

National Semiconductor

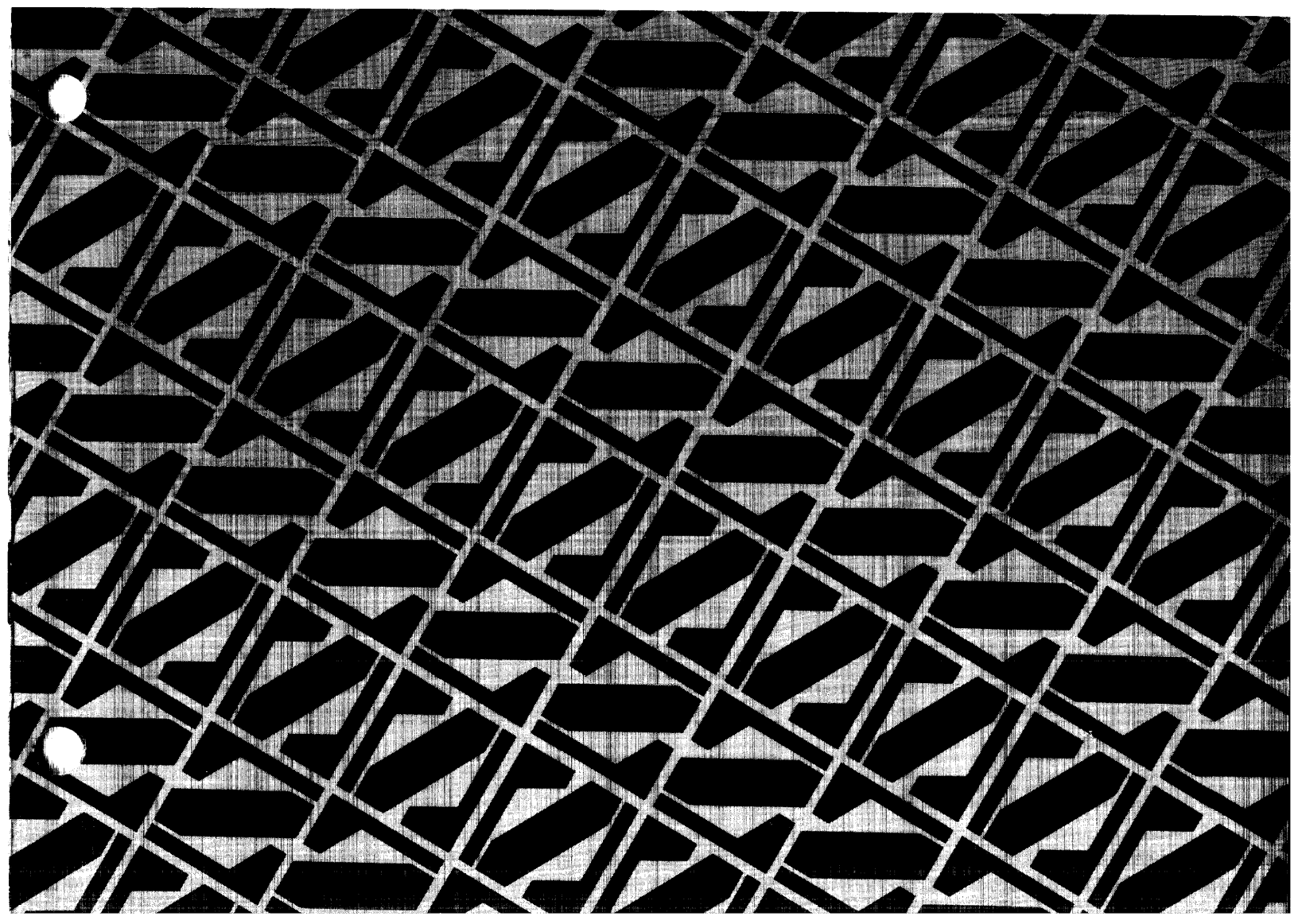
Order No. IMP-16S/501AA

Pub. No. 4210006A

IMP-16C

ROMDI and ROMDIX

Listings



Integrated MicroProcessor-16C

IMP-16C

ROMDI AND ROMDIX

LISTINGS

April 1974

© National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051

CONTENTS

	<u>Page</u>
ROMDI	
INTRODUCTION	1
ROMDI1 LISTING	2
ROMDI2 LISTING	9
ROMDIX	
INTRODUCTION	15
ROMDIX LISTING	16

ROMDI LISTINGS

INTRODUCTION

The listings of the Read Only Memory Diagnostic Program (ROMDI) is presented on the following pages as ROMDI1 and ROMDI2.

ROMDI is a compact CPU diagnostic that is written in the IMP-16 Assembler language, whose revision level appears at the top of the first page.

ROMDI tests the Basic Instruction Set of the Central Processing Unit (CPU) on the IMP-16C card on a "go/no-go" basis. If the test indicates a failure, the particular malfunctioning device is not designated. To test individual devices in the IMP-16C CPU, it must be assured that all other CPU devices and the other components on the IMP-16C are functioning.

There are four RALUs (Register, Arithmetic, and Logic Units) and one CROM (Control Read-Only Memory) in the IMP-16C CPU. For example, a particular RALU may be tested by first assuring that the CROM and three of the other RALUs on the IMP-16C are functioning. In this case, if the test passes, the RALU under test is functioning; but if the test fails, the RALU under test is malfunctioning.

Instructions for performing ROMDI are given in the listing.

ROMD11

REVISION-G 01/02/74
 ROMD11 000137B 03/22/74 PART 1

```

1 0000      ; .TITLE ROMD11,'000137B 03/22/74 PART 1'
2 0000      ;
3 0000      ; MAIN PROGRAM IN ROM, X'FE00-X'FFFF, ENTRY POINT X'FFF
4 0000      ;
5 0000      ; ROMD1 IS A GO-NOGO CPU DIAGNOSTIC DESIGNED FOR THE IMP-16C.
6 0000      ; THE PROGRAM FITS INTO 4 8X256-BIT PROMS IN THE FOLLOWING
7 0000      ; MANNER;
8 0000      ;
9 0000      ;
10 0000     ;
11 0000     ; ADDRESSES      BITS      ROM      DIAGRAM
12 0000     ; ROMD11 RIGHT FE00-FE0F 0-7      ATZ      4F
13 0000     ; ROMD11 LEFT  FE00-FE0F 8-15     AUA      4D
14 0000     ; ROMD12 RIGHT FF00-FFFF 0-7      AUB      4E
15 0000     ; ROMD12 LEFT  FF00-FFFF 8-15     AUC      4C
16 0000     ;
17 0000     ; ENTERING THE PROGRAM AT ADDRESS FFFE TRANSFERS CONTROL TO A
18 0000     ; PANEL ROUTINE WHICH PERFORMS THE FOLLOWING FUNCTIONS IN
19 0000     ; RESPONSE TO THE INDICATED SIGNALS:
20 0000     ;
21 0000     ; SIGNAL NAME      PIN NO.      FUNCTION
22 0000     ;
23 0000     ; JC12             129      (LOAD ADDRESS) LOADS THE CONTENTS OF
24 0000     ;                  SW00,SW01,...,SW15 SIGNALS INTO AC2
25 0000     ;                  REGISTER AND DISPLAYS THE CONTENTS
26 0000     ;                  OF THE MEMORY LOCATION ADDRESSED BY
27 0000     ;                  AC2 USING THE SIGNALS B0000,B0001,
28 0000     ;                  ...,B0015.
29 0000     ;
30 0000     ; JC13             119      (LOAD DATA) LOADS THE CONTENTS OF
31 0000     ;                  SW00,SW01,...,SW15 SIGNALS INTO AC0
32 0000     ;                  REGISTER, STORES AC0 INTO MEMORY
33 0000     ;                  LOCATION ADDRESSED BY AC2, & DISPLAYS
34 0000     ;                  AC0 USING THE SIGNALS B0000,B0001,
35 0000     ;                  ...,B0015.
36 0000     ;
37 0000     ; START           123      (EXECUTE) TRANSFERS CONTROL TO THE
38 0000     ;                  MEMORY LOCATION INDICATED BY THE
39 0000     ;                  SIGNALS SW00,SW01,...,SW15.
40 0000     ;
41 0000     ;
42 0000     ; UPON ENTRY TO THE PANEL ROUTINE, ACCUMULATORS 0,1,2, AND 3
43 0000     ; ARE SAVED IN MEMORY LOCATIONS 4,5,6 AND 7, RESPECTIVELY.
44 0000     ; THE CONTENTS OF AN ACCUMULATOR MAY BE SET BY ALTERING ITS
45 0000     ; CORRESPONDING MEMORY LOCATION.
46 0000     ;
47 0000     ; IF SIGNAL JC15 (JUMP CONDITION 15) IS HIGH WHEN 'EXECUTE',
48 0000     ; I. E. THE START SIGNAL, IS ON, THE PROGRAM WILL BE CONTINUOUSLY
49 0000     ; REEXECUTED UNTIL EITHER THE JUMP CONDITION GOES LOW OR AN
50 0000     ; ERROR CONDITION IS DETECTED.
51 0000     ;
52 0000     ; IF THE DIAGNOSTIC EXECUTES SUCCESSFULLY, THE PROGRAM SETS
53 0000     ; FLAG 8 (SIGNAL F8) AND RETURNS CONTROL TO THE PANEL ROUTINE.
54 0000     ; IF THE PROGRAM DETECTS AN ERROR CONDITION, IT SETS FLAG 15
55 0000     ; (SIGNAL F15) AND HALTS (SIGNAL HLT* (FLAG)).
56 0000     ;
57 0000     ; THE DIAGNOSTIC ENTRY POINT IS X'FE00.
58 0000     ;
59 0000     ; NOTE: SEE APPENDIX E IN IMP-16C APPLICATION MANUAL (PUB. NO.
60 0000     ; 4200021B) FOR THE LIST OF PIN CONNECTIONS AND SIGNALS
61 0000     ; ON IMP-16C CARD.
62 0000     ;

```

```

63 0000          .PAGE
64 0000          ; EQUATES
65 0000          ;
66 0000 0000 A R0 = 0
67 0000 0001 A R1 = 1
68 0000 0002 A R2 = 2
69 0000 0003 A R3 = 3
70 0000          ;
71 0000          ; ...BRANCH CONDITIONS
72 0000 0008 A STKFL = 8
73 0000 0001 A ZRO = 1
74 0000 0002 A PZRO = 2
75 0000 0003 A ODD = 3
76 0000 0004 A BIT1 = 4
77 0000 0005 A NZERO = 5 ; NOT EQUAL ZERO
78 0000 0009 A INTEN = 9
79 0000 000A A CYOV = 10
80 0000 000B A NZRO = 11 ; NEGATIVE OR ZERO
81 0000          ;
82 0000          ; ...FLAG ADDRESSES
83 0000 0001 A IENFL = 1
84 0000 0002 A SELX = 2
85 0000          ;
86 0000          ; LINK ADDRESSES FOR PART 2
87 0000          ;
88 0000 FE00 A ROMAD = X'FE00
89 0000 FFFE A ENTRY = ROMAD+X'1FE
90 0000 FF07 A D0 = ROMAD+X'107
91 0000 FF08 A D1 = D0+1
92 0000 FF09 A D2 = D1+1
93 0000 FF0A A D15 = D2+1
94 0000 FF0B A H5050 = D15+1
95 0000 FF0D A H5A5A = H5050+2
96 0000 FFOE A H7FFF = H5A5A+1
97 0000 FFOF A H8000 = H7FFF+1
98 0000 FF10 A HA0A0 = H8000+1
99 0000 FF11 A HA5A5 = HA0A0+1
100 0000 FF13 A HF0F0 = HA5A5+2
101 0000 FF15 A M1 = HF0F0+2
102 0000 FF8C A ERROR = ROMAD+X'18C
103 0000          ;
104 0000          ;
105 0000 000F A ILOC = 15 ; TEST LOC FOR INDIRECT TEST

