

ERR LINE ADDR B1 B2 B3 B4

6500 ASSEMBLER VER 1.OMR

Page 1

```
1      LIST X
2      ; THIS IS THE MODIFIED SEPTEMBER ATARI 400/800 COMPUTER OPERATING
3      ; SYSTEM LISTING. MODIFIED TO ASSEMBLE ON THE MICROTEC CROSS
4      ; ASSEMBLER.
5      ; THIS VERSION IS THE ONE WHICH WAS BURNED INTO ROM.
6      ; THERE IS A RESIDUAL PIECE OF CODE WHICH IS FOR LNDUG. THIS
7      ; IS AT LOCATION $9000 WHICH IS NOT IN ROM.
8      ;
9      ; THIS IS THE REVISION D EPROM VERSION
```

```

10      ;.PAGE
11      ;
12      ;
13      ; COLLEEN OPERATING      SYSTEM EQUATE FILE
14      ;
15      ; NTSC/PAL ASSEMBLY      FLAG
16      ;
17      0000      PALFLG      =      0      ; 0 = NTSC 1 = PAL
18      ;
19      ;
20      ;      MODULE ORIGIN TABLE
21      ;
22      E000      CHRORG      =      $E000      ;CHARACTER SET
23      E400      VECTBL      =      $E400      ;VECTOR TABLE
24      E480      VCTABL      =      $E480      ;RAM VECTOR INITIAL VALUE TABLE
25      E4A6      CIOORG      =      $E4A6      ;CENTRAL I/O HANDLER
26      E6D5      INTORG      =      $E6D5      ;INTERRUPT HANDLER
27      E944      SIOORG      =      $E944      ;SERIAL I/O DRIVER
28      EDEA      DSKORG      =      $EDEA      ;DISK HANDLER
29      EE78      PRNORG      =      $EE78      ;PRINTER HANDLER
30      EF41      CASORG      =      $EF41      ;CASSETTE HANDLER
31      F0E3      MONORG      =      $F0E3      ;MONITOR/POWER UP MODULE
32      F3E4      KBDORG      =      $F3E4      ;KEYBOARD/DISPLAY HANDLER
33      ;
34      ;
35      ;
36      ;
37      ;      VECTOR TABLE
38      ;
39      ; HANDLER ENTRY POINTS ARE CALLED OUT IN THE FOLLOWING VECTOR
40      ; TABLE. THESE ARE THE ADDRESSES MINUS ONE.
41      ;
42      ;
43      ; EXAMPLE FOR EDITOR
44      ;
45      ;      E400      OPEN
46      ;      2      CLOSE
47      ;      4      GET

```

```

48      ;      6      PUT
49      ;      8      STATUS
50      ;      A      SPECIAL
51      ;      C      JUMP TO POWER ON INITIALIZATION ROUTINE
52      ;      F      NOT USED
53      ;
54
55      E400      EDITRV      =      $E400      ;EDITOR
56      E410      SCRENV      =      $E410      ;TELEVISION SCREEN
57      E420      KEYBDV      =      $E420      ;KEYBOARD
58      E430      PRINTV      =      $E430      ;PRINTER
59      E440      CASETV      =      $E440      ;CASSETTE
60      ;
61      ;      JUMP VECTOR TABLE
62      ;
63      ; THE FOLLOWING IS A TABLE      OF JUMP INSTRUCTIONS
64      ; TO VARIOUS ENTRY POINTS IN      THE OPERATING SYSTEM.
65      ;
66      E450      DISKIV      =      $E450      ;DISK INITIALIZATION
67      E453      DSKINV      =      $E453      ;DISK INTERFACE
68      E456      CIOV      =      $E456      ;CENTRAL INPUT OUTPUT ROUTINE
69      E459      SIOV      =      $E459      ;SERIAL INPUT OUTPUT ROUTINE
70      E45C      SETVBV      =      $E45C      ;SET SYSTEM TIMERS ROUTINE
71      E45F      SYSVBV      =      $E45F      ;SYSTEM VERTICAL BLANK CALCULATIONS
72      E462      XITVBV      =      $E462      ;EXIT VERTICAL DLANK CALCULATIONS
73      E465      SIOINV      =      $E465      ;SERIAL INPUT OUTPUT INITIALIZATION
74      E468      SENDEV      =      $E468      ;SEND ENABLE ROUTINE
75      E46B      INTINV      =      $E46B      ;INTERRUPT HANDLER INITIALIZATION
76      E46E      CIOINV      =      $E46E      ;CENTRAL INPUT OUTPUT INITIALIZATION
77      E471      DLKDDV      =      $E471      ;BLACKBOARD MODE
78      E474      WARMSV      =      $E474      ;WARM START ENTRY POINT
79      E477      COLDSV      =      $E477      ;COLD START ENTRY POINT
80      E47A      RBLOKV      =      $E47A      ;CASSETTE READ BLOCK ENTRY POINT VECTOR
81      E47D      CSOPIV      =      $E47D      ;CASSETTE OPEN FOR INPUT VECTOR
82      ;VCTABL      =      $E480
83      ;
84      ;
85      ;      OPERATING SYSTEM EQUATES
86      ;

