

LOC OBJECT M STAT E LINE SOURCE LINE

```
1      TITLE 'EINSTEIN M.O.S.'
2 *
3 *
4 *****
5 *
6 *
7 *      CRYSTAL RESEARCH VERSION OF TATUNG HOME-COMPUTER
8 *      MACHINE OPERATING SYSTEM
9 *      =====
10 *
11 *      VERSION 1.2                                UPDATED 20/10/83
12 *      -----
13 *      (MODIFIED FROM XTAL MOS VERSION 1.13-BY Nigel Dedkin)
14 *      TATUNG SPECIFICATION 85-9647-6 ISSUE 2
15 *      -----
16 *
17 *****
18 *
19 *
20      EXTRN  PGTAB,PLDT,PLTXV,POINT,PNTXY,DRAWTO,POLYG,ORGO,FILL
21      EXTRN  CALADR,SETCLR,KTAB,ROSECT,WRSECT,COMMNO,HOMOSC,SETTRK
22      EXTRN  SETSEC,SETBUF,RDREC,WRREC,FDRST,GNSECT,SELDSC
23      ENTRY  BITS,CMDPRT,DATPRT,DBUF,DISC,DP8PTR,DSCPRT,ERENT,HSTACT
24 *
25 *
26 *
27      ENTRY  HSTDMA,HSTOSC,HSTSEC,HSTTRK,HSTWRT,NSECT,ROMEM,READOP
28      ENTRY  RSFLAG,RSTPRT,RWFLAG,SCTPRT,SCTSIZ,SECT,SEKHST
29      ENTRY  SIDFLG,STATUS,STEPR,TRAK,TRKPRT,TRKTBL,UNACNT,UNAOSC
30      ENTRY  UNASEC,WRTYPE,ZFDRST,ZHMOSEC,ZPINIT,ZRCPUV,ZRSECT,ZWSECT
31      ENTRY  CINC,CX,CY,DOTCNT,DOTON,FILMOD,FILSP,GCCLR,IMULT
32      ENTRY  ORGX,ORGY,RADX,RADY,SCRENT,STACK,STP,X1,X2,Y1,FSTACK
33      ENTRY  HSTBUF
34 *
35 *
36 XMO5  ASECT  RAM
37 *
38 *
39 MCAL  MACRO
40      RST    H'08'      MOS FUNCTION CALL ROUTINE
41      DATA  $1$      FUNCTION NUMBER
42      ENDM
43 *
44 *
45 DUTCH MACRO
46      RST    H'10'      CHARACTER OUTPUT ROUTINE
47      ENDM
48 *
49 *
50 FRM   MACRO
51      RST    H'18'      MESSAGE OUTPUT ROUTINE
52      ENDM
53 *
54 *
55 SREG  MACRO
56      RST    H'20'      SET VOP REGISTER ROUTINE
```

LOC OBJECT M STAT E LINE SOURCE LINE

57	ENDM		
58 *			
59 *			
60 VRIN	MACRO		
61	RST	H'28'	VIDEO RAM READ
62	ENDM		
63 *			
64 *			
65 VROUT	MACRO		
66	RST	H'30'	VIDEO RAM WRITE
67	ENDM		
68 *			
69 *			
70 CUR	EQU	H'04'	CONTROL CHARS.
71 DLR	EQU	H'06'	
72 BEL	EQU	H'07'	
73 BS	EQU	H'08'	
74 LF	EQU	H'0A'	
75 FF	EQU	H'0C'	
76 CR	EQU	H'0D'	
77 C40	EQU	H'0E'	
78 C32	EQU	H'0F'	
79 C80	EQU	H'10'	
80 DEL	EQU	H'19'	
81 INS	EQU	H'1A'	
82 ESC	EQU	H'1B'	
83 *			
84 *			
85 NTRAX	EQU	40	40 TRACKS/SIDE
86 NSECT	EQU	1	
86 NSECT	EQU	10	10 SECTORS/TRACK (DOUBLE DENSITY)
87 BUFSIZ	EQU	512	512 BYTES/SECTOR
88 STPRAT	EQU	1	DEFAULT STEPPING RATE (12ms/TRACK)
89 DEFMOD	EQU	H'C002'	DEFAULT VDP MODE (16k GRAPHICS II)
90 DFCOLR	EQU	H'F0'	DEFAULT COLOURS (WHITE/TRANSPARENT)
91 *			
92 *			
93 DEFVCT	EQU	H'0000'	DEFAULT 'BREAK' VECTOR
94 CALM	EQU	H'0008'	LOC. OF MCAL ROUTINE IN RAM (SAME AS ROM!)
95 VCGEN	EQU	H'1900'	VRAM CHAR GEN
96 VPATH	EQU	H'3300'	VRAM PATTERN NAME TABLE
97 VSPTBL	EQU	H'3800'	VRAM SPRITE ATTRIBUTE TABLE
98 VFHTBL	EQU	H'3880'	VRAM FN KEY TABLE
99 VTEXT	EQU	H'3C00'	VRAM TEXT POSITION TABLE
100 ROM2	EQU	H'4000'	2nd ROM START ADDRESS
101 R2INIT	EQU	ROM2+1	INITIALISE ROUTINE FOR 2nd ROM
102 R2XEC	EQU	ROM2+4	EXECUTION ROUTINE FOR 2nd ROM
103 SCRIP	EQU	H'FB00'	SCRATCH-PAD AREA
104 STACK	EQU	H'FCFF'	STACK TOP AT TOP OF SCRATCH-PAD STUFF
105 HSTBUF	EQU	H'FE00'	512-BYTE FILE DE-BLOCKING BUFFER
106 FSTACK	EQU	H'FFFD'	ALSO USED AS 'FILL' STACK
107 *			
108 *			
109 ZBAUD	EQU	H'81'	MODS FUNCTION-TABLE NUMBERS
110 ZREG	EQU	H'99'	
111 ZINIT	EQU	H'9A'	
112 ZSCON	EQU	H'9B'	

110 ZREG EQU H'99'  
 111 ZINIT EQU H'9A'  
 112 ZRSCAN EQU H'9B'

PMDS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.O.S.

MO512:5

00:17:44 PAGE: 3

LDC OBJECT M STAT E LINE SOURCE LINE

113 ZKEYIN EQU H'9C'  
 114 ZGETLN EQU H'9D'  
 115 ZOUTC EQU H'9E'  
 116 ZPQUT EQU H'9F'  
 117 ZSLQUT EQU H'A0'  
 118 ZSRLIN EQU H'A1'  
 119 ZRSECT EQU H'A2'  
 120 ZWSECT EQU H'A3'  
 121 ZRBLK EQU H'A4'  
 122 ZWBLK EQU H'A5'  
 123 ZCRLF EQU H'A6'  
 124 ZCRLFZ EQU H'A7'  
 125 ZSPACE EQU H'A8'  
 126 ZPR4HX EQU H'A9'  
 127 ZP2HXZ EQU H'AA'  
 128 ZPR2HX EQU H'AB'  
 129 ZFC4HX EQU H'AC'  
 130 ZFC2HX EQU H'AD'  
 131 ZDCMD EQU H'AE'  
 132 ZHMOSC EQU H'AF'  
 133 ZIGBLK EQU H'B0'  
 134 ZRDMM EQU H'B1'  
 135 ZRCPYU EQU H'B2'  
 136 ZRCPYD EQU H'B3'  
 137 ZMOUT EQU H'B4'  
 138 ZKSCAN EQU H'B5'

139 \*

140 \*

141 ZSLOSC EQU H'B6'  
 142 ZSETRK EQU H'B7'  
 143 ZSETSC EQU H'B8'  
 144 ZSETBF EQU H'B9'  
 145 ZRD128 EQU H'BA'  
 146 ZWR128 EQU H'BB'  
 147 ZZTIME EQU H'BC'  
 148 ZFORST EQU H'BD'  
 149 ZSYSRS EQU H'BE'  
 150 ZLOGO EQU H'BF'  
 151 ZPINIT EQU H'CO'  
 152 ZSREG EQU H'C1'  
 153 ZVRIN EQU H'C2'  
 154 ZVROUT EQU H'C3'  
 155 ZPLQT EQU H'C4'  
 156 ZPLTX EQU H'C5'  
 157 ZPOINT EQU H'C6'  
 158 ZPNTXY EQU H'C7'  
 159 ZDRWTD EQU H'C8'  
 160 ZPOLYG EQU H'C9'  
 161 ZORGCC EQU H'CA'  
 162 ZCALAD EQU H'CB'  
 163 ZSETCL EQU H'CC'  
 164 ZFILL EQU H'CD'  
 165 ZIMULT EQU H'CE'  
 166 ZPRM EQU H'CF'  
 167 ZVOUT EQU H'D0'  
 168 ZSCURS EQU H'D1'

DOS ROUTINE CALLS

REAL-TIME CLOCK STUFF

169 ZROM EQU	H'D2'	
170 ZIN80 EQU	H'D3'	
171 ZST80 EQU	H'D4'	
172 *		
173 *		
174 *		
175 *		ACTUAL SCRATCH-PAD AREA , COPIED FROM ROM ON START-UP
176 *		
177 *		INTERRUPT VECTORS FIRST
178 *		
179 ROMEM EQU	SCRIP+H'14'	LOCATION FOR MEMORY READ ROUTINE
180 XECEND EQU	SCRIP+H'17'	LOCATION FOR END OF "GO" ROUTINE
181 UPCOPY EQU	SCRIP+H'1A'	LOCATION FOR BLOCK MOVE
182 DHCOPY EQU	SCRIP+H'21'	LOCATION FOR BLOCK MOVE DOWN
183 MCXEC EQU	SCRIP+H'28'	LOCATION FOR EXTERNAL CALL INTO MOS
184 *		
185 *		
186 *		
187 *		REST OF SCRATCH-PAD INITIALISATION TABLE
188 *		
189 BRKVCT EQU	SCRIP+H'30'	
190 CLDVCT EQU	SCRIP+H'32'	
191 WRMVCT EQU	SCRIP+H'34'	
192 VDPMOD EQU	SCRIP+H'36'	
193 TCCLR EQU	SCRIP+H'38'	TEXT COLOURS
194 GCCLR EQU	SCRIP+H'39'	GRAPHICS COLOURS
195 FNVCT EQU	SCRIP+H'3A'	
196 FUVCT EQU	SCRIP+H'3C'	
197 FLAGS EQU	SCRIP+H'3E'	
198 CUSCDE EQU	SCRIP+H'3F'	CURSOR CHARACTER
199 PROMPT EQU	SCRIP+H'40'	PROMPT CHARACTER
200 BLINK EQU	SCRIP+H'41'	CURSOR BLINK RATE
201 RPTLNG EQU	SCRIP+H'42'	REPEAT DELAYS
202 RPTSHT EQU	SCRIP+H'43'	
203 SCTSIZ EQU	SCRIP+H'44'	SIZE OF DISC SECTOR IN 256-BYTE BLOCKS
204 PCFLGS EQU	SCRIP+H'45'	
205 LASTKY EQU	SCRIP+H'46'	
206 SCRNCD EQU	SCRIP+H'47'	
207 STORE EQU	SCRIP+H'48'	
208 ARGFLG EQU	SCRIP+H'49'	
209 VCOL EQU	SCRIP+H'4A'	
210 VROW EQU	SCRIP+H'4B'	
211 SCRLSZ EQU	SCRIP+H'4C'	
212 SCRCHT EQU	SCRIP+H'4E'	
213 XLEN EQU	SCRIP+H'4F'	
214 HSTDSC EQU	SCRIP+H'50'	DISC DRIVE (0-3)
215 HSTTRK EQU	SCRIP+H'51'	TRACK
216 HSTSEC EQU	SCRIP+H'52'	SECTOR
217 HSTDMA EQU	SCRIP+H'53'	DISC BUFFER AREA
218 RMFLAG EQU	SCRIP+H'55'	READ OR WRITE ?
219 STATUS EQU	SCRIP+H'56'	ERROR STATUS VALUE
220 ERCNT EQU	SCRIP+H'57'	ERROR COUNT
221 TRKTBL EQU	SCRIP+H'58'	STORE TRACK COUNT FOR UP TO 4 DISCS
222 BPVAL EQU	SCRIP+H'5C'	BREAKPOINT VALUE AND LOCATION
223 BPADR EQU	SCRIP+H'5D'	
224 VDPST EQU	SCRIP+H'5F'	VDP STATUS

LOC OBJECT M STAT E LINE SOURCE LINE

```

225 *
226 *
227 OLDI EQU SCRP+H'60' PROGRAM REGISTERS STORED FOR DEBUGGING
228 OLDIX EQU SCRP+H'61'
229 OLDIY EQU SCRP+H'63'
230 OLDSP EQU SCRP+H'65'
231 FILSP EQU OLDSP CAN USE "OLDSP" FOR "FILL" STACK
232 OLDAF1 EQU SCRP+H'67'
233 OLDSC1 EQU SCRP+H'69'
234 OLDDE1 EQU SCRP+H'68'
235 OLDHL1 EQU SCRP+H'6D'
236 OLDAF EQU SCRP+H'6F'
237 OLD8C EQU SCRP+H'71'
238 OLDDE EQU SCRP+H'73'
239 OLDHL EQU SCRP+H'75'
240 OLDPC EQU SCRP+H'77'
241 *
242 *
243 SAVHL EQU SCRP+H'79'
244 SAVSP EQU SCRP+H'78'
245 *
246 *
247 DISC EQU SCRP+H'7D' DOS VIRTUAL DISC INFORMATION
248 TRAK EQU SCRP+H'7E'
249 SECT EQU SCRP+H'7F' FOR 128-BYTE SECTOR IMPERSONATION
250 DBUF EQU SCRP+H'80'
251 SEKHST EQU SCRP+H'82' SEEK SHIFT-RIGHT COUNT
252 UNADSC EQU SCRP+H'83' LAST UNALLOC. DISC,
253 UNATRK EQU SCRP+H'84' TRACK,
254 UNASEC EQU SCRP+H'85' SECTOR.
255 HSTACT EQU SCRP+H'86' HOST ACTIVE FLAG
256 UNACNT EQU SCRP+H'87' UNALLOC.RECORD COUNT
257 HSTWRT EQU SCRP+H'88' HOST WRITTEN FLAG
258 READOP EQU SCRP+H'89' READ/WRITE FLAG (1=READ)
259 RSFLAG EQU SCRP+H'8A' READ SECTOR FLAG
260 WRTYPE EQU SCRP+H'8B' WRITE OP TYPE
261 *
262 *
263 TIME EQU SCRP+H'8C' TIME (SECONDS), MINS. AND HOURS
264 *
265 *
266 BUSY EQU SCRP+H'92' "BUSY" FLAG FOR INTERRUPTING DEVICES
267 FNKEY EQU SCRP+H'93' FUNCTION KEY POINTERS
268 FNPTR EQU SCRP+H'94'
269 X1 EQU SCRP+H'96' GRAPHICS SCRATCH-PAD
270 Y1 EQU SCRP+H'98'
271 X2 EQU Y1 X2 IS RIGHT END OF LINE IN FILL
272 ORGX EQU SCRP+H'9A'
273 ORGY EQU SCRP+H'9C'
274 CX EQU SCRP+H'9E'
275 CY EQU SCRP+H'A0'
276 RAOX EQU SCRP+H'A2'
277 RAOY EQU SCRP+H'A4'
278 CINC EQU SCRP+H'A6'
279 DOTON EQU SCRP+H'A8'
280 DOTOFF EQU SCRP+H'A9'

```

LOC OBJECT M STAT E LINE SOURCE LINE

```

281 DDTDN2 EQU   SCRP+H'AA'
282 DDTDF2 EQU   SCRP+H'AB'
283 DDTCHT EQU   SCRP+H'AC'
284 FILMOD EQU   SCRP+H'AD'
285 STP EQU      SCRP+H'AE'   POINTER TO CURRENT LEVEL OF "FILL" STACK
286 *
287 *
288 STEPR EQU     SCRP+H'BO'   STEPPING RATE (DEFAULT=1 i.e. 12ms)
289 SIDFLG EQU   SCRP+H'B1'   "DOUBLE-SIDE" FLAG (DEFAULT=0 i.e. SINGLE)
290 SCHKEY EQU   SCRP+H'B2'   KEY SCAN LOCATION
291 DPDPTR EQU   SCRP+H'B3'   DISC PARAMETER TABLE POINTER
292 KCNT EQU     SCRP+H'B5'   REPEAT KEY COUNT
293 *
294 *
295 BUFFER EQU    SCRP+H'B8'   SPACE FOR INPUT BUFFER (NCOLS)
296 *
297 *
298 VDPBUF EQU    SCRP+H'E0'   VDP BUFFER FOR COPYING TO/FROM VDP
299 *
300 *
301 BANKZ EQU     SCRP+H'109'   TIME INTERRUPT SERVICE ROUTINE
302 *
303 *
304 *   PORT DESIGNATIONS
305 *
306 *
307 RSTPRT EQU    H'00'   PSG/FDC RESET
308 PSGREG EQU    H'02'   PSG(RY-3-8910): REGISTER POINTER
309 PSGDAT EQU    H'03'   DATA
310 PCID EQU      H'10'   PCI(8251): DATA
311 PCIC EQU      H'11'   CONTROL/STATUS
312 CNDPRT EQU    H'18'   FDC(FD1770): COMMAND/STATUS
313 TRKPRT EQU    H'19'   TRACK REGISTER
314 SCTPRT EQU    H'1A'   SECTOR
315 DATPRT EQU    H'1B'   DATA
316 AUXREG EQU    H'20'   KBD INTERRUPT MASK / VARIOUS STATUS
317 ADCMSK EQU    H'21'   ADC
318 ALPHA EQU     H'22'   ALPHA-LOCK LED
319 DSCPRT EQU    H'23'   DRIVE SELECT
320 BANKSW EQU    H'24'   MEMORY BANK SWITCH
321 FIRMSK EQU    H'25'   FIRE BUTTON INTERRUPT MASK
322 CTC0 EQU      H'28'   CTC(280ACTC): CHANNEL 0
323 CTC1 EQU      H'29'   " 1
324 CTC2 EQU      H'2A'   " 2
325 CTC3 EQU      H'2B'   " 3
326 PIQAD EQU     H'30'   PIQ(230APIQ): PORT A DATA
327 PIQAC EQU     H'31'   " CONTROL
328 PIQBD EQU     H'32'   PORT B DATA
329 PIQBC EQU     H'33'   " CONTROL
330 DP80 EQU      H'40'   40H-47H ARE 80 COLUMN VIDEO RAM
331 RG6845 EQU    H'48'   6845 REGISTER ADDRESS PORT
332 DT6845 EQU    H'49'   6845 REGISTER DATA PORT
333 IP80 EQU      H'4C'   80 COLUMN INPUT PORT
334 *
335 *
336 *   ORG 0

```

000000

LOC OBJECT M STATE LINE SOURCE LINE

		337 *				
		338 *				
		339 *			RESTART VECTORS USED BY MONITOR ROUTINES	
		340 *				
		341 *				
000000 210080	1	342	LD	HL,H'8000'	CLEAR RAM AND INITIALISE	(RST H'00')
000003 110180	2	343	LD	DE,H'8001'		
000006 1838	3	344	JR	CLRRAM		
		345 *				
		346 *				
000008 C3C20F	4	347	JP	FXM0H	M05/005 FUNCTION JUMP	(RST H'08')
000008		348	RESB	5	(MCAL)	
		349 *				
		350 *				
000010 C31A0F	5	351	JP	OUTC	CHARACTER OUTPUT FROM A	(RST H'10')
000013		352	RESB	5	(OUTCH)	
		353 *				
		354 *				
000018 E3	6	355	PRMZ	EX (SP),HL	PRINT MESSAGE AFTER CALL	(RST H'18')
000019 CD3F0F	7	356	PRM1	CALL PRMCHR	(PRM)	
00001C 28FB	8	357	JR	Z,PRM1		
00001E E3	9	358	EX	(SP),HL		
00001F C9	10	359	RET			
		360 *				
		361 *				
000020 F5	11	362	SREG2	PUSH AF	WRITE VRAM ADDR GIVEN BY BC	(RST H'20')
000021 79	12	363	LD	A,C	(SREG)	
000022 0309	13	364	OUT	(H'09'),A		
000024 78	14	365	LD	A,8		
000025 C3980C	15	366	JP	SREG2		
		367 *				
		368 *				
000028 CD800C	16	369	VRINH	CALL SREG1	READ VRAM ADDR GIVEN BY BC TO A	(RST H'28')
000028 0808	17	370	IN	A,(H'08')	(VRIN)	
00002D C9	18	371	RET			
00002E		372	RESB	2		
		373 *				
		374 *				
		375	VROUTZ	SREG	WRITE A TO VRAM ADDR IN BC	(RST H'30')
000030 E7	19	+	RST	H'20'	SET VOP REGISTER ROUTINE	
		+	ENDM			
000031 0303	20	376	OUT	(H'08'),A	(VROUT)	
000033 C9	21	377	RET			
		378 *				
		379 *				
000034 01020408		380	BITS	DATA 1,2,4,8	DEFINE BITS FOR DRIVE SELECT	
		381 *				
		382 *				
000038 00	22	383	NOP		BREAK-POINT ROUTINE	(RST H'38')
000039 00	23	384	NOP		SPACE FOR BANK CHANGE	
00003A C30402	24	385	JP	BREAK		
00003D 62		386	DATA	H'62'		
00003E		387	RESB	2		
		388 *				
		389 *				
		390 *			COLD START INITIALISATION (0040H)	

LOC OBJECT M STAT E LINE SOURCE LINE

```

391 *
392 *
000040 01FF7F 25 393 CLR RAM LD 3C,H'7FFF' RAM CLEAR ROUTINE - START WITH NON-BANKED
000043 71 26 394 LD (HL),C AREA (ALLOW FOR 0-7FFF BANKED)
000044 ED80 27 395 LDIR
000046 210030 28 396 LD HL,H'8000' NOW ZAP LOWER 32K
000049 44 29 397 LD B,H
00004A ED80 30 398 LDIR
00004C 2100FB 31 399 LD HL,SCRIP
00004F 1101FB 32 400 LD DE,SCRIP+1
000052 34 33 401 INC (HL) ZERO SCRATCH-PAD RAM
000053 01FF04 34 402 LD BC,H'04FF'
000056 ED80 35 403 LDIR
000058 21F00F 36 404 LD HL,SCINIT
00005B 1100FB 37 405 LD DE,SCRIP
00005E 0E46 38 406 LD C,ZDFVCT-SCINIT INIT. SCRATCH-PAD
000060 ED80 39 407 LDIR
000062 213510 40 408 LD HL,ZDFVCT
000065 110000 41 409 LD DE,DEFVCT
000068 0E08 42 410 LD C,ZBANKZ-ZDFVCT
00006A ED80 43 411 LDIR COPY IN MCAL RAM ROUTINE
00006C 1109FC 44 412 LD DE,BANKZ
00006F 0E84 45 413 LD C,ZEND-ZBANKZ FOR PHILIPS!!
000071 ED80 46 414 LDIR COPY INTERRUPT ROUTINES
000073 218CF8 47 415 LD HL,TIME
000076 0606 48 416 LD B,6
000078 3630 49 417 TINIT LD (HL),0' ZERO TIME LOCATIONS
00007A 23 50 418 INC HL
00007B 10FB 51 419 DJNZ TINIT
00007D 31FFFC 52 420 INITH LD SP,STACK SET STACK AT TOP OF MEMORY
000080 3EF8 53 421 LD A,H'FB' SET INTERRUPT REGISTER TO HIGH BYTE OF
000082 ED47 54 422 LD I,A VECTOR AREA
000084 3260FB 55 423 LD (OLDI),A AND ITS COPY IN REGISTER STORAGE AREA
000087 ED7365FB 56 424 LD (OLDSP),SP SET UP SP IN STORAGE AREA
000088 3E01 57 425 LD A,STPRAT
00008D 32B0FB 58 426 LD (STEPR),A
000090 ED5E 59 427 IM 2 SET STEPPING RATE FOR FDC
000092 AF 60 428 XOR A MODE 2 INTERRUPTS
000093 0323 61 429 OUT (CTCO),A SET CTC INTERRUPT VECTOR (0-7)
430 MCAL ZZTIME
000095 CF 62 + RST H'08' MOS FUNCTION CALL ROUTINE
000096 8C + DATA ZZTIME FUNCTION NUMBER
+ ENDM
000097 3ECE 63 431 LD A,H'CE'
000099 0311 64 432 OUT (PCIC),A INIT. PCI FOR /16, 8-BITS, 2 STOP BITS
00009B 3E27 65 433 LD A,H'27' NO PARITY
00009D 0311 66 434 OUT (PCIC),A
00009F 210800 67 435 LD HL,8 SET 9600 BAUD
0000A2 54 68 436 LD D,H
0000A3 5C 69 437 LD E,H
438 MCAL ZBAUD
0000A4 CF 70 + RST H'08' MOS FUNCTION CALL ROUTINE
0000A5 81 + DATA ZBAUD FUNCTION NUMBER
+ ENDM
0000A6 3E10 71 439 LD A,H'10'
0000A8 0331 72 440 OUT (PIOAC),A SET UP PIO INTERRUPTS - PORT A