```

```

106 0000          .PAGE
107 0000          .ASECT
108 0000 FE00 A    .=RCMAD
109 FE00          ;
110 FE00          ; TEST REGISTER 0, SKG, SKNE, MEMORY TRANSFERS
111 FE00          ;
112 FE00 815C A START: LD R0,MIN1
113 FE01 F15B A SKNE R0,MIN1
114 FE02 2101 A JMP .+2
115 FE03 217E A JMP ERR1
116 FE04 F159 A SKNE R0,X7FFF
117 FE05 217C A JMP ERR1
118 FE06 A00A A ST R0,10 ;INITIALIZE FOR ISZ, DSZ TEST
119 FE07 E157 A SKG R0,ZERO
120 FE08 2101 A JMP .+2
121 FE09 2178 A JMP ERR1
122 FE0A 4801 A AISZ R0,1
123 FE0B 215F A JMP ERR
124 FE0C 1101 A BCC ZRO, .+2
125 FE0D 2174 A JMP ERR1

```

```

126 FE0E 5000 A      CAI      RO,0          ;ONE'S COMPLEMENT
127 FE0F F14D A      SKNE     RO,MIN1
128 FE10 2101 A      JMP      .+2
129 FE11 2170 A      JMP      ERR1
130 FE12 5000 A      CAI      RO,0          ; ONE'S COMPLEMENT
131 FE13 F14B A      SKNE     RO,ZERO
132 FE14 2101 A      JMP      .+2
133 FE15 216C A      JMP      ERR1
134 FE16 8149 A      LD       RO,ONE
135 FE17 5001 A      CAI      RO,1          ;TWO'S COMPLEMENT
136 FE18 F144 A      SKNE     RO,MIN1
137 FE19 2101 A      JMP      .+2
138 FE1A 2167 A      JMP      ERR1
139 FE1B 5001 A      CAI      RO,1          ;TWO'S COMPLEMENT
140 FE1C 48FF A      AISZ     RO,-1
141 FE1D 2164 A      JMP      ERR1

142 FE1E             .PAGE
143 FE1E             ;
144 FE1E             ;
145 FE1E             ;
146 FE1E 8542 A      LD       R1,XFACE     ;INITIALIZE ILOC
147 FE1F A40F A      ST       R1,ILOC
148 FE20 9146 A      LD       RO,@INDPNT
149 FE21 F30F A      SKNE     RO,ILOC
150 FE22 2101 A      JMP      .+2
151 FE23 215E A      JMP      ERR1
152 FE24 4E01 A      LI       R2,1
153 FE25 C940 A      ADD      R2,LINDPT   ;ADD ADDRESS OF INDPNT
154 FE26 813B A      LD       RO,XFOFJ
155 FE27 B200 A      ST       RO,@(R2)
156 FE28 5001 A      CAI      RO,1
157 FE29 C011 A      ADD      RO,ILOC+2
158 FE2A 1557 A      BOC      NZERO,ERR1

159 FE2B             .PAGE
160 FE2B             ;
161 FE2B             ;
162 FE2B             ;
163 FE2B 0980 A      PFLG     IENFL        ; DISABLE INTERRUPTS
164 FE2C 1955 A      BOC      INTEN,ERR1
165 FE2D 0900 A      SFLG     IENFL        ; ENABLE INTERRUPTS
166 FE2E 1901 A      BOC      INTEN,+.2
167 FE2F 2152 A      JMP      ERR1
168 FE30 8138 A      LD       RO,IEP      ; INITIALIZE INTERRUPT ENTRY
169 FE31 A001 A      ST       RO,1
170 FE32 8137 A      LD       RO,IEP+1
171 FE33 A002 A      ST       RO,2
172 FE34 4EF0 A      LI       R2,-16     ; EMPTY STACK
173 FE35 4400 A      PULL     RO
174 FE36 4A01 A      AISZ     R2,1
175 FE37 21FD A      JMP      .-2
176 FE38 1849 A      BOC      STKFL,ERR1
177 FE39 4EF9 A      LI       R2,-7      ; PUSH X'A5A5 AND X'5A5A ONTO
178 FE3A 8128 A      LD       RO,XA5A5
179 FE3B 8528 A      LC       R1,X5A5A
180 FE3C 4000 A      PUSH     RO
181 FE3D 4100 A      PUSH     R1
182 FE3E 4A01 A      AISZ     R2,1
183 FE3F 21FC A      JMP      .-3
184 FE40 4000 A      PUSH     RO
185 FE41 4100 A      PUSH     R1          ; SHOULD FILL STACK, CAUSE IN
186 FE42 213F A      JMP      ERR1
187 FE43 1801 A      INENTY: BOC      STKFL,+.2   ; TEST JUMP CONDITION

```

ROMDI1

```

188 FE44 213D A      JMP      ERR1
189 FE45 193C A      BOC      INTEN,ERR1
190 FE46 5400 A      XCHRS   RO
191 FE47 4100 A      PUSH    R1
192 FE48 4500 A      PULL    R1          ; INTERRUPTS SHOULD BE DISABL
193 FE49 1838 A      BOC      STKFL,ERR1
194 FE4A 8120 A      LD       RO,ERR    ;IF INTERRUPT, TRANSFER TO ER
195 FE4B A002 A      ST      RO,2
196 FE4C 4100 A      PUSH    R1          ;REFILL STACK; ERROR IF INTER
197 FE4D 1801 A      BOC      STKFL,++2
198 FE4E 2133 A      JMP
199 FE4F 4EF8 A      LI      R2,-8      ; TEST CONTENTS OF STACK
200 FE50 4400 A PLOOP: PULL    RO
201 FE51 F112 A      SKNE    RO,XA5A5
202 FE52 2101 A      JMP     .+2
203 FE53 212E A      JMP     ERR1
204 FE54 4400 A      PULL    RO
205 FE55 F10D A      SKNE    RO,XA5A5
206 FE56 2101 A      JMP     .+2
207 FE57 212A A      JMP     ERR1
208 FE58 4A01 A      AISZ   R2,1
209 FE59 21F6 A      JMP     PLOOP
210 FE5A 4400 A      PULL    RO          ; STACK SHOULD BE EMPTY
211 FE5B 1526 A      BOC      NZERO,ERR1
212 FE5C 210F A      JMP     SBRTST
213 FE5D FFFF A MINI:  .WORD   -1
214 FE5E 7FFF A X7FFF: .WORD   X'7FFF
215 FE5F 0000 A ZERO:  .WORD   0
216 FE60 0001 A ONE:   .WORD   1
217 FE61 FACE A XFACE: .WORD   X'FACE
218 FE62 F0F0 A XF0F0: .WORD   X'F0F0
219 FE63 A5A5 A XA5A5: .WORD   X'A5A5
220 FE64 5A5A A X5A5A: .WORD   X'5A5A
221 FE65 000F A I15:  .WORD   15
222 FE66 FE67 A LINDPT: .WORD   INDPNT
223 FE67 000F A INDPT: .WORD   ILOC,ILOC+2
FE68 0011 A
224 FE69 2500 A IEP:   JMP     @.+1      ; TO INITIALIZE LOCS 1 AND 2
225 FE6A FE43 A      .WORD   INENTY   ; MUST BE IN LOC IEP+1
226 FE6B FF8C A ERR:   .WORD   ERROR

227 FE6C      ;      .PAGE
228 FE6C      ;
229 FE6C      ;      TEST SUBROUTINE LINKAGES(JSR, JSR@, RTS, RTI)
230 FE6C      ;
231 FE6C 4F00 A SBRTST: LI      R3,0
232 FE6D 4300 A      PUSH    R3
233 FE6E 290D A      JSR     SUBR
234 FE6F 4B0A A RETN:  AISZ   R3,10
235 FE70 FDF4 A      SKNE    R3,I15
236 FE71 2101 A      JMP     .+2
237 FE72 210F A      JMP     ERR1
238 FE73 4F00 A      LI      R3,0
239 FE74 2D12 A      JSR     @SBRAD
240 FE75 4B0A A      AISZ   R3,10
241 FE76 FDEE A      SKNE    R3,I15
242 FE77 2101 A      JMP     .+2
243 FE78 2109 A      JMP     ERR1
244 FE79 1901 A      BOC      INTEN,++2  ; INTERRUPTS SHOULD BE ENABLE
245 FE7A 2107 A      JMP     ERR1
246 FE7B 210D A      JMP     TMWRT

```



```

247 FE7C          .SPACE 2
248 FE7C 4500 A SUBR: PULL  R1          ; JSR TEST
249 FE7D F50A A     SKNE  R1,RTNAD     ; CHECK THAT RETURN ADDRESS S
250 FE7E 4B05 A     AISZ  R3,5
251 FE7F 8508 A     LD    R1,RTNAD     ; ENSURE CORRECT RETURN ADDRE
252 FE80 4100 A     PUSH  R1
253 FE81 0200 A     RTS

254 FE82          .SPACE 2
255 FE82 2500 A ERR1: JMP    @.+1
256 FE83 FF8C A     .WORD  ERROR

257 FE84          .SPACE 2
258 FE84 4B05 A SUBR1: AISZ  R3,5      ; FOR JSR*
259 FE85 0100 A     RTI
260 FE86 21FB A     JMP    ERR1

261 FE87          .SPACE 2
262 FE87 FE84 A SBRAD: .WORD  SUBR1
263 FE88 FE6F A RTNAD: .WORD  RETN

264 FE89          .PAGE
265 FE89          ;
266 FE89          ; TEST MEMORY WRITE AND INDEXING
267 FE89          ;
268 FE89 4E00 A TMWRT: LI    R2,0      ;WRITE MEMORY FROM LOC 32 THR
269 FE8A 4C01 A     LI    R0,1      ;WITH INTEGERS 1 THRU 16
270 FE8B A220 A MLCOP: ST    R0,32(R2)
271 FE8C 4801 A     AISZ  R0,1
272 FE8D 4A01 A     AISZ  R2,1
273 FE8E E97B A     SKG   R2,D15
274 FE8F 21FB A     JMP   MLOOP
275 FE90 4FF0 A     LI    R3,-16     ;COMPARE LOCS 32 THRU 47 WITH
276 FE91 4C01 A     LI    R0,1      ;USING NEGATIVE INDEXING
277 FE92 F330 A CLOOP: SKNE  R0,48(R3)
278 FE93 2101 A     JMP   .+2
279 FE94 21ED A     JMP   ERR1
280 FE95 4801 A     AISZ  R0,1
281 FE96 4B01 A     AISZ  R3,1
282 FE97 21FA A     JMP   CLOOP

283 FE98          .PAGE
284 FE98          ;
285 FE98          ; TEST REGISTER-TO-REGISTER PLUS AND, XOR FUNCTIONS
286 FE98          ;
287 FE98 817C A     LD    R0,M1      ;R0=-1
288 FE99 856E A     LD    R1,D1      ;R1=1
289 FE9A 3400 A     RADD  R1,R0      ;R0=0
290 FE9B 15E6 A     BOC   NZERO,ERR1
291 FE9C 8174 A     LD    R0,HA5A5     ;R0=X'A5A5
292 FE9D 3281 A     RCPY  R0,R2      ;R2=X'A5A5
293 FE9E 3181 A     RCPY  R0,R1      ;R1=X'A5A5
294 FE9F 5200 A     CAI   R2,0      ;R2=X'5A5A
295 FEA0 3882 A     RXOR  R2,R0     ;R0=-1
296 FEA1 4801 A     AISZ  R0,1      ;R0=0
297 FEA2 21DF A     JMP   ERR1
298 FEA3 3983 A     RAND  R2,R1      ;R1=0
299 FEA4 4900 A     AISZ  R1,0
300 FEA5 21DC A     JMP   ERR1
301 FEA6 4FFF A     LI    R3,-1      ;R3=-1