```

```

87      ;      COMMAND CODES FOR IOCB
BB      0003      OPEN      =      3      ;OPEN FOR INPUT/OUTPUT
89      0005      GETREC     =      5      ;GET RECORD (TEXT)
90      0007      GETCHR     =      7      ;GET CHARACTERS)
91      0009      PUTREC     =      9      ;PUT RECORD (TEXT)
92      000B      PUTCHR     =      $B      ;PUT CHARACTERS)
93      000C      CLOSE     =      $C      ;CLOSE DEVICE
94      000D      STATIS    =      $D      ;STATUS REQUEST
95      000E      SPECIL    =      $E      ;BEGINNING OF SPECIAL ENTRY COMMANDS
96      ;
97      ;      SPECIAL ENTRY COMMANDS
98      0011      DRAWLN     =      $11      ;DRAW LINE
99      0012      FILLIN     =      $12      ;DRAW LINE WITH RIGHT FILL
100     0020      RENAME     =      $20      ;RENAME DISK FILE
101     0021      DELETE     =      $21      ;DELETE DISK FILE
102     0022      FORMAT     =      $22      ;FORMAT
103     0023      LOCKFL     =      $23      ;LOCK FILE TO READ ONLY
104     0024      UNLOCK     =      $24      ;UNLOCK LOCKED FILE
105     0025      POINT      =      $25      ;POINT SECTOR
106     0026      NOTE       =      $26      ;NOTE SECTOR
107     D0FF      IOCFRE     =      $FF      ;IOCB "FREE"
108     ;
109     ;      AUX1 EQUATES
110     ;      ( ) INDICATES WHICH DEVICES USE BIT
111     0001      APPEND      =      $1      ;OPEN FOR WRITE APPEND (D), OR SCREEN READ
112     0002      DIRECT     =      $2      ;OPEN FOR DIRECTORY ACCESS (D)
113     0004      OPNIN      =      $4      ;OPEN FOR INPUT (ALL DEVICES)
114     000B      OPNOT      =      $8      ;OPEN FOR OUTPUT (ALL DEVICES)
115     000C      OPNINO     =      OPNIN+OPNOT ;OPEN FOR INPUT AND OUTPUT (ALL DEVICES)
116     0010      MXDMOD     =      $10      ;OPEN FOR MIXED MODE (E, S)
117     0020      INSCLR     =      $20      ;OPEN WITHOUT CLEARING SCREEN (E, S)
118     ;
119     ;      DEVICE NAMES
120     0045      SCREDT      =      'E      ;SCREEN EDITOR (R/W)
121     004B      KBD        =      'K.      ;KEYBOARD (R ONLY)
122     0053      DISPLY     =      'S      ;SCREEN DISPLAY (R/W)
123     0050      PRINTR     =      'P      ;PRINTER (W ONLY)
124     0043      CASSET     =      'C      ;CASSETTE
125     004D      MODEM      =      'M      ;MODEM

```

```

126 0044 DISK      =      'D          ;DISK (R/W)
127      ;
128      ; SYSTEM EOL (CARRIAGE RETURN)
129 009D CR        =      $9B
130      ;
131      ;
132      ; OPERATING SYSTEM STATUS CODES
133      ;
134 0001 SUCCES    =      $01          ;SUCCESSFUL OPERATION
135      ;
136 0080 BRKABT    =      $80          ;BREAK KEY ABORT
137 0081 PRVOPN    =      $81          ;IOCB ALREADY OPEN
138 0082 NONDEV    =      $82          ;NON-EXISTANT DEVICE
139 0083 WRONLY    =      $83          ;IOCD OPENED FOR WRITE ONLY
140 0084 NVALID    =      $84          ;INVALID COMMAND
141 0085 NOTOPN    =      $85          ;DEVICE OR FILE NOT OPEN
142 0086 BADIOC    =      $86          ;INVALID IOCB NUMBER
143 0087 RDONLY    =      $87          ;IOCB OPENED FOR READ ONLY
144 0088 EOFERR    =      $88          ;END OF FILE
145 0089 TRNRCD    =      $89          ;TRUNCATED RECORD
146 008A TIMOUT    =      $8A          ;PERIPHERAL DEVICE TIME OUT
147 008B DNACK     =      $8B          ;DEVICE DOES NOT ACKNOWLEDGE COMMAND
148 008C FRMERR    =      $8C          ;SERIAL BUS FRAMING ERROR
149 008D CRSROR    =      $8D          ;CURSOR OVERRANGE
150 008E OVRRUN    =      $8E          ;SERIAL DUS DATA OVERRUN
151 008F CHKERR    =      $8F          ;SERIAL BUS CHECKSUM ERROR
152      ;
153 0090 DERROR     =      $90          ;PERIPHERAL DEVICE ERROR (OPERATION NOT COMP
154 0091 BADMOD     =      $91          ;BAD SCREEN MODE NUMBER
155 0092 FNCNOT     =      $92          ;FUNCTION NOT IMPLEMENTED IN HANDLER
156 0093 SCRMEM     =      $93          ;INSUFICIENT MEMORY FOR SCREEN MODE
157      ;
158      ;
159      ;
160      ;
161      ;
162      ;
163      ; PAGE ZERO RAM ASSIGNMENTS
164      ;