```

LOC OBJECT M STAT E LINE SOURCE LINE

0000AA	3E12	73	441	LD	A,H'12'	
0000AC	D333	74	442	OUT	(PI0BC),A	- PORT B
0000AE	3E0F	75	443	LD	A,H'0F'	
0000B0	D331	76	444	OUT	(PI0AC),A	MODE 0 (OUTPUT) FOR PORT A
0000B2	3E87	77	445	LD	A,H'87'	
0000B4	D331	78	446	OUT	(PI0AC),A	ENABLE INTERRUPT ON PORT A
0000B6	3E08	79	447	LD	A,8	SET UP VDP REGISTERS
0000B8	0680	80	448	LD	B,H'80'	
0000BA	21E30F	81	449	LD	HL,ITAB	
0000BD	4E	82	450	LD	C,(HL)	
0000BE	CD8D0C	83	451	CALL	SREG1	
0000C1	23	84	452	INC	HL	
0000C2	04	85	453	INC	B	
0000C3	3D	86	454	DEC	A	
0000C4	20F7	87	455	JR	NZ,INH1	
0000C6	018038	88	456	LD	BC,VFNTBL	INIT. FN KEY TABLE
			457	SREG		
0000C9	E7	89	+	RST	H'20'	SET VDP REGISTER ROUTINE
			+	ENDM		
0000CA	018000	90	458	LD	BC,H'80'	
0000CD	59	91	459	LD	E,C	
0000CE	CD860C	92	460	CALL	CLR	
			461 *			
			462 *			
			463 MOS0	MCAL	ZSYSRS	RESET DISPLAY, PSG, & FDC
0000D1	CF	93	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0000D2	BE		+	DATA	ZSYSRS	FUNCTION NUMBER
			+	ENDM		
			464	MCAL	ZIN80	
0000D3	CF	94	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0000D4	D3		+	DATA	ZIN80	FUNCTION NUMBER
			+	ENDM		
			465	MCAL	ZLOG0	PRINT LOG0
0000D5	CF	95	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0000D6	BF		+	DATA	ZLOG0	FUNCTION NUMBER
			+	ENDM		
0000D7	FB	96	466	EI		
0000D8	3A0040	97	467	LD	A,(ROM2)	SEE IF 2nd ROM PRESENT
0000DB	87	98	468	OR	A	
0000DC	CC0140	99	469	CALL	Z,R2INIT	IF IT IS, CALL INITIALISE ROUTINE IN-ROM2
0000DF	3E01	100	470	LD	A,1	SELECT DRIVE A
0000E1	D323	101	471	OUT	(DSCPR),A	
			472	MCAL	ZHMDSC	BOOT DISC IN DRIVE A IF AVAILABLE
0000E3	CF	102	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0000E4	AF		+	DATA	ZHMDSC	FUNCTION NUMBER
			+	ENDM		
0000E5	CAS002	103	473	JP	Z,BOOT	
			474	MCAL	ZFDRST	
0000E8	CF	104	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0000E9	8D		+	DATA	ZFDRST	FUNCTION NUMBER
			+	ENDM		
			475	PRM		
0000EA	DF	105	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
			+	ENDM		
0000EB	000A2020		476	DATA	CR,LF,'	Insert disc in drive 0 and'
	20202020					

LOC	OBJECT	M	STAT	E	LINE	SOURCE	LINE
	20496E73						
	65727420						
	64697363						
	20696E20						
	64726976						
	65203020						
	616E64						
00010E	000A2020		477		DATA	CR,LF,' press Ctrl-BREAK to load'	
	20202020						
	20207072						
	65737320						
	4374726C						
	2D425245						
	41482074						
	6F206C6F						
	6164						
000130	000A0A54		478		DATA	CR,LF,LF,'TATUNG/Xtal MOS 1.2 (C) 1983 198',H'B4'	
	4154554E						
	472F5874						
	616C204D						
	4F532031						
	2E322020						
	20202020						
	20202843						
	29203139						
	38332031						
	393884						
00015B	180A	106	479		JR	MOS2	
			480 *				
			481 *				
			482 *				
			483 *				
			484 *				
			485 MOS1		PRM		
00015D	DF	107	+		RST	H'18' MESSAGE OUTPUT ROUTINE	
			+		ENDM		
00015E	000A4D4F		486		DATA	CR,LF,'MOS 1.',H'B2'	
	5320312E						
	B2						
			487 MOS2		PRM		
000167	DF	108	+		RST	H'18' MESSAGE OUTPUT ROUTINE	
			+		ENDM		
000168	000A5265		488		DATA	CR,LF,'Ready',CR,LF+H'80'	
	6164790D						
	8A						
			489 *				
			490 *				
			491 *				
			492 *				
			493 *				
						NORMAL ENTRY TO MONITOR FROM OUTSIDE	
000171	31FFFC	109	494 MONITR	LD	SP,STACK		
000174	FB	110	495	EI			
			496	MCAL	ZCRLFZ		
000175	CF	111	+	RST	H'08'	MOS FUNCTION CALL ROUTINE	
000176	A7		+	DATA	ZCRLFZ	FUNCTION NUMBER	
			+	ENDM			

LOC OBJECT N STAT E LINE SOURCE LINE

000177	3A40FB	112	497	MONO	LD	A,(PROMPT)	SHOW PROMPT
			498		OUTCH		
00017A	D7	113	+		RST	H'10'	CHARACTER OUTPUT ROUTINE
			+		ENDM		
00017B	1183FB	114	499		LD	DE,BUFFER	
			500		MCAL	ZGETLN	
00017E	CF	115	+		RST	H'03'	MDS FUNCTION CALL ROUTINE
00017F	9D		+		DATA	ZGETLN	FUNCTION NUMBER
			+		ENDM		
000180	1A	116	501		LD	A,(DE)	GET COMMAND LETTER
000181	87	117	502		OR	A	
000182	28F3	118	503		JR	Z,MONO	DO AGAIN IF CR
000184	CDDC01	119	504		CALL	MON6	
000187	13	120	505		INC	DE	
000188	E65F	121	506		AND	H'5F'	IGNORE SHIFT
00018A	D641	122	507		SUB	'A'	
00018C	3867	123	508		JR	C,ERROR	
00018E	218202	124	509		LD	HL,FMTBL	
000191	87	125	510		ADD	A,A	GET VECTOR
000192	4F	126	511		LD	C,A	
000193	0600	127	512		LD	B,H'00'	
000195	09	128	513		ADD	HL,BC	
000196	46	129	514		LD	B,(HL)	
000197	23	130	515		INC	HL	
000198	66	131	516		LD	H,(HL)	
000199	68	132	517		LD	L,B	
00019A	013404	133	518		LD	BC,HEX	
00019D	ED42	134	519		SBC	HL,BC	
00019F	09	135	520		ADD	HL,BC	SPECIAL FOR HEX (H) COMMAND
0001A0	2005	136	521		JR	NZ,MON1	
0001A2	01F201	137	522		LD	BC,MONOUN	
0001A5	C5	138	523		PUSH	BC	
0001A6	E9	139	524		JP	(HL)	
0001A7	E5	140	525	MON1	PUSH	HL	
0001A8	D9	141	526		EXX		
0001A9	210000	142	527		LD	HL,H'0000'	
0001AC	54	143	528		LD	D,H	
0001AD	5D	144	529		LD	E,L	
0001AE	44	145	530		LD	B,H	
0001AF	4D	146	531		LD	C,L	
0001B0	D9	147	532		EXX		
0001B1	3E03	148	533		LD	A,H'03'	
0001B3	3249FB	149	534	MON2	LD	(ARGFLG),A	
			535		MCAL	ZIGBLK	
0001B6	CF	150	+		RST	H'03'	MDS FUNCTION CALL ROUTINE
0001B7	80		+		DATA	ZIGBLK	FUNCTION NUMBER
			+		ENDM		
0001B8	281A	151	536		JR	Z,MONEX	
			537		MCAL	ZFC4HX	
0001BA	CF	152	+		RST	H'03'	MDS FUNCTION CALL ROUTINE
0001BB	AC		+		DATA	ZFC4HX	FUNCTION NUMBER
			+		ENDM		
0001BC	E5	153	538		PUSH	HL	
0001BD	D9	154	539		EXX		
0001BE	3A49FB	155	540		LD	A,(ARGFLG)	
0001C1	FE03	156	541		CP	H'03'	

LOC OBJECT M STAT E LINE SOURCE LINE

0001C3	2001	157	542	JR	NZ,M0N3	
0001C5	E1	158	543	POP	HL	
0001C6	FE02	159	544	M0N3	CP	H'02'
0001C8	2001	160	545	JR	NZ,M0N4	
0001CA	D1	161	546	POP	DE	
0001C8	FE01	162	547	M0N4	CP	H'01'
0001CD	2001	163	548	JR	NZ,M0N5	
0001CF	C1	164	549	POP	BC	
0001D0	D9	165	550	M0N5	EXX	
0001D1	3D	166	551	DEC	A	
0001D2	20DF	167	552	JR	NZ,M0N2	
0001D4	87	168	553	M0NEX	OR	A
0001D5	21F201	169	554	LD	HL,M0NDUN	
0001D8	E3	170	555	EX	(SP),HL	
0001D9	E5	171	556	PUSH	HL	
0001DA	D9	172	557	EXX		
0001D8	C9	173	558	RET		
			559 *			
0001DC	F5	174	560	M0N6	PUSH	AF
0001D0	2145FB	175	561	LD	HL,PCFLG5	
0001E0	C856	176	562	BIT	2,(HL)	
0001E2	230C	177	563	JR	Z,M0N8	
0001E4	0650	178	564	LD	B,H'50'	
0001E6	2183FB	179	565	LD	HL,BUFFER	
0001E9	3E7F	180	566	M0N7	LD	A,H'7F'
0001EB	A6	181	567	AND	(HL)	
0001EC	77	182	568	LD	(HL),A	
0001ED	23	183	569	INC	HL	
0001EE	10F9	184	570	DJNZ	M0N7	
0001F0	F1	185	571	M0N8	POP	AF
0001F1	C9	186	572	RET		
			573 *			
			574 *			
0001F2	027101	187	575	M0NDUN	JP	NC,M0N1R
			576	ERROR	PRM	
0001F5	DF	188	+	RST	H'18'	
			+	ENDM		
0001F6	20457272		577	DATA	'Erro',H'F2'	
	6FF2					
			578	ERR1	PRM	
0001FC	DF	189	+	RST	H'18'	
			+	ENDM		
0001FD	2107008A		579	DATA	'!',BEL,CR,LF+H'80'	
000201	C37101	190	580	JP	M0N1R	
			581 *			
			582 *			
			583 *			
			584 *			
			585 *			
000204	ED7365FB	191	586	BREAK	LD	(OLDSP),SP
000208	F3	192	587	DI		SAVE STACK
000209	3177FB	193	588	LD	SP,OLDPC	NO INTERRUPTIONS
00020C	E5	194	589	PUSH	HL	
00020D	D5	195	590	PUSH	DE	
00020E	C5	196	591	PUSH	BC	
00020F	F5	197	592	PUSH	AF	

FETCH PARAMETERS

EXECUTE INSTRUCTION VIA RETURN

ROUTINE TO REMOVE MSB,S FROM BUFFER

JUMP IF NOT 30 COL

ZERO MSB ON ALL BUFFER

MESSAGE OUTPUT ROUTINE

MESSAGE OUTPUT ROUTINE

BREAK - DO BREAKPOINT STUFF

NORMAL REGISTERS ALL SAVED

LOC OBJECT # STAT E LINE SOURCE LINE

000210	2A65FB	198	593	LD	HL,(OLDSP)	
			594	MCAL	ZROMEM	GET OLD PC (MAY BE IN LOW RAM)
000213	CF	199	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000214	B1		+	DATA	ZROMEM	FUNCTION NUMBER
			+	ENDM		
000215	5F	200	595	LD	E,A	N.B. USING ALT. REGISTER AREA AS
000216	23	201	596	INC	HL	TEMP. STACK
			597	MCAL	ZROMEM	
000217	CF	202	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000218	B1		+	DATA	ZROMEM	FUNCTION NUMBER
			+	ENDM		
000219	57	203	598	LD	D,A	
00021A	23	204	599	INC	HL	
000218	2265FB	205	600	LD	(OLDSP),HL	RESET STACK TO CORRECT POSITION
00021E	18	206	601	DEC	DE	
00021F	ED5377FB	207	602	LD	(OLDPC),DE	SAVE BREAKPOINT LOCATION
000223	D9	208	603	EXX		
000224	08	209	604	EX	AF,AF'	
000225	E5	210	605	PUSH	HL	
000226	05	211	606	PUSH	DE	
000227	C5	212	607	PUSH	BC	
000228	F5	213	608	PUSH	AF	ALT REGISTER SAVED
000229	3165FB	214	609	LD	SP,OLDSP	
00022C	FDE5	215	610	PUSH	IV	
00022E	0DE5	216	611	PUSH	IX	
000230	2A5FFB	217	612	LD	HL,(VDPST)	
000233	ED57	218	613	LD	A,I	
000235	F5	219	614	PUSH	AF	SPECIAL REGISTERS SAVED
000236	7D	220	615	LD	A,L	
000237	325FFB	221	616	LD	(VDPST),A	PRESERVE VDPST
00023A	31FFFC	222	617	LD	SP,STACK	
00023D	2A50FB	223	618	LD	HL,(BPADR)	
000240	7C	224	619	LD	A,H	
000241	85	225	620	OR	L	
000242	3A5CFB	226	621	LD	A,(BPVAL)	SEE IF BREAK-POINT TO BE CLEARED
000245	2801	227	622	JR	Z,BRK1	
000247	77	228	623	LD	(HL),A	
000248	210000	229	624 BRK1	LD	HL,H'0000'	DISPLAY REGISTERS AT BREAKPOINT (20)
			625	MCAL	ZREG	
000248	CF	230	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
00024C	99		+	DATA	ZREG.	FUNCTION NUMBER
			+	ENDM		
00024D	C37101	231	626	JP	MONITR	
			627 *			
			628 *			
			629 *			BOOT - ACTIVATE XTALDOS, IF AVAILABLE
			630 *			
			631 *			
000250	31FFFC	232	632 BOOT	LD	SP,STACK	
000253	AF	233	633	XOR	A	
000254	67	234	634	LD	H,A	
000255	6F	235	635	LD	L,A	
000256	3250FB	236	636	LD	(HSTDSC),A	
000259	2251FB	237	637	LD	(HSTTRK),HL	
00025C	2100FE	238	638	LD	HL,HSTBUF	
00025F	2253FB	239	639	LD	(HSTDMA),HL	

LOC OBJECT M STAT E LINE SOURCE LINE

```

000262 E5      240  640  PUSH  HL
                  641  MCL  ZRSECT      READ TRACK 0, SECTOR 0
000263 CF      241    +   RST   H'08'      MOS FUNCTION CALL ROUTINE
000264 A2              +   DATA ZRSECT      FUNCTION NUMBER
                  +   ENDM
000265 E1      242  642  POP   HL
000266 5E      243  643  LD    E,(HL)      GET START ADDRESS OF DOS
000267 23      244  644  INC   HL
000268 56      245  645  LD    D,(HL)
000269 23      246  646  INC   HL
00026A 05      247  647  PUSH  DE      SAVE START ADDRESS OF DOS
00026B 5E      248  648  LD    E,(HL)      GET END ADDRESS OF DOS TO LOAD
00026C 23      249  649  INC   HL
00026D 56      250  650  LD    D,(HL)
00026E 23      251  651  INC   HL
00026F 4E      252  652  LD    C,(HL)      GET BOOT JUMP ADDRESS
000270 23      253  653  INC   HL
000271 46      254  654  LD    B,(HL)
000272 23      255  655  INC   HL
000273 7E      256  656  LD    A,(HL)      SET UP SINGLE/DOUBLE-SIDE FLAGS
000274 32B1F8  257  657  LD    (SIDFLG),A
000277 E1      258  658  POP   HL
000278 C5      259  659  PUSH  BC      SAVE JUMP ADDRESS
000279 AF      260  660  XOR    A
00027A 47      261  661  LD    B,A
00027B 4F      262  662  LD    C,A
00027C C08C04  263  663  CALL  RDBL1
00027F C317F8  264  664  JP     XECEND      EXECUTE DOS BY RETURN

```

665 \*

666 \*

667 \*

TABLE FOR MONITOR COMMANDS

668 \*

669 \*

```

000282 2C03      670 FMTBL ACON  ARITH      MOS FUNCTION TABLE START(A)
000284 4903      671      ACON  BAUD              (B)
000286 9D03      672      ACON  COPY              (C)
000288 8903      673      ACON  DECIML           (D)
00028A 0204      674      ACON  EXEC              (E)
00028C EC03      675      ACON  MFILL            (F)
00028E FE03      676      ACON  GOTO             (G)
000290 3404      677      ACON  HEX              (H)
000292 F501      678      ACON  ERROR
000294 F501      679      ACON  ERROR
000296 F501      680      ACON  ERROR
000298 F501      681      ACON  ERROR
00029A 5C04      682      ACON  MODIFY            (M)
00029C F501      683      ACON  ERROR
00029E F501      684      ACON  ERROR
0002A0 F501      685      ACON  ERROR
0002A2 F501      686      ACON  ERROR
0002A4 8904      687      ACON  RDBLOCK           (R)
0002A6 F501      688      ACON  ERROR
0002A8 4D05      689      ACON  TBLATE            (T)
0002AA F501      690      ACON  ERROR
0002AC F501      691      ACON  ERROR
0002AE 9104      692      ACON  WRBLOCK           (W)

```

LOC OBJECT M STAT E LINE SOURCE LINE

0002B0	8705		693	ACON	COLD	(X)
0002B2	8C05		694	ACON	WARM	(Y)
0002B4	9305		695	ACON	REGSTR	(Z)
			696 *			
			697 *			
0002B6	5001		698	ACON	MOS1	REST OF MOS FUNCTION TABLE (H'9A')
0002B8	0706		699	ACON	RSCAN	
0002BA	5E06		700	ACON	KEYIN	
0002BC	1406		701	ACON	GETLN	
0002BE	1A0F		702	ACON	OUTC	
0002C0	5C0F		703	ACON	POUT	
0002C2	480F		704	ACON	SRLDUT	(H'80')
0002C4	520F		705	ACON	SRLIN	
0002C6	0000	E	706	ACON	RDSECT	
0002C8	0000	E	707	ACON	WRSECT	
0002CA	E804		708	ACON	RDBLK	
0002CC	F804		709	ACON	WRBLK	
0002CE	140F		710	ACON	CRLF	
0002D0	0F0F		711	ACON	CRLFZ	
0002D2	180F		712	ACON	SPACE	(H'83')
0002D4	9D0E		713	ACON	PR4HX	(H'83')
0002D6	A10E		714	ACON	PR2HXZ	
0002D8	A50E		715	ACON	PR2HX	
0002DA	E40E		716	ACON	FCH4HX	
0002DC	D10E		717	ACON	FCH2HX	
0002DE	0000	E	718	ACON	COMMND	
0002E0	0000	E	719	ACON	HOMOSC	
0002E2	F00E		720	ACON	IGBLK	(H'80')
0002E4	14F8		721	ACON	RDMEH	
0002E6	1AF8		722	ACON	UPCOPY	
0002E8	21F8		723	ACON	DNCOPY	
0002EA	6A0F		724	ACON	MUSQUT	
0002EC	6F07		725	ACON	KSCAN	
0002EE	0000	E	726	ACON	SELDSC	
0002F0	0000	E	727	ACON	SETTRK	
0002F2	0000	E	728	ACON	SETSEC	(H'88')
0002F4	0000	E	729	ACON	SETBUF	
0002F6	0000	E	730	ACON	ROREC	
0002F8	0000	E	731	ACON	WRREC	
0002FA	AE0F		732	ACON	ZTIME	
0002FC	0000	E	733	ACON	FORST	
0002FE	C70D		734	ACON	SYSRST	
000300	780E		735	ACON	LOGO	
000302	920F		736	ACON	PGINIT	(H'CO')
000304	2000		737	ACON	SREGZ	
000306	E108		738	ACON	VIDIN	
000308	3000		739	ACON	VRQUTZ	
00030A	0000	E	740	ACON	PLOT	
00030C	0000	E	741	ACON	PLTXV	
00030E	0000	E	742	ACON	POINT	
000310	0000	E	743	ACON	PNTXY	
000312	0000	E	744	ACON	DRAWTO	(H'CS')
000314	0000	E	745	ACON	POLYG	
000316	0000	E	746	ACON	ORGC0	
000318	0000	E	747	ACON	CALADR	
00031A	0000	E	748	ACON	SETCLR	

LOC OBJECT M STAT E LINE SOURCE LINE

00031C	0000	E	749	ACON	FILL	
00031E	600E		750	ACON	IMULT	
000320	260F		751	ACON	MPRM	
000322	CE0A		752	ACON	VOUT	
000324	450C		753	ACON	SETCZ	(H'00')
000326	0440		754	ACON	R2XEC	
000328	080E		755	ACON	INIT30	
00032A	4100		756	ACON	STAT80	
			757 *			
			758 *			
			759 *			
			760 *			
			761 *			
					ARITH - DISPLAY SUM, DIFFERENCE, AND OFFSET OF ARGUMENTS	
00032C	EB	265	762	ARITH	EX	DE, HL
00032D	E5	266	763	PUSH		HL
00032E	19	267	764	ADD		HL, DE
			765	MCAL		ZPR4HX
00032F	CF	268	+	RST		H'08'
000330	A9		+	DATA		ZPR4HX
			+	ENDM		
000331	E1	269	766	POP		HL
000332	B7	270	767	OR		A
000333	ED52	271	768	SBC		HL, DE
			769	MCAL		ZPR4HX
000335	CF	272	+	RST		H'08'
000336	A9		+	DATA		ZPR4HX
			+	ENDM		
000337	28	273	770	DEC		HL
000338	28	274	771	DEC		HL
000339	7C	275	772	LD		A, H
00033A	C805	276	773	RLC		L
00033C	CE00	277	774	ADC		A, H'00'
00033E	2005	278	775	JR		NZ, AERR
000340	7D	279	776	LD		A, L
000341	0F	280	777	RRCA		
000342	C3A50E	281	778	JP		PR2HX
			779	AERR		PRM
000345	0F	282	+	RST		H'18'
			+	ENDM		
000346	2DA0		780	DATA		'-', H'AD'
000348	C3	283	781	RET		
			782 *			
			783 *			
			784 *			
			785 *			
			786 *			
			787 *			
			788 *			
					BAUD - SET Tx & Rx BAUD RATES. B rt xxxx WHERE r & t	
					ARE CODES CORRESPONDING TO THE TABLE OF BAUD	
					RATES xxxx IS OPTIONAL MODE CHANGE ON PCI	
000349	7D	284	789	BAUD	LD	A, L
00034A	F5	285	790	PUSH		AF
00034B	E60F	286	791	AND		H'0F'
00034D	6F	287	792	LD		L, A
00034E	67	288	793	LD		H, A
00034F	F1	289	794	POP		AF
000350	E6F0	290	795	AND		H'F0'
000352	2805	291	796	JR		Z, BAUI
						Rx RATE 0, ASSIGN Tx RATE TO Rx
						L(HIGH)=Rx RATE, L(LOW)=Tx RATE

LOC OBJECT M STAT E LINE SOURCE LINE

000354	OF		292	797	RRCA		
000355	OF		293	798	RRCA		
000356	OF		294	799	RRCA		
000357	OF		295	800	RRCA		
000358	67		296	801	LD	H,A	
000359	E5		297	802	PUSH	HL	
00035A	7D		298	803	LD	A,L	Tx SETTING
00035B	0E28		299	804	LD	C,CTC0	
00035D	CD7703		300	805	CALL	SETBD	
000360	E1		301	806	POP	HL	
000361	D8		302	807	RET	C	
000362	7C		303	808	LD	A,H	Rx RATE NOW IN A
000363	0E29		304	809	LD	C,CTC1	
000365	CD7703		305	810	CALL	SETBD	
000368	D8		306	811	RET	C	
000369	7A		307	812	LD	A,D	
00036A	B3		308	813	OR	E	SEE IF PCI TO BE SET UP
00036B	C8		309	814	RET	Z	
00036C	3F40		310	815	LD	A,H'40'	PUT PCI INTO MODE FORMAT
00036E	D311		311	816	OUT	(PCIC),A	
000370	7A		312	817	LD	A,D	SET MODE
000371	D311		313	818	OUT	(PCIC),A	
000373	7B		314	819	LD	A,E	DO COMMAND
000374	D311		315	820	OUT	(PCIC),A	
000376	C9		316	821	RET		
			322 *				
			823 *				
000377	FE09		317	824	SETBD	CP	9
000379	3F		318	825	CCF		
00037A	D8		319	826	RET	C	MUST BE 0-9
00037B	218803		320	827	LD	HL,BDTBL	ACCESS BAUD RATE TABLE
00037E	87		321	828	ADD	A,A	
00037F	85		322	829	ADD	A,L	
000380	6F		323	830	LD	L,A	
000381	3002		324	831	JR	NC,58D1	
000383	87		325	832	OR	A	
000384	24		326	833	INC	H	
000385	46		327	834	LD	B,(HL)	
000386	23		328	835	INC	HL	
000387	7E		329	836	LD	A,(HL)	
000388	C3BC0F		330	837	JP	SETCTC	SET CTC CLOCKS FOR SERIAL PORT
			838 *				
			839 *				
			840 *				
			841 *				
			842 *				
			843 *				
			844 *				
			845 *				
			846 *				
			847 *				
000388	D01D		348	BDTBL	DATA	H'00',H'10'	0 - 75 BAUD
00038D	3E1D		349		DATA	H'8E',H'10'	1 - 110
00038F	681D		350		DATA	H'68',H'10'	2 - 150
000391	341D		351		DATA	H'34',H'10'	3 - 300
000393	1A1D		352		DATA	H'1A',H'10'	4 - 600 (CTC IN TIMER MODE)