```

ROMDII

```

302 FEA7 3C80 A      RXCH   R3,R0          ;R3=0,R0=-1
303 FEA8 3282 A      RXOR   R0,R2          ;R2=X'A5A5
304 FEA9 4801 A      AISZ   R0,1          ;R0=0
305 FEAA 21D7 A      JMP    ERR1
306 FEAB 4800 A      AISZ   R3,0
307 FEAC 21D5 A      JMP    ERR1
308 FEAD F963 A      SKNE   R2,HA5A5
309 FEAE 2101 A      JMP    .+2
310 FEAF 21D2 A      JMP    ERR1
311 FEB0 8D62 A      LD     R3,HFOFO      ;R3=X'FOFO
312 FEB1 4DFF A      LI     R1,-1         ;R1=-1
313 FEB2 3D83 A      RAND   R3,R1         ;R1=X'FOFO
314 FEB3 3782 A      RXOR   R1,R3         ;R3=0
315 FEB4 3881 A      RCPY   R2,R0         ;R0=X'A5A5
316 FEB5 3380 A      RXCH   R0,R3         ;R3=X'A5A5,R0=0
317 FEB6 15CB A      BOC    NZERO,ERR1
318 FEB7 3680 A      RXCH   R1,R2         ;R2=X'FOFO,R1=X'A5A5
319 FEB8 3883 A      RAND   R2,R3         ;R3=X'A0A0
320 FEB9 FD56 A      SKNE   R3,HA0A0
321 FEBA 2101 A      JMP    .+2
322 FEBB 21C6 A      JMP    ERR1
323 FEBC           ;
324 FEBC           ;      TEST LOGICAL-OR FUNCTION
325 FEBC           ;
326 FEBC 8150 A      LD     R0,H5A5A      ;TEST 1V0 AND 0V1
327 FEBD 6953 A      OR     R0,HA5A5
328 FEBE 4801 A      AISZ   R0,1
329 FEBF 21C2 A      JMP    ERR1
330 FEC0 854A A      LD     R1,H5050      ;TEST 0V0 AND 1V1
331 FEC1 6D51 A      OR     R1,HFOFO
332 FEC2 F550 A      SKNE   R1,HFOFO
333 FEC3 2101 A      JMP    .+2
334 FEC4 21BD A      JMP    ERR1

335 FEC5           .PAGE
336 FEC5           ;
337 FEC5           ;      TEST ARITHMETIC FUNCTIONS
338 FEC5           ;
339 FEC5 4C01 A      LI     R0,1          ;1+(-1)
340 FEC6 C14E A      ADD    R0,M1
341 FEC7 15BA A      BOC    NZERO,ERR1
342 FEC8 0A00 A      SFLG   SELX         ; TEST FOR OVERFLOW
343 FEC9 1AB8 A      BOC    CYOV,ERR1
344 FECA 0A80 A      PFLG   SELX         ; TEST FOR CARRY
345 FECB 1A01 A      BOC    CYOV,+.+2
346 FECC 21B5 A      JMP    ERR1
347 FECD 8141 A      LD     R0,H8000      ;X'8000-1
348 FECE D139 A      SUB    R0,D1
349 FECF F13E A      SKNE   R0,H7FFF
350 FED0 2101 A      JMP    .+2
351 FED1 21B0 A      JMP    ERR1
352 FED2 1A01 A      BOC    CYOV,+.+2      ; TEST FOR CARRY
353 FED3 21AE A      JMP    ERR1
354 FED4 0A00 A      SFLG   SELX         ; TEST FOR OVERFLOW
355 FED5 1A01 A      BOC    CYOV,+.+2
356 FED6 21AB A      JMP    ERR1
357 FED7 8136 A      LD     R0,H7FFF      ;X'7FFF+1
358 FED8 C12F A      ADD    R0,D1
359 FED9 F135 A      SKNE   R0,H8000
360 FEDA 2101 A      JMP    .+2
361 FEDB 21A6 A      JMP    ERR1
362 FEDC 1A01 A      BOC    CYOV,+.+2      ; TEST FOR OVERFLOW
363 FEDD 21A4 A      JMP    ERR1
364 FEDE 0A80 A      PFLG   SELX         ; TEST FOR CARRY
365 FEDF 1AA2 A      BOC    CYOV,ERR1

```

ROMDI1

```

366 FEE0 812D A      LD      R0,H7FFF      ; X'7FFF-X'8000
367 FEE1 D12D A      SUB     R0,H8000
368 FEE2 F132 A      SKNE   R0,M1
369 FEE3 2101 A      JMP     .+2
370 FEE4 219D A      JMP     ERR1
371 FEE5 1A9C A      BOC    CYOV,ERR1      ;ANOMALY - NO CARRY GENERATED
372 FEE6 0A00 A      SFLG   SELX
373 FEE7 1A01 A      BOC    CYOV,+.2      ;SHOULD HAVE OVERFLOW
374 FEE8 2199 A      JMP     ERR1

```

```

375 FEE9              .PAGE
376 FEE9              ;
377 FEE9              ; TEST SKIP IF GREATER
378 FEE9              ;
379 FEE9 8D1E A      LD      R3,D1          ;1>-1
380 FEEA ED2A A      SKG    R3,M1
381 FEEB 2196 A      JMP     ERR1
382 FEEC ED1A A      SKG    R3,D0          ;1>0
383 FEED 2194 A      JMP     ERR1
384 FEEE ED19 A      SKG    R3,D1          ;1>1
385 FEEF 2101 A      JMP     .+2
386 FEF0 2191 A      JMP     ERR1
387 FEF1 ED17 A      SKG    R3,D2          ;1>2
388 FEF2 2101 A      JMP     .+2
389 FEF3 218E A      JMP     ERR1
390 FEF4 8919 A      LD      R2,H7FFF
391 FEF5 E919 A      SKG    R2,H8000      ;X'7FFF>X'8000
392 FEF6 218B A      JMP     ERR1
393 FEF7 8917 A      LD      R2,H8000      ;X'8000>X'7FFF
394 FEF8 E915 A      SKG    R2,H7FFF
395 FEF9 2101 A      JMP     .+2
396 FEFA 2187 A      JMP     ERR1
397 FEFB 8519 A      LD      R1,M1
398 FEFC E50A A      SKG    R1,D0          ;-1>0
399 FEFD 2101 A      JMP     .+2
400 FEFE 2183 A      JMP     ERR1
401 FEFF 3081 A      NOP
402 FF00              .END

```

***** 0 ERRORS IN ASSEMBLY *****

```

BIT1  CLOOP  CYOV  DO    D1    D15   D2    ENTRY  ERR   ERR1
0004 A FE92 A 000A A FF07 A FF08 A FF0A A FF09 A FFFE A FE68 A FE82 A

ERROR  H5050  H5A5A  H7FFF  H8000  HA0A0  HA5A5  HFOF0  I15   IENFL
FF8C A FF0B A FF0D A FFOE A FFOF A FF10 A FF11 A FF13 A FE65 A 0001 A

IEP    ILOC   INDPNT INENTY INTEN  LINDPT M1     MIN1   MLOOP  NZERO
FE69 A 000F A FE67 A FE43 A 0009 A FE66 A FF15 A FE5D A FE88 A 0005 A

NZRO   ODD    ONE    PLOOP  PZRO   RJ     R1     R2     R3     RETN
000B A 0003 A FE60 A FE50 A 0002 A 0000 A 0001 A 0002 A 0003 A FE6F A

ROMAD  RTNAD  SBRAD  SBRTST SELX   START  STKFL  SUBR   SUBR1  TMWRT
FE00 A FE88 A FE87 A FE6C A 0002 A FE00 A 0008 A FE7C A FE84 A FE89 A

X5A5A  X7FFF  XA5A5  XF0F0  XFACE  ZERO   ZRO
FE64 A FE5E A FE63 A FE62 A FE61 A FE5F A 0001 A

```

6C2A 539B

ROMDI2

REVISION-G 01/02/74
 ROMDI2 000137B 03/22/74 PART 2

```

1 0000          .TITLE ROMDI2,'000137B 03/22/74 PART 2'
2 0000          ;
3 0000          ; MAIN PROGRAM IN ROM, X*FE00-X*FFFF, ENTRY POINT X*FFF
4 0000          ;
5 0000          ; EQUATES
6 0000          ;
7 0000 0000 A  RO  =      0
8 0000 0001 A  R1  =      1
9 0000 0002 A  R2  =      2
10 0000 0003 A  R3  =      3
11 0000          ;
12 0000          ; ...BRANCH CONDITIONS
13 0000 0008 A  STKFL =      8
14 0000 0001 A  ZRO  =      1
15 0000 0002 A  PZRO  =      2
16 0000 0003 A  ODD   =      3
17 0000 0004 A  BIT1  =      4
18 0000 0005 A  NZERO =      5          ; NOT EQUAL ZERO
19 0000 0009 A  INTEN =      9
20 0000 000A A  CYOV  =     10
21 0000 000B A  NZRO  =     11
22 0000          ;
23 0000          ; ...FLAG ADDRESSES
24 0000 0001 A  IENFL =      1
25 0000 0002 A  SELX  =      2
26 0000          ;
27 0000          ; LINK ADDRESSES FOR PART 1
28 0000          ;
29 0000 FE00 A  ROMAD  =     X*FE00
30 0000 FE00 A  START  =     ROMAD
31 0000          ;
32 0000          ;
33 0000          .ASECT
34 0000 FFFE A  .=ROMAD+X*1FE
35 FFFE 21E0 A  ENTRY: JMP    PANEL          ;TRANSFER TO PANEL SERVICE RO
36 FFFF          ;
37 FFFF          ;
38 FFFF FF00 A  .=ROMAD+X*100
39 FF00 E514 A  SKG   R1,M1          ;-1>-1
40 FF01 2102 A  JMP    .+3
41 FF02 2500 A  JMP    @.+1
42 FF03 FF8C A  .WORD  ERROR
43 FF04 E511 A  SKG   R1,M2          ;-1>-2
44 FF05 21FD A  JMP    .-2          ;REF .-2 IN ORDER TO REACH ER
45 FF06 2110 A  JMP    TISZ          ;CONTINUE TESTING

46 FF07          .PAGE
47 FF07          ;
48 FF07          ; CONSTANTS
49 FF07          ;
50 FF07 0000 A  D0:   .WORD  0
51 FF08 0001 A  D1:   .WORD  1
52 FF09 0002 A  D2:   .WORD  2
53 FF0A 000F A  D15:  .WORD  15
54 FF0B 5050 A  H5050: .WORD  X*5050
55 FF0C 5555 A  H5555: .WORD  X*5555
56 FF0D 5A5A A  H5A5A: .WORD  X*5A5A          ; ALTERNATING ONES
57 FF0E 7FFF A  H7FFF: .WORD  X*7FFF          ; MAX POSITIVE VALUE
58 FF0F 8000 A  H8000: .WORD  X*8000          ; MAX NEGATIVE VALUE
59 FF10 A0A0 A  HA0A0: .WORD  X*A0A0
60 FF11 A5A5 A  HA5A5: .WORD  X*A5A5
  
```

```

61 FF12 AAAA A HAAA4: .WORD X'AAAA
62 FF13 FOFO A HFOFO: .WORD X'FOFO
63 FF14 FACE A HFACE: .WORD X'FACE
64 FF15 FFFF A M1: .WORD -1
65 FF16 FFFE A M2: .WORD -2

66 FF17 ; .PAGE
67 FF17 ;
68 FF17 ; ...TEST ISZ AND DSZ
69 FF17 780A A TISZ: ISZ 10 ;TEST VALUE -1
70 FF18 2173 A JMP ERROR
71 FF19 780A A ISZ 10 ; TEST VALUE 0
72 FF1A 2101 A JMP .+2
73 FF1B 2170 A JMP ERROR
74 FF1C 7C0A A DSZ 10 ;TEST VALUE 1
75 FF1D 216E A JMP ERROR
76 FF1E 7C0A A DSZ 10 ;TEST VALUE 0
77 FF1F 2101 A JMP .+2 ;LOC 10 VALUE = -1
78 FF20 216B A JMP ERROR

79 FF21 ; .PAGE
80 FF21 ;
81 FF21 ; TEST FLAGS
82 FF21 ;
83 FF21 4CFF A LI RO,-1 ; TEST VALUE -1
84 FF22 1169 A BOC ZRO,ERROR
85 FF23 1268 A BOC PZRO,ERROR
86 FF24 1301 A BOC ODD, .+2
87 FF25 2166 A JMP ERROR
88 FF26 1401 A BOC BIT1, .+2
89 FF27 2164 A JMP ERROR
90 FF28 1B01 A BOC NZRO, .+2
91 FF29 2162 A JMP ERROR
92 FF2A 1501 A BOC NZERO, .+2
93 FF2B 2160 A JMP ERROR
94 FF2C 4801 A AISZ RO,1
95 FF2D 215E A JMP ERROR ;AISZ FAILED TO SKIP ON ZERO
96 FF2E 1101 A BOC ZRO, .+2 ; TEST VALUE 0
97 FF2F 215C A JMP ERROR
98 FF30 1201 A BOC PZRO, .+2
99 FF31 215A A JMP ERROR
100 FF32 1359 A BOC ODD,ERRCR
101 FF33 1458 A BOC BIT1,ERROR
102 FF34 1B01 A BOC NZRO, .+2
103 FF35 2156 A JMP ERROR
104 FF36 1555 A BOC NZERO,ERROR
105 FF37 A000 A ST RO,0
106 FF38 4801 A AISZ RO,1 ;ASSUME CARRY-IN WORKS
107 FF39 A000 A ST RO,0
108 FF3A 7C00 A DSZ 0 ;TEST IF AISZ CAUSED SKIP
109 FF3B 2150 A JMP ERROR ;AISZ SKIPPED ON NON-ZERO
110 FF3C 114F A BOC ZRO,ERROR ;TEST VALUE 1
111 FF3D 1201 A BOC PZRO, .+2
112 FF3E 214D A JMP ERROR
113 FF3F 1301 A BOC ODD, .+2
114 FF40 214B A JMP ERROR
115 FF41 144A A BOC BIT1,ERROR
116 FF42 1849 A BOC NZRO,ERRCR
117 FF43 1501 A BOC NZERO, .+2
118 FF44 2147 A JMP ERROR
119 FF45 4801 A AISZ RO,1
120 FF46 1345 A BOC ODD,ERROR ;TEST VALUE 2
121 FF47 1401 A BOC BIT1, .+2
122 FF48 2143 A JMP ERROR