```

```

165          *=$0000
166 0000      LINZBS:      .RES 2          ;LINBUG RAM (WILL DE REPLACED BY MONITOR RAM
167          ;
168          ; THESE LOCATIONS ARE NOT CLEARED
169 0002      CASINI:      .RES 2          ;CASSETTE INIT LOCATION
170 0004      RAMLO:      .RES 2          ;RAM POINTER FOR MEMORY TEST
171 0006      TRAMSZ:     .RES 1          ;TEMPORARY REGISTER FOR RAM SIZE
172 0007      TSTDAT:     .RES 1          ;RAM TEST DATA REGISTER
173          ;
174          ; CLEARED ON COLDSTART ONLY
175 0008      WARMST:     .RES 1          ;WARM START FLAG
176 0009      BOOT?:     .RES 1          ;SUCCESSFUL BOOT FLAG
177 000A      DOSVEC:     .RES 2          ;DISK SOFTWARE START VECTOR
178 000C      DOSINI:     .RES 2          ;DISK SOFTWARE INIT ADDRESS
179 000E      APPMHI:     .RES 2          ;APPLICATIONS MEMORY HI LIMIT
180          ;
181          ; CLEAR ED ON COLD OR WARM START
182 0010      INTZBS      =*              ;INTERRUPT HANDLER
183 0010      POKMSK:     .RES 1          ;SYSTEM MASK FOR POKEY IRa ENABLE
184 0011      BRKKEY:     .RES 1          ;BREAK KEY FLAG
185 0012      RTCLOK:     .RES 3          ;REAL TIME CLOCK (IN 16 MSEC UNITS)
186          ;
187 0015      BUFADR:     .RES 2          ;INDIRECT BUFFER ADDRESS REGISTER
188          ;
189 0017      ICCOMT:     .RES 1          ;COMMAND FOR VECTOR
190          ;
191 0018      DSKFMS:     .RES 2          ;DISK FILE MANAGER POINTER
192 001A      DSKUTL:     .RES 2          ;DISK UTILITIES POINTER
193          ;
194 001C      PTIMOT:     .RES 1          ;PRINTER TIME OUT REGISTER
195 001D      PBPNT:     .RES 1          ;PRINT BUFFER POINTER
196 001E      PBUFSZ:     .RES 1          ;PRINT DUFFER SIZE
197 001F      PTEMP:     .RES 1          ;TEMPORARY REGISTER
198          ;
199 0020      ZIOCB       =*              ;ZERO PAGE I/O CONTROL DLOCK
200 0010      IOCBSZ     = 16            ;NUMBER OF BYTES PER IOCB
201 0080      MAXIOC     = 8*IOCBSZ      ;LENGTH OF THE IOCB AREA
202 0020      IOCBAS     =*
203 0020      ICHIDZ:     .RES 1          ;HANDLER INDEX NUMBER (FF = IOCB FREE)

```

```

204 0021 ICDNOZ: .RES 1 ;DEVICE NUMBER (DRIVE NUMBER)
205 0022 ICCOMZ: .RES 1 ;COMMAND CODE
206 0023 ICSTAZ: .RES 1 ;STATUS OF LAST IOCB ACTION
207 0024 ICBALZ: .RES 1 ;BUFFER ADDRESS LOW BYTE
208 0025 ICBAHZ: .RES 1
209 0026 ICPTLZ: .RES 1 ;PUT BYTE ROUTINE ADDRESS - 1
210 0027 ICPTHZ: .RES 1
211 0028 ICBLLZ: .RES 1 ;BUFFER LENGTH LOW BYTE
212 0029 ICBLHZ: .RES 1
213 002A ICAXIZ: .RES 1 ;AUXILIARY INFORMATION FIRST BYTE
214 002B ICAX21: .RES 1
215 002C ICSPRZ: .RES 4 ;TWO SPARE BYTES (CIO LOCAL USE)
216 002E ICIDNO = ICSPRZ+2 ;IOCB NUMBER X 16
217 002E CIOCHR = ICSPRZ+3 ;CHARACTER BYTE FOR CURRENT OPERATION
218 ;
219 0030 STATUS: .RES 1 ;INTERNAL STATUS STORAGE
220 0031 CHKSUM: .RES 1 ;CHECKSUM (SINGLE BYTE SUM WITH CARRY)
221 0032 BUFRLO: .RES 1 ;POINTER TO DATA BUFFER (LO BYTE)
222 0033 BUFRHI: .RES 1 ;POINTER TO DATA BUFFER (HI BYTE)
223 0034 BFENLO: .RES 1 ;NEXT BYTE PAST END OF THE DATA DUFFER (LO BYTE)
224 0035 BFENHI: .RES 1 ;NEXT BYTE PAST END OF THE DATA BUFFER (HI BYTE)
225 0036 CRETRY: .RES 1 ;NUMBER OF COMMAND FRAME RETRIES
226 0037 DRETRY: .RES 1 ;NUMBER OF DEVICE RETRIES
227 0038 BUFRFL: .RES 1 ;DATA BUFFER FULL FLAG
228 0039 RECVDN: .RES 1 ;RECEIVE DONE FLAG
229 003A XMTDON: .RES 1 ;TRANSMISSION DONE FLAG
230 003E CHKSNT: .RES 1 ;CHECKSUM SENT FLAG
231 003C NOCKSM: .RES 1 ;NO CHECKSUM FOLLOWS DATA FLAG
232 ;
233 ;
234 003D BPTR: .RES 1
235 003E FTYPE: .RES 1
236 003F FE0F: .RES 1
237 0040 FREQ: .RES 1
238 0041 SOUNDN: .RES 1 ;NOISY I/O FLAG. (ZERO IS QUIET)
239 0042 CRITIC: .RES 1 ;DEFINES CRITICAL SECTION (CRITICAL IF NON-Z)
240 ;
241 0043 FMSZPG: .RES 7 ;DISK FILE MANAGER SYSTEM ZERO PAGE
242 ;