BDTBL - TABLE OF BAUD RATE INFORMATION. THERE ARE FOUR VARIABLES:

1. COUNTER/TIMER MODE SETTING OF CTC's
2. RATE MULTIPLIER ON CTC's
3. 16/256 DIVISION ON CTC's IF IN TIMER MODE
4. 16/64 DIVISION ON PCI

ONLY 1. AND 2. NEEDED FOR SETTINGS GIVEN BELOW

LOC OBJECT M STAT E LINE SOURCE LINE

```

000395 635D      353      DATA  H'63',H'5D'      5 - 1200
000397 345D      354      DATA  H'34',H'5D'      6 - 2400
000399 1A5D      355      DATA  H'1A',H'5D'      7 - 4800
00039B 0D5D      356      DATA  H'0D',H'5D'      8 - 9600
      357 *
      358 *
      359 *      COPY - INTELLIGENT COPY , SAFE IN EITHER DIRECTION
      360 *
      361 *
00039D EB        331      362 COPY  EX      DE,HL
00039E B7        332      363      OR      A
00039F ED52      333      364      SBC     HL,DE
0003A1 D8        334      365      RET     C      END MUST BE >= START
0003A2 C5        335      366      PUSH   BC
0003A3 E3        336      367      EX      (SP),HL
0003A4 C1        337      368      POP     BC
0003A5 EB        338      369      EX      DE,HL
0003A6 03        339      370      INC     BC
0003A7 B7        340      371      OR      A
0003A8 ED52      341      372      SBC     HL,DE
0003AA 19        342      373      ADD     HL,DE
0003AB 3B03      343      374      JR      C,COPYUP
      375 COPY1  MCAL    ZRCPYU
0003AD CF        344      +      RST     H'08'      MOS FUNCTION CALL ROUTINE
0003AE B2        +      DATA  ZRCPYU      FUNCTION NUMBER
      +      ENDM
0003AF C3        345      376      RET
      377 *
      378 *
0003B0 EB        346      379 COPYUP EX      DE,HL
0003B1 09        347      380      ADD     HL,BC
0003B2 EB        348      381      EX      DE,HL
0003B3 09        349      382      ADD     HL,BC
0003B4 2B        350      383      DEC     HL
0003B5 1B        351      384      DEC     DE
      385      MCAL    ZRCPYD
0003B6 CF        352      +      RST     H'08'      MOS FUNCTION CALL ROUTINE
0003B7 B3        +      DATA  ZRCPYD      FUNCTION NUMBER
      +      ENDM
0003B8 C3        353      386      RET
      387 *
      388 *
      389 *      DECIML - CONVERT HEX ARGUMENT TO DECIMAL DISPLAY
      390 *
      391 *
0003B9 1E00      354      392 DECIML LD      E,0
0003BB 011027    355      393      LD      BC,10000      TENS OF THOUSANDS
0003BE C0D603    356      394      CALL   DIGIT
0003C1 01E803    357      395      LD      BC,1000      THOUSANDS
0003C4 C0D603    358      396      CALL   DIGIT
0003C7 016400    359      397      LD      BC,100      HUNDREDS
0003CA C0D603    360      398      CALL   DIGIT
0003CD 010A00    361      399      LD      BC,10      TENS
0003D0 C0D603    362      900      CALL   DIGIT
0003D3 7D        363      901      LD      A,L      MUST HAVE ONLY BOTTOM DIGIT LEFT
0003D4 1B10      364      902      JR      002

```

LOC OBJECT M STATE LINE SOURCE LINE

```

          903 *
          904 *
0003D6 3EFF      365  905 DIGIT LD    A,H'FF'      DO SINGLE DIGIT BY SUBTRACTION OF 8C
0003D8 B7        366  906 DG1  OR    A
0003D9 ED42      367  907      SBC    HL,BC
0003D8 3C        368  908      INC    A
0003DC 30FA      369  909      JR     NC,DG1
0003DE 09        370  910      ADD    HL,BC      RESTORE REMAINDER
0003DF 1C        371  911      INC    E
0003E0 1D        372  912      DEC    E
0003E1 2003      373  913      JR     NZ,DG2
0003E3 3C        374  914      INC    A
0003E4 3D        375  915      DEC    A
0003E5 C3        376  916      RET     Z          FORGET LEADING ZEROES
0003E6 C630      377  917 DG2  ADD    A,'0'      GET INTO ASCII
0003E8 1E01      378  918      LD     E,1        SET FLAG FOR LEADING ZEROES
          919      DUTCH
0003EA D7        379      +      RST    H'10'      CHARACTER OUTPUT ROUTINE
          +      ENDM
0003EB C9        380  920      RET
          921 *
          922 *
          923 *      MFILL - FILL BLOCK OF MEMORY WITH CONSTANT VALUE
          924 *
          925 *
0003EC EB        381  926 MFILL EX    DE,HL
0003ED B7        382  927      OR     A
0003EE ED52      383  928      SBC    HL,DE
0003F0 D8        384  929      RET     C          ERROR IF TOP(BOTTOM OF BLOCK
0003F1 79        385  930      LD     A,C
0003F2 2002      386  931      JR     NZ,MFILL1     JUMP IF START <> FINISH
0003F4 12        387  932      LD     (DE),A
0003F5 C9        388  933      RET
0003F6 44        389  934 MFILL1 LD    B,H
0003F7 4D        390  935      LD     C,L
0003F8 62        391  936      LD     H,D
0003F9 68        392  937      LD     L,E
0003FA 13        393  938      INC    DE
0003FB 77        394  939      LD     (HL),A
0003FC 18AF      395  940      JR     CPV1
          941 *
          942 *
          943 *      GOTO - EXECUTE PROGRAM AT GIVEN MEMORY LOCATION
          944 *      EXEC - EXECUTE PROGRAM FROM BREAKPOINT
          945 *
          946 *
0003FE 2277FB    396  947 GOTO  LD    (OLDPC),HL
000401 EB        397  948      EX     DE,HL
000402 F3        398  949 EXEC  DI
          950      LD     A,H          DON'T INTERRUPT WHILE SETTING UP
000403 7C        399  951      OR     L
000404 B5        400  952      JR     Z,EXEC1     NO BREAKPOINT IF ZERO
000405 230A      401  953      MCALL ZRDMM
000407 CF        402      +      RST    H'08'      MOS FUNCTION CALL ROUTINE
000408 B1        +      DATA ZRDMM          FUNCTION NUMBER
          +      ENDM

```

LOC OBJECT M STATE LINE SOURCE LINE

000409	225DFB	403	954	LD	(BPADR),HL	
00040C	325CFB	404	955	LD	(BPVAL),A	
00040F	36FF	405	956	LD	(HL),H'FF'	SET BREAKPOINT
000411	315FFB	406	957 EXEC1	LD	SP,OLDI-1	
000414	F1	407	958	POP	AF	
000415	ED47	408	959	LD	I,A	SET INTERRUPT REGISTER
000417	DDE1	409	960	POP	IX	
000419	FDE1	410	961	POP	IY	
00041B	F1	411	962	POP	AF	OLDSP NOT WANTED YET
00041C	F1	412	963	POP	AF	GET ALTERNATE REGISTERS
00041D	C1	413	964	POP	BC	
00041E	D1	414	965	POP	DE	
00041F	E1	415	966	POP	HL	
000420	D9	416	967	EXX		
000421	08	417	968	EX	AF,AF'	
000422	F1	418	969	POP	AF	GET NORMAL REGISTERS
000423	C1	419	970	POP	BC	
000424	D1	420	971	POP	DE	
000425	ED7B65FB	421	972	LD	SP,(OLDSP)	RETRIEVE STACK POINTER
000429	2A77FB	422	973	LD	HL,(OLDPC)	GET PC ADDRESS
00042C	E5	423	974	PUSH	HL	
00042D	2A75FB	424	975	LD	HL,(OLDHL)	
000430	FB	425	976	EI		CAN INTERRUPT NOW
000431	C317FB	426	977	JP	XECEND	END WITH SCRATCH-PAD ROUTINE
			978 *			
			979 *			
			980 *			
			981 *			
			982 *			
			983 HEX	MCAL	ZIGBLK	FETCH DIGITS
000434	CF	427	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000435	80		+	DATA	ZIGBLK	FUNCTION NUMBER
			+	ENDM		
000436	210000	428	984	LD	HL,0	
000439	1A	429	985 HC1	LD	A,(DE)	
00043A	87	430	986	OR	A	
00043B	281C	431	987	JR	Z,HC2	DONE IF CR
00043D	13	432	988	INC	DE	
00043E	063C	433	989	SUB	'0'	SEE IF 0-9
000440	FE0A	434	990	CP	H'0A'	
000442	3F	435	991	CCF		
000443	08	436	992	RET	C	NOT A DIGIT - ERROR
000444	019A19	437	993	LD	BC,H'199A'	ERROR IF > 6553
000447	ED42	438	994	SBC	HL,BC	
000449	09	439	995	ADD	HL,BC	
00044A	3F	440	996	CCF		
00044B	D8	441	997	RET	C	
00044C	29	442	998	ADD	HL,HL	MULTIPLY HL BY 10
00044D	44	443	999	LD	B,H	
00044E	40	444	1000	LD	C,L	
00044F	29	445	1001	ADD	HL,HL	
000450	29	446	1002	ADD	HL,HL	
000451	09	447	1003	ADD	HL,BC	
000452	4F	448	1004	LD	C,A	
000453	0600	449	1005	LD	B,0	
000455	09	450	1006	ADD	HL,BC	ADD NEW DIGIT

LDC	OBJECT	M	STATE	LINE	SOURCE	LINE
000456	30E1	451	1007	JR	HC,HC1	
000458	C9	452	1008	RET		
			1009	MCAL	ZPR4HX	FINISH BY OUTPUTTING 4-DIGIT HEX
000459	CF	453	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
00045A	A9		+	DATA	ZPR4HX	FUNCTION NUMBER
			+	ENDM		
00045B	C9	454	1010	RET		
			1011	*		
			1012	*		
			1013	*	MODIFY - CHANGE MEMORY LOCATION	
			1014	*		
			1015	*		
			1016	MODIFY MCAL	ZPR4HX	DISPLAY LOCATION
00045C	CF	455	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
00045D	A9		+	DATA	ZPR4HX	FUNCTION NUMBER
			+	ENDM		
			1017	MCAL	ZRDMM	READ IT
00045E	CF	456	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
00045F	B1		+	DATA	ZRDMM	FUNCTION NUMBER
			+	ENDM		
			1018	MCAL	ZPR2HX	DISPLAY CONTENTS
000460	CF	457	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
000461	A8		+	DATA	ZPR2HX	FUNCTION NUMBER
			+	ENDM		
			1019	POP		
000462	DF	458	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
			+	ENDM		
000463	0388		1020	DATA	BS,BS+H'80'	PUT CURSOR OVER CONTENTS
000465	1188F8	459	1021	LD	DE,BUFFER	
			1022	MCAL	ZGETLN	
000468	CF	460	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
000469	9D		+	DATA	ZGETLN	FUNCTION NUMBER
			+	ENDM		
00046A	E5	461	1023	PUSH	HL	SAVE DISPLAYED LOCATION
00046B	C00C01	462	1024	CALL	M0N6	REMOVE MSB'S FROM BUFFER(00001 INVERSE)
			1025	MCAL	ZFC4HX	FETCH AGAIN IN CASE USER CHANGED IT
00046E	CF	463	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
00046F	AC		+	DATA	ZFC4HX	FUNCTION NUMBER
			+	ENDM		
000470	3310	464	1026	JR	C,MOD3	
000472	C1	465	1027	POP	BC	POP RUBBISH
			1028	MOD1 MCAL	ZIG8LK	
000473	CF	466	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
000474	80		+	DATA	ZIG8LK	FUNCTION NUMBER
			+	ENDM		
000475	28E5	467	1029	JR	Z,MODIFY	
000477	FE2E	468	1030	CP	'.'	USE "." TO TERMINATE
000479	C8	469	1031	RET	Z	
			1032	MCAL	ZFC2HX	
00047A	CF	470	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
00047B	AD		+	DATA	ZFC2HX	FUNCTION NUMBER
			+	ENDM		
00047C	3305	471	1033	JR	C,MOD4	
00047E	77	472	1034	LD	(HL),A	UPDATE LOCATION
00047F	23	473	1035	INC	HL	
000480	18F1	474	1036	JR	MOD1	

LOC OBJECT # STATE LINE SOURCE LINE

000482 E1	475	1037	MOD3	POP	HL	
		1038	MOD4	PRM		
000483 DF	476	+	RST	H'18'		MESSAGE OUTPUT ROUTINE
		+	ENDM			
000484 3F008A		1039	DATA	'2',CR,LF,H'80'		
000487 18D3	477	1040	JR	MODIFY		
		1041 *				
		1042 *				
		1043 *				
		1044 *				
		1045 *				
000489 CD3605	478	1046	ROBLK CALL	DSCSET		GET 4th ARGUMENT AND SET UP
		1047	ROBLI	MCAL	ZRBLK	
00048C CF	479	+	RST	H'08'		MOS FUNCTION CALL ROUTINE
00048D A4		+	DATA	ZRBLK		FUNCTION NUMBER
		+	ENDM			
00048E C3	480	1048	RET	Z		
00048F 1306	481	1049	JR	DSCERR		
		1050 *				
		1051 *				
000491 CD3605	482	1052	WRBLK CALL	DSCSET		GET 4th ARGUMENT AND SET UP
		1053	MCAL	ZWBLK		
000494 CF	483	+	RST	H'08'		MOS FUNCTION CALL ROUTINE
000495 A5		+	DATA	ZWBLK		FUNCTION NUMBER
		+	ENDM			
000496 C3	484	1054	RET	Z		
		1055 *				
		1056 *				
		1057	DSCERR	PRM		MOS DISC ERROR HANDLER
000497 DF	485	+	RST	H'18'		MESSAGE OUTPUT ROUTINE
		+	ENDM			
000498 44697363 3AA0		1058	DATA	'Disc:',H'20'		
00049E 3A56FB	486	1059	LD	A,(STATUS)		HANDLE DISC ERROR
0004A1 3D	487	1060	DEC	A		
0004A2 2820	488	1061	JR	Z,DATERR		ERROR No. 1 - BAD DATA ERROR
0004A4 3D	489	1062	DEC	A		
0004A5 2828	490	1063	JR	Z,WPERR		2 - WRITE PROTECTED
0004A7 3D	491	1064	DEC	A		
0004A8 2831	492	1065	JR	Z,SNF		3 - SECTOR NOT FOUND
0004AA 3D	493	1066	DEC	A		
0004AB 2808	494	1067	JR	Z,NRDY		4 - DRIVE NOT READY
		1068	NDRY	PRM		- OTHERWISE NO DRIVE
0004AD DF	495	+	RST	H'18'		MESSAGE OUTPUT ROUTINE
		+	ENDM			
0004AE 4E6F2044 726975E5		1069	DATA	'No Driv',H'E5'		
0004B6 182D	496	1070	JR	JERR		
		1071	NRDY	PRM		- OTHERWISE JUST NOT READY
0004B8 DF	497	+	RST	H'18'		MESSAGE OUTPUT ROUTINE
		+	ENDM			
0004B9 4E6F7420 52656164 F9		1072	DATA	'Not Read',H'F9'		
0004C2 1821	498	1073	JR	JERR		
		1074	DATERR	PRM		

LOC	OBJECT	STAT	E	LINE	SOURCE	LINE
0004C4	DF	499	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
			+	ENDM		
0004C5	42616420 446174E1	1075		DATA	'Bad Dal',H'E1'	
0004C0	1816	500	1076	JR	JERR	
			1077	WPERR	PRM	
0004CF	DF	501	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
			+	ENDM		
0004D0	50726F74 65637465 E4	1078		DATA	'Protecte',H'E4'	
0004D9	180A	502	1079	JR	JERR	
			1080	SNF	PRM	
0004D8	DF	503	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
			+	ENDM		
0004DC	4E6F2053 6563746F F2	1081		DATA	'No Secto',H'F2'	
0004E5	C3FC01	504	1082	JERR	JP	ERR1
			1083	*		
			1084	*		
0004E8	CD4205	505	1085	R0BLK	CALL	DSETP
0004E8	D5	506	1086	R0B1	PUSH	DE
			1087	MCAL	ZRSECT	READ A SECTOR
0004EC	CF	507	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
0004ED	A2		+	DATA	ZRSECT	FUNCTION NUMBER
			+	ENDM		
0004EE	D1	508	1088	POP	DE	
0004EF	2015	509	1089	JR	HZ,RWBEND	
0004F1	CD0805	510	1090	CALL	INCSCCT	
0004F4	38F5	511	1091	JR	C,R0B1	
0004F6	180E	512	1092	JR	RWBEND	
			1093	*		
			1094	*		
0004F8	CD4205	513	1095	WR0LK	CALL	DSETP
0004F8	D5	514	1096	WR01	PUSH	DE
			1097	MCAL	ZWSECT	WRITE A SECTOR
0004FC	CF	515	+	RST	H'08'	M05 FUNCTION CALL ROUTINE
0004FD	A3		+	DATA	ZWSECT	FUNCTION NUMBER
			+	ENDM		
0004FE	D1	516	1098	POP	DE	
0004FF	2005	517	1099	JR	HZ,RWBEND	
000501	CD0805	518	1100	CALL	INCSCCT	
000504	38F5	519	1101	JR	C,WR01	
000506	3A56FB	520	1102	RWBEND	LD	A,(STATUS)
000509	B7	521	1103	OR	A	
00050A	C9	522	1104	RET		
			1105	*		
			1106	*		
00050B	2152FB	523	1107	INCSCCT	LD	HL,HSTSEC
00050E	34	524	1108	INC	(HL)	MOVE TO NEXT SECTOR
00050F	CD0000	525	1109	CALL	GNSECT	GET No. SECTORS FOR DRIVE
000512	4F	526	1110	LD	C,A	
000513	7E	527	1111	LD	A,(HL)	
000514	89	528	1112	CP	C	
000515	380E	529	1113	JR	C,IS1	

LOC OBJECT M STAT E LINE SOURCE LINE

000517 3600	530	1114	LD	(HL),0	SECTOR OVERFLOW - MOVE TO NEXT TRACK
000519 28	531	1115	DEC	HL	
00051A 34	532	1116	INC	(HL)	
00051B 7E	533	1117	LD	A,(HL)	
00051C FE28	534	1118	CP	NTRAX	
00051E 3805	535	1119	JR	C,IS1	
000520 E1	536	1120	POP	HL	TRACK OVERFLOW - ERROR
000521 3E03	537	1121	LD	A,H'03'	MARK AS SECTOR NOT FOUND (NO CARRY)
000523 1800	538	1122	JR	IS2	
000525 2A53F8	539	1123 IS1	LD	HL,(HSTDMA)	
000528 010002	540	1124	LD	SC,BUFSIZ	
00052B 09	541	1125	ADD	HL,BC	
00052C 2253F8	542	1126	LD	(HSTDMA),HL	
00052F AF	543	1127	XOR	A	
000530 ED52	544	1128	SBC	HL,DE	CARRY SET IF MORE TO GO
000532 3256F8	545	1129 IS2	LD	(STATUS),A	
000535 C9	546	1130	RET		
		1131 *			
		1132 *			
000536 09	547	1133 DSCSET	EXX		
		1134	MCAL	ZIGBLK	
000537 CF	548	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000538 B0		+	DATA	ZIGBLK	FUNCTION NUMBER
		+	ENDM		
		1135	MCAL	ZFC4HX	GET DISC NUMBER (0-3)
000539 CF	549	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
00053A AC		+	DATA	ZFC4HX	FUNCTION NUMBER
		+	ENDM		
00053B 7D	550	1136	LD	A,L	
00053C 09	551	1137	EXX		
00053D FE04	552	1138	CP	4	4 DISCS ONLY
00053F D8	553	1139	RET	C	
000540 18A3	554	1140	JR	JERR	
		1141 *			
		1142 *			
000542 3250F8	555	1143 DSETP	LD	(HSTDSC),A	
000545 2253F8	556	1144	LD	(HSTDMA),HL	SET UP START ADDRESS
000548 ED4351F8	557	1145	LD	(HSTTRK),BC	SET UP TRACK/SECTOR
00054C C9	558	1146	RET		
		1147 *			
		1148 *			
		1149 *			
		1150 *			
		1151 *			
00054D 79	559	1152 TBLATE	LD	A,C	
00054E B7	560	1153	OR	A	
00054F 2002	561	1154	JR	NZ,TBL1	
000551 3E08	562	1155	LD	A,H'08'	3-BYTE ROW DEFAULT
000553 4F	563	1156 TBL1	LD	C,A	
		1157 TBL2	MCAL	ZRSCAN	ALLOW BREAK FROM KEYBOARD
000554 CF	564	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000555 98		+	DATA	ZRSCAN	FUNCTION NUMBER
		+	ENDM		
000556 FE18	565	1158	CP	ESC	BREAK IF 'ESC'
000558 C8	566	1159	RET	Z	
		1160	MCAL	ZPR4HX	

LOC OBJECT M STATE LINE SOURCE LINE

000559 CF	567	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
00055A A9		+	DATA	ZPR4HX	FUNCTION NUMBER
		+	ENDM		
00055B 41	568	1161	LD	B,C	
00055C E5	569	1162	PUSH	HL	
		1163 TBL3	MCAL	ZRDMEM	
00055D CF	570	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
00055E 81		+	DATA	ZRDMEM	FUNCTION NUMBER
		+	ENDM		
		1164	MCAL	ZP2HX2	
00055F CF	571	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000560 AA		+	DATA	ZP2HX2	FUNCTION NUMBER
		+	ENDM		
000561 87	572	1165	OR	A	
000562 ED52	573	1166	SBC	HL,DE	
000564 19	574	1167	ADD	HL,DE	
000565 23	575	1168	INC	HL	
000566 3002	576	1169	JR	NC,TBL4	
000568 10F3	577	1170	DJNZ	TBL3	
00056A E1	578	1171 TBL4	POP	HL	
00056B 41	579	1172	LD	B,C	
		1173 TBL5	MCAL	ZRDMEM	
00056C CF	580	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
00056D 81		+	DATA	ZRDMEM	FUNCTION NUMBER
		+	ENDM		
00056E E67F	581	1174	AND	H'7F'	MASK MSB FOR DISPLAY
000570 FE20	582	1175	CP	' '	
000572 3804	583	1176	JR	C,TBL6	
000574 FE7F	584	1177	CP	H'7F'	DON'T ALLOW CTRL-CHARS OR H'7F' (DELETE ON
000576 2002	585	1178	JR	NZ,TBL7	PRINTER)
000578 3E2E	586	1179 TBL6	LD	A,' '	
		1180 TBL7	OUTCH		
00057A D7	587	+	RST	H'10'	CHARACTER OUTPUT ROUTINE
		+	ENDM		
00057B 87	588	1181	OR	A	
00057C ED52	589	1182	SBC	HL,DE	
00057E 19	590	1183	ADD	HL,DE	
00057F 23	591	1184	INC	HL	
000580 D0	592	1185	RET	NC	
000581 10E9	593	1186	DJNZ	TBL5	
		1187	MCAL	ZCRLF	
000583 CF	594	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000584 A6		+	DATA	ZCRLF	FUNCTION NUMBER
		+	ENDM		
000585 18CD	595	1188	JR	TBL2	
		1189 *			
		1190 *			
		1191 *			
		1192 *			
		1193 *			
000587 2A32F8	596	1194 COLD	LD	HL,(CLOVCT)	COLD START ENTRY
00058A 1803	597	1195	JR	VCTJMP	
		1196 *			
		1197 *			
00059C 2A34F8	598	1198 WARM	LD	HL,(WRMVCT)	WARM START ENTRY
00058F E5	599	1199 VCTJMP	PUSH	HL	

COLD, WARM - COLD AND WARM START ENTRIES

LOC OBJECT M STAT E LINE SOURCE LINE

```

000590 C317F8      600 1200      JP      XECEND      JUMP VIA RAM SWAP
                   1201 *
                   1202 *
                   1203 *      REGSTR - DISPLAY REGISTERS FOR DEBUGGING
                   1204 *      Z0 - NORMAL REGISTER SET
                   1205 *      Z1 - ALTERNATIVE SET
                   1206 *      Z2 - INDEX, INTERRUPT & STACK REGISTERS
                   1207 *
                   1208 *

000593 7D          601 1209 REGSTR LD      A,L
000594 FE02        602 1210      CP      H'02'
000596 2848        603 1211      JR      Z,REGIX
000598 3F          604 1212      CCF
000599 08          605 1213      RET      C      DON'T ALLOW > 2
00059A B7          606 1214      OR      A
00059B 116FFB      607 1215      LD      DE,DLDAF
00059E 3E20        608 1216      LD      A,' '
0005A0 2805        609 1217      JR      Z,REG1
0005A2 1167FB      610 1218      LD      DE,DLDAF1  DD ALTERNATE SET
0005A5 3E27        611 1219      LD      A,H'27'
0005A7 F5          612 1220 REG1  PUSH    AF
                   1221      NCAL    ZCRLF2
0005A8 CF          613      +      RST     H'08'      MOS FUNCTION CALL ROUTINE
0005A9 A7          +      DATA   ZCRLF2      FUNCTION NUMBER
                   +      ENDM
0005AA F1          614 1222      POP     AF
                   1223      OUTCH
0005AB 07          615      +      RST     H'10'      CHARACTER OUTPUT ROUTINE
                   +      ENDM
0005AC F5          616 1224      PUSH    AF
                   1225      PRM
0005AD DF          617      +      RST     H'18'      MESSAGE OUTPUT ROUTINE
                   +      ENDM
0005AE 41202042    1226      DATA   'A BC DE HL PC SZ-H-PN',H'C3'
                   43202020
                   44452020
                   20484C20
                   20205043
                   2020535A
                   20482050
                   4EC3
0005CC F1          618 1227      POP     AF
                   1228      OUTCH
0005CD 07          619      +      RST     H'10'      CHARACTER OUTPUT ROUTINE
                   +      ENDM
                   1229      NCAL    ZCRLF
0005CE CF          620      +      RST     H'08'      MOS FUNCTION CALL ROUTINE
0005CF A6          +      DATA   ZCRLF      FUNCTION NUMBER
                   +      ENDM
0005D0 05          621 1230      PUSH    DE
0005D1 CDFE05      622 1231      CALL    DSPREG
0005D4 E1          623 1232      POP     HL
0005D5 4E          624 1233      LD      C,(HL)
0005D6 0608        625 1234      LD      B,H'08'
0005D8 3E18        626 1235      LD      A,H'18'      DISPLAY FLAG BITS
0005DA C801        627 1236 REG2  RLC      C

