES PC DID
38 ; COUNT PROPERLY.
39 01056 020406 LDA 0,PC3 ; ADDR. WHICH WAS DECREMENTED
40 01057 101004 MOV 0,0,SZR ; SKIP IF 0, OK
41 01060 063077 HALT ; PC COUNT ERROR SINCE ERROR
42 ; RETURN WAS NOT TAKEN YET
43 ; (PC3) NOT = 0
44 01061 000406 JMP PCA
45 01062 015000 PC1: DSZ 0,2
46 01063 001400 PC2: JMP 0,3
47 01064 000000 PC3: 0
48 01065 001064 PC4: PC3
49 01066 011000 PC5: ISZ 0,2
50

```

↑ 0019 .MAIN

PAGE 18

```
01 ;
02 ;
03 ; ANOTHER TEST OF THE PC COUNTER ALONG
04 ; WITH THE ISZ, LDA, AND STA INSTRUCTIONS.
05 ;
06 01067 024043 PCA: LDA 1,DIF ;ITERATION COUNT
07 01070 124400 NEG 1,1 ;NEGATE TO USE WITH ISZ
08 01071 044773 STA 1,PC3 ;INITIALIZE ADDR. TO BE INCREMENTED
09 01072 030774 LDA 2,PC5 ;ISZ 0,2 INSTRUCTION
10 01073 034041 LDA 3,BUF ;BUFFER START ADDR.
11 01074 051400 STA 2,0,3 ;STORE INSTRUCTION IN BUFFER
12 01075 175400 INC 3,3 ;INCREMENT BUFFER ADDR.
13 01076 125400 INC 1,1 ;INCREMENT STORE COUNT
14 01077 031777 LDA 2,-1,3 ;LOAD LAST VALUE STORED
15 01100 051400 STA 2,0,3 ;STORE IN CURRENT LOCATION
16 01101 175400 INC 3,3 ;INCREMENT BUFFER ADDR.
17 01102 125404 INC 1,1,SZR ;INCREMENT COUNT, SKIP IF DONE
18 01103 000774 JMP -4 ;CONTINUE STORING
19 01104 024762 LDA 1,PC5 ;LOAD ORIGINAL VALUE STORED
20 01105 132414 SUB# 1,2,SZR ;SKIP IF LAST SAME AS FIRST
21 01106 063077 HALT ;LOAD OR STORE ERROR SINCE
22 ; EACH VALUE IN THE BUFFER
23 ; DEPENDS ON THE ONE BEFORE IT
24 01107 030754 LDA 2,PC2 ;JMP 0,3 INSTRUCTION
25 01110 051400 STA 2,0,3 ;NEXT TO LAST IN BUFFER
26 01111 175400 INC 3,3 ;INCREMENT BUFFER ADDR.
27 01112 151400 INC 2,2 ;CREATE JMP 1,3 INSTRUCTION
28 01113 051400 STA 2,0,3 ;LAST IN BUFFER
29 01114 030751 LDA 2,PC4 ;ADDR. TO BE INCREMENTED
30 01115 034041 LDA 3,BUF ;BUFFER START ADDR.
31 01116 005400 JSR 0,3 ;GO TO BUFFER
32 01117 063077 HALT ;(PC3)=0 IMPLIES ISZ DID NOT SKIP
33 ; ON 0. (PC3)≠0 IMPLIES PC DID
34 ; NOT COUNT PROPERLY
35 01120 020744 LDA 0,PC3 ;ADDR. WHICH WAS DECREMENTED
36 01121 101004 MOV 0,0,SZR ;SKIP IF 0, OK
37 01122 063077 HALT ;PC COUNT ERROR SINCE ERROR
38 ; RETURN WAS NOT TAKEN YET
39 ; (PC3) NOT = 0
40 ;
41 ; TEST ABILITY TO LOAD AND STORE IN EACH
42 ; CELL OF THE BUFFER BY STORING A ONE BIT
43 ; IN EACH BIT POSITION OF THE WORD SURROUNDED
44 ; BY A FIELD OF ZERO BITS AND CHECKING
45 ; THE READBACK. THEN A ZERO BIT IS STORED
46 ; IN EACH BIT POSITION SURROUNDED BY A
47 ; FIELD OF ONES.
