

```

1  SIGMETICS 2650 ASSEMBLER VERSION 3 LEVEL 1
2
3  ADDR  B1 B2 B3      LABEL  OPCODE  OPERAND  COMMENTS
4
5          UN          EQU      0003      UNCONDITIONAL
6          EQ          EQU      0000      EQUAL CONDITION
7          GT          EQU      0001      GREATER THAN CONDITION
8          LT          EQU      0002      LESS THAN CONDITION
9          WC          EQU      0008      WITH CARRY BIT OF PSL
10         RS          EQU      0010      REGISTER SELECT BIT OF PSL
11         INCD        EQU      0010      CHARACTER INCREMENT CODE
12         BDL         EQU      0004      BEGINNING OF LINE IS ON CHAR 4
13         SPCD        EQU      0020      SPACECODE
14         CUCD        EQU      001C      CURSORCODE
15         SENS        EQU      0080      SENSE BIT OF PSU
16         FLAG        EQU      0040      FLAG BIT OF PSU
17         CUR1        EQU      17FE      CURSOR POINTER, BYTE 1
18         CUR2        EQU      17FF      CURSOR POINTER, BYTE 2
19         ADD1        EQU      17FA      FIRST ADDRESS BYTE
20         ADD2        EQU      17FB      SECOND ADDRESS BYTE
21         BKP1        EQU      17F4      FIRST BREAKPOINT ADDRESS
22         BKP2        EQU      17F5      SECOND BREAKPOINT ADDRESS
23         BPD1        EQU      17F6      BREAKPOINT BYTE 1 SAVE POS
24         BPD2        EQU      17F7      BREAKPOINT BYTE 2 SAVE POS
25         CUCD        EQU      003A      ASCII FOR COLON
26         SUMC        EQU      17F9      SUMCHECK STORAGE LOCATION
27         LENT        EQU      17EA      LENGTH STORAGE LOCATION
28         *
29         0000 04 00      INIT      LODI,R0  00      ***R0,R1,R2 DESTROYED
30         0002 F0          WRTD,R0          TD OFF
31         0003 04 02          LODI,R0  02      SETUP FOR LPSL
32         0005 93          LPSL          LOAD PSL=RS0,LOGICAL COMPARE
33         0006 76 40          PPSU      FLAG      FLAG=1
34         0008 05 10          LODI,R1  10      COUNTER
35         000A CD 17 FE      STRA,R1  CUR1      PUT CURSOR IN DISP AREA
36         000D 05 01          LODI,R1  01      COUNTER FOR ERASE=16 HEX
37         000F 1B 02          BCTR,UN  L1      JUMP OVER BREAKPOINT COMMANDS
38         0011 00 AC          ORG      0011      ORIGINATING ADDR FOR BRKPT RETN
39         0011 00 AC          ACON      BPSV      BREAKPOINT RETURN ADDRESS=BPSV
40         *
41         0013 3B 0F          L1      BSTR,UN  LFCH      ERASE LINES, ONE AT A TIME
42         0015 F9 7C          BDR,R1  L1      DECREMENT THE COUNTER, JNZ
43         0017 07 20          LODI,R3  SPCD      SPACECODE
44         0019 CF 97 FE      STRA,R3  ICUR1     STORE THE SPACE AT CURSOR POS.
45         001C 05 40          LODI,R1  40      SET BEGINNING OF LINE ADDRESS
46         001E CD 17 FF      STRA,R1  CUR2      STORE THAT
47         0021 1F 03 C0          BCTA,UN  COMD      GO AND GET A SUPERVISOR COMMAND
48         *
49         0024 07 20          LFCR     LODI,R3  SPCD      SPACE ***R0,R1,R3 DESTROYED
50         0026 CF 97 FE      STRA,R3  ICUR1     WRITE SPACE AT CURSOR POSITION
51         0029 0F 17 FF      LOUA,R3  CUR2      MOVE LOW ORDER CURSOR POS TO R3
52         002C 47 0F          ANDI,R3  0F      MASK OFF CHAR BITS
53         002E 87 01          ALOI,R3  01      ADD ONE TO LF COUNT
54         0030 47 0F          ANDI,R3  0F      MASK OFF ANY CARRY
55         0032 CF 17 FF      STRA,R3  CUR2      STORE NEW BYTE

```

56 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1

```

57
58 ADDP H1 H2 H3 LABEL OPCODE OPERAND COMMENTS
59
60 0035 07 10 LOUI,R3 10 START AT BEGINING OF LINE
61 0037 CF 17 FE STRA,R3 CUR1 STORE IN CURSOR POINTER LOCATION
62 003A 04 20 LOUI,R0 SPCD SPACECODE
63
64 003C CC 97 FE L2 STRA,R0 ICUR1 STORE A SPACE IN THE NEXT POS.
65 003F 07 10 LOUI,R3 INCD INCREMENT LINE POSITION
66 0041 8F 17 FF ADDA,R3 CUR2 TO THE RIGHT ONE POSITION.