```

```

243      ;
244      004A      CKEY:      .RES 1      ;FLAG SET WHEN GAME START PRESSED
245      004E      CASSBT:    .RES 1      ;CASSETTE BOOT FLAG
246      004C      DSTAT:     .RES 1      ;DISPLAY STATUS
247      ;
248      004D      ATTRACT:    .RES 1      ;ATTRACT FLAG
249      004E      DRKMSK:     .RES 1      ;DARK ATTRACT MASK
250      004E      COLRSH:     .RES 1      ;ATTRACT COLOR SHIFTER (EOR'ED WITH PLAYFIELD)
251      ;
252      0002      LEDGE       =      2      ;LMARGN'S VALUE AT COLD START
253      0027      REDGE       =      39     ;RMARGN'S VALUE AT COLD START
254      0050      TMPCHR:     .RES 1
255      0051      HOLD1:      .RES 1
256      0052      LMARGN:     .RES 1      ;LEFT MARGIN ISET TO i AT POWER ON>
257      0053      RMARGN:     .RES 1      ;RIGHT MARGIN (SET TO 38 AT POWER ON)
258      0054      ROWCRS:     .RES 1      ;CURSOR COUNTERS
259      0055      COLCRS:     .RES 2
260      0057      DINDEXT:    .RES 1
261      0058      SAVMSC:     .RES 2
262      005A      OLDROW:     .RES 1
263      005B      OLDCOL:     .RES 2
264      005D      OLDCHR:     .RES 1      ;DATA UNDER CURSOR
265      005E      OLDADR:     .RES 2
266      0060      NEWROW:     .RES 1      ;POINT DRAW GOES TO
267      0061      NEWCOL:     .RES 2
268      0063      LOGCOL:     .RES 1      ;POINTS AT COLUMN IN LOGICAL LINE
269      0064      ADRESS:     .RES 2
270      0066      MLTTMP:     .RES 2
271      0066      DPNTMP      =      MLTTMP      ;FIRST BYTE IS USED IN OPEN AS TEMP
272      0068      SAVADR:     .RES 2
273      006A      RAMTOP:     .RES 1      ;RAM SIZE DEFINED BY POWER ON LOGIC
274      006B      BUFCNT:     .RES 1      ;BUFFER COUNT
275      006C      BUFSTR:     .RES 2      ;EDITOR GETCH POINTER
276      006E      BITMSK:     .RES 1      ;BIT MASK
277      006F      SHFAMT:     .RES 1
278      0070      ROWAC:      .RES 2
279      0072      COLAC:      .RES 2
280      0074      ENDPT:      .RES 2
281      0076      DELTAR:     .RES 1

```



```

282 0077 DELTAC: .RES 2
283 0079 ROWINC: .RES 1
284 007A COLINC: .RES 1
285 007E SWPFLG: .RES 1 ;NON-O IF TXT AND REGULAR RAM IS SWAPPED
286 007C HOLDCH: .RES 1 ;CH IS MOVED HERE IN KGETCH BEFORE CNTL & SH
287 007D INSDAT: .RES 1
288 007E COUNTR: .RES 2
289 ;
290 ;
291 ;
292 ;
293 ; 80 - FF ARE RESERVED FOR USER APPLICATIONS
294 ;
295 ;
296 ;
297 ; NOTE : SEE FLOATING POINT SUBROUTINE AREA FOR ZERO PAGE CELLS
298 ;
299 ;
300 ;
301 ;
302 ; PAGE 1 - STACK
303 ;
304 ;
305 ;
306 ;
307 ; PAGE TWO RAM ASSIGNMENTS
308 ;
309 ; *= $0200
310 0200 INTABS =* ;INTERRUPT RAM
311 0200 VDSLST: .RES 2 ;DISPLAY LIST NMI VECTOR
312 0202 VPRCED: .RES 2 ;PROCEED LINE IRQ VECTOR
313 0204 VINTER: .RES 2 ;INTERRUPT LINE IRQ VECTOR
314 0206 VBREAK: .RES 2 ;SOFTWARE BREAK (00) INSTRUCTION IRQ VECTOR
315 0208 VKEYBD: .RES 2 ;POKEY KEYBOARD IRQ VECTOR
316 020A VSERIN: .RES 2 ;POKEY SERIAL INPUT READY IRQ
317 020C VSEROR: .RES 2 ;POKEY SERIAL OUTPUT READY IRQ
318 020E VSEROC: .RES 2 ;POKEY SERIAL OUTPUT COMPLETE IRQ
319 0210 VTIMR1: .RES 2 ;POKEY TIMER 1 IRQ
320 0212 VTIMR2: .RES 2 ;POKEY TIMER 2 IRQ