48 ;
49 01123 034041 PATER: LDA 3,BUF ;BUFFER START ADDR.
50 01124 020042 LDA 0,BUFND ;BUFFER END ADDR.
51 01125 055400 STA 3,0,3 ;MAKE CONTENTS OF ADDR. = ADDR.
52 01126 175400 INC 3,3 ;INCREMENT BUFFER ADDR.
53 01127 162414 SUB# 3,0,SZR ;SKIP IF END OF BUFFER
54 01130 000775 JMP -3 ;CONTINUE STORING
55 01131 030041 LDA 2,BUF ;BUFFER START ADDR.
56
```

↑ 0020 .MAIN

```
01 ; PAGE 19
02 01132 102520 PAT1: SUBZL 0,0 ;INITIALIZE (AC0) TO 1
03 01133 025000 LDA 1,0,2 ;LOAD CONTENTS OF BUFFER ADDR.
04 01134 132414 SUB# 1,2,SZR ;SKIP IF CONTENTS EQUAL ADDR.
05 01135 063077 HALT ;ADDRESS HAD IMPROPER CONTENTS.
06 ; POSSIBLY ADDRESS DECODING
07 ; ERROR DURING LOAD OR STORE
08 01136 041000 STA 0,0,2 ;STORE BIT PATTERN
09 01137 000403 JMP .+3 ;DIVERSION BEFORE RELOADING
10 01140 025000 LDA 1,0,2 ;LOAD BIT PATTERN
11 01141 000402 JMP .+2
12 01142 000776 JMP .-2
13 01143 106414 SUB# 0,1,SZR ;SKIP IF BIT PATTERNS MATCH
14 01144 000435 JMP BITER ;BIT PATTERN NOT READ AS WRITTEN
15 01145 101123 MOVZL 0,0,SNC ;HAS 1 FLOATED ACROSS ENTIRE WORD
16 01146 000770 JMP .-10 ;NO
17 01147 102520 SUBZL 0,0
18 01150 100000 COM 0,0 ;INITIALIZE (AC0) TO 17776
19 01151 041000 STA 0,0,2 ;STORE BIT PATTERN
20 01152 000403 JMP .+3 ;DIVERSION BEFORE RELOADING
21 01153 025000 LDA 1,0,2 ;LOAD BIT PATTERN
22 01154 000402 JMP .+2
23 01155 000776 JMP .-2
24 01156 106414 SUB# 0,1,SZR ;SKIP IF BIT PATTERNS MATCH
25 01157 000422 JMP BITER ;BIT PATTERN NOT READ AS WRITTEN
26 01160 101142 MOVOL 0,0,SZC ;HAS 0 FLOATED ACROSS ENTIRE WORD
27 01161 000770 JMP .-10 ;NO
28 01162 102400 SUB 0,0
29 01163 041000 STA 0,0,2 ;STORE ALL ZEROS
30 01164 000403 JMP .+3 ;DIVERSION BEFORE RELOADING
31 01165 025000 LDA 1,0,2 ;LOAD BIT PATTERN
32 01166 000402 JMP .+2
33 01167 000776 JMP .-2
34 01170 106414 SUB# 0,1,SZR ;SKIP IF BIT PATTERNS MATCH
35 01171 000410 JMP BITER ;BIT PATTERN NOT READ AS WRITTEN
36 01172 100000 COM 0,0
37 01173 041000 STA 0,0,2 ;STORE ALL ONES
38 01174 000403 JMP .+3 ;DIVERSION BEFORE RELOADING
39 01175 025000 LDA 1,0,2 ;LOAD BIT PATTERN
40 01176 000402 JMP .+2
41 01177 000776 JMP .-2
42 01200 106414 SUB# 0,1,SZR ;SKIP IF BIT PATTERNS MATCH
43 01201 063077 BITER: HALT ;THIS BIT PATTERN COULD NOT BE
44 ; LOADED BACK THE SAME AS IT WAS
45 ; STORED, AC0 HAS STORED VALUE
46 ; AND AC1 HAS LOADED VALUE.