67 0044 CF 17 FF STRA,R3 CUR2 STORE THIS NEW POSITION
68 0047 07 00 LOUI,R3 00 ADD ZERO TO CUR1
69 0049 77 08 PPSL WC ALL ADDITIONS NOW WITH CARRY
70 004B 8F 17 FE ADDA,R3 CUR1 ADD IN CARRY BIT FROM 17FF
71 004E CF 17 FE STRA,R3 CUR1 STORE BYTE WITH CARRY ADDED
72 0051 75 08 CPSL WC ALL ADDITIONS NOW WITHOUT CRY
73 0053 F7 05 TMI ,R3 05 TEST TO SEE IF FULL PAGE IS DONE
74 0055 98 65 BCFR,EQ L2 BRANCH BACK TO L2 IF NOT DONE
75 0057 0F 17 FF LODA,R3 CUR2 OTHERWISE, CLEAR CHAR BITS
76 005A 67 40 ICHI,R3 40 AND SET FIRST POS TO NUMBER 4
77 005C CF 17 FF STRA,R3 CUR2 STORE THIS NEW INFORMATION
78 005F 07 10 LOUI,R3 10 CODE TO FORCE BEGINNING OF LINE
79 0061 CF 17 FE STRA,R3 CUR1 STORE THIS CODE
80 0064 07 1C LOUI,R3 CUCD CURSOR CODE
81 0066 CF 97 FE STRA,R3 ICUR1 WRITE CURSOR
82 0069 17 RETC,UN RETURN
83
84 006A 02 MXOT LOUZ,R2 LOAD CHAR INTO R0
85 006B C1 STRZ,R1 AND R1
86 006C 51 RHR ,R1 ROTATE HIGH ORDER BITS INTO LOW
87 006D 51 RRR ,R1
88 006E 51 RRR ,R1
89 006F 51 RRR ,R1
90 0070 45 0F ANDI,R1 0F CLEAR NOW HIGH BITS
91 0072 0D 61 A6 LODA,R0 HTBL,R1 LOAD R0 WITH TABLE PLUS R1
92 0075 C3 STRZ,R3 LOAD R0 INTO R3
93 0076 3F 03 96 BSTA,UN WCHR WRITE FIRST CHAR
94 0079 46 0F ANDI,R2 0F MASK OFF HIGH BITS
95 007H 0E 61 A6 LODA,R0 HTBL,R2 LOAD SECOND CHAR INTO R0
96 007E C3 STRZ,R3 STORE IN R3
97 007F 3F 03 96 BSTA,UN WCHR WRITE IT
98 0082 17 RETC,UN RETURN
99
100 0083 3H 03 RETU BSTR,UN SAVR HAVE YOUR PROGRAM BRANCH HERE ..
101
102
103 0085 1F 03 C0 BCTA,UN COMD GET A NEW COMMAND
104
105 008B CC 17 EB SAVR STRA,R0 17EB STORE R0
106 008B 13 SPPL,R0 PSL INTO R0
107 008C CC 17 F2 STRA,R0 17F2 STORE PSL
108 008F 12 SP5U,R0 LOAD PSU INTO R0
109 0090 CC 17 F3 STRA,R0 17F3 STORE PSU
110 0093 75 10 CPSL R5 R5=0
111 0095 CD 17 EC STRA,R1 17EC STORE R1

```

112 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1

113	ADDR	B1	B2	B3	LABEL	OPCODE	OPERAND	COMMENTS
116	0098	CE	17	ED		STRA,R2	17ED	STORE R2
117	0098	CF	17	EE		STRA,R3	17EE	STORE R3
118	009E	77	10			PPSL	RS	RS=1
119	00A0	CD	17	EF		STRA,R1	17EF	STORE R4
120	00A3	CE	17	F0		STRA,R2	17F0	STORE R5
121	00A6	CF	17	F1		STRA,R3	17F1	STORE R6
122	00A9	75	10			CPSL	RS	RS=0
123	00AB	17				RETC,UN		RETURN
124					*			
125	00AC	3B	5A		HPSV	BSTR,UN	SAVR	BREAKPOINT SERVICE ROUTINE
126	00AE	3F	00	24		BSTA,UN	LFCR	DO LINEFEED
127	00B1	07	42			LODI,R3	A#B#	LOAD R3 WITH ASCII B
128	00B3	3F	03	96		HSTA,UN	WCHR	WRITE THE CHARACTER
129	00B6	07	50			LODI,R3	A#P#	LOAD R3 WITH ASCII P
130	00BB	3F	03	96		BSTA,UN	WCHR	WRITE THE CHARACTER
131					*			
132	00BB	3F	01	70	CLRB	BSTA,UN	WRBL	WRITE BLANK ENTRY PT. FOR CLEAR
133	00BE	0E	17	F4		LODA,R2	BKP1	LOAD R2 WITH 1ST BREAKPOINT ADDR
134	00C1	3F	00	6A		BSTA,UN	HXOT	WRITE FIRST BREAKPOINT ADDRESS
135	00C4	0E	17	F5		LODA,R2	BKP2	LOAD R2 WITH 2ND BKPT ADDR
136	00C7	3F	00	6A		BSTA,UN	HXOT	WRITE SECOND BREAKPOINT ADDRESS
137	00CA	07	01			LODI,R3	01	LOAD R3 WITH INCREMENTER
138	00CC	0C	17	F6		LODA,R0	BPU1	FIRST BYTE OF BREAKPOINT DATA
139	00CF	CC	97	F4		STRA,R0	IBKP1	STORE FIRST BYTE IN CORRECT LOC
140	00D2	0C	17	F7		LODA,R0	BPO2	LOAD R0 WITH SECOND BYTE
141	00D5	CF	F7	F4		STRA,R0	IBKP1,R3	STORE IN ADDRESS PLUS 1
142	00DB	1F	03	C0		BCTA,UN	COMD	GET COMMAND
143					*			
144	00DB	07	00		WRAD	LODI,R3	00	ALL REGS DESTROYED
145	00DD	3F	00	24		BSTA,UN	LFCR	GO TO BEGINNING OF NEXT LINE
146	00E0	0E	17	FA		LODA,R2	ADD1	FIRST ADDR BYTE STORED IN ADD1