```

ROMDI2

```

123 FF49          .PAGE
124 FF49          ;
125 FF49          ; TEST IF RALU FLAGS CAN BE SET AND CLEARED VIA STACK.
126 FF49          ; TEST CYOV FLAG
127 FF49          ;
128 FF49 4C00 A   LI      R0,0
129 FF4A 4DFF A   LI      R1,-1
130 FF4B 4100 A   PUSH     R1
131 FF4C 0280 A   PULLF           ;FLAGS SHOULD ALL BE SET
132 FF4D 4000 A   PUSH     R0
133 FF4E 0080 A   PUSHF
134 FF4F 4600 A   PULL     R2           ;R2 SHOULD CONTAIN -1
135 FF50 F9C4 A   SKNE    R2,M1
136 FF51 2101 A   JMP     .+2
137 FF52 2139 A   JMP     ERROR
138 FF53 0A00 A   SFLG   SELX           ;TEST OVERFLOW
139 FF54 1A01 A   BOC    CYOV,+.2       ;SHOULD BE SET
140 FF55 2136 A   JMP     ERROR
141 FF56 0A80 A   PFLG   SELX           ;TEST CARRY
142 FF57 1A01 A   BOC    CYOV,+.2       ;SHOULD BE SET
143 FF58 2133 A   JMP     ERROR
144 FF59          ;
145 FF59          ; ...TRY TO CLEAR FLAGS
146 FF59 4000 A   PUSH     R0
147 FF5A 0280 A   PULLF
148 FF5B 4100 A   PUSH     R1
149 FF5C 0080 A   PUSHF
150 FF5D 4400 A   PULL     R0
151 FF5E 152D A   BOC    NZERO,ERROR    ;ASSUME JUMP CONDITION WORKS
152 FF5F 0A00 A   SFLG   SELX           ;TEST OVERFLOW
153 FF60 1A2B A   BOC    CYCV,ERROR    ;SHOULD NOT BE SET
154 FF61 0A80 A   PFLG   SELX           ;TEST CARRY
155 FF62 1A29 A   BOC    CYOV,ERROR    ;SHOULD NOT BE SET

156 FF63          .PAGE
157 FF63          ;
158 FF63          ; TEST SHIFT WITHOUT LINK
159 FF63          ;
160 FF63 0A80 A   PFLG   SELX           ; DO NOT INCLUDE LINK IN SHIF
161 FF64 4C00 A   LI      R0,0
162 FF65 4000 A   PUSH     R0
163 FF66 0280 A   PULLF           ; CLEAR LINK
164 FF67 81AD A   LD      R0,M1
165 FF68 5C0F A   SHL    R0,15
166 FF69 F1A5 A   SKNE    R0,H8000
167 FF6A 2101 A   JMP     .+2
168 FF6B 2120 A   JMP     ERROR
169 FF6C 0080 A   PUSHF
170 FF6D 4400 A   PULL     R0
171 FF6E 1201 A   BOC    PZRO,+.2       ; LINK SHOULD NOT BE SET
172 FF6F 211C A   JMP     ERROR
173 FF70 819E A   LD      R0,H8000
174 FF71 5CF1 A   SHR    R0,15
175 FF72 48FF A   AISZ   R0,-1
176 FF73 2119 A   JMP     ERROR
177 FF74 4CFF A   LI      R0,-1
178 FF75 5CF1 A   SHR    R0,15
179 FF76 48FF A   AISZ   R0,-1
180 FF77 2114 A   JMP     ERROR
181 FF78          ;
182 FF78          ; CHECK ROTATE WITHOUT LINK
183 FF78          ;
184 FF78 8199 A   LD      R0,HA4AA     ; LINK WAS CLEARED PREVIOUSLY
185 FF79 5801 A   ROL    R0,1
186 FF7A F191 A   SKNE    R0,H5555
187 FF7B 2101 A   JMP     .+2
188 FF7C 210F A   JMP     ERROR

```

```

189 FF7D 0080 A      PUSHF
190 FF7E 4500 A      PULL      R1
191 FF7F 758F A      SKAZ      R1,H8000
192 FF80 210B A      JMP      ERROR
193 FF81 58FE A      ROR      R0,2
194 FF82 F189 A      SKNE     R0,H5555
195 FF83 2101 A      JMP      .+2
196 FF84 2107 A      JMP      ERROR
197 FF85 0080 A      PUSHF
198 FF86 4500 A      PULL      R1
199 FF87 7587 A      SKAZ      R1,H8000
200 FF88 2103 A      JMP      ERROR
201 FF89 5801 A      ROL      R0,1
202 FF8A F187 A      SKNE     R0,HAAAA
203 FF8B 2105 A      JMP      SHLNK
204 FF8C
205 FF8C 4CFF A      ; ERROR: LI      R0,-1
206 FF8D 0F00 A      SFLG     7
207 FF8E 0000 A      HALT
208 FF8F 2500 A      JMP      @.+1
209 FF90 FFFE A      .WORD   ENTRY
; NO-GO INDICATOR
; ERROR OCCURRED
; REEXECUTE FROM BEGINNING

210 FF91
211 FF91
212 FF91
213 FF91
214 FF91 0A00 A      .PAGE
;
; TEST SHIFT WITH LINK
;
; SHLNK: SFLG     SELX
215 FF92 4C01 A      LI      R0,1
216 FF93 5C10 A      SHL     R0,16
217 FF94 1101 A      BOC     ZR0,+.2
218 FF95 21F6 A      JMP      ERROR
219 FF96 0080 A      PUSHF
220 FF97 4500 A      PULL      R1
221 FF98 750F A      SKAZ      R1,X8000
222 FF99 2101 A      JMP      .+2
223 FF9A 21F1 A      JMP      ERROR
224 FF9B 5CF0 A      SHR     R0,16
225 FF9C F109 A      SKNE     R0,I1
226 FF9D 2101 A      JMP      .+2
227 FF9E 21ED A      JMP      ERROR
228 FF9F 0080 A      PUSHF
229 FFA0 4400 A      PULL      R0
230 FFA1 7106 A      SKAZ      R0,X8000
231 FFA2 21E9 A      JMP      ERKOR
232 FFA3 2106 A      JMP      ROTLNK
; ASSUME SKAZ WORKS
; ASSUME SKNE WORKS

233 FFA4
234 FFA4
235 FFA4
236 FFA4
;
; TEST ROTATE WITH LINK
;
237 FFA4 5554 A      H5554: .WORD   X'5554
238 FFA5 D555 A      HD555: .WORD   X'D555
239 FFA6 0001 A      I1:    .WORD   1
240 FFA7 5555 A      X5555: .WORD   X'5555
241 FFA8 8000 A      X8000: .WORD   X'8000
242 FFA9 AAAA A      XAAAA: .WORD   X'AAAA
243 FFAA 89FE A      ROTLNK: LD      R2,XAAAA
244 FFAB 4D00 A      LI      R1,0
245 FFAC 4100 A      PUSH    R1
246 FFAD 0280 A      PULLF
247 FFAE 5A01 A      ROL     R2,1
248 FFAF F9F4 A      SKNE     R2,H5554
249 FF80 2101 A      JMP      .+2
250 FF81 210A A      JMP      ERROR
251 FF82 0080 A      PUSHF
252 FF83 4400 A      PULL     R0
; SET LINK=0

```

253	FFB4	12D7	A	BOC	PZRO,ERROR	; LINK SHOULD BE SET
254	FFB5	5AFE	A	ROR	R2,2	
255	FFB6	F9F0	A	SKNE	R2,X5555	
256	FFB7	2101	A	JMP	+.2	
257	FFB8	21D3	A	JMP	ERROR	
258	FFB9	0080	A	PUSHF		
259	FFBA	4400	A	PULL	RO	
260	FFBB	4800	A	AISZ	RO,0	
261	FFBC	21CF	A	JMP	ERROR	; LINK SHOULD NOT BE SET
262	FFBD	5A01	A	ROL	R2,1	
263	FFBE	F9EA	A	SKNE	R2,XAAAA	
264	FFBF	2101	A	JMP	+.2	
265	FFC0	21CB	A	JMP	ERROR	
266	FFC1	0080	A	PUSHF		
267	FFC2	4400	A	PULL	RO	
268	FFC3	4800	A	AISZ	RO,0	
269	FFC4	21C7	A	JMP	ERROR	; LINK SHOULD NOT BE SET
270	FFC5	4CFF	A	LI	RO,-1	
271	FFC6	4000	A	PUSH	RO	
272	FFC7	0280	A	PULLF		; SET LINK
273	FFC8	5A01	A	ROL	R2,1	
274	FFC9	F9DD	A	SKNE	R2,X5555	
275	FFCA	2101	A	JMP	+.2	
276	FFCB	21C0	A	JMP	ERROR	
277	FFCC	0080	A	PUSHF		
278	FFCD	4400	A	PULL	RO	
279	FFCE	12BD	A	BOC	PZRO,ERROR	; LINK SHOULD BE SET
280	FFCF	5AFE	A	ROR	R2,2	
281	FFD0	F9D4	A	SKNE	R2,HD555	
282	FFD1	2101	A	JMP	+.2	
283	FFD2	21B9	A	JMP	ERROR	
284	FFD3	0080	A	PUSHF		
285	FFD4	4400	A	PULL	RO	
286	FFD5	71D2	A	SKAZ	RO,X8000	
287	FFD6	21B5	A	JMP	ERROR	; LINK SHOULD BE ZERO
288	FFD7	5A01	A	ROL	R2,1	
289	FFD8	F9D0	A	SKNE	R2,XAAAA	
290	FFD9	2101	A	JMP	+.2	
291	FFDA	21B1	A	JMP	ERROR	
292	FFDB	0080	A	PUSHF		
293	FFDC	4400	A	PULL	RO	
294	FFDD	12AE	A	BOC	PZRO,ERROR	
295	FFDE	0800	A	SFLG	0	; INDICATE GOOD RESULTS
296	FFDF	A004	A	ST	0,4	; SAVE REGISTERS
297	FFE0	A405	A	ST	1,5	
298	FFE1	A806	A	ST	2,6	
299	FFE2	AC07	A	ST	3,7	
300	FFE3	8119	A	LD	0,STADD	
301	FFE4	1F10	A	BOC	15,EXIT	
302	FFE5	0600	A	ROUT:	ROUT	0
303	FFE6	1C03	A	WAIT:	BOC	12,LA
304	FFE7	1D07	A		BOC	13,LD
305	FFE8	170A	A		BOC	7,EX
306	FFE9	21FC	A		JMP	WAIT
307	FFEA	1CFF	A	LA:	BOC	12,LA
308	FFEB	0400	A		RIN	0
309	FFEC	3281	A		RCPY	0,2
310	FFED	8200	A		LD	0,(2)
311	FFEE	21F6	A		JMP	ROUT
312	FFEF	1DFF	A	LD:	BOC	13,LD
313	FFF0	0400	A		RIN	0
314	FFF1	A200	A		ST	0,(2)
315	FFF2	21F2	A		JMP	ROUT
316	FFF3	17FF	A	EX:	BOC	7,EX
317	FFF4	0400	A		RIN	0
318	FFF5	4000	A	EXIT:	PUSH	0
319	FFF6	8004	A		LD	0,4