```

LOC OBJECT M STATE LINE SOURCE LINE

0005DC 17	628	1237	RLA		
		1238	OUTCH		
0005DD 07	629	+	RST	H'10'	CHARACTER OUTPUT ROUTINE
		+	ENDM		
0005DE 1F	630	1239	RRR		
0005DF 10F9	631	1240	DJNZ	REG2	
0005E1 87	632	1241	OR	A	ENSURE CARRY RESET !
0005E2 C9	633	1242	RET		
0005E3 115FFB	634	1243 REGIX	LD	DE,OLDI-1	SPECIAL REGISTERS
		1244	PRM		
0005E6 0F	635	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
		+	ENDM		
0005E7 20492020		1245	DATA	' I IX IV SP PC',CR,LF+H'80'	
49582020					
20495920					
20205350					
20202050					
43008A					
		1246 *			
		1247 *			
0005FE CD080F	636	1248 DSPREG	CALL	GETHLZ	DISPLAY REGISTERS
000601 7C	637	1249	LD	A,H	
		1250	MCAL	ZP2HXZ	
000602 CF	638	+	RST	H'03'	MOS FUNCTION CALL ROUTINE
000603 AA		+	DATA	ZP2HXZ	FUNCTION NUMBER
		+	ENDM		
000604 0603	639	1251	LD	8,H'03'	
000606 CD080F	640	1252 DSR1	CALL	GETHLZ	
		1253	MCAL	ZPR4HX	
000609 CF	641	+	RST	H'03'	MOS FUNCTION CALL ROUTINE
00060A A9		+	DATA	ZPR4HX	FUNCTION NUMBER
		+	ENDM		
000608 10F9	642	1254	DJNZ	DSR1	
00060D 2A77FB	643	1255	LD	HL,(OLDPC)	END WITH PC IN EACH CASE
		1256	MCAL	ZPR4HX	
000610 CF	644	+	RST	H'03'	MOS FUNCTION CALL ROUTINE
000611 A9		+	DATA	ZPR4HX	FUNCTION NUMBER
		+	ENDM		
000612 87	645	1257	OR	A	
000613 C9	646	1258	RET		
		1259 *			
		1260 *			
		1261 *			
		1262 *			
		1263 *			
		1264 *			
		1265 GETLN	MCAL	ZKEYIN	GET KEY WITH REPEAT & FLASHING CURSOR
000614 CF	647	+	RST	H'03'	MOS FUNCTION CALL ROUTINE
000615 9C		+	DATA	ZKEYIN	FUNCTION NUMBER
		+	ENDM		
		1266	OUTCH		ECHO TYPED CHARACTER
000616 07	648	+	RST	H'10'	CHARACTER OUTPUT ROUTINE
		+	ENDM		
000617 FE0D	649	1267	CP	CR	'ENTER' KEY ?
000619 20F9	650	1268	JR	NZ,GETLN	
000618 C5	651	1269	PUSH	8C	

LOC	OBJECT	M	STAT	E	LINE	SOURCE	LINE
00061C	E5	652	1270			PUSH HL	
00061D	D5	653	1271			PUSH DE	
00061E	AF	654	1272			XOR A	
			1273			MCAL ZSCURS	
00061F	CF	655	+			RST H'08'	MOS FUNCTION CALL ROUTINE
000620	D1		+			DATA ZSCURS	FUNCTION NUMBER
			+			ENDM	
000621	E1	656	1274			POP HL	WRITE LINE TO BUFFER
000622	E5	657	1275			PUSH HL	
000623	3A45FB	658	1276			LD A,(PCFLG5)	SEE IF 80-COL MODE
000626	C857	659	1277			BIT 2,A	
000628	3A4FFB	660	1278			LD A,(XLEN)	
000628	5F	661	1279			LD E,A	
00062C	57	662	1280			LD D,A	
00062D	2009	663	1281			JR NZ,GL1A	
			1282	GL1		VRIN	GET CHARACTER FROM TEXT-POSITION TABLE
00062F	EF	664	+			RST H'28'	VIDEO RAM READ
			+			ENDM	
000630	77	665	1283			LD (HL),A	
000631	23	666	1284			INC HL	
000632	03	667	1285			INC BC	
000633	1D	668	1286			DEC E	
000634	20F9	669	1287			JR NZ,GL1	
000636	180B	670	1288			JR GL2	
000638	CDE308	671	1289	GL1A		CALL VIDINI	COPY 80-COL BUFFER
000638	77	672	1290			LD (HL),A	
00063C	23	673	1291			INC HL	
00063D	CDCF0B	674	1292			CALL INC80	
000640	1D	675	1293			DEC E	
000641	20F5	676	1294			JR NZ,GL1A	
000643	3600	677	1295	GL2		LD (HL),0	
000645	15	678	1296			DEC D	
000646	2806	679	1297			JR Z,GL3	
000648	28	680	1298			DEC HL	
000649	7E	681	1299			LD A,(HL)	
00064A	FE20	682	1300			CP	REMOVE TRAILING SPACES
00064C	28F5	683	1301			JR Z,GL2	
00064E	D1	684	1302	GL3		POP DE	
00064F	2140FB	685	1303			LD HL,PROMPT	SKIP PROMPT IF NECESSARY
000652	1A	686	1304			LD A,(DE)	
000653	E67F	687	1305			AND H'7F'	TO ALLOW INVERSE PROMPT(80 COL)
000655	8E	688	1306			CP (HL)	
000656	2001	689	1307			JR NZ,GL4	
000658	13	690	1308			INC DE	
			1309	GL4		PRM	DO LF ON END
000659	0F	691	+			RST H'18'	MESSAGE OUTPUT ROUTINE
			+			ENDM	
00065A	8A		1310			DATA LF+H'80'	
00065B	E1	692	1311			POP HL	
00065C	C1	693	1312			POP BC	
00065D	C9	694	1313			RET	
			1314	*			
			1315	*			
			1316	*			
			1317	*			
			1318	*			

KEYIN - INPUT KEY WITH FLASHING CURSOR

LOC OBJECT M STATE LINE SOURCE LINE

00065E 3A93FB	695	1319	KEYIN	LD	A,(FNKEY)	CHECK IF FUNCTION KEY PRESSED
000661 87	696	1320		OR	A	
000662 281B	697	1321	JR	Z,KEY2		
000664 C5	698	1322	PUSH	BC		
000665 ED4894FB	699	1323	LD	BC,(FNPTR)	YES - GET PTR TO FN TABLE	
		1324	VRIN			
000669 EF	700	+	RST	H'23'	VIDEO RAM READ	
		+	ENDM			
00066A 0C	701	1325	INC	C		
00066B C87F	702	1326	BIT	7,A	SEE IF AT TERMINATOR	
00066D C88F	703	1327	RES	7,A		
00066F ED4394FB	704	1328	LD	(FNPTR),BC		
000673 C1	705	1329	POP	BC		
000674 C8	706	1330	RET	Z		
000675 F5	707	1331	KEY1	PUSH	AF	
000676 AF	708	1332	XOR	A		
000677 3293FB	709	1333	LD	(FNKEY),A	TERMINATED - ZAP FLAG	
00067A 3282FB	710	1334	LD	(SCNKEY),A	AND KEY SCAN LOCATION	
00067D F1	711	1335	POP	AF		
00067E C9	712	1336	RET			
		1337 *				
		1338 *				
00067F E5	713	1339	KEY2	PUSH	HL	INPUT CHARACTER PRESERVING REGISTERS
000680 05	714	1340		PUSH	DE	
000681 C5	715	1341		PUSH	BC	
000682 3A82FB	716	1342	LD	A,(SCNKEY)	SEE IF KEY FROM A PREVIOUS SCAN	
000685 B7	717	1343	OR	A		
000686 F5	718	1344	PUSH	AF		
000687 2032	719	1345	JR	NZ,KEY6		
000689 F1	720	1346	POP	AF		
		1347	MCAL	ZSCURS	GET CURSOR LOCATION	
00068A CF	721	+	RST	H'08'	MOS FUNCTION CALL ROUTINE	
00068B 01		+	DATA	ZSCURS	FUNCTION NUMBER	
		+	ENDM			
		1348	KEY3	MCAL	ZVRIN	
00068C CF	722	+	RST	H'08'	MOS FUNCTION CALL ROUTINE	
00068D C2		+	DATA	ZVRIN	FUNCTION NUMBER	
		+	ENDM			
00068E 3247FB	723	1349	LD	(SCRNCD),A		
000691 C0830A	724	1350	KEY4	CALL	CURDSP	SHOW CURSOR IF WANTED
000694 C0C406	725	1351		CALL	CHCFLG	
000697 2008	726	1352	JR	NZ,KEY5		
000699 C0A80A	727	1353		CALL	SCRDSP	SHOW SCREEN CHARACTER
00069C C0C406	728	1354		CALL	CHCFLG.	
00069F 28EB	729	1355	JR	Z,KEY3		
0006A1 C0830A	730	1356	KEY5	CALL	CURDSP	SHOW CURSOR IF WANTED
		1357	MCAL	ZRSCAN	GET KEY WITH REPEAT IF NEEDED	
0006A4 CF	731	+	RST	H'08'	MOS FUNCTION CALL ROUTINE	
0006A5 98		+	DATA	ZRSCAN	FUNCTION NUMBER	
		+	ENDM			
0006A6 200F	732	1358	JR	NZ,KEY7	JUMP IF KEY VALUE	
0006A8 3A45FB	733	1359	LD	A,(PCFLGS)		
0006AB C857	734	1360	BIT	2,A	TEST FOR 80 COL	
0006AD 28E2	735	1361	JR	Z,KEY4	SKIP DELAY IF NOT	
0006AF 0600	736	1362	LD	B,0		
0006B1 E5	737	1363	KEY8	PUSH	HL	DELAY IF NULL CHAR ON 80 COL

LOC OBJECT M STAT E LINE SOURCE LINE

```

000682 E1      738 1364      POP  HL      (RPTNG=FF NOT LONG ENOUGH )
000683 10FC     739 1365      DJNZ KEY0
000685 180A     740 1366      JR     KEY4
000687 F5      741 1367 KEY7  PUSH  AF
000688 CDA80A   742 1368      CALL  SCRDSP      RESTORE SCREEN CODE
000688 AF      743 1369 KEY6  XOR   A
00068C 32B2FB   744 1370      LD    (SCNKEY),A      ZAP SCAN KEY LOCATION, SO NO DOUBLE-ENTRIES
00068F F1      745 1371      POP  AF
0006C0 C1      746 1372      POP  BC
0006C1 D1      747 1373      POP  DE
0006C2 E1      748 1374      POP  HL
0006C3 C9      749 1375      RET
                1376 *
                1377 *
0006C4 3A41FB   750 1378 CHCFLG LD    A,(BLINK)      TEST FOR KEY WHILE BLINKING CURSOR
0006C7 57      751 1379      LD    D,A
0006C8 1E00     752 1380      LD    E,D
0006CA C06307   753 1381 CHC1  CALL  KCHEK      DO QUICK STATUS CHECK
0006CD C0      754 1382      RET    NZ      RETURN IF KEY FOUND
0006CE 18      755 1383      DEC  DE
0006CF 7A      756 1384      LD    A,D
0006D0 83      757 1385      OR    E
0006D1 20F7     758 1386      JR    NZ,CHC1
0006D3 213EF8   759 1387      LD    HL,FLAGS
0006D5 C9      760 1388      RET
                1389 *
                1390 *
                1391 *      RSCAN - REPEAT KEY SCAN ROUTINE
                1392 *
                1393 *
0006D7 E5      761 1394 RSCAN  PUSH  HL
0006D8 D5      762 1395      PUSH DE
0006D9 C5      763 1396      PUSH BC
0006DA C06307   764 1397      CALL  KCHEK
0006DD CA5407   765 1398      JP    Z,RPT6
                1399      MCAL  ZKSCAN
0006E0 CF      766      +      RST   H'08'      MOS FUNCTION CALL ROUTINE
0006E1 B5      +      DATA  ZKSCAN      FUNCTION NUMBER
                +      ENDM
0006E2 34      767 1400      INC  (HL)      SEE IF LAST KEY WAS NULL
0006E3 35      768 1401      DEC  (HL)
0006E4 2818     769 1402      JR    Z,RPT1
0006E6 87      770 1403      OR    A      TEST FOR NULL
0006E7 200C     771 1404      JR    NZ,RPT7      JUMP IF A>NULL
0006E9 213EF8   772 1405      LD    HL,FLAGS
0006EC CB5E     773 1406      BIT  3,(HL)      CHECK FOR KEY PRESS
0006EE 2146FB   774 1407      LD    HL,LASTKY
0006F1 2061     775 1408      JR    NZ,RPT6      JUMP IF MORE THAN 1 KEY DOWN
0006F3 1309     776 1409      JR    RPT1
0006F5 C5      777 1410 RPT7  PUSH  BC
0006F6 47      778 1411      LD    B,A
0006F7 96      779 1412      SUB  (HL)      COMPARE WITH LASTKY
0006F8 E60F     780 1413      AND  H'0F'      IGNORE GRAPHICS/SHIFT/CONTROL
0006FA 78      781 1414      LD    A,B
0006FB C1      782 1415      POP  BC
0006FC 280A     783 1416      JR    Z,RPT2

```

LOC OBJECT M STAT E LINE SOURCE LINE

0006FE F5	784	1417	RPT1	PUSH	AF	
0006FF 3A42FB	785	1418		LD	A,(RPTLNG)	
000702 32B5FB	786	1419		LD	(KCNT),A	
000705 F1	787	1420		POP	AF	
000706 181C	788	1421		JR	RPT5	
000708 21B5FB	789	1422	RPT2	LD	HL,KCNT	
000708 35	790	1423		DEC	(HL)	
00070C 200E	791	1424		JR	NZ,RPT4	
00070E AF	792	1425		XOR	A	
00070F 3246FB	793	1426		LD	(LASTKY),A	ZAP LAST KEY VALUE
000712 3A43FB	794	1427		LD	A,(RPTSHT)	SET UP SHORT DELAY
000715 32B5FB	795	1428		LD	(KCNT),A	
		1429		MOVL	ZKSCAN	READ AGAIN
000718 CF	796	+		RST	H'08'	MOS FUNCTION CALL ROUTINE
000719 B5		+		DATA	ZKSCAN	FUNCTION NUMBER
		+		ENDM		
00071A 2003	797	1430		JR	NZ,RPT5	
00071C 213EFB	798	1431	RPT4	LD	HL,FLAGS	
00071F C836	799	1432		RES	0,(HL)	MARK HQ KEY
000721 AF	800	1433		XOR	A	
000722 1330	801	1434		JR	RPT6	
000724 77	802	1435	RPT5	LD	(HL),A	UPDATE LASTKY
000725 213EFB	803	1436		LD	HL,FLAGS	
000728 C84E	804	1437		BIT	1,(HL)	SEE IF FUNCTION KEY
00072A 2323	805	1438		JR	Z,RPT6	
		1439 *				
		1440 *				
00072C 5F	806	1441		LD	E,A	FUNCTION-KEY HANDLING
00072D C88B	807	1442		RES	7,E	REMOVE MSB
00072F 01803B	808	1443		LD	BC,VFNTBL	FIND FUNCTION-KEY COMBINATION
000732 1C	809	1444		INC	E	
000733 1D	810	1445	FNK1	DEC	E	
000734 280D	811	1446		JR	Z,FNK3	
000736 AF	812	1447	FNK2	XOR	A	
000737 0C	813	1448		INC	C	
000738 0D	814	1449		DEC	C	CHECK FOR END OF TABLE (ONLY GOES UP TO H'39FF')
000739 2319	815	1450		JR	Z,RPT6	
		1451		VRIN		
00073B EF	816	+		RST	H'23'	VIDEO RAM READ
		+		ENDM		
00073C 0C	817	1452		INC	C	
00073D C87F	818	1453		BIT	7,A	TEST MSB - SET IF LAST CHARACTER
00073F 23F5	819	1454		JR	Z,FNK2	
000741 18F0	820	1455		JR	FNK1	
		1456	FNK3	VRIN		GET FIRST KEY
000743 EF	821	+		RST	H'23'	VIDEO RAM READ
		+		ENDM		
000744 0C	822	1457		INC	C	
000745 C87F	823	1458		BIT	7,A	
000747 C88F	824	1459		RES	7,A	SEE IF ALSO THE LAST !
000749 ED4394FB	825	1460		LD	(FNPTR),BC	SET KEY POINTER
00074D 2005	826	1461		JR	NZ,RPT6	
00074F 2193FB	827	1462		LD	HL,FNKEY	SET FNKEY FLAG IF NOT AT TERMINATOR
000752 3601	828	1463		LD	(HL),1	
		1464 *				
		1465 *				

LOC OBJECT # STATE LINE SOURCE LINE

```

000754 B7      829 1466 RPT6 OR      A
000755 213EFB   830 1467      LD      HL,FLAGS
000758 CB46     831 1468      BIT      0,(HL)
00075A CB86     832 1469      RES      0,(HL)      CHECK CHARACTER REALLY TYPED
00075C 32B2FB   833 1470      LD      (SCNKEY),A
00075F C1       834 1471      POP      BC
000760 D1       835 1472      POP      DE
000761 E1       836 1473      POP      HL
000762 C9       837 1474      RET

1475 *
1476 *
1477 *      KCHEK - QUICK KEYBOARD CHECK RETURN Z AND A=0 IF NO
1478 *      KEY DOWN
1479 *
1480 *

000763 D5       838 1481 KCHEK PUSH    DE      CHECK FOR KEY STILL DOWN
000764 1EFF     839 1482      LD      E,H'FF'
000766 CDA20F   840 1483      CALL    PSGOI
000769 D1       841 1484      POP      DE
00076A C0       842 1485      RET      NZ
00076B 3246FB   843 1486      LD      (LASTKY),A      ZAP LAST KEY CODE
00076E C9       844 1487      RET      IF NO KEY

1488 *
1489 *
1490 *      KSCAN - FULL KEYBOARD SCAN ROUTINE, FLAG SET FOR TRUE KEY
1491 *
1492 *

00076F 3A3EFB   845 1493 KSCAN LD      A,(FLAGS)
000772 E680     846 1494      AND      H'80'
000774 323EFB   847 1495      LD      (FLAGS),A
000777 1E01     848 1496      LD      E,H'01'
000779 CDA20F   849 1497      CALL    PSGOI
00077C 1F       850 1498      RRA
00077D 47       851 1499      LD      B,A
00077E 301D     852 1500      JR      NC,SHIFT
000780 CDA50F   853 1501 BRKUP CALL    PSGIN
000783 1F       854 1502      RRA      WAIT UNTIL KEY RELEASED
000784 38FA     855 1503      JR      C,BRKUP
000786 CDA80A   856 1504      CALL    SCRDSP      ZAP CURSOR AS SOON AS RELEASED
000789 D82D     857 1505      IN      A,(AUXREG)   BREAK DETECTED - SEE IF SHIFT
00078B 07       858 1506      RLCA
00078C 3809     859 1507      JR      C,CTLBRK
00078E F1       860 1508      POP      AF
00078F ED5830FB  861 1509      LD      DE,(BRKVCT)
000793 D5       862 1510      PUSH    DE
000794 C317FB   863 1511      JP      XECEND      JUMP TO BREAK VECTOR IF SHIFT-BREAK
000797 07       864 1512 CTLBRK RLCA
000798 D25002   865 1513      JP      NC,BOOT      BOOT IF CTRL-BREAK
00079B 183F     866 1514      JR      KSC1
00079D 213EFB   867 1515 SHIFT LD      HL,FLAGS
0007A0 D82D     868 1516      IN      A,(AUXREG)
0007A2 2F       869 1517      CPL
0007A3 E6E0     870 1518      AND      H'E0'
0007A5 0F       871 1519      RRCA
0007A6 86       872 1520      OR      (HL)
0007A7 77       873 1521      LD      (HL),A      SHOW IF SHIFT, CTRL, OR C/A PRESSED

```

LOC OBJECT M STAT E LINE SOURCE LINE

0007A3 73	374	1522	FN0	LD	A,B	
0007A9 CB3F	375	1523		SRL	A	
0007AB 282F	376	1524		JR	Z,KSC1	SKIP ALL THIS JUNK IF NO SPECIAL KEY
0007AD 1F	377	1525		RRR		
0007AE 3002	378	1526		JR	NC, FN7	
0007B0 0680	379	1527		LD	B,H'80'	FN0
0007B2 1F	380	1528	FN7	RRR		
0007B3 3002	381	1529		JR	NC, ALPHLK	
0007B5 0687	382	1530		LD	B,H'87'	FN7
0007B7 1F	383	1531	ALPHLK	RRR		
0007B8 300F	384	1532		JR	NC, CARRET	
0007BA 7E	385	1533		LD	A,(HL)	
0007BB EE80	386	1534		XDR	H'80'	
0007BD 77	387	1535		LD	(HL),A	
0007BE 0822	388	1536		IN	A,(ALPHA)	TOGGLE ALPHA-LOCK LED
0007C0 CDA50F	389	1537	KYUP	CALL	PSGIN	
0007C3 CB67	390	1538		BIT	4,A	WAIT UNTIL KEY RELEASED
0007C5 20F9	391	1539		JR	NZ,KYUP	
0007C7 1813	392	1540		JR	KSC1	
0007C9 1F	393	1541	CARRET	RRR		
0007CA 3002	394	1542		JR	NC,SPB	
0007CC 0600	395	1543		LD	B,CR	CARRAIGE-RETURN
0007CE 0F	396	1544	SPB	RRR		
0007CF 3002	397	1545		JR	NC,ESCAP	
0007D1 0620	398	1546		LD	B,H'20'	SPACE-BAR
0007D3 1F	399	1547	ESCAP	RRR		
0007D4 3002	900	1548		JR	NC,SPCEND	
0007D6 0618	901	1549		LD	B,H'18'	ESCAPE
0007D8 CBC6	902	1550	SPCEND	SET	0,(HL)	END OF TESTS FOR SPECIAL CODES
0007DA C8DE	903	1551		SET	3,(HL)	
0007DC 110100	904	1552	KSC1	LD	DE,H'01'	
0007DF 0EFF	905	1553	NXTROW	LD	C,H'FF'	
0007E1 CB03	906	1554		RLC	E	
0007E3 CDA20F	907	1555		CALL	PSGDI	
0007E6 2878	908	1556		JR	Z,ROWDUN	
0007E8 0C	909	1557	NXTCLA	INC	C	
0007E9 0F	910	1558		RRR		
0007EA 3243FB	911	1559		LD	(STORE),A	
0007ED 3069	912	1560		JR	NC,CLMDUN	
0007EF 213EFB	913	1561		LD	HL,FLAGS	
0007F2 CB5E	914	1562		BIT	3,(HL)	
0007F4 C28303	915	1563		JP	NZ,INVLDK	
0007F7 C5	916	1564		PUSH	BC	SAVE COLUMN No.
0007F8 7A	917	1565		LD	A,D	PUT ROW INTO HL
0007F9 87	918	1566		ADD	A,A	DO MULTIPLY BY 24
0007FA 87	919	1567		ADD	A,A	
0007FB 87	920	1568		ADD	A,A	
0007FC 47	921	1569		LD	B,A	
0007FD 87	922	1570		ADD	A,A	
0007FE 30	923	1571		ADD	A,B	
0007FF 0600	924	1572		LD	B,0	
000801 210000	E 925	1573		LD	HL,KTAB	
000804 09	926	1574		ADD	HL,BC	
000805 4F	927	1575		LD	C,A	
000806 09	928	1576		ADD	HL,BC	
000807 3A3EF8	929	1577		LD	A,(FLAGS)	