147	00E3	3F	00	6A		BSTA,UN	HXOT	FIRST ADDRESS BYTE
148	00E6	0E	17	FB		LODA,R2	ADD2	SECOND ADDR BYTE STORED IN 17FB
149	00E9	3F	00	6A		BSTA,UN	HXOT	SECOND ADDRESS BYTE
150	00EC	3F	01	70		BSTA,UN	WRBL	WRITE A SPACE (BLANK)
151	00EF	17				RETC,UN		RETURN
152					*			
153	00F0	04	08		RUNT	LODI,R0	08	CODE TO TURN TD ON
154	00F2	F0				WRTO,R0		TURN THE TAPE DRIVE ON
155	00F3	3F	03	0F		BSTA,UN	KBIN	GET KBD DATA, RETURN IF ESCAPE
156	00F6	1B	78			BCTR,UN	RUNT	IF NOT ESCAPE, DO ROUTINE AGAIN
157					*			
158	00FB	3F	01	85	BKPT	BSTA,UN	ADDR	GET BREAKPOINT ADDRESS
159	00FB	0F	97	FA		LODA,R3	IADD1	GET BYTE AT BREAKPOINT ADDR
160	00FE	CF	17	F6		STRA,R3	BPU1	STORE BYTE AT BKPT ADDR
161	0101	07	01			LODI,R3	01	INCREMENT VALUE
162	0103	0F	F7	FA		LODA,R0	IADD1,R2	GET BYTE AT BREAKPOINT+1
163	0106	CC	17	F7		STRA,R0	BPU2	STORE SECOND BYTE IN SAVE AREA
164	0109	04	96			LODI,R0	96	ZBRR OPCODE
165	010B	CC	97	FA		STRA,R0	IADD1	STORE IN BKPT
166	010E	04	91			LODI,R0	91	INDIRECT ADDRESS, TO ADDR 11
167	0110	CF	F7	FA		STRA,R0	IADD1,R3	STORE IN BKPT+1

```

168 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1
169
170 ADDR B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
171
172 0113 0C 17 FA LOUA,R0 ADD1 STORE BKPT ADDR IN 17F4-5
173 0116 CC 17 F4 STRA,R0 BKP1 BKPT ADDR SAVE AREA, BYTE 1
174 0119 0C 17 FB LOUA,R0 ADD2 LOAD SECOND BYTE OF BKPT ADDR
175 011C CC 17 F5 STRA,R0 BKP2 STORE THIS BYTE IN SAVE AREA
176 011F 1F 03 C0 BCTA,UN COMD GO TO GET COMMAND ROUTINE
177
178 0122 3F 00 24 * INSP BSTA,UN LFCR SAVE ALL REG VALUES IN MEMORY
179 0125 07 52 LOUI,R3 A#R# PROMPTING CHAR
180 0127 3F 03 96 BSTA,UN WCHR WRITE THE CHARACTER
181 012A 3F 03 0F BSTA,UN KBIN GET REG VALUE
182 012D 3F 01 9A BSTA,UN TABL CONVERT THE INPUT TO HEX
183 0130 E7 06 CCM1,R3 08 COMPARE IT TO EIGHT
184 0132 10 01 22 BCTA,GT INSP IF GREATER THAN 8, JUMP BACK
185 0135 CF 17 FB STRA,R3 ADD2 STORE COMMAND IN TEMPS
186 0138 0F 77 EB LOUA,R0 17EB,R3 GET THE DATA IN REG SAVE AREA
187 013H C2 STRZ,R2 PUT IT IN R2 FOR HEXOUT ROUTINE
188 013C 3F 01 70 BSTA,UN WRBL WRITE A BLANK
189 013F 3F 00 6A BSTA,UN HX0T WRITE THE CONTENTS OF THE REG
190 0142 3F 03 0F BSTA,UN KBIN GET KEYBOARD DATA
191 0145 E7 43 CCM1,R3 A#C# COMPARE IT TO A #7C#7
192 0147 18 2D BCTR,EQ SETR IF C, BRANCH TO CHG REGISTER
193 0149 1F 01 22 BCTA,UN INSP OTHERWISE, DO AGAIN
194
195 014C 75 10 * SIHR CPSL RS RS=0
196 014E 0D 17 EC LOUA,R1 17EC LOAD R1
197 0151 0E 17 ED LOUA,R2 17ED LOAD R2
198 0154 0F 17 EE LOUA,R3 17EE LOAD R3
199 0157 77 10 PPSL RS RS=1
200 0159 0D 17 EF LOUA,R1 17EF LOAD R4
201 015C 0E 17 F0 LOUA,R2 17F0 LOAD R5
202 015F 0F 17 F1 LOUA,R3 17F1 LOAD R6
203 0162 0C 17 F3 LOUA,R0 17F3 LOAD PSU
204 0165 92 LPSU,R0 STORE R0 INTO PSU
205 0166 0C 17 F2 LOUA,R0 17F2 LOAD PSL
206 0169 93 LPSL,R0 STORE R0 INTO PSL
207 016A 0C 17 EB LOUA,R0 17EB LOAD R0
208 016D 1F 02 8B BCTA,UN RETN RETURN TO COMMAND ROUTINE
209
210 0170 07 20 * WRBL LOUI,R3 SPCD LOAD R3 WITH SPACECODE
211 0172 3F 03 96 BSTA,UN WCHR WRITE THE SPACE
212 0175 17 RETC,UN RETURN
213
214 0176 3B 7B * SETR BSTR,UN WRBL WRITE A BLANK
215 0176 3F 01 B6 BSTA,UN INHX GET A HEX DIGIT
216 017B 0D 17 FB LOUA,R1 ADD2 RETRIEVE REGISTER NUMBER
217 017E 03 LOUZ,R3 LOAD R0 WITH THE HEX DIGIT
218 017F CD 7, EB STRA,R0 17EB,R1 STORE NEW DATA IN CORRECT REG LO
219 0182 1F 01 22 BCTA,UN INSP JUMP BACK TO GET A NEW REG NO.