47 ; AC2 CONTAINS BUFFER ADDR.
48 01202 151400 INC 2,2 ;INCREMENT BUFFER ADDR.
49 01203 024042 LDA 1,BUFND ;BUFFER END ADDR.
50 01204 146414 SUB# 2,1,SZR ;SKIP IF END OF BUFFER
51 01205 000725 JMP PAT1
52
```

↑ 0021 .MAIN

PAGE 20

```
01 ;
02 ;
03 ; TEST ABILITY OF JSR INSTRUCTION TO JUMP TO
04 ; EACH CELL OF THE BUFFER AND RETURN WITH
05 ; PROPER ADDRESS IN AC3
06 ;
07 01206 024042 RETAD: LDA 1, BUFND ; BUFFER END ADDR.
08 01207 034041 LDA 3, BUF ; BUFFER START ADDR.
09 01210 020416 LDA 0, RET1 ; JSR 0,3 INSTRUCTION
10 01211 041400 STA 0,0,3 ; STORE INSTRUCTION IN BUFFER
11 01212 175400 INC 3,3 ; INCREMENT BUFFER ADDR.
12 01213 166414 SUB# 3,1, SZR ; SKIP IF END OF BUFFER
13 01214 000775 JMP .-3 ; CONTINUE STORING
14 01215 030041 LDA 2, BUF ; INIT. AC2 WITH BUFFER START ADDR.
15 01216 034411 LDA 3, RET2 ; SET UP RETURN ADDR.
16 01217 001000 JMP 0,2 ; JMP TO BUFFER
17 01220 151400 RETA: INC 2,2 ; INCREMENT BUFFER ADDR.
18 01221 156414 SUB# 2,3, SZR ; SKIP IF EQUAL, OK
19 01222 063077 HALT ; JSR INSTRUCTION FAILED TO
20 ; AC3 TO PC+1
21 01223 146414 SUB# 2,1, SZR ; SKIP IF END OF BUFFER
22 01224 000772 JMP .-6 ; CONTINUE TEST
23 01225 000403 JMP .+3 ; JUMP OVER CONSTANTS
24 01226 005400 RET1: JSR 0,3
25 01227 001220 RET2: RETA
26 ;
27 ; TEST ABILITY TO DECODE POSITIVE DISPLACEMENT
28 ; USING THE JMP, LDA, AND STA INSTRUCTIONS.
29 ;
30 01230 020427 PDISP: LDA 0, C377
31 01231 030041 LDA 2, BUF ; BUFFER START ADDR.
32 01232 024417 LDA 1, DIS1 ; JMP 0,2 INSTRUCTION
33 01233 044402 STA 1, PDIS1 ; INITIALIZE JMP INSTRUCTION
34 01234 034416 LDA 3, DIS2 ; SET UP RETURN ADDRESS
35 01235 001000 PDIS1: JMP 0,2 ; JMP TO BUFFER
36 01236 010777 ISZ PDIS1 ; INCREMENT DISPLACEMENT
37 01237 024776 LDA 1, PDIS1 ; LOAD THE INSTRUCTION
38 01240 107400 AND 0,1 ; ISOLATE THE DISPLACEMENT
39 01241 156400 SUB 2,3 ; DISPLACEMENT FROM START OF BUFFER
40 01242 136404 SUB 1,3, SZR ; SKIP IF EQUAL, OK
41 01243 063077 HALT ; JMP WENT TO WRONG PLACE IN
42 ; BUFFER OR THE JSR GIVING THE
43 ; PROPER RETURN ADDR. WOULD HAVE
44 ; BEEN EXECUTED
45 01244 024771 LDA 1, PDIS1 ; LOAD VARIABLE DISPLACEMENT INSTR.
46 01245 034413 LDA 3, JMPND ; MAXIMUM POS. DISPLACEMENT INSTRUCTIO.