LOC OBJECT # STATE LINE SOURCE LINE

00080A C87F	930	1578	BIT	7,A	
00080C 2816	931	1579	JR	Z,SHIFT2	
00080E C867	932	1580	SIT	4,A	
000810 2012	933	1581	JR	NZ,SHIFT2	DON'T LOCK SHIFT ON GRAPHICS KEYS
000812 7E	934	1582	LD	A,(HL)	
000813 FE61	935	1583	CP	'1'	
000815 380D	936	1584	JR	C,SHIFT2	
000817 FE78	937	1585	CP	H'78'	
000819 3009	938	1586	JR	NC,SHIFT2	
00081B E5	939	1587	PUSH	HL	
00081C 213EF8	940	1588	LD	HL,FLAGS	
00081F C806	941	1589	SET	Z,(HL)	ALPHA-LOCK ON ?
000821 E1	942	1590	POP	HL	
000822 1807	943	1591	JR	SHIFT3	
000824 3A3EF8	944	1592	SHIFT2 LD	A,(FLAGS)	
000827 C877	945	1593	BIT	6,A	SHIFT ?
000829 2804	946	1594	JR	Z,CTRL2	
00082B 010800	947	1595	SHIFT3 LD	BC,H'0008'	
00082E 09	948	1596	ADD	HL,BC	
00082F C1	949	1597	CTRL2 POP	BC	
000830 3A3EF8	950	1598	LD	A,(FLAGS)	
000833 C86F	951	1599	BIT	5,A	CTRL ?
000835 2818	952	1600	JR	Z,GTCD1	
000837 C877	953	1601	BIT	6,A	DON'T ALLOW CTRL-SHIFT
000839 2048	954	1602	JR	NZ,INVLOK	
00083B C5	955	1603	PUSH	BC	
00083C 010800	956	1604	LD	BC,H'0008'	
00083F 09	957	1605	ADD	HL,BC	
000840 3A3EF8	958	1606	LD	A,(FLAGS)	
000843 C857	959	1607	BIT	2,A	
000845 3A48FB	960	1608	LD	A,(STORE)	
000848 2001	961	1609	JR	NZ,GTCDZ	
00084A 09	962	1610	ADD	HL,BC	
00084B C1	963	1611	GTCDZ POP	BC	
00084C 3248FB	964	1612	GTCDZ LD	(STORE),A	
00084F 3A3EF8	965	1613	LD	A,(FLAGS)	
000852 F609	966	1614	GTCD1 JR	H'09'	
000854 323EF8	967	1615	LD	(FLAGS),A	
000857 46	968	1616	LD	B,(HL)	
000858 79	969	1617	CLMDUN LD	A,C	
000859 FE07	970	1618	CP	7	
00085B 3A48FB	971	1619	LD	A,(STORE)	
00085E 2088	972	1620	JR	NZ,NXTCLM	
000860 14	973	1621	ROWDUN INC	0	
000861 7A	974	1622	LD	A,D	
000862 FE07	975	1623	CP	7	
000864 C2DF07	976	1624	JP	NZ,NXTROW	GO BACK IF MORE ROWS TO DO
000867 CD970F	977	1625	CALL	PGOUTZ	
00086A 213EF8	978	1626	LD	HL,FLAGS	
00086D C878	979	1627	BIT	7,B	
00086F 280A	980	1628	JR	Z,GRAPS	DO SPECIAL BIT FOR FUNCTION KEYS
000871 C8CE	981	1629	SET	1,(HL)	
000873 C876	982	1630	BIT	6,(HL)	
000875 2816	983	1631	JR	Z,KEYDUN	
000877 C8D8	984	1632	SET	3,B	
000879 1812	985	1633	JR	KEYDUN	

LOC OBJECT N STATE LINE SOURCE LINE

000879	C866	986	1634	GRAFS	BIT	4,(HL)	GRAPHICS KEY PRESSED
00087D	280E	987	1635	JP		Z,KEYDUN	
00087F	C8F8	988	1636	SET		7,B	IF SO, SET MSB
000881	180A	989	1637	JP		KEYDUN	
000883	213EF8	990	1638	INVLK	LD	HL,FLAGS	RESET KEY-PRESSED FLAG
000886	C886	991	1639	RES		0,(HL)	
000888	CD970F	992	1640	CALL		PGOUT2	
00088B	AF	993	1641	XOR		A	
00088C	47	994	1642	LD		B,A	
00088D	C846	995	1643	KEYDUN	BIT	0,(HL)	SEE IF KEY-PRESS FOUND
00088F	2146F8	996	1644	LD		HL,LASTKY	
000892	C8	997	1645	RET		Z	ZERO IF SAME OR NO KEY
000893	78	998	1646	LD		A,B	GET KEY INTO A
000894	8E	999	1647	CP		(HL)	
000895	C9	1000	1648	RET			END OF KEYBOARD CODE
			1649 *				
			1650 *				SCREEN - OUTPUT CHARACTER IN A
			1651 *				
			1652 *				
000896	2145F8	1001	1653	SCREEN	LD	HL,PCFLGS	
000899	C85E	1002	1654	BIT		3,(HL)	TEST FOR COORDINATES AFTER CTRL 10H
00089B	C24E0A	1003	1655	JP		NZ,CUSAD1	JUMP IF COORDINATE
00089E	C84E	1004	1656	BIT		1,(HL)	SEE IF WE NEED TO O/P TO PRINTER ALSO
0008A0	C45C0F	1005	1657	CALL		NZ,POUT	
0008A3	FE20	1006	1658	CP		H'20'	JUMP IF NOT CONTROL CHARACTER
0008A5	3815	1007	1659	JR		C,CTRL	
0008A7	2145F8	1008	1660	LD		HL,PCFLGS	IF 80 COLUMN MODE & INVERT MODE
0008AA	C856	1009	1661	BIT		2,(HL)	TOGGLE MSB OF CHARACTER
0008AC	2806	1010	1662	JR		Z,V81	
0008AE	C87E	1011	1663	BIT		7,(HL)	
0008B0	2802	1012	1664	JR		Z,V81	
0008B2	EE30	1013	1665	XOR		H'80'	
0008B4	2148F8	1014	1666	V81	LD	HL,VROW	
0008B7	CDCE0A	1015	1667	CALL		VOUT	DISPLAY NORMAL CHAR.
0008BA	1813	1016	1668	JR		FWD	
			1669 *				
			1670 *				
0008BC	37	1017	1671	CTRL	ADD	A,A	
0008BD	6F	1018	1672	LD		L,A	
0008BE	2600	1019	1673	LD		H,0	
0008C0	11680A	1020	1674	LD		DE,VOUTBL	OFFSET TO CONTROL-CHARACTER ROUTINE TABLE
0008C3	19	1021	1675	ADD		HL,DE	
0008C4	CD090F	1022	1676	CALL		GETHL	
0008C7	E5	1023	1677	PUSH		HL	
0008C8	2148F8	1024	1678	LD		HL,VROW	
0008CB	3A45F8	1025	1679	LD		A,(PCFLGS)	LOAD FLAGS IN CASE WE WANT THEM
0008CE	C9	1026	1680	RET			
			1681 *				
			1682 *				
0008CF	2B	1027	1683	FWD	DEC	HL	CURSOR RIGHT
0008D0	34	1028	1684	INC		(HL)	
0008D1	3A4FF8	1029	1685	LD		A,(XLEN)	
0008D4	8E	1030	1686	CP		(HL)	CURSOR RIGHT
0008D5	C0	1031	1687	RET		NZ	
0008D6	3600	1032	1688	LD		(HL),0	
0008D8	23	1033	1689	INC		HL	

LOC OBJECT M STAT E LINE SOURCE LINE

```

1690 *
1691 *
000809 3E18 1034 1692 DOWN LD A,H'18' CURSOR DOWN
000808 3D 1035 1693 DEC A
00080C 8E 1036 1694 CP (HL)
00080D CA830C 1037 1695 JP Z,SCROLL
0008E0 34 1038 1696 INC (HL)
0008E1 C9 1039 1697 RET
1698 *
1699 *
0008E2 7E 1040 1700 UP LD A,(HL)
0008E3 87 1041 1701 OR A
0008E4 C8 1042 1702 RET Z
0008E5 35 1043 1703 UP1 DEC (HL)
0008E6 C9 1044 1704 RET
1705 *
1706 *
0008E7 2B 1045 1707 BKSP DEC HL BACKSPACE
0008E8 7E 1046 1708 LD A,(HL)
0008E9 87 1047 1709 OR A
0008EA 20F9 1048 1710 JR NZ,UP1
0008EC 3A4FFB 1049 1711 LD A,(XLEN)
0008EF 3D 1050 1712 DEC A
0008F0 77 1051 1713 LD (HL),A
0008F1 23 1052 1714 INC HL
0008F2 35 1053 1715 DEC (HL)
0008F3 F0 1054 1716 RET P
1717 *
1718 *
0008F4 3600 1055 1719 HOME LD (HL),0 CURSOR HOME
1720 *
1721 *
0008F6 2B 1056 1722 CRET DEC HL CARRIAGE RETURN
0008F7 3600 1057 1723 LD (HL),0
0008F9 C9 1058 1724 RET
1725 *
1726 *
0008FA 5F 1059 1727 CL30 LD E,A SET 80*24 MODE
0008FB C0310E 1060 1728 CALL CHK80
0008FE C0 1061 1729 RET NZ IF NO CARD, DISALLOW 80 COLUMN MODE
0008FF 78 1062 1730 LD A,E
000900 0E50 1063 1731 LD C,H'50'
000902 110000 1064 1732 LD DE,0
000905 C8D7 1065 1733 SET 2,A
000907 21A010 1066 1734 LD HL,H'10A0' REPEAT DELAY FOR 80 COLUMN MODE
00090A 1814 1067 1735 JR SET322
1736 *
1737 *
00090C 0E28 1068 1738 CL40 LD C,H'28' SET 40-COLUMN MODE
00090E 119303 1069 1739 LD DE,H'0398'
000911 21A010 1070 1740 LD HL,H'10A0' 40 COL REPEAT DELAYS
000914 1803 1071 1741 JR SET321
1742 *
1743 *
000916 0E20 1072 1744 CL32 LD C,H'20' SET 32-COLUMN MODE
000918 11E002 1073 1745 LD DE,H'02E0'

```

LOC OBJECT M STAT E LINE SOURCE LINE

000918 21A81C	1074	1746	LD	HL,H'1CA8'	32 COL REPEAT DELAYS
00091E C897	1075	1747 SET321	RES	2,A	ENSURE NOT 80 COLUMN MODE
000920 C88F	1076	1748 SET322	RES	7,A	
000922 3245F8	1077	1749	LD	(PCFLG5),A	
000925 2242F8	1078	1750	LD	(RPTLNG),HL	SET REPEAT DELAYS
000928 214FF8	1079	1751	LD	HL,XLEN	
000928 71	1080	1752	LD	(HL),C	
00092C ED534CF8	1081	1753	LD	(SCRLSZ),DE	
		1754 *			
		1755 *			
000930 C857	1082	1756 CLSCR	BIT	2,A	
000932 201E	1083	1757	JR	NZ,CL580	CKEAK FOR 80 COLUMN MODE
000934 01003C	1084	1758	LD	BC,VTEXT	CLEAR VDU TABLE AREA
		1759	SREG		
000937 E7	1085	+	RST	H'20'	SET VDP REGISTER ROUTINE
		+	ENDM		
000938 01C003	1086	1760	LD	BC,960	HAVE SIZE OF AREA IN BC
000938 1E20	1087	1761	LD	E,H'20'	
00093D CDA60C	1088	1762	CALL	CLR	
		1763	SREG		N.B. BC=0
000940 E7	1089	+	RST	H'20'	SET VDP REGISTER ROUTINE
		+	ENDM		
000941 0618	1090	1764	LD	B,H'18'	
000943 CDAF0C	1091	1765	CALL	CLR0	CLEAR PATTERN TABLE
000946 0620	1092	1766	LD	B,H'20'	
		1767	SREG		
000948 E7	1093	+	RST	H'20'	SET VDP REGISTER ROUTINE
		+	ENDM		
000949 CD9F0C	1094	1768	CALL	CLRC	RESET COLOUR TABLE
00094C AF	1095	1769	XOR	A	
00094D CDB00D	1096	1770	CALL	HANSET	INITIALISE PATTERN NAME TABLE
000950 1815	1097	1771	JR	C83	
		1772 *			
		1773 *			
000952 014000	1098	1774 CL580	LD	BC,DP80	SET BC TO 80 COLUMN RAM
000955 3E20	1099	1775 C81	LD	A,H'20'	
000957 ED79	1100	1776 C82	OUT	(C),A	
000959 10FC	1101	1777	DJNZ	C82	
00095B 0C	1102	1778	INC	C	
00095C 3E48	1103	1779	LD	A,DP80+8	
00095E B9	1104	1780	CP	C	
00095F 20F4	1105	1781	JR	NZ,C81	
000961 210000	1106	1782	LD	HL,0	
000964 CD3000	1107	1783	CALL	SETSCR	ZERO SCROLL OFFSET
000967 2148F8	1108	1784 C83	LD	HL,VROW	
00096A 1883	1109	1785	JR	H0NE	
		1786 *			
		1787 *			
00096C 2B	1110	1788 CLRLNE	DEC	HL	CLEAR LINE, DONE AS CR
00096D 3600	1111	1789	LD	(HL),0	FOLLOWED BY CANCEL
00096F 23	1112	1790	INC	HL	
		1791 *			
		1792 *			
000970 2B	1113	1793 CANCEL	DEC	HL	CLEAR TO END-OF-LINE
000971 E5	1114	1794 CANCEL	PUSH	HL	
000972 2A4AF8	1115	1795	LD	HL,(VCOL)	SAVE CURSOR LOCATION

LOC	OBJECT	H	STAT	E	LINE	SOURCE	LINE	
000975	E3	1116	1796	EX	(SP),HL			SAVE OLD CURSOR, GET POINTER BACK
000976	3E20	1117	1797	CANC2	LD	A,H'20'		
000978	CDCE0A	1118	1798	CALL	VDUT			PUT SPACE OUT
000978	3A4FFB	1119	1799	LD	A,(XLEN)			
00097E	34	1120	1800	INC	(HL)			
00097F	BE	1121	1801	CP	(HL)			
000980	20F4	1122	1802	JR	NZ,CANC2			
000982	E1	1123	1803	CANC3	POP	HL		
000983	224AFB	1124	1804	LD	(VCOL),HL			SET CURSOR BACK
000986	C9	1125	1805	RET				
			1806 *					
			1807 *					
000987	28	1126	1808	CLRTEN	DEC	HL		
000988	E5	1127	1809	PUSH	HL			
000989	CD7109	1128	1810	CALL	CANC1			CLEAR TO END-OF-SCREEN : END-OF-LINE FIRST
00098C	E3	1129	1811	EX	(SP),HL			SAVE CURSOR AGAIN
00098D	3600	1130	1812	LD	(HL),0			
00098F	3E18	1131	1813	CT1	LD	A,H'18'		
000991	23	1132	1814	INC	HL			
000992	34	1133	1815	INC	(HL)			
000993	96	1134	1816	SUB	(HL)			
000994	28EC	1135	1817	JR	Z,CANC3			
000996	28	1136	1818	DEC	HL			
000997	E5	1137	1819	PUSH	HL			
000998	CD7109	1138	1820	CALL	CANC1			THEN DO SOME CLRLINEs
000998	E1	1139	1821	POP	HL			
00099C	18F1	1140	1822	JR	CT1			
			1823 *					
			1824 *					
00099E	7E	1141	1825	CHDELL	LD	A,(HL)		DELETE CHARACTER TO LEFT
00099F	28	1142	1826	DEC	HL			
0009A0	B6	1143	1827	OR	(HL)			
0009A1	C3	1144	1828	RET	Z			NO DEL IF AT START OF SCREEN
0009A2	3E08	1145	1829	LD	A,B5			
			1830	OUTCH				
0009A4	D7	1146	+	RST	H'10'			CHARACTER OUTPUT ROUTINE
			+	ENDM				
0009A5	23	1147	1831	INC	HL			
			1832 *					
			1833 *					
0009A6	28	1148	1834	CHDELR	DEC	HL		DELETE CHARACTER FROM RIGHT
0009A7	E5	1149	1835	PUSH	HL			
			1836	MCAL	ZSCURS			
0009A8	CF	1150	+	RST	H'08'			MOS FUNCTION CALL ROUTINE
0009A9	D1		+	DATA	ZSCURS			FUNCTION NUMBER
			+	ENDM				
0009AA	2A4AFB	1151	1837	LD	HL,(VCOL)			
0009AD	E3	1152	1838	EX	(SP),HL			
0009AE	3A4FFB	1153	1839	CDEL2	LD	A,(XLEN)		
0009B1	3D	1154	1840	DEC	A			
0009B2	8E	1155	1841	CP	(HL)			
0009B3	2814	1156	1842	JR	Z,CDEL5			
0009B5	03	1157	1843	INC	BC			
0009B6	3A45FB	1158	1844	LD	A,(PCFLGS)			
0009B9	C857	1159	1845	BIT	2,A			
0009BB	2804	1160	1846	JR	Z,CDEL3			

LOC OBJECT M STATE LINE SOURCE LINE

0009BD 08	1161	1847	DEC	BC	
0009BE CDCF08	1162	1848	CALL	INC30	
		1849 CDEL3	MCAL	ZVRIN	
0009C1 CF	1163	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0009C2 C2		+	DATA	ZVRIN	FUNCTION NUMBER
		+	ENDM		
0009C3 CDCE0A	1164	1850	CALL	VOUT	PRINT NEXT CHAR. AT CURRENT POSITION
0009C6 34	1165	1851	INC	(HL)	
0009C7 18E5	1166	1852	JR	CDEL2	
0009C9 3E20	1167	1853 CDEL5	LD	A, ''	END WITH SPACE FILL
0009CB CDCE0A	1168	1854	CALL	VOUT	
0009CE 1882	1169	1855	JR	CANC3	
		1856 *			
		1857 *			
0009D0 28	1170	1858 INSERT	DEC	HL	INSERT CHARACTER
0009D1 E5	1171	1859	PUSH	HL	
		1860	MCAL	ZSCURS	
0009D2 CF	1172	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0009D3 D1		+	DATA	ZSCURS	FUNCTION NUMBER
		+	ENDM		
0009D4 2A4AFB	1173	1861	LD	HL, (VCOL)	
0009D7 E3	1174	1862	EX	(SP), HL	
0009D8 1E20	1175	1863	LD	E, ''	
		1864 CINS1	MCAL	ZVRIN	
0009DA CF	1176	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
0009DB C2		+	DATA	ZVRIN	FUNCTION NUMBER
		+	ENDM		
0009DC 03	1177	1865	INC	BC	
0009DD F5	1178	1866	PUSH	AF	
0009DE 3A45FB	1179	1867	LD	A, (PCFLGS)	
0009E1 C857	1180	1868	BIT	2, A	
0009E3 2804	1181	1869	JR	Z, CINS2	
0009E5 08	1182	1870	DEC	BC	
0009E6 CDCF08	1183	1871	CALL	INC30	
0009E9 3A4FFB	1184	1872 CINS2	LD	A, (XLEN)	
0009EC BE	1185	1873	CP	(HL)	
0009ED 2809	1186	1874	JR	Z, CINS3	
0009EF 78	1187	1875	LD	A, E	
0009F0 CDCE0A	1188	1876	CALL	VOUT	PRINT PREVIOUS LOCATION
0009F3 34	1189	1877	INC	(HL)	
0009F4 F1	1190	1878	POP	AF	
0009F5 5F	1191	1879	LD	E, A	SAVE NEXT LOCATION
0009F6 18E2	1192	1880	JR	CINS1	
0009F8 F1	1193	1881 CINS3	POP	AF	
0009F9 1887	1194	1882	JR	CANC3	
		1883 *			
		1884 *			
0009FB 2A4AFB	1195	1885 PRNTSC	LD	HL, (VCOL)	DUMP SCREEN CONTENTS TO PRINTER
0009FE E5	1196	1886	PUSH	HL	SAVE CURSOR POSITION
0009FF AF	1197	1887	XOR	A	ZERO VCOL & VROW
000A00 47	1198	1888	LD	B, A	
000A01 4F	1199	1889	LD	C, A	
000A02 CD270A	1200	1890 PSC1	CALL	PCALF	DO CR, LF TO PRINTER AFTER EACH LINE
000A05 ED434AFB	1201	1891 PSC2	LD	(VCOL), BC	
000A09 C5	1202	1892	PUSH	BC	
		1893	MCAL	ZSCURS	

LOC OBJECT # STAT E LINE SOURCE LINE

000A0A CF	1203	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000A0B D1		+	DATA	ZSCURS	FUNCTION NUMBER
		+	ENDM		
	1894		MCAL	ZVRIN	
000A0C CF	1204	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000A0D C2		+	DATA	ZVRIN	FUNCTION NUMBER
		+	ENDM		
	1895		MCAL	ZPQUT	
000A0E CF	1205	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000A0F 9F		+	DATA	ZPQUT	FUNCTION NUMBER
		+	ENDM		
000A10 C1	1206	1896	POP	9C	
000A11 ED584FFB	1207	1897	LD	DE,(XLEN)	GET COLUMN & ROW SIZES
000A15 1618	1208	1898	LD	D,H'18'	
000A17 0C	1209	1899	INC	C	
000A18 79	1210	1900	LD	A,C	
000A19 88	1211	1901	CP	E	
000A1A 20E9	1212	1902	JR	NZ,PSC2	
000A1C 0E00	1213	1903	LD	C,0	
000A1E 04	1214	1904	INC	B	
000A1F 78	1215	1905	LD	A,B	
000A20 8A	1216	1906	CP	D	DO ANOTHER ROW
000A21 200F	1217	1907	JR	NZ,PSC1	
000A23 E1	1218	1908	POP	HL	
000A24 224AFB	1219	1909	LD	(VCOL),HL	
000A27 F5	1220	1910 PCRLF	PUSH	AF	
000A28 3E0D	1221	1911	LD	A,CR	END WITH FINAL CR, LF TO PRINTER
		1912	MCAL	ZPQUT	
000A2A CF	1222	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000A2B 9F		+	DATA	ZPQUT	FUNCTION NUMBER
		+	ENDM		
000A2C 3E0A	1223	1913	LD	A,LF	
		1914	MCAL	ZPQUT	
000A2E CF	1224	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000A2F 9F		+	DATA	ZPQUT	FUNCTION NUMBER
		+	ENDM		
000A30 F1	1225	1915	POP	AF	
000A31 C9	1226	1916	RET		
		1917 *			
		1918 *			
000A32 EE80	1227	1919 INVERT XOR	H'80'	TOGGLE INVERSE VIDEO FLAG	
000A34 180E	1228	1920	JR	PCSR	
		1921 *			
		1922 *			
000A36 CBCF	1229	1923 PFSET SET	1,A	SET PRINTER FLAG	
000A38 180A	1230	1924	JR	PCSR	
		1925 *			
		1926 *			
000A3A C887	1231	1927 CFRST RES	0,A	RESET CURSOR FLASH	
000A3C 1806	1232	1928	JR	PCSR	
		1929 *			
		1930 *			
000A3E CBC7	1233	1931 CFSET SET	0,A	SET CURSOR FLASH	
000A40 1802	1234	1932	JR	PCSR	
		1933 *			
		1934 *			

LOC OBJECT \* STATE LINE SOURCE LINE

000A42 C88F	1235	1935	PFRST	RES	1,A	RESET PRINTER FLAG
000A44 3245FB	1236	1936	PCSR	LD	(PCFLGS),A	
000A47 C9	1237	1937	NODSP	RET		
		1938	*			
		1939	*			
000A48 2145FB	1238	1940	CUSADD	LD	HL,PCFLGS	DIRECT CURSOR ADDRESSING
000A48 C8DE	1239	1941		SET	3,(HL)	SET FLAG FOR X NEXT
000A4D C9	1240	1942		RET		
000A4E C866	1241	1943	CUSAD1	BIT	4,(HL)	TEST FOR X OR Y
000A50 2008	1242	1944	JR		HZ,CUSAD2	JUMP FOR Y
000A52 C8E6	1243	1945		SET	4,(HL)	SET FLAG FOR Y NEXT
000A54 214FFB	1244	1946		LD	HL,XLEN	
000A57 BE	1245	1947		CP	(HL)	CHECK FOR WITHIN RANGE
000A58 D0	1246	1948		RET	NC	RETURN IF OUTSIDE
000A59 324AFB	1247	1949		LD	(VCOL),A	SET X COORDINATE
000A5C C9	1248	1950		RET		
000A5D C89E	1249	1951	CUSAD2	RES	3,(HL)	SET Y COORDINATE
000A5F C8A6	1250	1952		RES	4,(HL)	-CANCEL FLAGS
000A61 FE18	1251	1953		CP	H'18'	CHECK FOR OUT OF RANGE
000A63 D0	1252	1954		RET	NC	
000A64 3248FB	1253	1955		LD	(VROW),A	SET Y COORDINATE
000A67 C9	1254	1956		RET		