ROMDI2

```

320 FFF7 8405 A      LD      1,5
321 FFF8 8806 A      LD      2,6
322 FFF9 8C07 A      LD      3,7
323 FFFA 0880 A      PFLG   0          ;INITIALIZE RESULT INDICATORS
324 FFFB 0F80 A      PFLG   7
325 FFFC 0200 A      RTS
326 FFFD FE00 A      STADD:  .WORD   START
327 FFFE FFFE A      .END    ENTRY

```

***** 0 ERRORS IN ASSEMBLY *****

```

BIT1  CYQV  D0      D1      D15     D2      ENTRY  ERROR  EX      EXIT
0004 A 000A A FF07 A FF08 A FFOA A FF09 A FFFE A FF8C A FFF3 A FFF5 A

H5050 H5554 H5555 H5A5A H7FFF H8000 HAOAO HASA5 HAAAA HD555
FF0B A FFA4 A FF0C A FF0D A FFOE A FF0F A FF10 A FF11 A FF12 A FFA5 A

HF0F0 HFACE I1      IENFL  INTEN  LA      LD      M1      M2      NZERO
FF13 A FF14 A FFA6 A 0001 A 0009 A FFEA A FFEF A FF15 A FF16 A 0005 A

NZRO  ODD  PANEL  PZRO  RO      R1      R2      R3      ROMAD  ROTLNK
000B A 0003 A FFDF A 0002 A 0000 A 0001 A 0002 A 0003 A FE00 A FFAA A

ROUT  SELX  SHLNK  STADD  START  STKFL  TISZ  WAIT  X5555  X8000
FFE5 A 0002 A FF91 A FFFD A FE00 A 0008 A FF17 A FFE6 A FFA7 A FFA8 A

XAAAA ZRO
FFA9 A 0001 A

```

43D6 BC2F

ROMDIX LISTING

INTRODUCTION

The listing of the Read Only Memory Diagnostic (eXtended) Program (ROMDIX) is presented on the following pages.

ROMDIX is a compact CPU diagnostic that is written in the IMP-16 Assembler language, whose revision level appears at the top of the first page.

ROMDIX tests the Extended Instruction Set option of the Central Processing Unit (CPU) on the IMP-16C card on a "go/no-go" basis. If the test indicates a failure, the particular malfunctioning device is not designated. To test individual devices in the IMP-16C, first, it must be assured that all other CPU devices and the other components on the IMP-16C are functioning.

It should be noted that ROMDIX only tests the Extended Instruction Set of the CPU. It is assumed that the ROMDI program is used to test the Basic Instruction Set, and the commonalities between the two instruction sets (for example, addressing).

There are four RALUs (Register, Arithmetic, and Logic Units) and two CROMs (Control Read-Only Memories) in the IMP-16C CPU. For example, a particular RALU may be tested by first assuring that the CROM I, CROM II, and three of the other RALUs on the IMP-16C are functioning. In this case, if the test passes, the RALU under test is functioning; but if the test fails the RALU under test is malfunctioning.

Instructions for performing ROMDIX are given in the listing.

ROMDIX

REVISION-F 10/02/73
 ROMDIX 000327A 01/03/74

```

1 0000      .TITLE 'ROMDIX,'000327A 01/03/74'
2 0000      ;
3 0000      ;
4 0000      ;
5 0000      ;
6 0000      ;
7 0000      ;
8 0000      ;
9 0000      ;
10 0000     ;
11 0000     ;
12 0000     ;
13 0000     ;
14 0000     ;
15 0000     ;
16 0000     ;
17 0000     ;
18 0000     ;
19 0000     ;
20 0000     ;
21 0000     ;
22 0000     ;
23 0000     ;
24 0000     ;
25 0000     ;
26 0000     ;
27 0000     ;
28 0000     ;
29 0000     ;
30 0000     ;
31 0000     ;
32 0000     ;
33 0000     ;
34 0000     ;
35 0000     ;
36 0000     ;
37 0000     ;
38 0000     ;
39 0000     ;
40 0000     ;
41 0000     ;
42 0000     ;
43 0000     ;
44 0000     ;
45 0000     ;
46 0000     ;
47 0000     ;
48 0000     ;
49 0000     ;
50 0000     ;
51 0000     ;
52 0000     ;
53 0000     ;
54 0000     ;
55 0000     ;
56 0000     ;
57 0000     ;
58 0000     ;
59 0000     ;
60 0000     ;

```

MAIN PROGRAM IN ROM, X'FE00-X'FFFF, ENTRY POINT X'FFFE

ROMDIX IS A GO-NOGO CPU DIAGNOSTIC BASICALLY DESIGNED TO TEST THE EXTENDED INSTRUCTION SET OPTION FOR THE IMP-16C THE PROGRAM FITS INTO 4 8X256-BIT PROMS IN THE FOLLOWING MANNER:

	ADDRESSES	BITS	PROM DESIG	DIAGRAM CO-ORDINATE
ROMDIX1 RIGHT	X'FE00-X'FEFF	0-7	BJA	4F
ROMDIX1 LEFT	X'FE00-X'FEFF	8-15	BJB	4D
ROMDIX2 RIGHT	X'FF00-X'FFFF	0-7	BJC	4E
ROMDIX2 LEFT	X'FF00-X'FFFF	8-15	BJD	4C

ENTERING THE PROGRAM AT ADDRESS FFFE TRANSFERS CONTROL TO A PANEL ROUTINE WHICH PERFORMS THE FOLLOWING FUNCTIONS IN RESPONSE TO THE INDICATED SIGNALS:

SIGNAL NAME	PIN NO.	FUNCTION
JC12	129	(LOAD ADDRESS) LOADS THE CONTENTS OF SW00,SW01,...,SW15 SIGNALS INTO AC2 REGISTER AND DISPLAYS THE CONTENTS OF THE MEMORY LOCATION ADDRESSED BY AC2 USING THE SIGNALS BDO00,BDO01,...,BDO15.
JC13	119	(LOAD DATA) LOADS THE CONTENTS OF SW00,SW01,...,SW15 SIGNALS INTO ACO REGISTER, STORES ACO INTO MEMORY LOCATION ADDRESSED BY AC2, & DISPLAYS ACO USING THE SIGNALS BDO00,BDO01,...,BDO15.
START	123	(EXECUTE) TRANSFERS CONTROL TO THE MEMORY LOCATION INDICATED BY THE SIGNALS SW00,SW01,...,SW15.

UPON ENTRY TO THE PANEL ROUTINE, ACCUMULATORS 0,1,2,& 3 ARE SAVED IN MEMORY LOCATIONS 4,5,6,& 7, RESPECTIVELY. THE CONTENTS OF ACCUMULATOR MAY BE SET BY ALTERING ITS CORRESPONDING MEMORY LOCATION.

IF SIGNAL JC15 (JUMP CONDITION 15) IS HIGH WHEN 'EXECUTE', I.E. THE START SIGNAL IS ON, THE DIAGNOSTIC WILL BE CONTINUOUSLY REEXECUTED UNTIL EITHER THE JUMP CONDITION GOES LOW OR AN ERROR CONDITION IS DETECTED.

IF THE DIAGNOSTIC EXECUTES SUCCESSFULLY, THE PROGRAM SETS FLAG 8 (SIGNAL F8) AND RETURNS CONTROL TO THE PANEL ROUTINE. IF THE PROGRAM DETECTS AN ERROR CONDITION, IT SETS FLAG 15 (SIGNAL F15) AND HALTS (SIGNAL HLT* (FLAG)).

THE DIAGNOSTIC ENTRY POINT IS X'FE00.

NOTE: SEE APPENDIX E IN THE IMP-16C APPLICATION MANUAL (PUB. NO. 4200021B) FOR THE LIST OF PIN CONNECTIONS AND SIGNALS ON IMP-16C CARD.

ROMDIX

```

61 0000                .PAGE
62 0000                ; SYMBOL EQUATES
63 0000 0000 A R0      0                ; AC0 NAME
64 0000 0001 A R1      1                ; AC1 NAME
65 0000 0002 A R2      2                ; AC2 NAME
66 0000 0003 A R3      3                ; AC3 NAME
67 0000                ;
68 0000                ; ... BRANCH CONDITIONS
69 0000 0008 A STKFL   8                ; STACK FULL
70 0000 0001 A ZRO     1                ; ZERO
71 0000 0002 A PZRO    2                ; POSITIVE OR ZERO
72 0000 0003 A ODD     3                ; ODD (BIT 0 TRUE)
73 0000 0004 A BIT1    4                ; BIT 1 TRUE
74 0000 0005 A NZERO   5                ; NOT EQUAL ZERO
75 0000 0009 A INTEN   9                ; INTERRUPT ENABLE
76 0000 000A A CYOV    10               ; CARRY/OVERFLOW
77 0000 000B A NZRO    11               ; NEGATIVE OR ZERO
78 0000 000D A SEL     13               ; SELECT FLAG TRUE
79 0000                ; ... FLAG ADDRESSES
80 0000 0001 A IENFL   1                ; INTERRUPT ENABLE FLAG
81 0000 0002 A SELX    2                ; SELECT CONTROL FLAG
82 0000                ;
83 0000 FE00 A ROMAD   X'FE00           ; ROM STARTING ADDRESS
84 0000                ;
85 0000                .EXTD

86 0000                .PAGE
87 0000                .ASECT           ;USE .TSECT WHEN ROMDIX IS RELOC
88 0000 FFFE A        . ROMAD+X'1FE     ;USED WHEN ROMDIX IS IN PROMS
89 FFFE 2500 A ENTRY: JMP @.+1         ; TRANSFER TO PANEL SERVICE ROUTINE
90 FFFF FF77 A        .WORD    PANEL
91 000                ;
92 000                ;
93 000 FE00 A        . ROMAD           ;STARTING LOC OF ROM (USED WHEN ROMDIX IS IN PROMS)
94 FE00                ;
95 FE00                ; SECTION 1.....
96 FE00                ; TEST MPY AND DIV INSTRUCTIONS
97 FE00                ;
98 FE00 4F01 A START: LI R3,1          ; ESTABLISH SECTION COUNTER IN R3
99 FE01 0A00 A SFLG SELX              ; SET SELECT FLAG
100 FE02 4D01 A LI R1,1                ; (AC1) NON-ZERO ( 1)
101 FE03 0480 A MPY D0                 ; X ZERO
    FE04 FE78 A
102 FE05 1D70 A BOC SEL,ERR1           ; SEL MUST BE RESET BY MPY
103 FE06 3400 A RADD R1,R0             ;
104 FE07 156E A BOC NZERO,ERR1        ; ERROR IF ANSWER NON-ZERO
105 FE08                ; (AC1) 0
106 FE08 0480 A MPY D4                 ; X 4
    FE09 FE79 A
107 FE0A 3400 A RADD R1,R0             ; ANSWER 0
108 FE0B 156A A BOC NZERO,ERR1
109 FE0C                ;
110 FE0C                ; (AC1) 0
111 FE0C 0480 A MPY D0                 ; X ZERO
    FE0D FE78 A
112 FE0E 3400 A RADD R1,R0
113 FE0F 1566 A BOC NZERO,ERR1        ; ERROR IF ANSWER NON-ZERO
114 FE10 4D01 A LI R1,1                ; (AC1) 1
115 FE11 0480 A MPY D4                 ; X 4
    FE12 FE79 A
116 FE13 3400 A RADD R1,R0             ; ANSWER 4
117 FE14 48FC A AISZ R0,-4
118 FE15 2160 A JMP ERR1
119 FE16 8563 A LD R1,X7FFF           ; (AC1) X'7FFF
120 FE17 0480 A MPY X7FFF             ; X X'7FFF
    FE18 FE7A A