220
221 0185 3F 01 70 * ADDR BSTA,UN WRBL WRITE A SPACE
222 0188 07 41 LOUI,R3 A#A# ADDRESS PLACED IN 17FA-FB
223 * PU,R1,R3 DESTROYED

```

224 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1

ADDR	R1	R2	R3	LABEL	OPCODE	OPERAND	COMMENTS
225							
226	018A	3F	03		BSTA,UN	WCHR	WRITE A
227	018D	3F	01		BSTA,UN	INHX	GET HIGH ORDER BYTE
228	0190	CF	17		STRA,R3	ADD1	STORE HIGH ORDER BYTE
229	0193	3F	01		BSTA,UN	INHX	GET LOW ORDER BYTE
230	0196	CF	17		STRA,R3	ADD2	STORE LOW ORDER BYTE
231	0199	17			RETC,UN		RETURN
232				*			
233	019A	03		TABL	LODZ,R3		LOAD HEX CHAR INTO R0
234				*			BYTE STARTS AND ENDS IN R3,
235				*			R0 DESTROYED
236	019B	07	10		LODI,R3	10	R3=16D
237	019D	EF	41	DECR	CCMA,R0	HTBL,R3,-	COMPARE NEXT BYTE OF HEXTABLE
238	01A0	14			RETC,EQ		RETURN IF EQUAL
239	01A1	E7	00		CCMI,R3	00	SEE IF R3 IS DOWN TO 0
240	01A3	14			RETC,EQ		IF IT IS, RETURN
241	01A4	1B	77		BCTR,UN	DECR	JUMP BACK
242	01A6			*			
243				HTBL	ALIT	#0123456789ABCDEF#	LITERAL VALUES FOR HEX CONV.
244				*			LITERAL VALUES FOR HEX CONV.
245	01B6	3F	03	INHX	BSTA,UN	KBIN	GET FIRST HEX CHAR
246				*			ASSEMBLED BYTE IN R3
247				*			R0,R1 DESTROYED
248				*			CONVERT HEX TO BINARY
249	01B9	3B	5F		BSTR,UN	TABL	SHIFT HIGH ORDER BITS
250	01BB	D3			RHL,R3		
251	01BC	D3			RHL,R3		
252	01BD	D3			RHL,R3		
253	01BE	D3			RHL,R3		
254	01BF	04	F0		LODI,R0	F0	MASK FOUR HIGH BITS
255	01C1	43			ANDZ,R3		CLR LOW BITS
256	01C2	C1			STRZ,R1		STORE R1
257	01C3	3F	03		BSTA,UN	KBIN	GET SECOND CHAR
258	01C6	3B	52		BSTR,UN	TABL	CONVERT
259	01C6	01			LODZ,R1		RETRIEVE R1
260	01C9	47	0F		ANDI,R3	0F	MASK OFF HIGH BITS
261	01CB	63			ICHZ,R3		COMBINE HIGH AND LOW BITS
262	01CC	C3			STRZ,R3		MOVE BYTE TO R3
263	01CD	17			RETC,UN		RETURN
264				*			
265	01CE	3F	01	KHAM	BSTA,UN	ADDR	GET HEX ADDR
266	01D1	3F	00	NAOD	BSTA,UN	WRAD	WRITE ADDRESS
267	01D4	6E	97		LUDA,R2	IADD1	LOAD R2 WITH CONTENTS OF MEM POI
268				*			TO BY 17FA
269	01D7	3F	00		BSTA,UN	HXOT	DISPLAY DATA IN MEM LOCATION
270	01DA	3F	03		BSTA,UN	KBIN	GET CHAR FROM KBD
271	01DD	E7	43		CCMI,R3	A=C#	IF A C, JUMP
272	01DF	98	05		BCTR,EQ	NOCH	BRANCH IF COMPARE ISNT EQUAL
273	01E1	3B	53		BSTR,UN	INHX	GET NEW DATA
274	01E3	CF	97		STRA,R3	IADD1	PUT DATA INTO MEM LOCATION
275				*			
276	01E6	0F	17	NOCH	LUDA,R3	ADD2	LOAD LOW ORDER ADDR INTO R3