1957 \*

1958 \*

1959 \*

1960 \*

1961 \*

VDUTBL - TABLE OF VDU ACTION ROUTINES

000A63 470A	1962	VDUTBL	ACON	NODSP	
000A6A FB09	1963		ACON	PRNTSC	1 - DUMP SCREEN TO PRINTER
000A6C 470A	1964		ACON	NODSP	2 - (RESERVED)
000A6E 470A	1965		ACON	NODSP	
000A70 CF08	1966		ACON	FWD	4 - CURSOR RIGHT
000A72 470A	1967		ACON	NODSP	
000A74 A609	1968		ACON	CHDELX	6 - DELETE FROM RIGHT
000A76 710F	1969		ACON	BELL	7 - BELL
000A78 E703	1970		ACON	BKSP	8 - CURSOR LEFT
000A7A 470A	1971		ACON	NODSP	
000A7C 0908	1972		ACON	DOWN	10- CURSOR DOWN
000A7E E208	1973		ACON	UP	11- CURSOR UP
000A80 3009	1974		ACON	CLSCR	12- CLEAR SCREEN
000A82 F608	1975		ACON	CRET	13- CARRIAGE RETURN
000A84 0C09	1976		ACON	CL40	14- CLEAR TO 40 COLUMNS
000A86 1609	1977		ACON	CL32	15- CLEAR TO 32 COLUMNS
000A88 FA03	1978		ACON	CL80	16- CLEAR TO 80 COLUMNS
000A8A 3E0A	1979		ACON	CFSET	17- CURSOR FLASHING
000A8C 360A	1980		ACON	PFSET	18- PRINTER ECHO
000A8E 420A	1981		ACON	PFRST	19- PRINTER OFF
000A90 3A0A	1982		ACON	CFRST	20- CURSOR OFF
000A92 7009	1983		ACON	CANCEL	21- CLEAR TO END-OF-LINE
000A94 8709	1984		ACON	CLRTEN	22- CLEAR TO END-OF-SCREEN
000A96 320A	1985		ACON	INVERT	23- INVERT VIDEO TEXT
000A98 6C09	1986		ACON	CLRLE	24- CLEAR WHOLE LINE
000A9A 9E09	1987		ACON	CHDELL	25- DELETE TO LEFT
000A9C D009	1988		ACON	INSERT	26- INSERT
000A9E 470A	1989		ACON	NODSP	
000AA0 470A	1990		ACON	NODSP	

LOC OBJECT M STAT E LINE SOURCE LINE

000A2 480A	1991	ACON	CUSADO	29- DIRECT CURSOR ADDRESSING
000A4 F403	1992	ACON	HONE	30- HONE CURSOR
000A6 470A	1993	ACON	HODSP	
	1994 *			
	1995 *			
	1996 *	SCRDSP	-- SHOW SCREEN CHARACTER, WHEN CURSOR OFF OR REMOVED	
	1997 *			
	1998 *			
000A8 3A45FB	1255	1999	SCRDSP LD A,(PCFLGS)	
000A8 C857	1256	2000	-BIT 2,A	
000AD 231C	1257	2001	JR Z,C02	
000AF 1E20	1258	2002	LD E,H'20'	
000B1 1808	1259	2003	JR CF80	
	2004 *			
	2005 *			
	2006 *	CURDSP	-- SHOW CURSOR CHARACTER UNLESS FLAG SET TO REMOVE IT	
	2007 *			
	2008 *			
000B3 3A45FB	1260	2009	CURDSP LD A,(PCFLGS)	
000B6 C857	1261	2010	BIT 2,A	
000B8 1F	1262	2011	RRR	
000B9 280B	1263	2012	JR Z,C01	
000B8 D0	1264	2013	RET NC	
000B8C 1E00	1265	2014	LD E,0	
000B8E 3E0A	1266	2015	CF80 LD A,10	
000AC0 D348	1267	2016	OUT (RG6845),A	SET/RESET CURSOR FOR 80-COL
000AC2 78	1268	2017	LD A,E	
000AC3 D349	1269	2018	OUT (OT6845),A	
000AC5 C9	1270	2019	RET	
000AC6 3A3FFB	1271	2020	CD1 LD A,(CUSCODE)	DISPLAY CURSOR CODE
000AC9 3803	1272	2021	JR C,VOUT	MOVE TO REQUIRED PG LOCATION
000ACB 3A47FB	1273	2022	CD2 LD A,(SCRNCD)	(BC) AT TEXT POSN LOC
	2023 *			
	2024 *			
	2025 *	VOUT	-- PLACE CHAR AT CURRENT POS. ACCORDING TO 32 OR 40 COL MODE	
	2026 *			
	2027 *			
000ACE E5	1274	2028	VOUT PUSH HL	
000ACF D5	1275	2029	PUSH DE	
000AD0 C5	1276	2030	PUSH BC	
000AD1 47	1277	2031	LD B,A	
000AD2 3A45FB	1278	2032	LD A,(PCFLGS)	
000AD5 C857	1279	2033	BIT 2,A	
000AD7 202D	1280	2034	JR NZ,VOUT80	
000AD9 3A4FFB	1281	2035	LD A,(XLEN)	
000ADC FE28	1282	2036	CP 40	
000ADE 78	1283	2037	LD A,B	
000ADF 282F	1284	2038	JR Z,VOUT40	JUMP IF 40-COLUMN MODE
000AE1 CD4C0C	1285	2039	CALL SETCR5	MOVE TO REQ'D PG LOCATION
	2040		VROUT	(BC) AT TEXT POSITION
000AE4 F7	1286	+	RST H'30'	VIDEO RAM WRITE
		+	ENDM	
000AE5 E5	1287	2041	PUSH HL	
000AE6 210800	1288	2042	LD HL,8	
000AE9 C1	1289	2043	POP BC	
000AEA E5	1290	2044	PUSH HL	

LOC OBJECT M STAT E LINE SOURCE LINE

000AEB D5	1291	2045	PUSH	DE	
000AEC 55	1292	2046	LD	D,L	
000AED CD900D	1293	2047	CALL	YMIN	
000AF0 45	1294	2048	LD	B,L	NB, L=8 PUT OUT PATTERN
000AF1 2145FB	1295	2049	LD	HL,PCFLGS	
000AF4 C37E	1296	2050	BIT	7,(HL)	
000AF6 2809	1297	2051	JR	Z,V0322	
		2052 *			
		2053 *			
000AF8 21E0FB	1298	2054	LD	HL,V0P8UF	
000AFB 7E	1299	2055 V0321	LD	A,(HL)	DO INVERSE FOR 32-COLS
000AFC 2F	1300	2056	CPL		
000AFD 77	1301	2057	LD	(HL),A	
000AFE 23	1302	2058	INC	HL	
000AFF 10FA	1303	2059	DJNZ	V0321	
00801 C1	1304	2060 V0322	POP	BC	
00802 E1	1305	2061	POP	HL	
00803 C38008	1306	2062	JP	VEND2	
		2063 *			
		2064 *			
00806 78	1307	2065 V0UT30	LD	A,B	
00807 CD130C	1308	2066	CALL	SETC80	
00808 CD0908	1309	2067	CALL	DISOUT	OUTPUT CHARACTER
00809 D3C208	1310	2068	JP	VEND3	
		2069 *			
		2070 *			
00810 ED484AF8	1311	2071 V0UT40	LD	BC,(V0COL)	GET 40-COLUMN ROW/COLUMN
00814 59	1312	2072	LD	E,C	SAVE TEXT COLUMN IN E
00815 F5	1313	2073	PUSH	AF	
00816 79	1314	2074	LD	A,C	
00817 E603	1315	2075	AND	3	
00819 C839	1316	2076	SRL	C	
0081B C839	1317	2077	SRL	C	NOW HAVE BLOCK IN C, REMAINDER IN A
0081D 08	1318	2078	EX	AF,AF'	
0081E 79	1319	2079	LD	A,C	
0081F 87	1320	2080	ADD	A,A	
00820 81	1321	2081	ADD	A,C	
00821 4F	1322	2082	LD	C,A	HAVE TRUE COLUMN FOR BLOCK START
00822 F1	1323	2083	POP	AF	RETRIEVE CHARACTER
00823 CD510C	1324	2084	CALL	SETC1	
		2085	VROUT		SET UP TEXT LOCATION
00826 F7	1325	+	RST	H'30'	VIDEO RAM WRITE
		+	ENDM		
00827 08	1326	2086	EX	AF,AF'	GET REMAINDER BACK
00828 010800	1327	2087	LD	BC,8	
00828 C84F	1328	2088	BIT	1,A	IF POSITION 2, OR 3, MOVE TO NEXT CHAR
0082D EB	1329	2089	EX	DE,HL	
0082E 2801	1330	2090	JR	Z,V401	
00830 09	1331	2091	ADD	HL,BC	
00831 FE03	1332	2092 V401	CP	3	
00833 282D	1333	2093	JR	Z,V4103	
00835 F5	1334	2094	PUSH	AF	
00836 4D	1335	2095	LD	C,L	
00837 44	1336	2096	LD	B,H	HAVE PATTERN PTR IN BC & HL
00838 D5	1337	2097	PUSH	DE	SAVE CHAR. POINTER
00839 1610	1338	2098	LD	D,16	

LOC OBJECT M STAT E LINE SOURCE LINE

000838	CD900D	1339	2099	CALL	VMIN	GET TWO PATTERNS INTO VOP8UF
00083E	C1	1340	2100	POP	BC	CHAR. GEN. STUFF INTO BC
00083F	F1	1341	2101	POP	AF	
000840	E5	1342	2102	PUSH	HL	SAVE PATTERN POINTER
000841	21E0FB	1343	2103	LD	HL,VOP8UF	
000844	1608	1344	2104	LD	D,3	
000846	B7	1345	2105	OR	A	
000847	233D	1346	2106	JR	Z,V4100	
000849	3D	1347	2107	DEC	A	
00084A	284F	1348	2108	JR	Z,V4101	
00084C	CDC60B	1349	2109 V4102	CALL	VRCIN	
00084F	E6FC	1350	2110	AND	H'FC'	
000851	C5	1351	2111	PUSH	BC	
000852	013FF0	1352	2112	LD	BC,H'F03F'	
000855	07	1353	2113	RLCA		
000856	07	1354	2114	RLCA		
000857	CDF80B	1355	2115	CALL	MASK	
00085A	C1	1356	2116	POP	BC	
00085B	23	1357	2117	INC	HL	
00085C	03	1358	2118	INC	BC	
00085D	15	1359	2119	DEC	D	
00085E	20EC	1360	2120	JR	NZ,V4102	
000860	184A	1361	2121	JR	YEND	
000862	09	1362	2122 V4103	ADD	HL,BC	GET 3rd COLUMN ONLY ON 4th CHARACTER
000863	44	1363	2123	LD	B,H	GET LAST POSITION ON 4th CHARACTER
000864	4D	1364	2124	LD	C,L	
000865	D5	1365	2125	PUSH	DE	
000866	1608	1366	2126	LD	D,3	
000868	CD900D	1367	2127	CALL	VMIN	
00086B	C1	1368	2128	POP	BC	CHAR. GEN. STUFF IN BC
00086C	E5	1369	2129	PUSH	HL	
00086D	21E0F9	1370	2130	LD	HL,VOP8UF	
000870	1608	1371	2131	LD	D,3	
000872	7E	1372	2132 V41031	LD	A,(HL)	
000873	E6C0	1373	2133	AND	H'CO'	MASK BOTTOM 6 BITS
000875	5F	1374	2134	LD	E,A	
000876	CDC60B	1375	2135	CALL	VRCIN	
000879	C83F	1376	2136	SRL	A	
00087B	C83F	1377	2137	SRL	A	SHIFT DOWN NEW CHARACTER
00087D	83	1378	2138	OR	E	
00087E	77	1379	2139	LD	(HL),A	
00087F	23	1380	2140	INC	HL	
000880	03	1381	2141	INC	BC	
000881	15	1382	2142	DEC	D	
000882	20EE	1383	2143	JR	NZ,V41031	
000884	1310	1384	2144	JR	YEND	COPY LAST 3 BYTES ONLY FOR 4th CHARACTER
000886	CDC60B	1385	2145 V4100	CALL	VRCIN	
000889	E6FC	1386	2146	AND	H'FC'	
00088B	5F	1387	2147	LD	E,A	
00088C	7E	1388	2148	LD	A,(HL)	
00088D	E603	1389	2149	AND	3	MASK TOP 6 BITS
00088F	83	1390	2150	OR	E	
000890	77	1391	2151	LD	(HL),A	
000891	23	1392	2152	INC	HL	
000892	03	1393	2153	INC	BC	
000893	15	1394	2154	DEC	D	

PMOS CROSS ASSEMBLER 280

REL: 4.0 EINSTEIN M.B.S.

M0512-5

00 17 44 PAGE 45

LOC OBJECT M STATE LINE SOURCE LINE

000894	20F0	1395	2155	JR	NZ,V4100	
000896	210800	1396	2156	VEND0 LD	HL,B	8-BYTE COPY FOR 1st AND 4th CHARACTERS
000899	1814	1397	2157	JR	VEND01	
00089B	CDC603	1398	2158	V4101 CALL	VR0IN	
00089E	C5	1399	2159	PUSH	BC	
00089F	010FFC	1400	2160	LD	BC,H'FC0F'	
0008A2	A0	1401	2161	AND	B	N.B. B=H'FC'
0008A3	CDFA08	1402	2162	CALL	MASK	
0008A6	C1	1403	2163	POP	BC	
0008A7	23	1404	2164	INC	HL	
0008A8	03	1405	2165	INC	BC	
0008A9	15	1406	2166	DEC	D	
0008AA	20EF	1407	2167	JR	NZ,V4101	
0008AC	211000	1408	2168	VEND LD	HL,16	
0008AF	C1	1409	2169	VEND1 POP	BC	RETRIEVE PATTERN PTR
0008B0	E5	1410	2170	VEND2 PUSH	HL	
0008B1	C5	1411	2171	PUSH	BC	
0008B2	55	1412	2172	LD	D,L	
0008B3	CD9100	1413	2173	CALL	VNDOUT	STICK THE PATTERNS BACK OUT
0008B6	C1	1414	2174	POP	BC	
0008B7	C8E8	1415	2175	SET	S,B	MOVE TO COLOUR TABLE
			2176	SREG		SET COLOUR OF DISPLAYED CHARACTER
0008B9	E7	1416	+	RST	H'20'	SET VOP REGISTER ROUTINE
			+	ENOM		
0008BA	C1	1417	2177	POP	BC	
0008BB	3A38FB	1418	2178	LD	A,(TCOLR)	
0008BE	5F	1419	2179	LD	E,A	
0008BF	CDA60C	1420	2180	CALL	CLR	
0008C2	C1	1421	2181	VEND3 POP	BC	
0008C3	D1	1422	2182	POP	DE	
0008C4	E1	1423	2183	POP	HL	
0008C5	C9	1424	2184	RET		FINISH AFTER RESTORING THE NECESSARIES
			2185 *			
			2186 *			
			2187 *	VR0IN -- DO VRIN AND COMPLEMENT IF IN INVERSE MODE		
			2188 *			
			2189 *			
0008C6	3A45FB	1425	2190	VR0IN LD	A,(PCFLOS)	
0008C9	C37F	1426	2191	SIT	7,A	COMPLEMENT IF INVERSE MODE
			2192	VRIN		ACCESS CHAR GEN TABLE
0008CB	EF	1427	+	RST	H'28'	VIDEO RAM READ
			+	ENOM		
0008CC	C3	1428	2193	RET	Z	
0008CD	2F	1429	2194	CPL		
0008CE	C9	1430	2195	RET		
			2196 *			
			2197 *			
0008CF	04	1431	2198	INC80 INC	B	INCREMENT 80 COLUMN ADDRESS
0008D0	C0	1432	2199	RET	NZ	
0008D1	0C	1433	2200	INC81 INC	C	
0008D2	C899	1434	2201	RES	3,C	KEEP ADDRESS IN PORT 40H-47H RANGE
0008D4	C8A1	1435	2202	RES	4,C	
0008D6	C8A9	1436	2203	RES	5,C	
0008D8	C9	1437	2204	RET		
			2205 *			
			2206 *			

LOC OBJECT M STAT E LINE SOURCE LINE

```

0008D9 F5      1438 2207 DISOUT PUSH AF      DO 80 COLUMN OUTPUT WHEN ALLOWED
0008DA CDEF08  1439 2208      CALL WAIT80
0008DD F1      1440 2209      POP  AF
0008DE ED79    1441 2210      OUT  (C),A
0008EO C9      1442 2211      RET
                2212 *
                2213 *
0008E1 3A45F8  1443 2214 VIDIN LD      A,(PCFLG5)  READ FROM 80 COLUMN VIDEO RAM
0008E4 C857    1444 2215      BIT  2,A
0008E6 CA2800  1445 2216      JP   Z,VRINH
0008E9 CDEF08  1446 2217 VIDINI CALL WAIT80
0008EC ED78    1447 2218      IN   A,(C)
0008EE C9      1448 2219      RET
                2220 *
                2221 *
0008EF D84C    1449 2222 WAIT80 IN   A,(IP80)  WAIT FOR DISPLAY ENABLE LINE TO GO LOW
0008F1 1F      1450 2223      RRA
0008F2 30F8    1451 2224      JR   NC,WAIT80
0008F4 D84C    1452 2225 W81  IN   A,(IP80)
0008F6 1F      1453 2226      RRA
0008F7 30F8    1454 2227      JR   C,W81
0008F9 C9      1455 2228      RET
                2229 *
                2230 *
0008FA D5      1456 2231 MASK PUSH  DE      ROUTINE TO FORM CHARACTER WITHIN OUR
0008FB 07      1457 2232      RLCA      2-CHARACTER BLOCK
0008FC 07      1458 2233      RLCA
0008FD F5      1459 2234      PUSH AF
0008FE E60F    1460 2235      AND  H'0F'
000C00 57      1461 2236      LD   D,A
000C01 F1      1462 2237      POP  AF
000C02 E6F0    1463 2238      AND  H'F0'
000C04 5F      1464 2239      LD   E,A
000C05 7E      1465 2240      LD   A,(HL)
000C06 A0      1466 2241      AND  B      MASK BOTTOM 2 BITS
000C07 82      1467 2242      OR   D
000C08 77      1468 2243      LD   (HL),A
000C09 C800    1469 2244      SET  3,L      MOVE TO NEXT POSITION
000C0B 7E      1470 2245      LD   A,(HL)
000C0C A1      1471 2246      AND  C      MASK TOP 4 BITS
000C0D 83      1472 2247      OR   E
000C0E 77      1473 2248      LD   (HL),A
000C0F C890    1474 2249      RES  3,L
000C11 D1      1475 2250      POP  DE
000C12 C9      1476 2251      RET
                2252 *
                2253 *
                2254 *      SETC80 -- SET 80 COLUMN CURSOR, GIVEN ROW, COLUMN IN (VCOL)
                2255 *      RETURN PORT ADDRESS IN BC, PRESERVE AF
                2256 *
                2257 *
000C13 ED48AFB 1477 2258 SETC80 LD   BC,(VCOL)
000C17 F5      1478 2259 SETC81 PUSH AF
000C18 2600    1479 2260      LD   H,0
000C1A 68      1480 2261      LD   L,B
000C1B 44      1481 2262      LD   B,H

```

LOC OBJECT M STATE LINE SOURCE LINE

```

000C1C 29      1482 2263   ADD   HL,HL
000C1D 29      1483 2264   ADD   HL,HL
000C1E 29      1484 2265   ADD   HL,HL
000C1F 29      1485 2266   ADD   HL,HL
000C20 54      1486 2267   LD     D,H
000C21 5D      1487 2268   LD     E,L
000C22 29      1488 2269   ADD   HL,HL
000C23 29      1489 2270   ADD   HL,HL
000C24 19      1490 2271   ADD   HL,DE
000C25 09      1491 2272   ADD   HL,BC
000C26 ED584CFB 1492 2273   LD     DE,(SCRLSZ)   ADD OFFSET FOR SCROLL
000C2A 19      1493 2274   ADD   HL,DE
000C2B 7C      1494 2275   LD     A,H
000C2C E63F    1495 2276   AND   H,'3F'
000C2E 67      1496 2277   LD     H,A
000C2F 3E0E    1497 2278   LD     A,14          SET CURSOR ADDRESS
000C31 D348    1498 2279   OUT    (RG6845),A
000C33 0E49    1499 2280   LD     C,DT6845
000C35 ED61    1500 2281   OUT    (C),H
000C37 3C      1501 2282   INC    A
000C38 D348    1502 2283   OUT    (RG6845),A
000C3A ED69    1503 2284   OUT    (C),L
000C3C 7C      1504 2285   LD     A,H
000C3D E607    1505 2286   AND    7
000C3F 45      1506 2287   LD     B,L
000C40 4F      1507 2288   LD     C,A
000C41 C8F1    1508 2289   SET    6,C          BC NOW AT REQUIRED PORT ADDRESS
000C43 F1      1509 2290   POP    AF
000C44 C9      1510 2291   RET
                2292 *
                2293 *
                2294 *   SETCR5 - SET CURSOR LOCATION. RETURNS BC AT YOU TABLE AREA
                2295 *   HL AT CHAR. GEN. LOCATION, AND DE AT PATTERN POSITION
                2296 *   ENTER AND RETURN WITH REQ'D CHARACTER IN A
                2297 *
                2298 *
000C45 2145FB  1511 2299 SETCZ LD     HL,PCFLGS
000C48 CB56    1512 2300 BIT    2,(HL)
000C4A 20C7    1513 2301 JR     NZ,SETC90
000C4C ED4B4AFB 1514 2302 SETCR5 LD     BC,(VCOL)
000C50 59      1515 2303 SETCO LD     E,C
                2304 *
                2305 *
000C51 F5      1516 2306 SETC1 PUSH   AF          ENTER HERE WITH B=VROW, C=TEXT COL, E=VCOL
000C52 2600    1517 2307 LD     H,0
000C54 6F      1518 2308 LD     L,A          FIND CHARACTER PATTERN IN CHARACTER GENERATOR
000C55 29      1519 2309 ADD   HL,HL
000C56 29      1520 2310 ADD   HL,HL
000C57 29      1521 2311 ADD   HL,HL          3 TIMES HL
000C58 3E18    1522 2312 LD     A,VCGEN/256
000C5A 84      1523 2313 ADD   A,H          NOW WE HAVE CHARACTER TABLE START
000C5B 67      1524 2314 LD     H,A
000C5C E5      1525 2315 SETC2 PUSH   HL          SAVE IT
000C5D 2600    1526 2316 LD     H,0
000C5F 68      1527 2317 LD     L,B          GET ROW & MULTIPLY BY 32
000C60 29      1528 2318 ADD   HL,HL

```

LOC OBJECT M STAT E LINE SOURCE LINE

000C61	29	1529	2319	ADD	HL,HL	
000C62	29	1530	2320	ADD	HL,HL	
000C63	E5	1531	2321	PUSH	HL	PUSH 8*VALUE
000C64	29	1532	2322	ADD	HL,HL	
000C65	29	1533	2323	ADD	HL,HL	
000C66	E5	1534	2324	PUSH	HL	PUSH 32*VALUE
000C67	0600	1535	2325	LD	B,0	
000C69	09	1536	2326	ADD	HL,BC	COLUMN OFFSET
000C6A	010038	1537	2327	LD	BC,VPATH	GET PATTERN NAME LOCATION
000C6D	E5	1538	2328	PUSH	HL	
000C6E	09	1539	2329	ADD	HL,BC	
000C6F	E3	1540	2330	EX	(SP),HL	
000C70	C1	1541	2331	POP	BC	
			2332	VRIN		GET PATTERN POINTER FROM NAME TABLE
000C71	EF	1542	+	RST	H'28'	VIDEO RAM READ
			+	ENDM		
000C72	6F	1543	2333	LD	L,A	NOW WE HAVE TRUE PATTERN NUMBER IN HL
000C73	29	1544	2334	ADD	HL,HL	
000C74	29	1545	2335	ADD	HL,HL	
000C75	29	1546	2336	ADD	HL,HL	MULTIPLY BY 3 FOR PATTERN POSITION
000C76	C1	1547	2337	POP	BC	RETRIEVE 32*VALUE IN BC
000C77	E3	1548	2338	EX	(SP),HL	
000C78	EB	1549	2339	EX	DE,HL	AND 8*VALUE IN DE
000C79	2600	1550	2340	LD	H,0	TEXT COLOUR NUMBER IN HL
000C7B	3A4FFB	1551	2341	LD	A,(XLEN)	
000C7E	FE28	1552	2342	CP	40	
000C80	2001	1553	2343	JR	NZ,SETC3	
000C82	19	1554	2344	ADD	HL,DE	ADD 3*VALUE FOR EFFECTIVE 40*VALUE
000C83	09	1555	2345 SETC3	ADD	HL,BC	
000C84	01003C	1556	2346	LD	BC,VTEXT	NOW GET TRUE TEXT POSITION
000C87	09	1557	2347	ADD	HL,BC	
000C88	D1	1558	2348	POP	DE	HAVE PATTERN POSITION IN D
000C89	E3	1559	2349	EX	(SP),HL	HAVE CHARACTER GENERATOR POSITION IN HL
000C8A	C1	1560	2350	POP	BC	HAVE TEXT POSITION IN BC
000C8B	F1	1561	2351	POP	AF	
000C8C	C9	1562	2352	RET		
			2353 *			
			2354 *			
			2355 *			SREG - LATCH VOP FOR VRIN
			2356 *			
			2357 *			
000C8D	F5	1563	2358 SREG1	PUSH	AF	
000C8E	79	1564	2359	LD	A,C	
000C8F	D309	1565	2360	OUT	(H'09'),A	
000C91	F5	1566	2361	PUSH	AF	SMALL DELAY
000C92	F1	1567	2362	POP	AF	
000C93	78	1568	2363	LD	A,B	
000C94	D309	1569	2364	OUT	(H'09'),A	
000C96	F1	1570	2365	POP	AF	
000C97	C9	1571	2366	RET		
			2367 *			
			2368 *			
000C98	CBF7	1572	2369 SREG2	SET	6,A	END OF SREG ROUTINE
000C9A	D309	1573	2370	OUT	(H'09'),A	
000C9C	00	1574	2371	NOP		
000C9D	F1	1575	2372	POP	AF	