```

```

321 0214 VTIMR4: .RES 2 ;POKEY TIMER 4 IRQ
322 0216 VIMIRQ: .RES 2 ;IMMEDIATE IRQ VECTOR
323 0218 CDTMV1: .RES 2 ;COUNT DOWN TIMER 1
324 021A CDTMV2: .RES 2 ;COUNT DOWN TIMER 2
325 021C CDTMV3: .RES 2 ;COUNT DOWN TIMER 3
326 021E CDTMV4: .RES 2 ;COUNT DOWN TIMER 4
327 0220 CDTMV5: .RES 2 ;COUNT DOWN TIMER 5
328 0222 VVBLKI: .RES 2 ;IMMEDIATE VERTICAL BLANK NMI VECTOR
329 0224 VVBLKD: .RES 2 ;DEFERRED VERTICAL BLANK NMI VECTOR
330 0226 CDTMA1: .RES 2 ;COUNT DOWN TIMER 1 JSR ADDRESS
331 0228 CDTMA2: .RES 2 ;COUNT DOWN TIMER 2 JSR ADDRESS
332 022A CDTMF3: .RES 1 ;COUNT DOWN TIMER 3 FLAG
333 022E SRTIMR: .RES 1 ;SOFTWARE REPEAT TIMER
334 022C CDTMF4: .RES 1 ;COUNT DOWN TIMER 4 FLAG
335 022D INTEMP: .RES 1 ;IAN'S TEMP (RENAMED FROM T1 BY POPULAR DEMA)
336 022E CDTMF5: .RES 1 ;COUNT DOWN TIMER FLAG 5
337 022E SDMCTL: .RES 1 ;SAVE DMACTL REGISTER
338 0230 SDLSTL: .RES 1 ;SAVE DISPLAY LIST LOW BYTE
339 0231 SDLSTH: .RES 1 ;SAVE DISPLAY LIST HI BYTE
340 0232 SSKCTL: .RES 1 ;SKCTL REGISTER RAM
341 0233 .RES 1 ;
342 ;
343 0234 LPENH: .RES 1 ;LIGHT PEN HORIZONTAL VALUE
344 0235 LPENV: .RES 1 ;LIGHT PEN VERTICAL VALUE
345 0236 BRKKY: .RES 2 ;BREAK KEY VECTOR
346 ;
347 0238 .RES 2 ;SPARE
348 ;
349 023A CDEVIC: .RES 1 ;COMMAND FRAME BUFFER - DEVICE
350 023E CCOMND: .RES 1 ;COMMAND
351 023C CAUX1: .RES 1 ;COMMAND AUX BYTE 1
352 023D CAUX2: .RES 1 ;COMMAND AUX BYTE 2
353 ; NOTE: MAY NOT BE THE LAST WORD ON A PAGE
354 023E TEMP: .RES 1 ;TEMPORARY RAM CELL
355 ; NOTE: MAY NOT BE THE LAST WORD ON A PAGE
356 023E ERRFLG: .RES 1 ;ERROR FLAG - ANY DEVICE ERROR EXCEPT TIME
357 ;
358 0240 DFLAGS: .RES 1 ;DISK FLAGS FROM SECTOR ONE
359 0241 DBSECT: .RES 1 ;NUMBER OF DISK BOOT SECTORS

```

ERR LINE ADDR B1 B2 B3 B4

6500 ASSEMBLER VER 1.OMR

Page 11

360	0242	B00TAD:	.RES	2	;ADDRESS WHERE DISK B00T LOADER	WILL BE PUT
361	0244	COLDST:	.RES	1	;COLDSTART FLAG (1=IN MIDDLE OF	COLDSTART)
362		;				
363	0245		.RES	1	;SPARE	
364		;				
365	0246	DSKTIM:	.RES	1	;DISK TIME OUT REGISTER	
366		;				
367	0247	LINBUF:	.RES	40	;CHAR LINE BUFFER	
368		;				
369	026E	GPRIOR:	.RES	1	;GLOBAL PRIORITY CELL	
370		;				
371	0270	PADDL0:	.RES	1	;POTENTIOMETER 0 RAM CELL	
372	0271	PADDL1:	.RES	1		
373	0272	PADDL2:	.RES	1		
374	0273	PADDL3:	.RES	1		
375	0274	PADDL4:	.RES	1		
376	0275	PADDL5:	.RES	1		
377	0276	PADDL6:	.RES	1		
378	0277	PADDL7:	.RES	1		
379	0278	STICK0:	.RES	1	;JOYSTICK 0 RAM CELL	
380	0279	STICK1:	.RES	1		
381	027A	STICK2:	.RES	1		
382	0278	STICK3:	.RES	1		
383	027C	PTRIGO:	.RES	1	;PADDLE TRIGGER 0	
384	027D	PTRIGI:	.RES	1		
385	027E	PTRI02:	RES	1		
386	027E	PTRIG3:	.RES	1		
387	0280	PTRIG4:	.RES	1		
388	0281	PTRIG5:	.RES	1		
389	0282	PTRIG6:	.RES	1		
390	0283	PTRIG7:	.RES	1		
391	0284	STRIG0:	.RES	1	;JOYSTICK TRIGGER 0	
392	0285	STRIG1:	.RES	1		
393	0286	STRIG2:	.RES	1		
394	0287	STRIG3:	.RES	1		
395		;				
396	0288	CSTAT:	.RES	1		
397	0289	WMODE:	.RES	1		
398	028A	SLIM:	.RES	1		

```

399 028E IMASK: .RES 1
400 028C JVECK: .RES 2
401 ;
402 028E .RES 2 ;SPARE
403 ;
404 ;
405 ;
406 ;
407 0290 TXTROW: .RES 1 ;TEXT ROWCRS
408 0291 TXTCOL: .RES 2 ;TEXT COLCRS
409 0293 TINDEX: .RES 1 ;TEXT INDEX
410 0294 TXTMSC: .RES 2 ;FOOLS CONVRT INTO NEW MSC
411 0296 TXTOLD: .RES 6 ;OLDROW & OLDCOL FOR TEXT (AND THEN SOME)
412 029C TMPX1: .RES 1
413 029D HOLD3: .RES 1
414 029E SUBTMP: .RES 1
415 029E HOL02: .RES 1
416 02A0 DMASK: .RES 1
417 02A1 TEMPLBT: .RES 1
418 02A2 ESCFLG: .RES 1 ;ESCAPE FLAG
419 02A3 TABMAP: .RES 15
420 0282 LOGMAP: .RES 4 ;LOGICAL LINE START BIT MAP
421 0286 INVFLG: .RES 1 ;INVERSE VIDEO FLAG (TOGGLED BY ATARI KEY)
422 0287 FILFLG: .RES 1 ;RIGHT FILL FLAG FOR DRAW
423 0288 TMPROW: .RES 1
424 0289 TMPCOL: .RES 2
425 02BB SCRFLG: .RES 1 ;SET IF SCROLL OCCURS
426 02BC HOLD4: .RES 1 ;TEMP CELL USED IN DRAW ONLY
427 02BD HOLD5: .RES 1 ;DITTO
428 02BE SHFLOK: .RES 1
429 02BF BOTSCR: .RES 1 ;BOTTOM OF SCREEN : 24 NORM 4 SPLIT
430 ;
431 ;
432 02C0 PCOLR0: .RES 1 ;P0 COLOR
433 02C1 PCOLR1: .RES 1 ;P1 COLOR
434 02C2 PCOLR2: .RES 1 ;P2 COLOR
435 02C3 PCOLR3: .RES 1 ;P3 COLOR
436 02C4 COLOR0: .RES 1 ;COLOR 0
437 02C5 COLOR1: .RES 1