```

ROMDIX

121	FE19	3400	A	RADD	R1,R0	:	ANSWER	X'3FFF0001
122	FE1A	D160	A	SUB	R0,X4000			
123	FE1B	155A	A	BOC	NZERO,ERR1			
124	FE1C	4D04	A	LI	R1,4	:	(AC1)	4
125	FE1D	0480	A	MPY	X7FFF	:	X	X'7FFF
	FE1E	FE7A	A					
126	FE1F	3100	A	RADD	R0,R1	:	ANSWER	X'1FFFC
127	FE20	4903	A	AISZ	R1,3			
128	FE21	2154	A	JMP	ERR1			
129	FE22	8557	A	LD	R1,X7FFF	:	(AC1)	X'7FFF
130	FE23	0480	A	MPY	D4	:	X	4
	FE24	FE79	A					
131	FE25	3100	A	RADD	R0,R1	:	ANSWER	X'1FFFC
132	FE26	4903	A	AISZ	R1,3			
133	FE27	214E	A	JMP	ERR1			
134	FE28	4D03	A	LI	R1,3	:	(AC1)	3
135	FE29	0480	A	MPY	D5	:	X	5
	FE2A	FE7C	A					
136	FE2B	3400	A	RADD	R1,R0	:	ANSWER	15
137	FE2C	48F1	A	AISZ	R0,-15			
138	FE2D	2148	A	JMP	ERR1			
139	FE2E	4D19	A	LI	R1,25	:	(AC0)	0, (AC1) 25
140	FE2F	0490	A	DIV	D5	:	/	5
	FE30	FE7C	A					
141	FE31	1544	A	BOC	NZERO,ERR1	:	ANSWER	5
142	FE32	49FB	A	AISZ	R1,-5			
143	FE33	2142	A	JMP	ERR1			
144	FE34	4D1B	A	LI	R1,27	:	(AC0)	0, (AC1) 27
145	FE35	0490	A	DIV	D5	:	/	5
	FE36	FE7C	A					
146	FE37	48FE	A	AISZ	R0,-2	:	QUO	5, REM 2
147	FE38	213D	A	JMP	ERR1			
148	FE39	49FB	A	AISZ	R1,-5			
149	FE3A	213B	A	JMP	ERR1			
150	FE3B	4C07	A	LI	R0,7	:	(AC0)	X'0007
151	FE3C	4DF0	A	LI	R1,-16	:	(AC1)	X'FFF0
152	FE3D	0490	A	DIV	D16	:	/	16
	FE3E	FE7D	A					
153	FE3F	1536	A	BOC	NZERO,ERR1	:	ANSWER	X'7FFF
154	FE40	F539	A	SKNE	R1,X7FFF			
155	FE41	2101	A	JMP	+.2			
156	FE42	2133	A	JMP	ERR1			
157	FE43	4C07	A	LI	R0,7	:	(AC0)	X'0007
158	FE44	4DF5	A	LI	R1,-11	:	(AC1)	X'FFF5
159	FE45	0490	A	DIV	D16	:	/	16
	FE46	FE7D	A					
160	FE47	48FB	A	AISZ	R0,-5	:	QUO	X'7FFF, REM 5
161	FE48	212D	A	JMP	ERR1			
162	FE49	F530	A	SKNE	R1,X7FFF			
163	FE4A	2101	A	JMP	+.2			
164	FE4B	212A	A	JMP	ERR1			
165	FE4C	4D03	A	LI	R1,3	:	(AC0)	0, (AC1) 3
166	FE4D	0490	A	DIV	D5	:	/	5
	FE4E	FE7C	A					
167	FE4F	48FD	A	AISZ	R0,-3	:	QUO	0, REM 3
168	FE50	2125	A	JMP	ERR1			
169	FE51	4900	A	AISZ	R1,0			
170	FE52	2123	A	JMP	ERR1			
171	FE53	4C06	A	LI	R0,6	:	(AC0)	6
172	FE54	4D01	A	LI	R1,1	:	(AC1)	1
173	FE55	0490	A	DIV	D5	:	/	5
	FE56	FE7C	A					
174	FE57	0A00	A	SFLG	SELX	:	SET SELECT FLAG FOR OV TEST	
175	FE58	1A01	A	BOC	CYOV,+.2	:	SHOULD HAVE OVERFLOW	

ROMDIX

```

176 FE59 211C A      JMP      ERR1
177 FE5A 4C0F A      LI       R0,15      ; (AC0)  X'000F
178 FE5B 4DF0 A      LI       R1,-16     ; (AC0)  X'FFFF
179 FE5C 0490 A      DIV      D16        ; /16
      FE5D FE7D A
180 FE5E 0A00 A      SFLG    SELX
181 FE5F 1A01 A      BOC     CYOV,..+2   ; SHOULD HAVE OVERFLOW
182 FE60 2115 A      JMP      ERR1
183 FE61 4C00 A      LI       R0,0       ; (AC0)  0
184 FE62 4D01 A      LI       R1,1       ; (AC1)  1
185 FE63 0490 A      DIV      D0         ; /0
      FE64 FE78 A
186 FE65 0A00 A      SFLG    SELX
187 FE66 1A01 A      BOC     CYOV,..+2   ; SHOULD HAVE OVERFLOW
188 FE67 210E A      JMP      ERR1
189 FE68 4C05 A      LI       R0,5       ; (AC0)  5
190 FE69 4D01 A      LI       R1,1       ; (AC1)  1
191 FE6A 0490 A      DIV      D5         ; /5
      FE6B FE7C A
192 FE6C 0A00 A      SFLG    SELX
193 FE6D 1A01 A      BOC     CYOV,..+2   ; SHOULD HAVE OVERFLOW
194 FE6E 2107 A      JMP      ERR1
195 FE6F 4C08 A      LI       R0,8
196 FE70 5C08 A      SHL     R0,8        ; (AC0)  X'0800
197 FE71 4D0F A      LI       R1,15      ; (AC1)  X'000F
198 FE72 0490 A      DIV      X1000      ; / X'1000
      FE73 FE7E A
199 FE74 0A00 A      SFLG    SELX
200 FE75 1A09 A      BOC     CYOV,SEC2   ;SHOULD HAVE OVERFLOW

201 FE76                .PAGE
202 FE76 2500 A  ERR1:  JMP      @.+1      ; GO TO ERROR EXIT
203 FE77 FF71 A      .WORD   ERROR
204 FE78                ;
205 FE78                ; CONSTANTS
206 FE78                ;
207 FE78 0000 A  D0:    .WORD   0          ;ZERO
208 FE79 0004 A  D4:    .WORD   4          ;FOUR
209 FE7A 7FFF A  X7FFF: .WORD   X'7FFF
210 FE7B 4000 A  X4000: .WORD   X'4000
211 FE7C 0005 A  D5:    .WORD   5          ;FIVE
212 FE7D 0010 A  D16:   .WORD   16         ;SIXTEEN
213 FE7E 1000 A  X1000: .WORD   X'1000

214 FE7F                .PAGE
215 FE7F                ; SECTION 2.....
216 FE7F                ; TEST DADD AND DSUB INSTRUCTIONS
217 FE7F                ;
218 FE7F 4F02 A  SEC2:  LI       R3,2      ;UPDATE SECTION COUNTER
219 FE80 4C00 A      LI       R0,0         ; (AC0)  0
220 FE81 4DFF A      LI       R1,-1        ; (AC1)  X'FFFF
221 FE82 04A0 A      DADD     DD1          ; +1
      FE83 FEF3 A
222 FE84 3400 A      RADD     R1,R0        ; ANSWER  X'10000
223 FE85 48FF A      AISZ    R0,-1
224 FE86 21EF A      JMP      ERR1
225 FE87 4D01 A      LI       R1,1         ; (AC0)  0, (AC1)  1
226 FE88 04A0 A      DADD     XFFFFFFF     ; +X'000FFFFFFF
      FE89 FEF5 A
227 FE8A 3400 A      RADD     R1,R0        ; ANSWER  X'00100000
228 FE8B 48F0 A      AISZ    R0,-16
229 FE8C 21E9 A      JMP      ERR1
230 FE8D 4D0F A      LI       R1,15        ; (AC0)  0, (AC1)  15

```

ROMDIX

```

231 FE8E 04A0 A      DADD    DD1          ; +1
    FE8F FEF3 A
232 FE90 0A00 A      SFLG    SELX
233 FE91 1AE4 A      BOC     CYOV,ERR1    ; ERROR IF OVERFLOW SET
234 FE92 0A80 A      PFLG    SELX
235 FE93 1AE2 A      BOC     CYOV,ERR1    ; ERROR IF CARRY SET
236 FE94 3400 A      RADD    R1,R0        ; ANSWER 16
237 FE95 48F0 A      AISZ    R0,-16
238 FE96 21DF A      JMP     ERR1
239 FE97 4CFF A      LI      R0,-1        ; (AC0) X'FFFF
240 FE98 4DFF A      LI      R1,-1        ; (AC1) X'FFFF
241 FE99 04A0 A      DADD    DD1          ; +1
    FE9A FEF3 A
242 FE9B 3400 A      RADD    R1,R0        ; ANSWER 0
243 FE9C 15D9 A      BOC     NZERO,ERR1
244 FE9D 4CFF A      LI      R0,-1        ; (AC0) -1
245 FE9E 4DFF A      LI      R1,-1        ; (AC1) -1
246 FE9F 04A0 A      DADD    DDMIN1      ; + (-1)
    FEA0 FEF7 A
247 FEA1 3400 A      RADD    R1,R0        ; ANSWER IS -2 (BUT MERGE INTO 1 REG)
248 FEA2 4803 A      AISZ    R0,3         ; MERGER -3 IN 1 REG
249 FEA3 21D2 A      JMP     ERR1
250 FEA4 81D5 A      LD      R0,X7FFF     ; (AC0) X'7FFF
251 FEA5 4DFF A      LI      R1,-1        ; (AC1) X'FFFF
252 FEA6 04A0 A      DADD    DD15        ; +15
    FEA7 FEF9 A
253 FEA8 0A00 A      SFLG    SELX        ; TEST OV
254 FEA9 1A01 A      BOC     CYOV,.-+2
255 FEA A 21CB A      JMP     ERR1        ; SHOULD HAVE OVERFLOW
256 FEAB 0A80 A      PFLG    SELX
257 FEAC 1AC9 A      BOC     CYOV,ERR1    ; NO CARRY EXPECTED
258 FEAD 4C01 A      LI      R0,1         ; (AC0) 1
259 FEAE 4DFF A      LI      R1,-1        ; (AC1) X'FFFF
260 FEAF 04B0 A      DSUB    X1FFFE      ; -X'1FFFE
    FEB0 FEFB A
261 FEB1 3400 A      PADD    R1,R0        ; ANSWER 1
262 FEB2 48FF A      AISZ    R0,-1
263 FEB3 21C2 A      JMP     ERR1
264 FEB4 4D01 A      LI      R1,1         ; (AC0) 0, (AC1) 1
265 FEB5 04B0 A      DSUB    DD2         ; -2
    FEB6 FEFD A
266 FEB7 3400 A      RADD    R1,R0        ; ANSWER -1 (BUT MERGE TO 1 REG)
267 FEB8 4802 A      AISZ    R0,2         ; MERGER -2 IN 1 REG
268 FEB9 21BC A      JMP     ERR1
269 FEBA 4D0F A      LI      R1,15        ; (AC0) 0, (AC1) 15
270 FEBB 04B0 A      DSUB    DD13        ; -13
    FEBC FEF3 A
271 FEBD 3400 A      RADD    R1,R0        ; ANSWER 2
272 FEBE 48FE A      AISZ    R0,-2
273 FEBF 21B6 A      JMP     ERR1
274 FEC0 4D01 A      LI      R1,1         ; (AC0) 0, (AC1) 1
275 FEC1 04B0 A      DSUB    DD1         ; -1
    FEC2 FEF3 A
276 FEC3 3400 A      RADD    R1,R0        ; ANSWER 0
277 FEC4 15B1 A      BOC     NZERO,ERR1
278 FEC5 4CFF A      LI      R0,-1        ; (AC0) X'FFFF
279 FEC6 4DFF A      LI      R1,-1        ; (AC1) X'FFFF
280 FEC7 04B0 A      DSUB    DD1         ; -1
    FEC8 FEF3 A
281 FEC9 3400 A      RADD    R1,R0        ; ANSWER -2 ( BUT MERGE INTO 1 REG)
282 FECA 4803 A      AISZ    R0,3         ; MERGER -3
283 FECB 21AA A      JMP     ERR1
284 FECC 4D0F A      LI      R1,15        ; (AC0) 0, (AC1) 15
285 FECD 04B0 A      DSUB    X8000       ; -X'80000000
    FECE FF01 A
286 FECE 0A00 A      SFLG    SELX        ; TEST OVERFLOW
287 FED0 1A01 A      BOC     CYOV,.-+2    ; SHOULD HAVE OVERFLOW