277	01E9	B7	01		ALDI,R3	01	ADD 1 TO THE ADDRESS
278	01EB	CF	17		STRA,R3	ADD2	STORE ADD2+1
279							

```

280 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1
281
282 ADDR B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
283
284 01EE 77 08 PPSL WC SET WC=1
285 01F0 20 ECRZ,R0 CLEAR R0
286 01F1 8C 17 FA AUDA,R0 ADD1 ADD HIGH ORDER ADDR BYTE TO CRY
287 01F4 CC 17 FA STRA,R0 ADD1 STORE THE NEW HIGH BYTE
288 01F7 75 08 CPSL WC RETURN TO OPERATIONS WITOUT CRY
289 01F9 18 56 BCTR,UN NAAD DO NEXT CHAR
290
291 *
291 01FB 07 08 TPOT LODI,R3 08 TD ON
292 01FD FJ WRD,R3 WRITE PORT TO TURN TAPE ON
293 01FE 3F 01 B5 BSTA,UN ADDR GET ADDRESS
294 0201 3F 01 70 BSTA,UN WRBL WRITE A BLANK
295 0204 07 4C LODI,R3 A≠L≠ PROMPTING CHARACTER FOR LENGTH
296 0206 3F 03 96 BSTA,UN WCHR WRITE THE CHARACTER
297 0209 3F 01 B6 BSTA,UN INHX GET LENGTH IN HEX
298 020C CF 17 EA STRA,R3 LENT STORE LENGTH
299 020F 20 ECRZ,R0 CLEAR R0
300 0210 CC 17 F9 STRA,R0 SUMC STORE 00 IN SUMCHECK CHAR
301 0213 07 3A LODI,R3 COCD COLON TO INDICATE START
302 0215 38 38 BSTR,UN SERO SEND CHAR OUT
303 0217 0F 17 FA LUDA,R3 ADD1 GET HIGH ADDR BYTE
304 021A 38 33 BSTR,UN SERO SEND IT OUT
305 021C 0F 17 FB LUDA,R3 ADD2 GET LOW ADDR BYTE
306 021F 38 2E BSTR,UN SERO SEND IT OUT
307 0221 0F 17 EA LUDA,R3 LENT GET LENGTH
308 0224 38 29 BSTR,UN SERO SEND IT OUT
309 0226 0F 17 F9 LUDA,R3 SUMC GET SUMCHECK BYTE
310 0229 38 24 BSTR,UN SERO SEND IT OUT
311 022B 20 ECRZ,R0 CLR R0
312 022C CC 17 F9 STRA,R0 SUMC STORE 00 IN SUMCHECK
313 022F EC 17 EA COMA,R0 LENT COMPARE LENGTH WITH 00
314 0232 18 15 BCTR,EQ ENDT IF 00, END ROUTINE
315 0234 05 FF LODI,R1 FF LOAD R1 WITH A FF
316 0236 3F 02 78 BSTA,UN D6 DO A SHORT DELAY
317
318 *
318 0239 0D B7 FA NEWL LUDA,R0 IADD1,R1,+ LOAD R0 WITH NEXT BYTE, INCR R1
319 023C C3 STRZ,R3 STORE THE BYTE IN R3
320 023D 38 10 BSTR,UN SERO SEND IT OUT
321 023F E0 17 EA COMA,R1 LENT COMPARE R1 WITH LENGTH
322 0242 98 75 BCFR,EQ NEWL IF EQUAL, END ROUTINE
323 0244 0F 17 F9 LUDA,R3 SUMC WHEN DONE, SEND OUT SUMCHECK
324 0247 38 06 BSTR,UN SERO SEND OUT THE SUMCHECK
325
326 *
326 0249 04 00 ENDT LODI,R0 00 TURN TAPE DRIVE OFF
327 024B F0 WRD,R0 WRITE IT TO THE PORT
328 024C 1F 03 C0 RCTA,UN COMD GET NEW COMMAND
329
330 *
330 024F 74 40 SERO CPSU 40 FLAG=0 (START BIT) **R0,R2 DEST.