47 01246 136404 SUB 1,3, SZR ; SKIP IF TEST DONE
48 01247 000765 JMP PDIS1-1 ; CONTINUE TEST
49 01250 000411 JMP NDISP ; JUMP OVER CONSTANTS
50 01251 001000 DIS1: JMP 0,2
51 01252 001236 DIS2: PDIS1+1
52 01253 001200 DIS3: JMP -200,2
53 01254 001271 DIS4: NDIS1+1
54 01255 000200 C200: 200
55 01256 177400 LF: 177400
56 01257 000377 C377: 377
57 01260 001177 JMPND: JMP 177,2
58
```

↑ 0022 .MAIN

PAGE 21

```
01 ;
02 ;
03 ; TEST ABILITY TO DECODE NEGATIVE DISPLACEMENT
04 ; USING THE JMP, LDA, AND STA INSTRUCTIONS.
05 ;
06 01261 020776 NDISP: LDA 0,C377 ;DISPLACEMENT MASK
07 01262 030041 LDA 2,BUF ;BUFFER START ADDR.
08 01263 024772 LDA 1,C200
09 01264 133000 ADD 1,2 ;POSITION POINTER TO 200TH LOC.
10 01265 024766 LDA 1,DIS3 ;JMP -200,2 INSTRUCTION
11 01266 044402 STA 1,NDIS1 ;INIT. INSTR. TO MAX. NEG. DISPLACEMENT
12 01267 034765 LDA 3,DIS4 ;SET UP RETURN ADDR.
13 01270 001200 NDIS1: JMP -200,2 ;GO TO BUFFER
14 01271 156400 SUB 2,3
15 01272 117400 AND 0,3 ;KEEP 8 BIT RESULT ONLY
16 01273 024775 LDA 1,NDIS1 ;LOAD INSTRUCTION
17 01274 107400 AND 0,1 ;ISOLATE DISPLACEMENT
18 01275 125400 INC 1,1 ;INCREMENT DISPLACEMENT
19 01276 136414 SUB# 1,3,SZR ;SKIP IF EQUAL, OK
20 01277 063077 HALT ;JMP WENT TO WRONG PLACE IN
21 ; BUFFER OR THE JSR GIVING THE
22 ; PROPER RETURN ADDR. WOULD
23 ; HAVE BEEN EXECUTED
24 01300 034770 LDA 3,NDIS1 ;LOAD INSTRUCTION
25 01301 020755 LDA 0,LF ;LEFT 8 BITS MASK
26 01302 117400 AND 0,3 ;MASK OFF OLD DISPLACEMENT
27 01303 137000 ADD 1,3 ;ADD IN NEW DISPLACEMENT
28 01304 054764 STA 3,NDIS1 ;STORE INSTRUCTION BACK
29 01305 020752 LDA 0,C377 ;-1 DISPLACEMENT
30 01306 106414 SUB# 0,1,SZR ;SKIP IF DONE
31 01307 000760 JMP NDIS1-1 ;CONTINUE TEST
32
```

```

01 ;
02 ;
03 ; TEST THE INDIRECT FUNCTION USING THE
04 ; LDA AND STA INSTRUCTIONS.
05 ;
06 01310 030041 INDIR: LDA 2,BUF ;BUFFER START ADDR.
07 01311 034042 LDA 3,BUFND ;BUFFER END ADDR.
08 01312 051000 STA 2,0,2 ;MAKE ADDR. CONTENTS = ADDR.
09 01313 151400 INC 2,2 ;INCREMENT BUFFER ADDR.
10 01314 156414 SUB# 2,3,SZR ;SKIP IF END OF BUFFER
11 01315 000775 JMP ,-3 ;CONTINUE STORING
12 01316 030041 LDA 2,BUF ;BUFFER START ADDR.