```

```

438 02C6 COLOR2: .RES 1
439 02C7 COLOR3: .RES 1
440 02CB COLOR4: .RES 1
441 ;
442 ;
443 02C9 .RES 23 ; SPARE
444 ;
445 ;
446 ;
447 02E0 GLBABS =* ; GLOBAL VARIABLES
448 ;
449 02E0 .RES 4 ; SPARE
450 ;
451 02E4 RAMSIZ: .RES 1 ; RAM SIZE (HI BYTE ONLY)
452 02E5 MEMTOP: .RES 2 ; TOP OF AVAILABLE USER MEMORY
453 02E7 MEMO: .RES 2 ; BOTTOM OF AVAILABLE USER MEMORY
454 02E9 .RES 1 ; SPARE
455 02EA DVSTAT: .RES 4 ; STATUS BUFFER
456 02EE CBAUDL: .RES 1 ; CASSETTE BAUD RATE LOW BYTE
457 02EF CBAUDH: .RES 1
458 ;
459 02F0 CRSINH: .RES 1 ; CURSOR INHIBIT (00 = CURSOR ON)
460 02F1 KEYDEL: .RES 1 ; KEY DELAY
461 02F2 CH1: .RES 1
462
463 02F3 CHACT: .RES 1 ; CHACTL REGISTER RAM
464 02F4 CHBAS: .RES 1 ; CHBAS REGISTER RAM
465
466 02F5 .RES 5 ; SPARE BYTES
467
468 02FA CHAR: .RES 1
469 02FB ATACHR: .RES 1 ; ATASCII CHARACTER
470 02FC CH: .RES 1 ; GLOBAL VARIABLE FOR KEYBOARD
471 02FD FILDAT: .RES 1 ; RIGHT FILL DATA (DRAW)
472 02FE DSPFLG: .RES 1 ; DISPLAY FLAG : DISPLAY CNTLS IF NON-ZERO
473 02FF SSFLAG: .RES 1 ; START/STOP FLAG FOR PAGING (CNTL 1). CLEARE
474 ;
475 ;
476 ;

```

```

477      ;
478      ;
479      ;
480      ;
481      ;      PAGE      THREE RAM      ASSIGNMENTS
482      ;
483      0300      DCB      =*              ;DEVICE CONTROL BLOCK
484      0300      DDEVIC:      .RES      1      ;PERIPHERAL UNIT 1 BUS I. D. NUMBER
485      0301      DUNIT:      .RES      1      ;UNIT NUMBER
486      0302      DCOMND:      .RES      1      ;BUS COMMAND
487      0303      DSTATS:      .RES      1      ;COMMAND TYPE/STATUS RETURN
488      0304      DBUFLO:      .RES      1      ;DATA BUFFER POINTER LOW BYTE
489      0305      DBUFHI:      .RES      1
490      0306      DTIMLO:      .RES      i      ;DEVICE TIME OUT IN 1 SECOND UNITS
491      0307      DUNUSE:      .RES      1      ;UNUSED BYTE
492      0308      DBYTLO:      .RES      1      ;NUMBER OF BYTES TO BE TRANSFERRED LOW BYTE
493      0309      DBYTHI:      .RES      1
494      030A      DAUX1:      .RES      1      ;COMMAND AUXILIARY BYTE 1
495      030E      DAUX2:      .RES      1
496      ;
497      030C      TIMER1:      .RES      2      ;INITIAL TIMER VALUE
498      030E      ADDCOR:      .RES      1      ;ADDITION CORRECTION
499      030E      CASFLG:      .RES      1      ;CASSETTE MODE WHEN SET
500      0310      TIMER2:      .RES      2      ;FINAL TIMER VALUE. THESE TWO TIMER VALUES
501      ; ARE USED TO COMPUTE INTERVAL FOR BAUD RATE
502      0312      TEMP1:      .RES      2      ;TEMPORARY STORAGE REGISTER
503      0314      TEMP2:      .RES      1      ;TEMPORARY STORAGE REGISTER
504      0315      TEMP3:      .RES      1      ;TEMPORARY STORAGE REGISTER
505      0316      SAVIO:      .RES      1      ;SAVE SERIAL IN DATA PORT
506      0317      TIMFLG:      .RES      1      ;TIME OUT FLAG FOR BAUD RATE CORRECTION
507      0318      STACKP:      .RES      1      ;SIO STACK POINTER SAVE CELL
508      0319      TSTAT:      .RES      1      ;TEMPORARY STATUS HOLDER
509      ;
510      ;
511      ;
512      031A      HATABS:      .RES      38      ;HANDLER ADDRESS TABLE
513      0021      MAXDEV      =*-HATABS-5 ;MAXIMUM HANDLER ADDRESS INDEX
514
515      ;      NOTE :      THE ENTIRE IOCB DEFINITIONS HAVE BEEN MODIFIED