LOC OBJECT M STAT E LINE SOURCE LINE

000C9E C9	1576	2373	RET		
		2374 *			
		2375 *			
		2376 *		CLEAR - VRAM CLEAR ROUTINES	
		2377 *			
		2378 *			
000C9F 010018	1577	2379 CLRC	LD	BC,H'1800'	START AT PATTERN TABLE
000CA2 3A38F8	1578	2380 CLRC1	LD	A,(TCCLR)	
000CA5 5F	1579	2381	LD	E,A	RESET COLOUR TABLE
000CA6 78	1580	2382 CLR	LD	A,E	
000CA7 D308	1581	2383	OUT	(H'08'),A	
000CA9 08	1582	2384	DEC	BC	
000CAA 78	1583	2385	LD	A,B	
000CAB 81	1584	2386	OR	C	
000CAC 20F8	1585	2387	JR	NZ,CLR	
000CAE C9	1586	2388	RET		
		2389 *			
		2390 *			
000CAF 1E00	1587	2391 CLRO	LD	E,0	
000CB1 18F3	1588	2392	JR	CLR	
		2393 *			
		2394 *			
		2395 *		SCROLL - SCROLL VRAM AND CLEAR BOTTOM LINE	
		2396 *			
		2397 *			
000CB3 2145FB	1589	2398 SCROLL	LD	HL,PCFLGS	
000CB6 C856	1590	2399	BIT	2,(HL)	
000CB8 2043	1591	2400	JR	NZ,SCRL80	
000CBA 010038	1592	2401	LD	BC,VPATH	GET START OF PATTERN NAME TABLE
		2402	VRIN		
000CBD EF	1593	+	RST	H'28'	VIDEO RAM READ
		+	ENDM		
000CBE F5	1594	2403	PUSH	AF	
000CBF C620	1595	2404	ADD	A,32	ADD 32 TO FIRST CHARACTER VALUE
000CC1 C08000	1596	2405	CALL	NAMSET	AND RE-WRITE TABLE
000CC4 F1	1597	2406	POP	AF	GET FIRST CHARACTER OF NAME TABLE
000CC5 2600	1598	2407	LD	H,0	
000CC7 6F	1599	2408	LD	L,A	CALCULATE ADDR. OF TOP LINE START
000CC8 29	1600	2409	ADD	HL,HL	
000CC9 29	1601	2410	ADD	HL,HL	
000CCA 29	1602	2411	ADD	HL,HL	
000CCB 44	1603	2412	LD	B,H	
000CCC 4D	1604	2413	LD	C,L	TOP-LINE IN BC
000CCD C5	1605	2414	PUSH	BC	
000CCE C05D00	1606	2415	CALL	VCOPYZ	NOW MUST COPY LINES WHICH HAVE CHANGED
		2416	SREG		
000CD1 E7	1607	+	RST	H'20'	SET VDP REGISTER ROUTINE
		+	ENDM		
000CD2 010001	1608	2417	LD	BC,H'0100'	
000CD5 C0AF0C	1609	2418	CALL	CLRO	CLEAR BOTTOM LINE
000CD8 C1	1610	2419	POP	BC	
000CD9 C8E8	1611	2420	SET	S,B	NOW DO COLOUR TABLE TOO
000CDB C05D00	1612	2421	CALL	VCOPYZ	
		2422	SREG		
000CDE E7	1613	+	RST	H'20'	SET VDP REGISTER ROUTINE
		+	ENDM		

LOC	OBJECT	M	STAT	E LINE	SOURCE LINE	
000CDF	010001	1614	2423	LD	BC,H'0100'	
000CE2	CDR20C	1615	2424	CALL	CLRC1	
000CE5	2A4FFB	1616	2425	LD	HL,(XLEN)	
000CE8	2600	1617	2426	LD	H,0	
000CEA	E5	1618	2427	PUSH	HL	SAVE LINE LENGTH
000CEB	11003C	1619	2428	LD	DE,VTEXT	
000CEE	19	1620	2429	ADD	HL,DE	
000CEF	E8	1621	2430	EX	DE,HL	
000CF0	44	1622	2431	LD	B,H	
000CF1	40	1623	2432	LD	C,L	
000CF2	2A4CFB	1624	2433	LD	HL,(SCRLSZ)	
000CF5	CD680D	1625	2434	CALL	VCOPY	SCROLL TEXT POSITION TABLE
000CF8	C1	1626	2435	POP	BC	
000CF9	1E20	1627	2436	LD	E,H'20'	
000CFB	18A9	1628	2437	JR	CLR	SPACES AT BOTTOM
			2438 *			
			2439 *			
000CFD	2A4CFB	1629	2440	SCRL80 LD	HL,(SCRLSZ)	HERE FOR 80 COLUMN STUFF
000D00	015000	1630	2441	LD	BC,80	
000D03	09	1631	2442	ADD	HL,BC	
000D04	7C	1632	2443	LD	A,H	
000D05	E607	1633	2444	AND	7	
000D07	67	1634	2445	LD	H,A	ENSURE START WITHIN MEMORY AREA
000D08	CD300D	1635	2446	CALL	SETSCR	SET NEW SCROLL OFFSET
000D08	010017	1636	2447	LD	BC,H'1700'	COPY 24th ROW TO 25th AND BLANK 24th ROW
000D0E	CD170C	1637	2448	CALL	SETC81	
000D11	1E50	1638	2449	LD	E,80	
000D13	C5	1639	2450	SC83 PUSH	BC	
000D14	CDE908	1640	2451	CALL	VIDINI	GET CHARACTERS
000D17	57	1641	2452	LD	D,A	
000D18	3E20	1642	2453	LD	A,H'20'	
000D1A	CD0908	1643	2454	CALL	DISOUT	OUTPUT SPACE
000D1D	3E50	1644	2455	LD	A,80	ADD 80 TO BC FOR 25th ROW
000D1F	80	1645	2456	ADD	A,B	
000D20	47	1646	2457	LD	B,A	
000D21	DCD108	1647	2458	CALL	C,INC81	
000D24	7A	1648	2459	LD	A,D	
000D25	CD0908	1649	2460	CALL	DISOUT	OUTPUT CHARACTER TO 25th ROW
000D28	C1	1650	2461	POP	BC	
000D29	CDCF0B	1651	2462	CALL	INC80	NEXT SCREEN POSITION
000D2C	1D	1652	2463	DEC	E	
000D2D	20E4	1653	2464	JR	NZ,SC83	
000D2F	C9	1654	2465	RET		
			2466 *			
			2467 *			
000D30	3E0C	1655	2468	SETSCR LD	A,12	SET SCROLL OFFSET BY PLUGGING IN
000D32	D348	1656	2469	OUT	(RG6845),A	TO R12, R13 OF 6845 CTC
000D34	0E49	1657	2470	LD	C,DT6845	
000D36	ED61	1658	2471	OUT	(C),H	
000D38	3C	1659	2472	INC	A	
000D39	D348	1660	2473	OUT	(RG6845),A	
000D3B	ED69	1661	2474	OUT	(C),L	
000D3D	224CFB	1662	2475	LD	(SCRLSZ),HL	
000D40	C9	1663	2476	RET		
			2477 *			
			2478 *			
				STATUS LINE-	COPY FROM RAM (DE) TO STATUS LINE (25th ROW) 80 COL	

PMOS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.O.S.

MO512:5

00:17:44 PAGE: 51

LQC OBJECT M STAT E LINE SOURCE LINE

```

2479 *
000041 2145F8 1664 2480 STAT80 LD HL,PCFLG5 TEST FOR 80 CUL
000044 C856 1665 2481 BIT 2,(HL)
000046 C8 1666 2482 RET Z RETURN IF NOT
000047 010013 1667 2483 LD BC,H'1800' 25th ROW,COLUMN 0
00004A D5 1668 2484 PUSH DE
00004B CD170C 1669 2485 CALL SETC21 SET CURSOR
00004E D1 1670 2486 POP DE
00004F 2650 1671 2487 LD H,80
000051 1A 1672 2488 STAT1 LD A,(DE) COPY LINE FROM (DE) TO 80 CUL
000052 CD0908 1673 2489 CALL DISOUT OUTPUT CHARACTER
000055 CDCF08 1674 2490 CALL INC30
000058 13 1675 2491 INC DE
000059 25 1676 2492 DEC H
00005A 20F5 1677 2493 JR NZ,STAT1
00005C C9 1678 2494 RET

2495 *
2496 * VCPY - COPY VOP AREA AT (DE) TO (BC), (HL) IN LENGTH
2497 *
2498 *
00005D CD600D 1679 2499 VCPYZ CALL VCPZ1 ENTER HERE TO COPY TWO PATTERN LINES
000060 59 1680 2500 VCPZ1 LD E,C DE=8C+H'800'
000061 78 1681 2501 LD A,B
000062 C603 1682 2502 ADD A,B
000064 57 1683 2503 LD D,A
000065 210001 1684 2504 LD HL,H'0100'
000068 D5 1685 2505 VCPY PUSH DE DO COPY 32 BYTES AT A TIME TO SPEED
000069 C5 1686 2506 VC1 PUSH BC THINGS UP
00006A 42 1687 2507 LD B,D
00006B 48 1688 2508 LD C,E
00006C 112000 1689 2509 LD DE,32
00006F 87 1690 2510 OR A
000070 ED52 1691 2511 SBC HL,DE
000072 E5 1692 2512 PUSH HL
000073 19 1693 2513 ADD HL,DE
000074 3001 1694 2514 JR NC,VC2
000076 5D 1695 2515 LD E,L IF <32 BYTES, REDUCE COPY ACCORDINGLY
000077 03 1696 2516 VC2 EX AF,AF'
000078 53 1697 2517 LD D,E
000079 D5 1698 2518 PUSH DE
00007A CD900D 1699 2519 CALL VMIN
00007D D1 1700 2520 POP DE
00007E E1 1701 2521 POP HL SWAP TOP TWO STACK POSITIONS
00007F E3 1702 2522 EX (SP),HL
000080 C5 1703 2523 PUSH BC
000081 E3 1704 2524 EX (SP),HL
000082 C1 1705 2525 POP BC
000083 E5 1706 2526 PUSH HL
000084 CDA10D 1707 2527 CALL VMDUT
000087 D1 1708 2528 POP DE
000088 E1 1709 2529 POP HL
000089 03 1710 2530 EX AF,AF'
00008A 2802 1711 2531 JR Z,VC3
00008C 3008 1712 2532 JR NC,VC1
00008E C1 1713 2533 VC3 POP BC
00008F C9 1714 2534 RET

```

LOC OBJECT M STAT E LINE SOURCE LINE

```

2535 *
2536 *
2537 *      VMIN, VMOUT - READ/WRITE D BYTES BETWEEN BUFFER AND VDP
2538 *
2539 *
000090 C0800C 1715 2540 VMIN CALL SREG1
000093 E5      1716 2541 PUSH HL
000094 21E0FB 1717 2542 LD HL,VDPBUF
000097 D808    1718 2543 VMIN1 IN A,(8)
000099 77      1719 2544 LD (HL),A
00009A 23      1720 2545 INC HL
00009B 03      1721 2546 INC BC
00009C 15      1722 2547 DEC D
00009D 20F8    1723 2548 JR NZ,VMIN1
00009F E1      1724 2549 POP HL
0000A0 C9      1725 2550 RET

2551 *
2552 *
2553 VMOUT SREG
0000A1 E7      1726 + RST H'20' SET VDP REGISTER ROUTINE
+ ENDM
0000A2 E5      1727 2554 PUSH HL
0000A3 21E0FB 1728 2555 LD HL,VDPBUF
0000A6 7E      1729 2556 VMOUT LD A,(HL)
0000A7 D308    1730 2557 OUT (8),A
0000A9 23      1731 2558 INC HL
0000AA 03      1732 2559 INC BC
0000AB 15      1733 2560 DEC D
0000AC 20F8    1734 2561 JR NZ,VMOUT
0000AE E1      1735 2562 POP HL
0000AF C9      1736 2563 RET

2564 *
2565 *
2566 *      NAMSET - SET UP PATTERN NAME TABLE, STARTING WITH No. IN A
2567 *
2568 *
000080 010038 1737 2569 NAMSET LD BC,VDPATH SET TABLE ADDRESS
2570 SREG
000083 E7      1738 + RST H'20' SET VDP REGISTER ROUTINE
+ ENDM
000084 F5      1739 2571 PUSH AF
000085 07      1740 2572 RLCA QUICK DIVIDE-BY-32
000086 07      1741 2573 RLCA
000087 07      1742 2574 RLCA
000088 324EF8 1743 2575 LD (SCRCNT),A SET SCROLL OFFSET LOCATION
000088 F1      1744 2576 POP AF
00008C 0603    1745 2577 LD B,3 THREE LOTS OF 256 TO WRITE (C=0)
00008E D308    1746 2578 CSI OUT (8),A START WRITING
0000C0 3C      1747 2579 INC A
0000C1 0C      1748 2580 INC C
0000C2 20FA    1749 2581 JR NZ,CSI
0000C4 10F8    1750 2582 DJNZ CSI
0000C6 C9      1751 2583 RET

2584 *
2585 *
2586 *      SYSRST -- RESET SYSTEM PARAMETERS, 40-COLS, BLUE BACKDROP.

```

PMOS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.Q.S.

MOS12:5

00:17:44 PAGE: 53

LOC OBJECT M STATE LINE SOURCE LINE

```

2587 *      WHITE TEXT AND GRAPHICS, SPRITES OFF, CLEAR SCREEN
2588 *
2589 *
2590 SYSRST MCAL   ZFORST      RESET PSG & FOC
000DC7 CF        1752      +      RST      H'08'      MOS FUNCTION CALL ROUTINE
000DC8 BD        +      DATA      ZFORST      FUNCTION NUMBER
                        +      ENDM
000DC9 010038    1753  2591      LD      8C,VSP8L
                        2592      SREG
000DCC E7        1754      +      RST      H'20'      SET VDP REGISTER ROUTINE
                        +      ENDM
000DCD 018000    1755  2593      LD      8C,H'80'
000DD0 CDAFOC    1756  2594      CALL      CLRO      CLEAR SPRITE ATTRIBUTE TABLE
000DD3 010018    1757  2595      LD      8C,VCGEN    SET UP CHARACTER GEN
                        2596      SREG
000DD6 E7        1758      +      RST      H'20'      SET VDP REGISTER ROUTINE
                        +      ENDM
000DD7 0601      1759  2597      LD      8,1
000DD9 CDAFOC    1760  2598      CALL      CLRO
000DDC 210000    E 1761  2599      LD      HL,PGTAB
000DDF 3E07      1762  2600      LD      A,7      PLUG IN 128 CHARACTERS
000DE1 010800    1763  2601      LD      8C,8      SET UP PORT 8
000DE4 EDA3      1764  2602 SYR1    OUTI
000DE6 F5        1765  2603      PUSH     AF      TEENY DELAY
000DE7 F1        1766  2604      POP      AF
000DE8 20FA      1767  2605      JR      NZ,SYR1
000DEA 3D        1768  2606      DEC      A
000DEB 20F7      1769  2607      JR      NZ,SYR1
000DED 210000    1770  2608      LD      HL,DEFVCT
000DF0 2230F8    1771  2609      LD      (BRKVCT),HL
000DF3 2232F8    1772  2610      LD      (CLDVCT),HL
000DF6 2234F8    1773  2611      LD      (WRMVCT),HL
000DF9 21D324    1774  2612      LD      HL,H'24D3'    SET UP BREAK FROM RAM (OUT (BANKSW),A)
000DFC 223800    1775  2613      LD      (H'38'),HL
000DFF 3E01      1776  2614      LD      A,1
000E01 D320      1777  2615      OUT      (AUXREG),A    DISABLE KEY INTERRUPT
000E03 D325      1778  2616      OUT      (FIRMSK),A    DISABLE ADC & FIRE BUTTON INTERRUPTS
000E05 D321      1779  2617      OUT      (ADCMASK),A
000E07 3E0E      1780  2618 SYR2    LD      A,C40    CLEAR TO 40 COLUMNS
                        2619      OUTCH
000E09 D7        1781      +      RST      H'10'    CHARACTER OUTPUT ROUTINE
                        +      ENDM
000E0A C9        1782  2620      RET
2621 *
2622 *
2623 *      INIT80 -- INITIALISE 80-COLUMN EXPANSION CARD, IF FITTED
2624 *
2625 *
000E08 CD310E    1783  2626 INIT80 CALL    CHK80    READ 80 COLUMN RAM
000E0E 20F7      1784  2627      JR      NZ,SYR2    SKIP IF NO RAM PRESENT
000E10 084C      1785  2628      IN      A,(IP80)
000E12 C857      1786  2629      BIT      2,A
000E14 21400E    1787  2630      LD      HL,IN525    525 LINE MODE
000E17 2003      1788  2631      JR      NZ,IN81
000E19 21500E    1789  2632      LD      HL,IN625
000E1C 014910    1790  2633 IN81    LD      BC,DT6845+H'1000'

```

LOC OBJECT M STAT E LINE SOURCE LINE

```

000E1F AF      1791 2634    XOR    A          C AT 6845 REGISTER DATA PORT
000E20 0348    1792 2635 IN82    OUT    (RG6845),A
000E22 3C      1793 2636    INC    A
000E23 ED83    1794 2637    OUTI
000E25 20F9    1795 2638    JR     NZ,IN82
000E27 3E10    1796 2639    LD     A,C80
                2640    OUTCH
000E29 07      1797    +    RST    H'10'    CHARACTER OUTPUT ROUTINE
                +    ENDM
000E2A 084C    1798 2641    IN     A,(IP80)
000E2C C84F    1799 2642    BIT    1,A
000E2E 28D7    1800 2643    JR     Z,SYR2    CHECK FOR 'AUTO-80' BOOT LINE
000E30 C9      1801 2644    RET
                2645 *
                2646 *
000E31 C5      1802 2647 CHK80  PUSH   8C
000E32 014000  1803 2648    LD     8C,H'40'
000E35 ED73    1804 2649    IN     A,(C)    TEST ON 1st BYTE OF 80-COL RAM
000E37 2F      1805 2650    CPL
000E38 ED79    1806 2651    OUT    (C),A
000E3A 57      1807 2652    LD     0,A
000E3B ED73    1808 2653    IN     A,(C)
000E3D BA      1809 2654    CP     0
000E3E C1      1810 2655    POP    8C
000E3F C9      1811 2656    RET
                2657 *
                2658 *
                2659 *    525 & 625 LINE OPERATING MODE INITIALISATION TABLES
                2660 *
                2661 *
000E40 7E506138 2662 IN525 DATA H'7E',H'50',H'61',H'38',H'1A',H'13',H'19',H'1A'
                1A13191A
000E48 00082008 2663    DATA H'00',H'08',H'20',H'08',0,0,0,0
                00000000
000E50 7F506238 2664 IN525 DATA H'7F',H'50',H'62',H'38',H'1E',H'02',H'19',H'18'
                1E021918
000E58 00092009 2665    DATA H'00',H'09',H'20',H'09',0,0,0,0
                00000000
                2666 *
                2667 *
                2668 *    IMULT - INTEGER MULTIPLY ROUTINE, DEHL := DE*HL
                2669 *
                2670 *
000E60 78      1812 2671 IMULT  LD     A,8
000E61 210000  1813 2672    LD     HL,0
000E64 0610    1814 2673    LD     8,16
000E66 1F      1815 2674    RRA
000E67 C819    1816 2675    RR     C
000E69 3001    1817 2676 IML1  JR     NC,IML2
000E6B 19      1818 2677    ADD    HL,DE
000E6C C81C    1819 2678 IML2  RR     H
000E6E C81D    1820 2679    RR     L
000E70 1F      1821 2680    RRA
000E71 C819    1822 2681    RR     C
000E73 10F4    1823 2682    DJNZ  IML1
000E75 EB      1824 2683    EX     DE,HL

```

LOC OBJECT M STAT E LINE SOURCE LINE

000E76 67	1825	2684	LD	H,A	
000E77 69	1826	2685	LD	L,C	
000E78 7A	1827	2686	LD	A,D	
000E79 83	1828	2687	OR	E	
000E7A C9	1829	2688	RET		
		2689 *			
		2690 *			
		2691 *			
		2692 *			
		2693 *			
		2694 LOGO	PRM		
000E7B DF	1830	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
		+	ENDM		
000E7C 0C202020		2695	DATA	FF.'	*** EINSTEIN ***
20202020					
20202020					
2A2A2A20					
2045494E					
53544549					
4E20202A					
2A2A					
000E9A 008A		2696	DATA	CR,LF+H'80'	
000E9C C9	1831	2697	RET		
		2698 *			
		2699 *			
		2700 *			
		2701 *			
		2702 *			
000E9D 7C	1832	2703 PR4HX	LD	A,H	
		2704	MCAL	ZPR2HX	
000E9E CF	1833	+	RST	H'08'	MDS FUNCTION CALL ROUTINE
000E9F AB		+	DATA	ZPR2HX	FUNCTION NUMBER
		+	ENDM		
000EAO 7D	1834	2705	LD	A,L	
		2706 PR2HXZ	MCAL	ZPR2HX	
000EA1 CF	1835	+	RST	H'08'	MDS FUNCTION CALL ROUTINE
000EA2 AB		+	DATA	ZPR2HX	FUNCTION NUMBER
		+	ENDM		
000EA3 1873	1836	2707	JR	SPACE	
		2708 *			
		2709 *			
		2710 *			
		2711 *			
		2712 *			
000EA5 F5	1837	2713 PR2HX	PUSH	AF	
000EA6 0F	1838	2714	RRCA		
000EA7 0F	1839	2715	RRCA		
000EA8 0F	1840	2716	RRCA		
000EA9 0F	1841	2717	RRCA		
000EAA C0AE0E	1842	2718	CALL	PRHEX	
000EAD F1	1843	2719	POP	AF	
000EAE E60F	1844	2720 PRHEX	AND	H'0F'	PRINT ASCII OF HEX NUMBER IN BOTTOM HALF OF A
000EB0 C690	1845	2721	ADD	A,H'90'	
000EB2 27	1846	2722	DAA		
000EB3 CE40	1847	2723	ADC	A,H'40'	
000EB5 27	1848	2724	DAA		

LOC OBJECT M STAT E LINE SOURCE LINE

```

      2725      DUTCH
000EB6 07      1349      +      RST      H'10'      CHARACTER OUTPUT ROUTINE
      +      ENDM
000EB7 87      1350      2726      OR      A      ENSURE CARRY RESET !
000EB8 C9      1351      2727      RET
      2728 *
      2729 *
      2730 *      GET HEX DIGIT FROM TEXT
      2731 *
      2732 *
000EB9 EB      1352      2733 FCHHEX EX      DE,HL
      2734      MCAI      ZROMEM
000EBA CF      1353      +      RST      H'08'      MOS FUNCTION CALL ROUTINE
000EBB B1      +      DATA      ZROMEM      FUNCTION NUMBER
      +      ENDM
000EBC EB      1354      2735      EX      DE,HL
000EBD 13      1355      2736      INC      DE
000EBE D630     1356      2737      SUB      '0'
000EC0 FE0A     1357      2738      CP      H'0A'
000EC2 3F      1358      2739      CCF
      2740      RET      NC      TEST FOR 0-9
000EC3 D0      1359      2741      ADD      A,'0'
000EC4 C630     1360      2742      AND      H'5F'      MASK LOWER CASE
000EC6 E65F     1361      2743      SUB      H'37'      GET A-F (OR a-f)
000EC8 D637     1362      2744      CP      H'10'
000ECA FE10     1363      2745      CCF
000ECC 3F      1364      2746      RET      NC
000ECD D0      1365      2747      LD      A,H'00'
000ECE 3E00     1366      2748      RET
      2749 *
      2750 *
      2751 *      GET 2-DIGIT HEX NUMBER FROM TEXT INTO A
      2752 *
      2753 *
000ED1 C5      1368      2754 FCH2HX PUSH      BC
000ED2 CD890E   1369      2755      CALL      FCHHEX
000ED5 3808     1370      2756      JR      C,FCH21
000ED7 07      1371      2757      RLCA
000ED8 07      1372      2758      RLCA
000ED9 07      1373      2759      RLCA
000EDA 07      1374      2760      RLCA
000EDB 4F      1375      2761      LD      C,A
000EDC CD890E   1376      2762      CALL      FCHHEX
000EDF 3801     1377      2763      JR      C,FCH21
000EE1 B1      1378      2764      OR      C
000EE2 C1      1379      2765 FCH21 POP      BC
000EE3 C3      1380      2766      RET
      2767 *
      2768 *
      2769 *      GET HEX NUMBER FROM TEXT INTO HL (UP TO 4 DIGITS)
      2770 *
      2771 *
000EE4 3E04     1381      2772 FCH4HX LD      A,H'04'
000EE6 C5      1382      2773 FCH      PUSH      BC
000EE7 47      1383      2774      LD      B,A
000EE8 AF      1384      2775 FCH1      XOR      A

```

PMDS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.D.S.