13 01317 052422 INDR1: STA 2,@IND1 ;SHOULD END UP IN IND2
14 01320 024422 LDA 1,IND2 ;INVESTIGATE IND2
15 01321 132414 SUB# 1,2,SZR ;SKIP IF DESIRED VALUE THERE
16 01322 063077 HALT ;STORE INDIRECT ERROR HAS OCCURED
17 ; ; DISPLAY LOCATION IND2 AC2
18 ; ; CONTAINS EXPECTED VALUE
19 01323 020420 LDA 0,IND3
20 01324 024415 LDA 1,IND1
21 01325 106414 SUB# 0,1,SZR ;SKIP IF ADDR. REMAINED CORRECT
22 01326 063077 HALT ;STORE INDIRECT HAS SOMEHOW
23 ; ; MODIFIED THE ADDR. THROUGH
24 ; ; WHICH IT STORED INDIRECTLY
25 01327 026413 LDA 1,@IND2 ;LOAD INDIRECTLY FROM BUFFER
26 01330 020412 LDA 0,IND2 ;INVESTIGATE CONTENTS OF IND2
27 01331 112414 SUB# 0,2,SZR ;SKIP IF STILL CORRECT
28 01332 063077 HALT ;LOAD INDIRECT HAS SOMEHOW
29 ; ; MODIFIED THE ADDR. THROUGH
30 ; ; WHICH IT LOADED INDIRECTLY
31 01333 132414 SUB# 1,2,SZR ;SKIP IF VALUE LOADED CORRECTLY
32 01334 063077 HALT ;LOAD INDIRECT DID NOT LOAD
33 ; ; CORRECTLY. AC2 CONTAINS THE
34 ; ; BUFFER ADDR. WHICH IS ALSO THE
35 ; ; EXPECTED VALUE. AC1 HAS LOADED
36 ; ; VALUE
37 01335 151400 INC 2,2 ;INCREMENT BUFFER ADDR.
38 01336 156414 SUB# 2,3,SZR ;SKIP IF END OF BUFFER
39 01337 000760 JMP INDR1 ;CONTINUE TEST
40 01340 000411 JMP DBLIN ;JUMP OVER CONSTANTS
41 01341 001342 IND1: IND2
42 01342 000000 IND2: 0
43 01343 001342 IND3: IND2
44 01344 101345 IND4: @IND5
45 01345 001347 IND5: IND7
46 01346 101347 IND6: @IND7
47 01347 000000 IND7: 0
48 01350 125252 CALT1: 125252
49

```

```

01 ;
02
03 ; TEST DOUBLE INDIRECTS AND CHAINED INDIRECTS
04 ; USING THE LDA AND STA INSTRUCTIONS.
05 01351 102620 DBLIN: SUBZR 0,0 ;(AC0) = 100000
06 01352 024042 LDA 1,BUFND ;BUFFER END ADDR.
07 01353 030041 LDA 2,BUF ;BUFFER START ADDR.
08 01354 052770 DBL1: STA 2,IND4 ;SHOULD END UP IN IND7
09 01355 034772 LDA 3,IND7 ;INVESTIGATE IND7
10 01356 156414 SUB# 2,3,SZR ;SKIP IF STORED CORRECTLY
11 01357 063077 HALT ;STORE INDIRECT INDIRECT
12 ; FAILED TO STORE PROPER
13 ; VALUE
14 01360 155400 INC 2,3 ;(AC3) = BUFFER ADDR. +1
15 01361 117000 ADD 0,3 ;FORM INDIRECT ADDR. + 1
16 01362 056764 STA 3,IND6 ;STORE INTO BUFFER
17 01363 151400 INC 2,2 ;INCREMENT BUFFER ADDR.
18 01364 132414 SUB# 1,2,SZR ;SKIP IF END OF BUFFER
19 01365 000767 JMP DBL1 ;CONTINUE STORING
20 01366 035377 LDA 3,-1,2 ;STRIP INDIRECT BIT OFF ADDR. IN
21 01367 116400 SUB 0,3 ; NEXT TO THE LAST ADDR. IN
22 01370 055377 STA 3,-1,2 ; BUFFER
23 01371 034757 LDA 3,CALT1 ;STORE UNIQUE BIT PATTERN IN
24 01372 055000 STA 3,0,2 ; LAST LOCATION OF BUFFER
25 01373 030041 LDA 2,BUF ;BUFFER START ADDR.