331 0251 03 LODZ,R3 LOAD BYTE INTO R0
332 0252 2C 17 F9 EDRA,R0 SUMC FIGURE NEW SUMCHECK
333 0255 D0 RHL ,R0 FIGURE NEW SUMCHECK
334 0256 CC 17 F9 STRA,R0 SUMC STORE NEW SUMCHECK
335 0259 06 08 LODI,R2 08 BIT COUNTER

```

```

336 SIGNETICS 2650 ASSEMBLER VERSION 3 LEVEL 1
337
338 ADDR B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
339
340 *
341 0253 38 12 NXTB BSTR,UN D2 WAIT ONE BIT TIME
342 0250 53 RRR,R3 MOVE NEXT BIT INTO B7
343 025E 1A 04 HCTR,LT ONE B7=1
344 0260 74 40 CFSU 40 SET FLAG TO ONE
345 0202 18 02 BCTR,UN ZERO JUMP
346 *
347 0264 76 40 ONE PFSU 40 SET FLAG TO 1
348 *
349 0206 FA 73 ZERO BRR,R2 NXTB IF R2-1 IS NOT EQUAL TO 0, JUMP
350 0208 38 05 HSTR,UN D2 WAIT ONE BIT TIME
351 020A 76 40 PFSU 40 STOP BIT
352 026C 38 0A BSTR,UN D6 STOP BIT DELAY, 2 BIT TIMES
353 020L 17 RETC,UN RETURN
354 *
355 026F 04 08 D2 LOUI,R0 DB 1850--FULL BIT TIME DELAY
356 0271 F8 7E BRR,R0 $ DECREMENT R0 UNTIL 0
357 *
358 0273 04 08 D1 LOUI,R0 DB HALF BIT TIME DELAY
359 0275 F8 7E BRR,R0 $ 256 MORE
360 0277 17 RETC,UN RETURN
361 *
362 0278 38 75 D6 BSTR,UN D2 DO D2 THREE TIMES
363 027A 38 73 BSTR,UN D2 THIS EXTRA DELAY IS REQUIRED
364 027C 38 71 BSTR,UN D2 WHEN LOADING FROM TAPE
365 027E 17 RETC,UN RETURN
366 *
367 027F 3F 01 B5 EXEC BSTA,UN ADDR GET EXECUTE ADDRESS
368 0282 3F 03 0F BSTA,UN KBIN GET KEYBOARD DATA
369 0285 1F 01 4C BCTA,UN STRR LOAD REGISTERS INTO CPU
370 *
371 0288 1F 97 FA RETN BCTA,UN IADD1 BRANCH TO ADDRESS IN ADD1
372 *
373 0283 07 08 COLU LOUI,R3 08 LOAD R3 WITH TD ON CODE
374 0280 F3 WTD,R3 TURN TAPE DRIVE ON
375 *
376 WAIT ECRZ,R0 CLEAR R0
377 028F CC 17 F4 STRA,R0 SUMC STORE R0 IN SUMCHECK POSITION
378 0292 73 RLOU,R3 HEAD THE KBD PORT
379 0293 E7 18 CCM1,R3 18 COMPARE IT WITH ESCAPE CHAR
380 0295 1C 02 49 BCTA,EQ ENUT IF ESCAPE, RETURN
381 0298 3F 02 E9 BSTA,UN SERI GET SERIAL INPUT
382 0296 E7 3A CCM1,R3 COCD COMPARE IT TO A COLON
383 029D 98 0F BCFR,EQ WAIT IF NOT A COLON, JUMP BACK
384 029F 17 RETC,UN RETURN IF A COLON
385 *
386 02A0 3F 01 B5 VER1 BSTA,UN ADDR GET ADDRESS OF VERIFY
387 02A3 31 06 BSTR,UN COLU GET COLON FROM TAPE
388 02A5 3F 02 E9 BSTA,UN SERI GET FIRST SERIAL CHAR
389 02AB EF 17 FA CLMA,R3 ADD1 COMPARE IT TO FIRST ADDR BYTE
390 02AB 9C 03 7C BCPA,EQ ERHR IF NOT EQUAL, DO ERH
391 02AL 3F 02 E9 BSTR,UN SERI GET SECOND SERIAL INPUT

```

392 SIGNETICS 2050 ASSEMBLER VERSION 3 LEVEL 1

```

393
394 ADDR B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
395
396 02B1 EF 17 FB CUMA,R3 ADD2 COMPARE IT TO SECOND ADDR BYTE
397 02B4 9C 03 7C BCFA,EQ ERRR IF NOT EQUAL, DO ERR
398 02B7 3F 02 E9 BSTA,UN SERI GET THIRD SERIAL INPUT
399 02BA CF 17 EA STRA,R3 LENT STORE IT IN LENGTH
400 02BD 3F 02 E9 BSTA,UN SERI GET FOURTH (SUMCHECK) INPUT
401 02C0 0C 17 F9 L0DA,R0 SUMC LOAD SUMCHECK CHAR
402 02C3 9C 03 7C BCFA,EQ ERRR BRANCH TO ERR IF NOT ZERO
403 02C6 EC 17 EA CUMA,R0 LENT COMPARE THE LENGTH TO 0
404 02C9 1C 02 49 BCFA,EQ ENOT IF ZERO, END VERIFY
405 02CC 05 FF LUDI,R1 FF LOAD R1 WITH FF, COUNTER
406
407 02CE 3F 02 E9 * NEWC BSTA,UN SERI GET NEXT CHAR IN
408 02D1 03 L0DZ,R3 STORE IN R0