MDS12:5

00:17:44 PAGE: 57

LOC OBJECT \* STATE LINE SOURCE LINE

000EE9 67	1885	2776	LD	H,A	
000EEA 6F	1886	2777	LD	L,A	
000EEB CDB90E	1887	2778 FCH2	CALL	FCHHEX	
000EEE 3809	1888	2779 FCH3	JR	C,FCH4	
000EF0 29	1889	2780	ADD	HL,HL	
000EF1 29	1890	2781	ADD	HL,HL	
000EF2 29	1891	2782	ADD	HL,HL	
000EF3 29	1892	2783	ADD	HL,HL	
000EF4 85	1893	2784	OR	L	
000EF5 6F	1894	2785	LD	L,A	
000EF6 10F3	1895	2786	DJNZ	FCH2	
000EF8 13	1896	2787	INC	DE	
000EF9 18	1897	2788 FCH4	DEC	DE	NO MORE DIGITS
000EFA C1	1898	2789	POP	BC	
000EFB C9	1899	2790	RET		
		2791 *			
		2792 *			
		2793 *			
		2794 *			
		2795 *			
000EFC 13	1900	2796 IGBZ	INC	DE	
000EFD 1A	1901	2797 IGBLK	LD	A,(DE)	
000EFE FE20	1902	2798	CP	' '	
000F00 28FA	1903	2799	JR	Z,IGBZ	
000F02 FE2C	1904	2800	CP	' '	
000F04 28F6	1905	2801	JR	Z,IGBZ	SKIP VALID DELIMITERS
000F06 87	1906	2802	OR	A	
000F07 C9	1907	2803	RET		
		2804 *			
		2805 *			
		2806 *			
		2807 *			
		2808 *			
000F08 EB	1908	2809 GETHLZ	EX	DE,HL	GET HL=(DE)
000F09 5E	1909	2810 GETHL	LD	E,(HL)	
000FOA 23	1910	2811	INC	HL	
000F0B 56	1911	2812	LD	D,(HL)	
000FOC 23	1912	2813	INC	HL	
000F0D EB	1913	2814	EX	DE,HL	
000FOE C9	1914	2815	RET		
		2816 *			
		2817 *			
		2818 *			
		2819 *			
		2820 *			
000F0F 3A4AFB	1915	2821 CRLFZ	LD	A,(VCOL)	OUTPUT CR,LF IF NOT AT COLUMN ZERO
000F12 87	1916	2822	OR	A	
000F13 C8	1917	2823	RET	Z	
		2824 CRLF	PRM		OUTPUT CR,LF
000F14 0F	1918	+	RST	H'18'	MESSAGE OUTPUT ROUTINE
		+	ENDM		
000F15 0D8A		2825	DATA	CR,LF+H'30'	
000F17 C9	1919	2826	RET		
		2827 *			
		2828 *			
000F18 3E20	1920	2829 SPACE	LD	A,' '	OUTPUT SPACE

IGBLK - IGNORE SPACES &amp; COMMAS IN TEXT

SKIP VALID DELIMITERS

GETHL,GETHLZ - LOAD HL WITH MEMORY CONTENTS AT (HL) OR (DE)

GET HL=(DE)

EXTRA I/O ROUTINES

C OBJECT M STATE LINE SOURCE LINE

0F1A E5	1921	2830	OUTC	PUSH	HL	OUTPUT CHARACTER PRESERVING REGISTERS
0F18 05	1922	2831		PUSH	DE	
0F1C C5	1923	2832		PUSH	BC	
0F1D F5	1924	2833		PUSH	AF	
0F1E CD9608	1925	2834		CALL	SCREEN	
0F21 F1	1926	2835		POP	AF	
0F22 C1	1927	2836	POPS	POP	BC	
0F23 D1	1928	2837		POP	DE	
0F24 E1	1929	2838		POP	HL	
0F25 C9	1930	2839		RET		
		2840 *				
		2841 *				
0F26 E5	1931	2842	MPRM	PUSH	HL	'MCAL' PRINT MESSAGE ROUTINE
0F27 2A75FB	1932	2843		LD	HL,(SAVSP)	GET RETURN ADDRESS OF ROUTINE
		2844		MCAL	ZROMEM	FROM OLD SP POSITION
0F28	1933	+		RST	H'08'	MOS FUNCTION CALL ROUTINE
0F2B		+		DATA	ZROMEM	FUNCTION NUMBER
		+		ENDM		
0F2C 5F	1934	2845		LD	E,A	
0F2D 23	1935	2846		INC	HL	
		2847		MCAL	ZROMEM	
0F2E CF	1936	+		RST	H'08'	MOS FUNCTION CALL ROUTINE
0F2F B1		+		DATA	ZROMEM	FUNCTION NUMBER
		+		ENDM		
0F30 57	1937	2848		LD	D,A	
0F31 EB	1938	2849		EX	DE,HL	
		2850	MPRM1	MCAL	ZROMEM	
0F32 CF	1939	+		RST	H'08'	MOS FUNCTION CALL ROUTINE
0F33 B1		+		DATA	ZROMEM	FUNCTION NUMBER
		+		ENDM		
00F34 CD400F	1940	2851		CALL	PRMCH1	
00F37 28F9	1941	2852		JR	Z,MPRM1	
00F39 EB	1942	2853		EX	DE,HL	
00F3A 72	1943	2854		LD	(HL),D	PUT BACK RETURN ADDRESS
00F3B	1944	2855		DEC	HL	
00F3C 73	1945	2856		LD	(HL),E	
00F3D E1	1946	2857		POP	HL	
00F3E C9	1947	2858		RET		
		2859 *				
		2860 *				
00F3F 7E	1948	2861	PRMCHR	LD	A,(HL)	INTERNAL ROUTINE FOR PRM AND MPRM
00F40 CB7F	1949	2862	PRMCH1	BIT	7,A	
00F42 CB8F	1950	2863		RES	7,A	
		2864		OUTCH		
00F44 D7	1951	+		RST	H'10'	CHARACTER OUTPUT ROUTINE
		+		ENDM		
00F45 23	1952	2865		INC	HL	
00F46 C9	1953	2866		RET		
		2867 *				
		2868 *				
		2869 *				SERIAL I/O ROUTINES
		2870 *				
		2871 *				
00F47 79	1954	2872	SLOUTZ	LD	A,C	
00F48 F5	1955	2873	SRLQUT	PUSH	AF	
00F49 DB11	1956	2874	SRLQ1	IN	A,(PCIC)	

PMDS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.O.S.

MOS12:5

00:17:44 PAGE:

59

LOC OBJECT M STATE LINE SOURCE LINE

000F4B 1F	1957	2875	RRR		CHECK Tx STATUS
000F4C 30F8	1958	2876	JR	NC,SRL01	
000F4E F1	1959	2877	POP	AF	
000F4F D310	1960	2878	OUT	(PCID),A	
000F51 C9	1961	2879	RET		
		2880 *			
		2881 *			
000F52 DB11	1962	2882	SRLIN	IN A,(PCIC)	
000F54 1F	1963	2883	RRR		
000F55 1F	1964	2884	RRR		CHECK Rx STATUS
000F56 30FA	1965	2885	JR	NC,SRLIN	
000F58 DB10	1966	2886	IN	A,(PCID)	GET Rx CHARACTER
000F5A C9	1967	2887	RET		(NULL PRINTER ROUTINE)
		2888 *			
		2889 *			
		2890 *			PRINTER OUTPUT ROUTINE (CENTRONICS) - NO BUFFER
		2891 *			
		2892 *			
000F58 79	1968	2893	POUTZ	LD A,C	
000F5C E5	1969	2894	POUT	PUSH HL	
000F5D 2192FB	1970	2895	LD	HL,BUSY	CHECK FLAG FOR PRINTER BUSY
000F60 CB46	1971	2896	POI	BIT 0,(HL)	IF SQ, WAIT FOR INTERRUPT
000F62 20FC	1972	2897	JR	NZ,POI	
000F64 D330	1973	2898	OUT	(PI0A0),A	OUTPUT OUR CHARACTER
000F66 CB06	1974	2899	SET	0,(HL)	SHOW BUSY
000F68 E1	1975	2900	POP	HL	
000F69 C9	1976	2901	RET		
		2902 *			
		2903 *			
		2904 *			MUSQUT - OUTPUT B TO PSG PORT GIVEN IN C
		2905 *			
		2906 *			
000F6A 79	1977	2907	MUSQUT	LD A,C	
000F6B D302	1978	2908	OUT	(PSGREG),A	
000F6D 78	1979	2909	LD	A,B	
000F6E D303	1980	2910	OUT	(PSGDAT),A	
000F70 C9	1981	2911	RET		
		2912 *			
		2913 *			
		2914 *			BELL -- ROUTINE TO BEEP 880Hz FOR 0.1s
		2915 *			
		2916 *			
000F71 01008E	1982	2917	BELL	LD BC,H'8E00'	LOW BYTE OF TUNING TO R0
		2918	MCAL	ZMQUT	
000F74 CF	1983	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F75 B4		+	DATA	ZMQUT	FUNCTION NUMBER
		+	ENDM		
000F76 010100	1984	2919	LD	BC,H'0001'	HIGH BYTE OF TUNING TO R1
		2920	MCAL	ZMQUT	
000F79 CF	1985	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F7A B4		+	DATA	ZMQUT	FUNCTION NUMBER
		+	ENDM		
000F7B 01080A	1986	2921	LD	BC,H'0A08'	SET MAX. AMPLITUDE CHANNEL A
		2922	MCAL	ZMQUT	
000F7E CF	1987	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F7F B4		+	DATA	ZMQUT	FUNCTION NUMBER

LOC OBJECT M STAT E LINE SOURCE LINE

		+	ENDM		
000F80 01077E	1988	2923	LD	8C,H'7E07'	TURN ON CHANNEL A TONE
		2924	MCAL	ZMOUT	
000F83 CF	1989	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F84 B4		+	DATA	ZMOUT	FUNCTION NUMBER
		+	ENDM		
000F85 012E3A	1990	2925	LD	8C,H'3A2E'	0.1s DELAY FOR SOUND
000F88 08	1991	2926	DEC	8C	
000F89 78	1992	2927	LD	A,B	
000F8A 81	1993	2928	OR	C	
000F8B 20FB	1994	2929	JR	NZ,BEL1	
000F8D 010300	1995	2930	LD	8C,3	TURN OFF CHANNEL A
		2931	MCAL	ZMOUT	
000F90 CF	1996	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F91 B4		+	DATA	ZMOUT	FUNCTION NUMBER
		+	ENDM		
		2932 *			
		2933 *			
		2934 *			PGINIT - SET UP PSG PORTS FOR START-UP
		2935 *			PSGOUT - OUTPUT COMPLEMENT OF E TO PSG PORT 'A'
		2936 *			
		2937 *			
000F92 01077F	1997	2938	PGINIT LD	8C,H'7F07'	ENSURE ALL CHANNELS OFF
		2939	MCAL	ZMOUT	
000F95 CF	1998	+	RST	H'08'	MOS FUNCTION CALL ROUTINE
000F96 B4		+	DATA	ZMOUT	FUNCTION NUMBER
		+	ENDM		
000F97 1EFF	1999	2940	PGOUTZ LD	E,H'FF'	
000F99 3E0E	2000	2941	PSGOUT LD	A,H'0E'	
000F9B D302	2001	2942	OUT	(PSGREG),A	
000F9D 78	2002	2943	LD	A,E	
000F9E 2F	2003	2944	CPL		
000F9F D303	2004	2945	PSGDI OUT	(PSGDAT),A	
000FA0 79	2005	2946	RET		
		2947 *			
		2948 *			
		2949 *			PSGIN - GET PSG PORT 'B' INTO A
		2950 *			
		2951 *			
000FA2 CD390F	2006	2952	PSGDI CALL	PSGOUT	
000FA5 3E0F	2007	2953	PSGIN LD	A,H'0F'	
000FA7 D302	2008	2954	OUT	(PSGREG),A	
000FA9 DB02	2009	2955	IN	A,(PSGREG)	
000FAB 2F	2010	2956	CPL		
000FAC B7	2011	2957	OR	A	
000FAD C9	2012	2958	RET		
		2959 *			
		2960 *			
		2961 *			ZTIME - SET UP PORTS CTC2 & CTC3 FOR REAL-TIME CLOCK
		2962 *			
		2963 *			
000FAE 0E2A	2013	2964	ZTIME LD	C,CTC2	
000FBO 3E3F	2014	2965	LD	A,H'3F'	
000FB2 067D	2015	2966	LD	B,H'7D'	
000FB4 CD8C0F	2016	2967	CALL	SETCTC	CTC2 & 3 SET UP FOR REAL-TIME CLOCK
000FB7 0C	2017	2968	INC	C	

C OBJECT M STATE LINE SOURCE LINE

```

0FB8 3EDF      2018 2969      LD      A,H'DF'      ALLOW CTC3 TO INTERRUPT
0FB8 067D      2019 2970      LD      B,H'7D'      DROP THROUGH TO SETCTC
                2971 *
                2972 *
                2973 *      SETCTC - SET MODE OF DESIRED CTC CHANNEL,PORT IN C,DIVIDER IN 8
                2974 *
                2975 *
0FBC ED79      2020 2976 SETCTC OUT  (C),A      SET MODE FOR CTC CHANNEL
0FBE 00        2021 2977      NOP
0FBF ED41      2022 2978      OUT     (C),B      SELECT TIME CONSTANT FOR CTC
0FC1 C3        2023 2979      RET
                2980 *
                2981 *
                2982 *      FXMON - MOS/DOS FUNCTION TABLE ACCESS ROUTINE
                2983 *
                2984 *
0FC2 2279FB    2024 2985 FXMON LD      (SAVHL),HL      SAVE HL FOR NOW
0FC5 E1        2025 2986      POP     HL
0FC6 F5        2026 2987      PUSH    AF
0FC7 7E        2027 2988      LD      A,(HL)      GET FUNCTION NUMBER
0FC8 23        2028 2989      INC     HL
0FC9 E3        2029 2990      EX      (SP),HL      PUT RETURN BACK OVER AF
0FCA E5        2030 2991 FMC1 PUSH    HL      AND PUSH AF VALUE AGAIN
0FCB C87F      2031 2992      BIT     7,A
0FCD C8BF      2032 2993      RES     7,A      MOS FUNCTION IF TOP BIT SET
0FCE 2313      2033 2994      JR      Z,FNUSR      JUMP IF USER MOS FUNCTION
0FD1 2A3AFB    2034 2995      LD      HL,(FMVCT)
0FD4 87        2035 2996 FMC2 ADD     A,A
0FD5 85        2036 2997      ADD     A,L
0FD6 6F        2037 2998      LD      L,A
0FD7 3001      2038 2999      JR      NC,FMC3
0FD9 24        2039 3000      INC     H
0FDA 7E        2040 3001 FMC3 LD      A,(HL)
0FDB          2041 3002      INC     HL
0FDC          2042 3003      LD      H,(HL)
0FDD 6F        2043 3004      LD      L,A      HL CONTAINS FUNCTION ADDRESS
0FDE F1        2044 3005      POP     AF
0FDF E5        2045 3006      PUSH    HL
0FE0 2A79FB    2046 3007      LD      HL,(SAVHL)      RESTORE HL, AND EXECUTE FUNCTION
0FE3 C3        2047 3008      RET
                3009 *
                3010 *
FE4 2A3CF8     2048 3011 FNUSR LD      HL,(FUVCT)      JUMP TO USER ROUTINES
FE7 E3        2049 3012      JP      (HL)
                3013 *
                3014 *
                3015 *      VDP INITIALISATION TABLE
                3016 *
                3017 *
FE8 02C0       2018 3018 ITAB ACON    DEFMOD
FEA 0EFF0376   2019      DATA    H'0E',H'FF',H'03',H'76',H'03',H'F4'
03F4
                3020 *
                3021 *
                3022 *      SCRATCH-PAD INITIALISATION TABLE
                3023 *

```

LOC OBJECT M STAT E LINE SOURCE LINE

3024 \*

000FF0 4EFC	3025	SCINIT	ACON	DUMINT	CTCO INTERRUPT	
000FF2 4EFC	3026		ACON	DUMINT	CTC1	
000FF4 4EFC	3027		ACON	DUMINT	CTC2	
000FF6 47FC	3028		ACON	TIMINT	REAL TIME CLOCK INTERRUPT	H'F806'
000FF8 4EFC	3029		ACON	DUMINT	KEYBOARD INTERRUPT VECTOR	H'F808'
000FFA 4EFC	3030		ACON	DUMINT	ADC INTERRUPT	
000FFC 4EFC	3031		ACON	DUMINT	'FIRE BUTTON' INTERRUPT	
000FFE 4EFC	3032		ACON	DUMINT	SPARE INTERRUPT (NO HARDWARE)	
001000 84FC	3033		ACON	PIOA	PRINTER INTERRUPT	H'F810'
001002 4EFC	3034		ACON	DUMINT	PIO 'B' INTERRUPT	

3035 \*

3036 \*

3037 \*

REST OF SCRATCH-PAD INITIALISATION TABLE

3038 \*

3039 \*

001004 D324	2050	3040	OUT	(BANKSW),A	READ MEMORY CALLED FROM ROM (RDMEM)
001006 7E	2051	3041	LD	A,(HL)	
001007 D324	2052	3042	OUT	(BANKSW),A	TOGGLE BACK MEMORY (XECEND=SCRIP+H'17')
001009 C9	2053	3043	RET		

3044 \*

3045 \*

00100A D324	2054	3046	OUT	(BANKSW),A	BLOCK COPY (UPCOPY=SCRIP+H'1A')
00100C ED80	2055	3047	LDIR		
00100E D324	2056	3048	OUT	(BANKSW),A	TOGGLE BACK TO ROM
001010 C9	2057	3049	RET		

3050 \*

3051 \*

001011 D324	2058	3052	OUT	(BANKSW),A	BLOCK COPY (DNCOPY=SCRIP+H'21')
001013 ED80	2059	3053	LDOR		
001015 D324	2060	3054	OUT	(BANKSW),A	TOGGLE BACK TO ROM
001017 C9	2061	3055	RET		

3056 \*

3057 \*

001018 D324	2062	3058	OUT	(BANKSW),A	XDOS ROUTINE CALL (MCXEC=SCRIP+H'23')
00101A CDCA0F	2063	3059	CALL	FMC1	
00101D D324	2064	3060	OUT	(BANKSW),A	
00101F C9	2065	3061	RET		

3062 \*

3063 \*

001020 0000	3064	ACON	DEFVCT	BRKVLW (SCRIP+H'30')
001022 0000	3065	ACON	DEFVCT	CLOVCT
001024 0000	3066	ACON	DEFVCT	WRMVCT
001026 02C0	3067	ACON	DEFMOD	VOPMOD
001028 F0	3068	DATA	OFCOLR	TCOLR
001029 F0	3069	DATA	OFCOLR	CCOLR
00102A 9202	3070	ACON	FMTSL	FNVCT
00102C 7D00	3071	ACON	INITM	FUVCT
00102E 80	3072	DATA	H'80'	FLAGS
00102F 7F	3073	DATA	H'7F'	CUSCODE
001030 3E	3074	DATA	'>'	PROMPT
001031 20	3075	DATA	H'20'	BLINK
001032 A0	3076	DATA	H'A0'	RPTLNG
001033 10	3077	DATA	H'10'	RPTSHT
001034 02	3078	DATA	BUF512/256	SCTSIZ
001035 01	3079	DATA	1	PCFLGS

LOC OBJECT M STATE LINE SOURCE LINE

```

3080 *
3081 *
3082 ZDFVCT MCAL ZINIT
001036 CF 2066 + RST H'08' MOS FUNCTION CALL ROUTINE
001037 9A + DATA ZINIT FUNCTION NUMBER
+ ENDM
001038 FF 3083 DATA H'FF'
001039 0000 3084 ACON 0
00103B FFFFFFFF 3085 DATA H'FF',H'FF',H'FF'
00103E C322FC 2067 3086 JP CALM1
3087 *
3088 *
3089 * ROUTINES FOR SETTING BANKS AND RESTORING AFTER INTERRUPT
3090 * (COPIED UP)
3091 *
3092 *
001041 3A3800 2068 3093 ZBANKZ LD A,(H'38') SET ROM BANK
001044 87 2069 3094 OR A
001045 C8 2070 3095 RET Z
001046 130F 2071 3096 JR ZOUT
001048 3A3800 2072 3097 ZSANK0 LD A,(H'38') SET BANK 0 (MAIN RAM)
001048 87 2073 3098 OR A
00104C C0 2074 3099 RET NZ
00104D 1303 2075 3100 JR ZOUT
00104F C5 2076 3101 ZRBANK PUSH BC RESTORE BANK AS AT START OF INTERRUPT
001050 4F 2077 3102 LD C,A
001051 3A3800 2078 3103 LD A,(H'38')
001054 89 2079 3104 CP C
001055 C1 2080 3105 POP BC
001056 C3 2081 3106 RET Z
001057 D324 2082 3107 ZOUT OUT (BANKSW),A
001059 C9 2083 3108 RET
3109 *
3110 *
3111 * ZCALM - COPY OF MCAL ACTION WHEN CALLED FROM RAM
3112 *
3113 *
00105A 2279FB 2084 3114 ZCALM1 LD (SAYHL),HL PUT HL SOMEWHERE SAFE
00105D E1 2085 3115 POP HL
00105E F5 2086 3116 PUSH AF
00105F 7E 2087 3117 LD A,(HL) GET ROUTINE NUMBER
001060 23 2088 3118 INC HL
001061 E3 2089 3119 EX (SP),HL SAVE INCREMENTED RETURN
001062 ED7378FB 2090 3120 LD (SAYSP),SP SAVE RAM SP, WHICH WE CAN'T USE IN ROM
001066 E5 2091 3121 PUSH HL
001067 B7 2092 3122 OR A
001068 2100FB 2093 3123 LD HL,H'FB00'
00106B ED72 2094 3124 SBC HL,SP CHECK IF STACK ABOVE FB00
00106D E1 2095 3125 POP HL
00106E 3803 2096 3126 JR C,CALM2 JUMP IF STACK > FB00
001070 31FFFC 2097 3127 LD SP,STACK
001073 D324 2098 3128 CALM2 OUT (BANKSW),A DO SWITCH
001075 C0CA0F 2099 3129 CALL FNC1
001078 D324 2100 3130 OUT (BANKSW),A
00107A ED7878FB 2101 3131 LD SP,(SAYSP) RESTORE STACK
00107E C9 2102 3132 RET

```

LOC OBJECT N STATE LINE SOURCE LINE

```

3133 *
3134 *
3135 *          INTERRUPT SERVICE ROUTINES
3136 *
3137 *
00107F E5      2103 3138 ZTINT PUSH HL          INTERRUPTS EVERY SECOND
001030 F5      2104 3139      PUSH AF
001081 C051FC   2105 3140      CALL TINT
001034 F1      2106 3141      POP AF
001085 E1      2107 3142      POP HL
001036 FB      2108 3143 ZDMINT EI          DUMMY INTERRUPT ROUTINE
001087 ED40     2109 3144      RETI
3145 *
3146 *
3147 *          TINT - MAIN PART OF TIMER SERVICE ROUTINE
3148 *
3149 *
001089 2191FB   2110 3150 ZTINT LD HL,TIME+5    MAIN PART OF TIMER SERVICE ROUTINE
00103C 3E39     2111 3151      LD A,'9'        ONE INTERRUPT EVERY SECOND
00108E C07DFC   2112 3152      CALL INCTIM
001091 00      2113 3153      RET NC
001092 3E35     2114 3154      LD A,'5'
001094 C07DFC   2115 3155      CALL INCTIM
001097 00      2116 3156      RET NC
001098 3E39     2117 3157      LD A,'9'        MINS
00109A C07DFC   2118 3158      CALL INCTIM
00109D 00      2119 3159      RET NC
00109E 3E35     2120 3160      LD A,'5'
0010A0 C07DFC   2121 3161      CALL INCTIM
0010A3 00      2122 3162      RET NC
0010A4 28      2123 3163      DEC HL
0010A5 3E32     2124 3164      LD A,'2'        SEE IF IN 20's OF HOURS
0010A7 8E      2125 3165      CP (HL)
0010A8 23      2126 3166      INC HL
0010A9 3E39     2127 3167      LD A,'9'
0010AB 2002     2128 3168      JR NZ,TINT2
0010AD 3E33     2129 3169      LD A,'3'        IF 50 CLOCK ON 24
0010AF C07DFC   2130 3170 TINT2 CALL INCTIM    HOURS
0010B2 00      2131 3171      RET NC
0010B3 3E32     2132 3172      LD A,'2'
3173 *
3174 *
0010B5 34      2133 3175 ZINCTM INC (HL)
0010B6 9E      2134 3176      CP (HL)
0010B7 00      2135 3177      RET NC
0010B8 3630     2136 3178      LD (HL),'0'
0010BA 28      2137 3179      DEC HL
0010BB C9      2138 3180      RET
3181 *
3182 *
0010BC E5      2139 3183 ZPIOR PUSH HL
0010BD 2192FB   2140 3184      LD HL,BUSY
0010C0 C835     2141 3185      RES 0,(HL)    JUST RESET BUSY FLAG
0010C2 E1      2142 3186      POP HL
0010C3 18C1     2143 3187      JR ZDMINT
3188 ZEND EQU $

```

PMOS CROSS ASSEMBLER Z80

REL: 4.0 EINSTEIN M.O.S.

MO512 S

00:17:44 PAGE

65

LOC OBJECT \* STAT E LINE SOURCE LINE

```

3189 *
3190 *
3191 *
3192 OFFSET EQU      BANKZ-ZBANKZ      COPY OFFSET
3193 BANKO EQU      ZBANKO+OFFSET
3194 RBANK EQU      ZRBANK+OFFSET
3195 CALM1 EQU      ZCALM1+OFFSET
3196 TIMINT EQU     ZTIMINT+OFFSET
3197 DUMINT EQU     ZDUMINT+OFFSET
3198 TINT EQU       ZTINT+OFFSET
3199 INSTIM EQU     ZINSTIM+OFFSET
3200 PIDA EQU       ZPIDA+OFFSET
3201 *
3202 *
3203      END
    
```

71005

NO ERRORS DETECTED