26 01374 035000 LDA 3,0,2 ;GET CONTENTS OF ADDR.
27 01375 151400 INC 2,2 ;INCREMENT BUFFER ADDR.
28 01376 175120 MOVZL 3,3
29 01377 175220 MOVZR 3,3 ;STRIP OFF INDIRECT BIT
30 01400 156414 SUB# 2,3,SZR ;SKIP IF CORRECT VALUE STORED
31 01401 063077 HALT ;EITHER LOAD OR STORE DOUBLE
32 ; INDIRECT FAILED. PROBABLY LOAD
33 ; SINCE STORE WORKED EARLIER
34 01402 132414 SUB# 1,2,SZR ;SKIP IF END OF BUFFER
35 01403 000771 JMP .-7 ;CONTINUE CHECKING
36 01404 030041 LDA 2,BUF ;BUFFER START ADDR.
37 01405 113000 ADD 0,2 ;ADD INDIRECT BIT
38 01406 050741 STA 2,IND7
39 01407 036737 LDA 3,IND6 ;SHOULD CHAIN THROUGH INDIRECTS
40 01410 030740 LDA 2,CALT1 ;LOAD EXPECTED VALUE
41 01411 156414 SUB# 2,3,SZR ;SKIP IF CORRECT LOAD
42 01412 063077 HALT ;LOAD INDIRECT FAILED WHEN
43 ;EFFECTIVE INDIRECT ADDRESSES
44 ;WERE CHAINED THROUGH THE BUFFER
45

```



```

01 ;
02
03 ; TEST CELLS 20-37 FOR CORRECT AUTO-INCREMMENTING
04 ; USING THE LDA AND STA INSTRUCTIONS.
05 01413 024042 AUTIN: LDA 1, BUFND ; BUFFER END ADDR.
06 01414 030462 LDA 2, C20 ; FIRST CELL OF AUTO INCREMENT
07 01415 034041 LDA 3, BUF ; BUFFER START ADDR.
08 01416 055000 STA 3, 0, 2 ; INIT. AUTO INCR. CELL WITH ADDR.
09 01417 175400 INC 3, 3 ; START WITH BUF+ADDR.+1 = ADDR.+1
10 01420 057000 STA 3, 0, 2 ; STORE IN BUFFER VIA AUTO INCREMENT
11 01421 175400 INC 3, 3 ; INCREMENT VALUE TO BE STORED
12 01422 136414 SUB# 1, 3, SZR ; SKIP IF END OF BUFFER
13 01423 000775 JMP .-3 ; CONTINUE STORING
14 01424 034041 LDA 3, BUF ; BUFFER START ADDR.
15 01425 175400 INC 3, 3 ; INCREMENT BUFFER ADDR.
16 01426 021400 LDA 0, 0, 3
17 01427 116414 SUB# 0, 3, SZR ; SKIP IF CONTENTS EQUAL. ADDR., OK
18 01430 063077 HALT ; AUTO INCR. FAILED. AC2 CONTAINS
19 ; THE CELL WHICH SHOULD HAVE BEEN
20 ; AUTO INCREMENTING. AC3 HAS EXPECTED
21 ; VALUE AND AC0 HAS LOADED VALUE
22 01431 175400 INC 3, 3 ; INCREMENT BUFFER ADDR.
23 01432 136414 SUB# 1, 3, SZR ; SKIP IF END OF BUFFER
24 01433 000773 JMP .-5 ; CONTINUE CHECKING BUFFER
25 01434 151400 INC 2, 2 ; INCR. TO NEXT AUTO INCREMENT LOC.
26 01435 020442 LDA 0, C30 ; FIRST CELL OF AUTO DECREMENT
27 01436 112414 SUB# 0, 2, SZR ; SKIP IF END OF AUTO INCREMENT
28 01437 000756 JMP AUTIN+2 ; CHECK NEXT AUTO INCREMENT CELL
29 01440 024042 LDA 1, BUFND ; BUFFER END ADDR.
30 01441 030041 LDA 2, BUF ; BUFFER START ADDR.