409 02D2 ED 07 FA CUMA,R0 IADD1,R1,+ COMPARE IT TO NEXT ADDR LOCATION
410 02D5 9C 03 7C BCFA,EQ ERRR IF NOT EQUAL, JUMP TO ERR
411 02D8 ED 17 EA CUMA,R1 LENT COMPARE R1 TO LENGTH
412 02DB 9B 71 BCFA,EQ NEWC IF EQUAL, END VERIFY ROUTINE
413 02DD 3F 02 E9 BSTA,UN SERI GET SUMCHECK CHAR
414 02E0 0C 17 F9 L0DA,R0 SUMC LOAD R0 WITH SUMCHK FOR DATA
415 02E3 9C 03 7C BCFA,EQ ERRR IF NOT ZERO, DO ERR
416 02E6 1F 02 49 BCTA,UN ENOT OTHERWISE END VERIFY ROUTINE
417
418 02E9 07 00 * SERI LUDI,R3 00 CLEAR R3, WILL BE DESTINATION
419 02EB 06 08 LUDI,R2 08 BIT COUNTER
420 02ED 04 00 * NOST TPSU R0 SENSE 0=START
421 02EF 10 7C BCTR,EQ NOST IF NO START BIT, JUMP BACK
422 02F1 3F 02 73 BSTA,UN 01 WAIT FOR MIDDLE OF START BIT
423 02F4 04 80 TPSU R0 TEST TO SEE IF STILL START BIT
424 02F6 10 75 BCTR,EQ NOST IF NOT, JUMP BACK
425
426 02F8 3F 02 6F * NEXT BSTA,UN D2 GO TO MIDDLE OF NEXT BIT
427 02FB 12 SPSU SAMPLE PSU INTO R0
428 02FC 44 80 ANDI,R0 R0 BLOCK EVERYTHING BUT SENSE
429 02FE 53 RRR,R3 ROTATE BYTE
430 02FF 63 ICRZ,R3 ADD NEXT BIT
431 0300 C3 STRZ,R3 MOVE R0 INTO R3
432 0301 F2 75 BRRR,R2 NEXT DECR R2, BRANCH IF NOT 0
433 0303 3F 02 73 BSTA,UN 01 DELAY 1/2 BIT TIME
434 0306 03 L0DZ,R3 LOAD BYTE INTO R0
435 0307 2C 17 F9 EUMA,R0 SUMC FIGURE SUMCHECK
436 030A 00 RHL,R0 FIGURE SUMCHECK
437 030B CC 17 F9 STRA,R0 SUMC STORE NEW SMCHECK
438 030E 17 RETC,UN
439
440 030F 73 * KBIN REUD,R3 READ_KEYBOARD, DATA-IN-R3
441 0310 F7 80 TMI,R3 R0 TEST B7 FOR A 1
442 0312 10 7B BCTR,EQ KBIN IF B7=1, JUMP BACK
443 0314 CF 17 FB STRA,R3 17FB STORE DATA IN TEMPORARY STORAGE
444 0317 3F 03 9B BSTA,UN WCHR WRITE THE CHAR ON THE SCREEN
445 031A 73 * L11 REUD,R3 HEAD THE KEYBOARD PORT AGAIN
446 031B F7 80 TMI,R3 R0 TEST STRUBE BIT
447 031E 9B 7B BCFA,EQ L11 BRANCH IF STILL 0

```



```

448 SIGNIFICS 2650 ASSEMBLER VERSION 3 LEVEL 1
449
450 ADDR B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
451
452 031F 0F 17 F8 LUDA,R3 17F8 RECAL DATA FROM TEMP STORAGE
453 0322 E7 18 CCM1,R3 18 COMPARE DATA FOR ESCAPE CODE
454 0324 JC 02 49 BCTA,EU ENDT BRANCH TO ENDT IF ESCAPE
455 0327 17 RETC,UN
456
457 0328 3F 02 8B * TPIN BSTA,UN COLD GET COLON
458 0328 3F 02 E9 BSTA,UN SERI GET FIRST BYTE IN
459 032L CF 17 FA STRA,R3 ADD1 STORE BYTE IN HIGH ORDER ADDRESS
460 0331 3F 02 E9 BSTA,UN SERI GET SECOND BYTE IN
461 0334 CF 17 F8 STRA,R3 ADD2 STORE BYTE IN LOW ORDER ADDR POS
462 0337 3F 02 E9 BSTA,UN SERI GET THIRD BYTE IN
463 033A CF 17 EA STRA,R3 LENT STORE IN LENGTH STORAGE
464 033D 3F 02 E9 BSTA,UN SERI GET FOURTH BYTE IN
465 0340 3F 00 0H BSTA,UN WRAD WRITE THE ADDRESS JUST RECEIVED
466 0343 0C 17 F9 LUDA,R0 SUMC LOAD R0 WITH THE SUMCHECK
467 0346 98 34 BCFR,0 ERRR BRANCH TO ERR IF SUMCK ISNT 0
468 0349 0C 17 EA CCM1,R0 LENT COMPARE THE LENGTH TO 0
469 034H 1C 02 49 BCTA,EU ENDT IF IT IS 0, BRANCH TO ENDT
470 034E 3F 00 24 BSTA,UN LFCH DO A LINEFEED
471 0351 05 0F LCU1,R1 FF LOAD R1 WITH FF
472
473 0353 3F 02 E9 * NADC BSTA,UN SERI GET BYTE FORM TAPE
474 0356 03 LCU2,R3 R3 INTO R0
475 0357 CD H7 FA STRA,R0 IADD1,R1,+ STORE BYTE JUST RECD IN NEXT POS
476 035A C2 STRZ,R2 CHAR INTO R2