```

```

516      ;
517      IOCB:      .ORG  *      ;I/O CONTROL BLOCKS
518      0340      ICHID:      .RES  1      ;HANDLER INDEX NUMBER (FF = IOCB FREE)
519      0341      ICDNO:      .RES  1      ;DEVICE NUMBER (DRIVE NUMBER)
520      0342      ICCOM:      .RES  1      ;COMMAND CODE
521      0343      ICSTA:      .RES  1      ;STATUS OF LAST IOCB ACTION
522      0344      ICBAL:      .RES  1      ;BUFFER ADDRESS LOW BYTE
523      0345      ICBAH:      .RES  1
524      0346      ICPTL:      .RES  1      ;PUT BYTE ROUTINE ADDRESS - 1
525      0347      ICPH:      .RES  1
526      0348      ICBLL:      .RES  1      ;BUFFER LENGTH LOW BYTE
527      0349      ICBHL:      .RES  1
528      034A      ICAX1:      .RES  1      ;AUXILIARY INFORMATION FIRST BYTE
529      034E      ICAX2:      .RES  1
530      034C      ICSPR:      .RES  4      ;FOUR SPARE BYTES
531      0350      .RES  MAXIOC-IOCBSZ
532      ;
533      03C0      PRNBUF:      .RES  40      ;PRINTER BUFFER
534      ;
535      03E8      .RES  21      ;SPARE BYTES
536      ;
537      ;
538      ;
539      ;
540      ;
541      ;
542      ;
543      ;      PAGE  FOUR RAM ASSIGNMENTS
544      ;
545      03FD      CASBUF:      .RES  131      ;CASSETTE BUFFER
546      ;
547      ; USER  AREA STARTS HERE AND      GOES TO END OF PAGE FIVE
548      0480      USAREA:      .RES  128      ;SPARE
549      ;
550      ;
551      ;
552      ;
553      ;
554      ;

```

```

555 ;
556 ; PAGE FIVE RAM ASSIGNMENTS
557 ;
558 ; PAGE FIVE IS RESERVED AS A USER WORK SPACE
559 ;
560 ; NOTE: SEE FLOATING POINT SUBROUTINE AREA FOR PAGE FIVE CELLS
561 ;
562 ;
563 ; PAGE SIX RAM ASSIGNMENTS
564 ;
565 ; PAGE SIX IS RESERVED AS A USER'S USER WORK SPACE
566 ;
567 ;
568 ;
569 ;
570 ; FLOATING POINT SUBROUTINES
571 ;
572 0006 FPREC = 6 ;FLOATING POINT PRECISION (# OF BYTES)
573 ; IF CARRY USED THEN CARRY CLEAR -> NO ERROR, CARR
574 D800 AFP = $D800 ;ASCII->FLOATING POINT (FP)
575 ; INBUFF+CIX -> FR0, CIX, CARRY
576 DBE6 FASC = $D8E6 ;FP -> ASCII FR0 -> LBUFF (INBUFF)
577 D9AA IFP = $D9AA ;INTEGER -> FP
578 ; 0-$FFFF (LSB, MSB) IN FR0, FR0+1 -> FR0
579 D9D2 FPI = $D9D2 ;FP -> INTEGER FR0 -> FR0, FR0+1, CARRY
580 DA60 FSUB = $DA60 ;FR0 <- FR0 - FR1 CARRY
581 DA66 FADD = $DA66 ;FR0 <- FR0 + FR1 CARRY
582 DADS FMUL = $DADB ;FR0 <- FR0 * FR1 CARRY
583 DB28 FDIV = $DB28 ;FR0 <- FR0 / FR1 .CARRY
584 DD89 FLD0R = $DD89 ;FLOATING LOAD REG0 FR0 <- (X, Y)
585 DD8D FLD0P = $DD8D ; " " REG0 FR0 <- (FLPTR)
586 DD98 FLD1R = $DD98 ; " " REG1 FR1 <- (X, Y)
587 DD9C FLD1P = $DD9C ; " " REG1 FR1 <- (FLPTR)
588 DDA7 FST0R = $DDA7 ;FLOATING STORE REG0 (X, Y) <- FR0
589 DDAB FST0P = $DDAB ; " " REG0 (FLPTR) <- FR0
590 DDB6 FMOVE = $DDB6 ;FR1 <- FR0
591 DD40 PLYEVL = $DD40 ;FR0 <- P(Z) = SUM(I=N TO 0) (A(I)*Z**I) CAR
592 ; INPUT: (X, Y) = A(N), A(N-1). . . A(0) -> PLYARG
593 ; ACC = # OF COEFFICIENTS = DEGREE+1