31 01442 102400 SUB 0, 0
32 01443 041400 STA 0, 0, 3 ; ZERO FILL BUFFER
33 01444 151400 INC 2, 2 ; INCREMENT BUFFER ADDR.
34 01445 132414 SUB# 1, 2, SZR ; SKIP IF END OF BUFFER
35 01446 000775 JMP .-3 ; CONTINUE ZERO FILLING
36 01447 041400 STA 0, 0, 3 ; ZERO FILL LAST CELL IN BUFFER
37

```

```

01 ;
02
03 ; TEST CELLS 30-37 FOR CORRECT AUTO-DECREMENTING
04 ; USING THE LDA AND STA INSTRUCTIONS.
05 01450 030427 AUTDC: LDA 2,C30 ;FIRST CELL OF AUTO DECR.
06 01451 024041 LDA 1,BUF ;BUFFER START ADDR.
07 01452 034042 LDA 3,BUFND ;BUFFER END ADDR.
08 01453 055000 STA 3,0,2 ;INIT. AUTO DECR. CELL WITH ADDR.
09 01454 102520 SUBZL 0,0 ;(AC0) = 1
10 01455 116400 SUB 0,3 ;DECREMENT VALUE TO BE STORED
11 01456 057000 STA 3,0,2 ;STORE IN BUFFER VIA AUTO DECR.
12 01457 136414 SUB# 1,3,SZR ;SKIP IF BEGINNING OF BUFFER
13 01460 000775 JMP .-3
14 01461 024042 LDA 1,BUFND ;BUFFER END ADDR.
15 01462 034041 LDA 3,BUF ;BUFFER START ADDR.
16 01463 021400 LDA 0,0,3 ;VALUE SHOULD EQUAL BUFFER ADDR.
17 01464 116414 SUB# 0,3,SZR ;SKIP EQUAL, OK
18 01465 063077 HALT ;ERROR IN AUTO DECREMENT. AC3
19 ; CONTAINS EXPECTED VALUE, AC0
20 ; LOADED VALUE. AC2 CONTAINS THE
21 ; ADDR. THAT WAS LOADED THROUGH
22 01466 175400 INC 3,3 ;INCREMENT BUFFER ADDR.
23 01467 136414 SUB# 1,3,SZR ;SKIP IF END OF BUFFER
24 01470 000773 JMP .-5 ;CONTINUE CHECKING BUFFER
25 01471 151400 INC 2,2 ;INCREMENT TO NEXT AUTO DECR. CELL
26 01472 020406 LDA 0,C40 ;ONE PAST AUTO DECREMENT
27 01473 112414 SUB# 0,2,SZR ;SKIP IF NOT AUTO DECREMENT
28 01474 000755 JMP AUTDC+1 ;CONTINUE TEST
29 01475 000404 JMP .+4 ;JUMP OVER CONSTANTS
30 01476 000020 C20: 20
31 01477 000030 C30: 30
32 01500 000040 C40: 40
33 01501 000100 JMP PERIP+2
34 000200 BUFF: .BLK 200
35 01702 000000 BFEND: 0
36

```

↑ 0027 .MAIN
 01 000076
 0028 .MAIN

.END PERIP

ALU	000400	6/07	11/10						
ARLOG	000044	6/07	7/32						
AUTDC	001450	26/05	26/28						
AUTIN	001413	25/05	25/28						
BFEND	001702	6/05	6/06	26/35					
BITER	001201	20/14	20/25	20/35	20/43				
BUF	000041	6/04	18/14	18/34	19/10	19/30	19/49	19/55	21/08
		21/14	21/31	22/07	23/06	23/12	24/07	24/25	24/36
		25/07	25/14	25/30	26/06	26/15			
BUFF	001502	6/04	6/06	26/34					
BUFND	000042	6/05	19/50	20/49	21/07	23/07	24/06	25/05	25/29
		26/07	26/14						
C10	000052	6/13	10/51						
C15	000053	6/14	10/52						
C17	000054	6/15	10/49						
C20	001476	25/06	26/30						
C200	001255	21/54	22/08						
C3	000051	6/12	7/33						
C30	001477	25/26	26/05	26/31					
C37	000055	6/16	7/06						
C377	001257	21/30	21/56	22/06	22/29				
C40	001500	26/26	26/32						