```

ROMDIX

```

288 FED1 21A4 A      JMP      ERR1
289 FED2 812E A      LD        R0,X8000      ;(AC0)  X'8000
290 FED3 4D00 A      LI        R1,0          ; (AC1)  0
291 FED4 04B0 A      DSUB     X8000          ; -X'80000000
      FED5 FF01 A
292 FED6 1A01 A      BOC      CYOV,SEC3     ;TEST CARRY
293 FED7 219E A      JMP      ERR1          ; SHOULD HAVE CARRY

```

```

294 FED8
295 FED8      ; SECTION 3.....
296 FED8      ; TEST LDB & STB INST. USING ASSEM. MNEMONICS LLB,LRB,SRB,SLB
297 FED8      ;
298 FED8 4F03 A SEC3:  LI      R3,3          ;UPDATE SECTION COUNTER
299 FED9 4CFF A      LI      R0,-1         ; (AC0)  X'FFFF
300 FEDA 8971 A      LD      R2,BASPT1     ;(AC2)  X'100 (LOC. IN BASE PAGE)
301 FEDB 8527 A      LD      R1,X0F0A
302 FEDC A600 A      ST      R1,(R2)      ;PUT X'0F0A IN LOC. X'100
303 FEDD 5E01 A      SHL     R2,1          ;SET UP EA IN R2 FOR LLB & LRB INST.
304 FEDE 06C0 A      LLB     (R2)          ;LOAD LEFT BYTE, I.E. X'0F
      FEDF 0000 A
305 FEE0 48F1 A      AISZ   R0,-15        ; CHECK FOR X'000F
306 FEE1 2194 A      JMP     ERR1
307 FEE2 4CFF A      LI     R0,-1         ; (AC0)  X'FFFF
308 FEE3 06C0 A      LRB    (R2)          ;LOAD RIGHT BYTE, I.E. X'0A
      FEE4 0001 A
309 FEE5 48F6 A      AISZ   R0,-10        ; CHECK FOR X'000A
310 FEE6 218F A      JMP     ERR1
311 FEE7 4C0A A      LI     R0,10         ; (AC0)  X'000A
312 FEE8 4DFF A      LI     R1,-1         ; (AC1)  X'FFFF
313 FEE9 B562 A      ST     R1,@BASPT1    ;LOC. X'100 NOW CONTAINS X'FFFF
314 FEEA 06D0 A      SRB    (R2)          ;STORE RIGHT BYTE, I.E. X'0A INTO X'100
      FEEB 0001 A
315 FEEC 4C0F A      LI     R0,15         ; (AC0)  X'000F
316 FEED 06D0 A      SLB    (R2)          ;STORE LEFT BYTE, I.E. X'0F
      FEEE 0000 A
317 FEFF 915C A      LD     R0,@BASPT1    ;FETCH CONTENTS OF LOC. X'100
318 FEF0 F112 A      SKNE   R0,X0F0A
319 FEF1 2113 A      JMP     SEC4
320 FEF2 2183 A      JMP     ERR1

```

```

321 FEF3      ; SECTION 4.....
322 FEF3      ;
323 FEF3      ; CONSTANTS
324 FEF3      ;
325 FEF3 0000 A DD1:  .WORD  0          ;DBL PREC 1
326 FEF4 0001 A      .WORD  1
327 FEF5 000F A XFFFFF: .WORD  X'000F      ;X'000FFFFF
328 FEF6 FFFF A      .WORD  X'FFFF
329 FEF7 FFFF A DDMIN1: .WORD  -1        ;DBL PREC (-1)
330 FEF8 FFFF A      .WORD  -1
331 FEF9 0000 A DD15:  .WORD  0          ;DBL PREC 15
332 FEFA 000F A      .WORD  15
333 FEFB 0001 A X1FFFF: .WORD  1          ;X'0001FFFF
334 FEFC FFFE A      .WORD  X'FFFE
335 FEFD 0000 A DD2:  .WORD  0          ;DBL PREC 2
336 FEFE 0002 A      .WORD  2
337 FEFF 0000 A DD13:  .WORD  0          ;DBL PREC 13
338 FF00 000D A      .WORD  13
339 FF01 8000 A X8000:  .WORD  X'8000      ;X'80000000
340 FF02 0000 A      .WORD  0
341 FF03 0F0A A X0F0A:  .WORD  X'0F0A
342 FF04 0008 A D8:   .WORD  8          ;EIGHT

```


ROMDIX

```

343 FF05          .PAGE
344 FF05          ; SECTION 4.....
345 FF05          ; TEST SETST, CLRST, SETBIT, CLRBIT, AND CMPBIT
346 FF05          ;
347 FF05 4F04 A SEC4:  LI      R3,4          ; UPDATE SECTION COUNTER
348 FF06 0080 A          PUSHF          ; STATUS FLAGS TO STACK
349 FF07 4C00 A          LI       R0,0          ; (ACO) 0
350 FF08 4000 A          PUSH      R0          ; 0 WORD TO STACK
351 FF09 0280 A          PULLF         ; 0 WORD TO STATUS FLAGS WORD
352 FF0A 0703 A          SETST     3          ; SET STATUS FLAG 3
353 FF0B 0080 A          PUSHF         ;
354 FF0C 4400 A          PULL      R0          ; PUT GENERATED FLAGS IN ACO
355 FF0D 0280 A          PULLF         ; GET SAVED FLAGS BACK
356 FF0E 48F8 A          AISZ     R0,-8        ; CHECK FOR FLAGS 3 TRUE IN GEN'D FLAGS
357 FF0F 2161 A          JMP      ERR2
358 FF10 0080 A          PUSHF         ; SAVE STATUS FLAGS
359 FF11 4C08 A          LI       R0,8          ; (ACO) HAS BIT 3 SET
360 FF12 4000 A          PUSH      R0          ; (ACO) TO STACK
361 FF13 0280 A          PULLF         ; (ACO) TO FLAG WORD
362 FF14 0713 A          CLRST     3          ; CLEAR BIT 3 IN STATUS WORD
363 FF15 0080 A          PUSHF         ;
364 FF16 4400 A          PULL      R0          ;
365 FF17 0280 A          PULLF         ;
366 FF18 1558 A          BOC      NZERO,ERR2   ; TEST FOR CLEARED BIT 3 IN GEN'D STATUS
367 FF19          ;          (ACO) 0
368 FF19 0723 A          SETBIT    3          ; BIT 3 TRUE
369 FF1A 48F8 A          AISZ     R0,-8        ; CHECK BIT 3
370 FF1B 2155 A          JMP      ERR2
371 FF1C 4C08 A          LI       R0,8          ; (ACO) HAS BIT 3 SET
372 FF1D 0733 A          CLRBIT    3          ; CLEAR BIT 3
373 FF1E 1552 A          BOC      NZERO,ERR2   ; TEST FOR CLEAR
374 FF1F 0763 A          CMPBIT    3          ; (ACO) 0, COMPLEMENT BIT 3
375 FF20 F1E3 A          SKNE     R0,D8        ; CHECK BIT 3 FOR TRUE
376 FF21 2101 A          JMP      .+2
377 FF22 214E A          JMP      ERR2
378 FF23 0763 A          CMPBIT    3          ; COMPLIMENT BIT 3 AGAIN
379 FF24 154C A          BOC      NZERO,ERR2   ; ERROR IF NOT CLEARED

```

```

380 FF25          .PAGE
381 FF25          ; SECTION 5.....
382 FF25          ; TEST SKSTF, SKBIT INSTRUCTIONS
383 FF25          ;
384 FF25 4F05 A          LI       R3,5          ; UPDATE SECTION COUNTER
385 FF26 0703 A          SETST     3          ; SET STATUS FLAG 3
386 FF27 0743 A          SKSTF     3          ;
387 FF28 2148 A          JMP      ERR2          ; ERROR IF NO SKIP
388 FF29 0713 A          CLRST     3          ; CLEAR STATUS FLAG 3
389 FF2A 0743 A          SKSTF     3          ;
390 FF2B 2101 A          JMP      .+2
391 FF2C 2144 A          JMP      ERR2          ; ERROR IF SKIP
392 FF2D 4C00 A          LI       R0,0          ; CLEAR ACO
393 FF2E 072B A          SETBIT    11         ; SET BIT 11
394 FF2F 075B A          SKBIT     11         ;
395 FF30 2140 A          JMP      ERR2          ; ERROR IF NO SKIP
396 FF31 073B A          CLRBIT    11         ; CLEAR BIT 11
397 FF32 075B A          SKBIT     11         ;
398 FF33 2101 A          JMP      SEC6
399 FF34 213C A          JMP      ERR2          ; ERROR IF SKIP

```

ROMDIX

```

400 FF35                .PAGE
401 FF35                ; SECTION 6.....
402 FF35                ; TEST JSRP AND JMPP INSTRUCTIONS
403 FF35                ;
404 FF35 4F06 A SEC6:  LI      R3,6                ;UPDATE SECTION COUNTER
405 FF36 4C00 A        LI      R0,0                ; (AC0) 0
406 FF37 8507 A        LD      R1,PTRADR          ; INIT LOC X'100 WITH (ADR-DISP) OF PTR
407 FF38 B513 A        ST      R1,@BASPT1
408 FF39 0305 A        JSRP   PDISP              ; GO TO SUBR POINTED TO
409 FF3A 0758 A        SKBIT  8                  ; CHECK FOR BIT SET IN SUBR
410 FF3B 2135 A        JMP     ERR2              ; ERROR IF NO SKIP
411 FF3C 2104 A        JMP     SEC6A             ; JUMP AROUND SUBR
412 FF3D 0728 A S6SUBR: SETBIT 8                ; SET BIT FOR TEST ON RETURN FROM SUBR
413 FF3E 0200 A        RTS     0                  ; RETURN
414 FF3F FF3B A PTRADR: .WORD  SUBLOC-5
415 FF40 0005 A PDISP  5
416 FF40 FF3D A SUBLOC: .WORD  S6SUBR
417 FF41 4C00 A SEC6A:  LI      R0,0                ; (AC0) 0
418 FF42 8507 A        LD      R1,JMPPTR          ; INIT LOC X'100 WITH (ADDR-DISP) OF PTR
419 FF43 B508 A        ST      R1,@BASPT1
420 FF44 0505 A        JMPP   PDISP              ; GO TO LOC POINTED TO
421 FF45 0758 A SEC6B:  SKBIT  8                  ; CHECK FOR BIT SET AT LOC
422 FF46 212A A        JMP     ERR2              ; ERROR IF NO SKIP
423 FF47 2105 A        JMP     SEC7              ; GO TO NEXT TEST SECTION
424 FF48 0728 A SEC6C:  SETBIT 8
425 FF49 21FB A        JMP     SEC6B
426 FF4A FF46 A JMPPTR: .WORD  JMPLOC-5
427 FF4B FF48 A JMPLOC: .WORD  SEC6C
428 FF4C 0100 A BASPT1: .WORD  X'100

```

```

429 FF4D                .PAGE
430 FF4D                ; SECTION 7.....
431 FF4D                ; TEST ISCAN AND JINT INSTRUCTIONS
432 FF4D                ;
433 FF4D 4F07 A SEC7:  LI      R3,7                ; UPDATE SECTION COUNTER
434 FF4E 4D00 A        LI      R1,0                ; CLEAR AC1 AND AC2
435 FF4F 4E00 A        LI      R2,0
436 FF50 0510 A        ISCAN  ; CHECK ISCAN WITH ZERO IN AC1
437 FF51 4A00 A        AISZ   R2,0                ; CHECK FOR CHANGE IN AC2
438 FF52 211E A        JMP     ERR2              ; ERROR IF NO SKIP
439 FF53 4D08 A        LI      R1,8                ; SET BIT 3 IN AC1
440 FF54 0510 A        ISCAN  ; EXECUTE INTERRUPT SCAN
441 FF55 211B A        JMP     ERR2              ; ERROR IF NO SKIP AFTER ISCAN
442 FF56 4AFC A        AISZ   R2,-4              ; CHECK AC2 FOR CORRECT SHIFT COUNT 4
443 FF57 2119 A        JMP     ERR2              ; ERROR IF NO SKIP
444 FF58 4D40 A        LI      R1,64             ; SET BIT 6 IN AC1
445 FF59 0510 A        ISCAN
446 FF5A 2116 A        JMP     ERR2
447 FF5B 4AF9 A        AISZ   R2,-7
448 FF5C 2114 A        JMP     ERR2
449 FF5D 4D40 A        LI      R1,64
450 FF5E 5D09 A        SHL    R1,9                ; SET BIT 15 IN AC1
451 FF5F 0510 A        ISCAN
452 FF60 2110 A        JMP     ERR2
453 FF61 4AF0 A        AISZ   R2,-16
454 FF62 210E A        JMP     ERR2
455 FF63 4C00 A        LI      R0,0                ; (AC0) 0
456 FF64 8507 A        LD      R1,INTPTR
457 FF65 B50A A        ST      R1,@BASPT2          ; INIT. LOC (X'120+DISP) WITH PTR.
458 FF66 0525 A        JINT   PDISP              ; JUMP TO DUMMY INTERRUPT ROUTINE
459 FF67 1901 A        BOC    INTEN, +2
460 FF68 2108 A        JMP     ERR2              ; ERROR IF INTEN NOT SET
461 FF69 48F8 A        AISZ   R0,-8                ; CHECK FOR SWITCH SET IN INT. ROUTINE
462 FF6A 2106 A        JMP     ERR2              ; ERROR IF BIT 3 NOT SET
463 FF6B 210A A        JMP     TESTEND             ; DONE WITH LAST TEST SECTION