477 035B CD 17 FC STRA,R1 17FC SAVE R1
478 035E 3F 00 0A BSTA,UN HXOT WRITE CHARACTER JUST READ
479 0361 0C 17 FC LUDA,R1 17FC RESTORE R1
480 0364 3F 01 7C BSTA,UN WRBL WRITE BLANK
481 0367 F5 0F TPI ,R1 0F TEST FOR 16TH CHARACTER
482 0369 3C 00 24 BSTA,EU LFCH IF 16TH, DO LINEFEED
483 036C 1D 17 EA CCM1,R1 LENT COMPARE R1 TO THE LENGTH
484 036F 9H 02 HCFR,EU NXTC IF EQUAL, GO ON
485 0371 3F 02 E9 BSTA,UN SERI GET LAST, SUMCHECK, BYTE
486 0374 0C 17 F9 LUDA,R0 SUMC LOAD R0 WITH SUMCHECK CHAR
487 0377 98 03 BCFR,EU ERRR WRITE ERROR MESSG IF CHECK NOT 0
488 0379 1F 03 2H BCTA,UN TPIN BRANCH BACK TO GET NEW BLOCK
489
490 037C 3F 01 70 * ERRR BSTA,UN WRBL WRITE #ERROR#
491 037F 07 45 LCU1,R3 A#E#
492 0381 36 13 BSTR,UN WCHR
493 0383 07 52 LCU1,R3 A#R#
494 0385 3H 0F BSTR,UN WCHR
495 0387 07 52 LCU1,R3 A#R#
496 0389 38 0B BSTR,UN WCHR
497 038H 07 4F LCU1,R3 A#O#
498 038D 3H 07 BSTR,UN WCHR
499 038F 07 52 LCU1,R3 A#R#
500 0391 3H 03 BSTR,UN WCHR
501 0393 1F 02 49 BCTA,UN ENDT
502
503 0396 27 FF FF * WCHR LCU1,R3 FF INVERT R3

```

```

504 SIGMETICS 2050 ASSEMBLER VERSION 3 LEVEL 1
505
506 ALCP B1 B2 B3 LABEL OPCODE OPERAND COMMENTS
507
508 039H F7 E0 TMI,R3 E0 TEST FOR CONTROL CHARACTER
509 039A 14 RETC,E0 RETURN IF CONTROL
510 039B 27 FF EORI,R3 FF INVERT R3 AGAIN
511 039D 47 3F ANVI,R3 3F SELECT CHAR GEN NUMBER 1
512
513 039F CF 97 FE 60 STRA,R3 ICUR1 STORE CHAR IN CURSOR POSITION
514 03A2 07 10 LUDI,R3 10 INCR CHARACTER CODE
515 03A4 0F 17 FF ALVA,R3 CUR2 MOVE CURSOR POSITION RIGHT ONE
516 03A7 CF 17 FF STRA,R3 CUR2 MOVE NEW LOW ORDER POSITION BACK
517 03AA 77 08 PPSL WC OPERATIONS WITH CARRY
518 03AC 20 ECHZ,R0 CLEAR R0
519 03AD 0C 17 FE ALVA,R0 CUR1 ADD CRY BIT TO CUR1, PUT IN R3
520 03B0 0C 17 FC STRA,R0 CUR1 STORE NEW HIGH CURSOR POSITION
521 03B3 F4 05 TMI,R0 05 TEST FOR BOTH CHARACTER
522 03B5 3C 00 24 BSTA,E0 LFCH IF THE END OF LINE, DO CHLF
523 03B8 07 1C LUDI,R3 CUCD LOAD R3 WITH CURSOR CODE
524 03BA CF 97 FE STRA,R3 ICUR1 WRITE NEW CURSOR
525 03BD 75 08 CPSC WC WC=0
526 03BF 17 RETC,UN
527
528 03C0 3F 00 24 CUMD BSTA,UN LFCH LINEFEED ***R3 DESTROYED
529 03C3 07 2E LUDI,R3 A#,#
530 03C5 3F 03 96 BSTA,UN WCHR WRITE THE PROMPTING CHARACTER
531 03C8 3F 03 0F BSTA,UN KBIN WAIT FOR COMMAND
532 03CB E7 43 CUMI,R3 A#C# CLEAR BREAKPOINT SETTING
533 03CD 1C 00 08 BCTA,E0 CLR#
534 03D0 E7 52 CUMI,R3 A#R# RUN TAPE UNIT
535 03D2 1C 00 F0 BCTA,E0 RUNT
536 03D5 E7 4C CUMI,R3 A#L# LOAD FROM TAPE
537 03D7 1C 03 28 BCTA,E0 TPIN
538 03DA E7 50 CUMI,R3 A#V# VERIFY TAPE CONTENTS
539 03DC 1C 02 40 BCTA,E0 VERI
540 03DF E7 45 CUMI,R3 A#E# EXECUTE FROM RAM
541 03E1 1C 02 7F BCTA,E0 EXEC
542 03E4 E7 41 CUMI,R3 A#A# ALTER OR DISPLAY RAM FROM KBD
543 03E8 1C 01 CE BCTA,E0 XRAM
544 03E9 E7 44 CUMI,R3 A#D# DUMP TO TAPE
545 03EB 1C 01 F4 BCTA,E0 TPOI
546 03EE E7 49 CUMI,R3 A#I# INSPECT CPU REGISTERS
547 03F0 1C 01 22 BCTA,E0 INSP
548 03F3 E7 42 CUMI,R3 A#B# BREAKPOINT SETTING
549 03F5 1C 00 Fd BCTA,E0 BKPT
550 03F8 07 3F LUDI,R3 3F IF INVALID, PRINT QUESTION MARK
551 03FA 3F 03 96 BSTA,UN WCHR WRITE THE CHARACTER
552 03FD 1B 41 BCTR,UN CUMD
553 TOTAL ERRORS= 0

```