C400	000056	6/17	8/03						
CALT1	001350	23/48	24/23	24/40					
DATA	000045	6/08	8/05	9/09	9/12	9/15	10/08	10/28	10/29
DBL1	001354	24/08	24/19						
DBLIN	001351	23/40	24/05						
DIF	000043	6/06	18/10	19/06					
DIS1	001251	21/32	21/50						
DIS2	001252	21/34	21/51						
DIS3	001253	21/52	22/10						
DIS4	001254	21/53	22/12						
ENTR1	000065	6/24	7/12	7/36	7/40	9/22			
ENTR2	000066	6/25	7/17	8/11	8/15	10/15			
ENTR3	000067	6/26	7/22	8/20	8/26	9/25			
ENTR4	000070	6/27	7/27	8/28	8/33	10/18			
HSP1	000147	8/02	8/19	8/27	8/34				
HSPON	000140	7/09	7/36						
HSRON	000160	7/14	8/11						
IND1	001341	23/13	23/20	23/41					
IND2	001342	23/14	23/25	23/26	23/41	23/42	23/43		
IND3	001343	23/19	23/43						
IND4	001344	23/44	24/08						
IND5	001345	23/44	23/45						
IND6	001346	23/46	24/16	24/39					
IND7	001347	23/45	23/46	23/47	24/09	24/38			
INDIR	001310	23/06							
INDR1	001317	23/13	23/39						
INP	000325	10/07	10/22						
INP1	000331	10/04	10/26						
INPRC	000351	6/03	10/42						
INPUT	000301	6/28	6/30	10/02					
IPROC	000040	6/03	7/02						
IVECT	000071	6/28	10/56						
JMPND	001260	21/46	21/57						
LEAD	000046	6/09	8/04	9/02	9/08	10/22			
LF	001256	21/55	22/25						
LFHAF	000677	16/03	16/07	16/08	16/09	16/22			
NDIS1	001270	21/53	22/11	22/13	22/16	22/24	22/28	22/31	

0029 ,MAIN

NDISP	001261	21/49	22/06						
PAT1	001132	20/02	20/51						
PATER	001123	19/49							
PC	001025	18/10							
PC1	001062	18/13	18/23	18/45					
PC2	001063	18/28	18/46	19/24					
PC3	001064	18/11	18/39	18/47	18/48	19/08	19/35		
PC4	001065	18/33	18/48	19/29					
PC5	001066	18/49	19/09	19/19					
PCA	001067	18/44	19/06						
PD1S1	001235	21/33	21/35	21/36	21/37	21/45	21/48	21/51	
PDISP	001230	21/30							
PERIP	000076	7/02	26/33	27/01					
PFLAG	000057	6/18	8/02	9/19	9/32	10/12	10/33		
POUT	000253	9/06	9/13	9/30	9/32				
PUN1	000223	9/04	9/08						
PUN2	000232	9/11	9/15						
PUN3	000250	9/17	9/29						
PUNCH	000215	6/29	6/31	9/02					
QUIT	000370	8/39	9/07	9/14	9/28	9/31	10/21	10/23	10/32
		11/02							
READ	000340	10/02	10/33						
RET1	001226	21/09	21/24						
RET2	001227	21/15	21/25						
RETA	001220	21/17	21/25						
RETAD	001206	21/07							
RTCLK	000210	6/32	8/35						
RTCON	000135	7/29	7/33						
SAVC	000064	6/23	10/47	11/02					
SREG0	000060	6/19	10/42	11/04					
SREG1	000061	6/20	10/43	11/05					
SREG2	000062	6/21	10/44	11/06					
SREG3	000063	6/22	10/45	11/07					
STRT	000050	6/11	8/09	8/18	10/05	10/27			
TRAIL	000047	6/10	8/07	9/16					
TTION	000201	7/24	8/28						
TTOON	000171	7/19	8/20						

4

1