```

ROMDIX

```

464 FF6C FF6D A INTPTR: .WORD S7SUBR ; PTR TO DUMMY INTERRUPT ROUTINE
465 FF6D 1903 A S7SUBR: BOC INTEN,ERR2 ; ERROR IF INTEN SET
466 FF6E 0723 A SETBIT 3 ; SET BIT 3 FOR SWITCH
467 FF6F 0100 A RTI 0 ; RETURN FROM DUMMY INTERRUPT ROUTINE
468 FF70 0125 A BASPT2: .WORD X'120+PDISP

```

```

469 FF71 ; .PAGE
470 FF71 ;
471 FF71 ; ERROR EXIT
472 FF71 ;
473 FF71 FF71 A ERR2 -
474 FF71 4CFF A ERROR: LI R0,-1
475 FF72 0F00 A SFLG 7 ; NO-GO INDICATOR
476 FF73 0000 A HALT ; ERROR OCCURRED
477 FF74 2500 A JMP @.+1 ; REEXECUTE FROM BEGINNING
478 FF75 FF7E A .WORD ENTRY

```

```

479 FF76 ; .PAGE
480 FF76 0800 A TESTEND:SFLG 0 ; INDICATE GOOD RESULTS
481 FF77 A004 A PANEL: ST 0,4 ; SAVE REGISTERS
482 FF78 A405 A ST 1,5
483 FF79 A806 A ST 2,6
484 FF7A AC07 A ST 3,7
485 FF7B 9119 A LD 0,@STADD
486 FF7C 1F10 A BOC 15,EXIT
487 FF7D 0600 A ROUT: ROUT 0
488 FF7E 1C03 A WAIT: BOC 12,LA
489 FF7F 1D07 A BOC 13,LD
490 FF80 170A A BOC 7,EX
491 FF81 21FC A JMP WAIT
492 FF82 1CFF A LA: BOC 12,LA
493 FF83 0400 A RIN 0
494 FF84 3281 A RCPY 0,2
495 FF85 8200 A LD 0,(2)
496 FF86 21F6 A JMP ROUT
497 FF87 1DFE A LD: BOC 13,LD
498 FF88 0400 A RIN 0
499 FF89 A200 A ST 0,(2)
500 FF8A 21F2 A JMP ROUT
501 FF8B 17FF A EX: BOC 7,EX
502 FF8C 0400 A RIN 0
503 FF8D 4000 A EXIT: PUSH 0
504 FF8E 8004 A LD 0,4 ; RESTORE REGISTERS
505 FF8F 8405 A LD 1,5
506 FF90 8806 A LD 2,6
507 FF91 8C07 A LD 3,7
508 FF92 0880 A PFLG 0 ; INITIALIZE RESULT INDICATORS
509 FF93 0F80 A PFLG 7
510 FF94 0200 A RTS
511 FF95 FE00 A STADD: .WORD START
512 FF96 FF7E A .END ENTRY

```

***** 0 ERRORS IN ASSEMBLY *****

ROMDIX

BASPT1	BASPT2	BIT1	CYOV	D0	D16	D4	D5	D8	DD1
FF4C A	FF70 A	0004 A	000A A	FE78 A	FE7D A	FE79 A	FE7C A	FF04 A	FEF3 A
DD13	DD15	DD2	DDMIN1	ENTRY	ERR1	ERR2	ERROR	EX	EXIT
FEFF A	FEF9 A	FEFD A	FEF7 A	FFFE A	FE76 A	FF71 A	FF71 A	FF8B A	FF8D A
IENFL	INTEN	INTPTR	JMPLOC	JMPPTR	LA	LD	NZERO	NZRO	ODD
0001 A	0009 A	FF6C A	FF4B A	FF4A A	FF82 A	FF87 A	0005 A	000B A	0003 A
PANEL	PDISP	PTRADR	PZRO	R0	R1	R2	R3	ROMAD	ROUT
FF77 A	0005 A	FF3F A	0002 A	0000 A	0001 A	0002 A	0003 A	FE00 A	FF7D A
S6SUBR	S7SUBR	SEC2	SEC3	SEC4	SEC6	SEC6A	SEC6B	SEC6C	SEC7
FF3D A	FF6D A	FE7F A	FFD8 A	FF05 A	FF35 A	FF41 A	FF45 A	FF48 A	FF4D A
SEL	SELY	STADD	START	STKFL	SUBLOC	TESTEN	WAIT	X0F0A	X1000
000D A	0002 A	FF95 A	FE00 A	0008 A	FF40 A	FF76 A	FF7E A	FF03 A	FE7E A
X1FFFE	X4000	X7FFF	X8000	XFFFFFF	ZRO				
FEFB A	FE7B A	FE7A A	FF01 A	FEF5 A	0001 A				

565A



National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
(408) 732-5000
TWX: 910-339-9240

National Semiconductor Electronics SDNBHD
Batu Berendam
Free Trade Zone
Malacca, Malaysia
Telephone: 5171
Telex: NSELECT 519 MALACCA (c/o Kuala Lumpur)

National Semiconductor GmbH
D 808 Fuerstenfeldbruck
Industriestrasse 10
West Germany
Telephone: (08141) 1371
Telex: 27649

National Semiconductor (UK) Ltd.
Larkfield Industrial Estates
Greenock, Scotland
Telephone: (0475) 33251
Telex: 778 632

NS Electronics (PTE) Ltd.
No. 1100 Lower Delta Rd.
Singapore 3
Telephone: 630011
Telex: 21402

REGIONAL AND DISTRICT SALES OFFICES

ALABAMA

DIXIE DISTRICT OFFICE
3322 Memorial Parkway, S.W. #67
Huntsville, Alabama 35802
(205) 881-0622
TWX: 810-726-2207

ARIZONA

*ROCKY MOUNTAIN REGIONAL OFFICE
7349 Sixth Avenue
Scottsdale, Arizona 85251
(602) 945-8473
TWX: 910-950-1195

CALIFORNIA

*NORTH-WEST REGIONAL OFFICE
2680 Bayshore Frontage Road, Suite 112
Mountain View, California 94043
(415) 961-4740
TWX: 910-379-6432

NATIONAL SEMICONDUCTOR
DISTRICT SALES OFFICE
Valley Freeway Center Building
15300 Ventura Boulevard, Suite 305
Sherman Oaks, California 91403
(213) 783-8272
TWX: 910-495-1773

NATIONAL SEMICONDUCTOR
SOUTH-WEST REGIONAL OFFICE
17452 Irvine Boulevard, Suite M
Tustin, California 92680
(714) 832-8113
TWX: 910-595-1523

CONNECTICUT

AREA OFFICE
Commerce Park
Danbury, Connecticut 06810
(203) 744-2350

*DISTRICT SALES OFFICE
25 Sylvan Road South
Westport, Connecticut 06880
(203) 226-6833

INTERNATIONAL SALES OFFICES

AUSTRALIA

NS ELECTRONICS PTY. LTD.
Cnr. Stud Road & Mountain Highway
Bayswater, Victoria 3153
Australia
Telephone: 729-6333
Telex: 32096

CANADA

*NATIONAL SEMICONDUCTOR CORP.
1111 Finch Avenue West
Downsview, Ontario, Canada
(416) 635-9880
TWX: 610-492-1334

DENMARK

NATIONAL SEMICONDUCTOR
SCANDINAVIA
Vordingborggade 22
2100 Copenhagen
Denmark
Telephone: (01) 92-OBRO-5610
Telex: DK 6827 MAGNA

FLORIDA

*AREA SALES OFFICE
2721 South Bayshore Drive, Suite 121
Miami, Florida 33133
(305) 446-8309
TWX: 810-848-9725

CARIBBEAN REGIONAL SALES OFFICE

P.O. Box 6335
Clearwater, Florida 33518
(813) 441-3504

ILLINOIS

NATIONAL SEMICONDUCTOR
WEST-CENTRAL REGIONAL OFFICE
800 E. Northwest Highway, Suite 203
Mt. Prospect, Illinois 60056
(312) 394-8040
TWX: 910-689-3346

INDIANA

NATIONAL SEMICONDUCTOR
NORTH-CENTRAL REGIONAL OFFICE
P.O. Box 40073
Indianapolis, Indiana 46240
(317) 255-5822

KANSAS

DISTRICT SALES OFFICE
13201 West 82nd Street
Lenexa, Kansas 66215
(816) 358-8102

MARYLAND

CAPITAL REGIONAL SALES OFFICE
300 Hospital Drive, No. 232
Glen Burnie, Maryland 21061
(301) 760-5220
TWX: 710-861-0519

MASSACHUSETTS

*NORTH-EAST REGIONAL OFFICE
No. 3 New England, Exec. Office Park
Burlington, Massachusetts 01803
(617) 273-1350
TWX: 710-332-0166

ENGLAND

NATIONAL SEMICONDUCTOR (UK) LTD.
The Precinct,
Broxbourne, Hertfordshire
England
Telephone: Hoddesdon 69571
Telex: 267-204

FRANCE

NATIONAL SEMICONDUCTOR
FRANCE S.A.R.L.
28, Rue de la Redoute
92260-Fontenay-Aux-Roses
Telephone: 660-81-40
TWX: NSF 25956F

HONG KONG

*NATIONAL SEMICONDUCTOR
HONG KONG LTD.
9 Lai Yip Street
Kwun Tung, Kowloon
Hong Kong
Telephone: 3-458888
Telex: HX3866

MICHIGAN

*DISTRICT SALES OFFICE
23629 Liberty Street
Farmington, Michigan 48024
(313) 477-0400

MINNESOTA

DISTRICT SALES OFFICE
8053 Bloomington Freeway, Suite 101
Minneapolis, Minnesota 55420
(612) 888-3060
Telex: 290766

NEW JERSEY/NEW YORK CITY

MID-ATLANTIC REGIONAL OFFICE
301 Sylvan Avenue
Englewood Cliffs, New Jersey 07632
(201) 871-4410
TWX: 710-991-9734

NEW YORK (UPSTATE)

CAN-AM REGIONAL SALES OFFICE
104 Pickard Drive
Syracuse, New York 13211
(315) 455-5858

OHIO/PENNSYLVANIA/ W. VIRGINIA/KENTUCKY

EAST-CENTRAL REGIONAL OFFICE
Financial South Building
5335 Far Hills, Suite 214
Dayton, Ohio 45429
(513) 434-0097

TEXAS

*SOUTH-CENTRAL REGIONAL OFFICE
5925 Forest Lane, Suite 205
Dallas, Texas 75230
(214) 233-6801
TWX: 910-860-5091

WASHINGTON

DISTRICT OFFICE
300 120th Avenue N.E.
Building 2, Suite 205
Bellevue, Washington 98005
(206) 454-4600

JAPAN

*NATIONAL SEMICONDUCTOR JAPAN
Nakazawa Building**
1-19 Yotsuya, Shinjuku-Ku
Tokyo, Japan 160
Telephone: 03-359-4571
Telex: J 28592

SWEDEN

NATIONAL SEMICONDUCTOR SWEDEN
Sikvagen 17
13500 Tyreso
Stockholm
Sweden
Telephone: (08) 712-04-80

WEST GERMANY

*NATIONAL SEMICONDUCTOR GMBH
8000 Munchen 81
Cosimstrasse 4
Telephone: (0811) 915-027