```



```

594      ;          FR0 = Z
595      DDC0      EXP      =      $DDCO      ;FR0 <- E**FR0 = EXP10(FR0 * LOG10(E)) CARRY
596      DDCC      EXP10    =      $DDCC      ;FR0 <- 10**FR0 CARRY
597      DECD      LOG      =      $DECD      ;FR0 <- LN(FR0) = LOG10(FR0)/LOG10(E) CARRY
598      DED1      LOG10    =      $DED1      ;FR0 <- LOG10(FR0) CARRY
599      ;          THE FOLLOWING ARE IN BASIC CARTRIDGE:
600      BD81      SIN      =      $BD81      ;FR0 <- SIN(FR0) DEGFLG=0 =>RADS, 6=>DEG. CA
601      BD73      COS      =      $BD73      ;FR0 <- COS(FR0) CARRY
602      BE43      ATAN     =      $BE43      ;FR0 <- ATAN(FR0) CARRY
603      BEB1      SQR      =      $BEB1      ;FR0 <- SQUAREROOT(FR0) CARRY
604      ; FLOATING POINT ROUTINES ZERO PAGE (NEEDED ONLY IF F. P. ROUTINES ARE CA
605      ;          *=$D4
606      00D4      FR0:      .RES  FPREC      ;FP REG0
607      00DA      FRE:      .RES  FPREC
608      00E0      FRI:      .RES  FPREC      ;FP REG1
609      00E6      FR2:      .RES  FPREC
610      00EC      FRX:      .RES  1          ;FP SPARE
611      00ED      EEXP:     .RES  1          ;VALUE OF E
612      00EE      NSIGN:    .RES  1          ;SIGN OF #
613      00EF      ESIGN:    .RES  1          ;SIGN OF EXPONENT
614      00F0      FCHRFLG:  .RES  1          ;1ST CHAR FLAG
615      00F1      DIGRT:    .RES  1          ;# OF DIGITS RIGHT OF DECIMAL
616      00F2      CIX:      .RES  1          ;CURRENT INPUT INDEX
617      00F3      INBUFF:   .RES  2          ;POINTS TO USER'S LINE INPUT BUFFER
618      00F5      ZTEMP1:   .RES  2
619      00F7      ZTEMP4:   .RES  2
620      00F9      ZTEMP3:   .RES  2
621      00FB      DEGFLG
622      00FB      RADFLG:   .RES  1          ;0-RADIANS 6=DEGREES
623      0000      RADON     =      0          ;INDICATES RADIANS
624      0006      DEGON     =      6          ;INDICATES DEGREES
625      00FC      FLPTR:    .RES  2          ;POINTS TO USER'S FLOATING PT NUMBER
626      00FE      FPTR2:    .RES  2
627      ; FLOATING PT ROUTINES' NON-ZERO PAGE RAM
628      ; (NEEDED ONLY IF F. P. ROUTINES CALLED)
629      ;          *=$57E
630      057E      LBPR1:    .RES  1          ;LBUFF PREFIX 1
631      057E      LBPR2:    .RES  1          ;LBUFF PREFIX 2
632      0580      LBUFF:    .RES  128       ;LINE BUFFER

```

```

633 05E0      PLYARG      =      LBUFF+860      ;POLYNOMIAL ARGUMENTS
634 05E6      FPSCR       =      PLYARG+FPREC
635 05EC      FPSCR1      =      FPSCR+FPREC
636 05E6      FSCR        =      FPSCR
637 05EC      FSCR1       =      FPSCRI
638 05FF      LBFEND      =      *-1      ;END OF LBUFF
639           ;
640           ;
641           ;
642           ;
643           ;
644           ;
645           ;
646           ;
647           ;
648           ;      COLLEEN      MNEMONICS
649           ;
650 D200      POKEY        =      $D200      ;VBLANK ACTION:      DESCRIPTION:
651 D200      POT0         =      POKEY+0      ;POT0-->PADDL0      0-227 IN RAM CELL
652 D201      POT1         =      POKEY+1      ;POT1-->PADDL1      0-227 IN RAM CELL
653 D202      POT2         =      POKEY+2      ;POT2-->PADDL2      0-227 IN RAM CELL
654 D203      POTS         =      POKEY+3      ;POT3-->PADDL3      0-227 IN RAM CELL
655 D204      POT4         =      POKEY+4      ;POT4-->PADDL4      0-227 IN RAM CELL
656 D205      POT5         =      POKEY+5      ;POT5-->PADDL5      0-227 IN RAM CELL
657 D206      POT6         =      POKEY+6      ;POT6-->PADDL6      0-227 IN RAM CELL
658 D207      POT7         =      POKEY+7      ;POT7-->PADDL7      0-227 IN RAM CELL
659 D208      ALLPOT       =      POKEY+8      :???
660 D209      KBCODE       =      POKEY+9
661 D20A      RANDOM       =      POKEY+10
662 D20B      POTGO        =      POKEY+11      ;STROBED
663 D20D      SERIN        =      POKEY+13
664 D20E      IRQST       =      POKEY+14
665 D20F      SKSTAT      =      POKEY+15
666 D200      AUDF1       =      POKEY+0
667 D201      AUDC1       =      POKEY+1
668 D202      AUDF2       =      POKEY+2
669 D203      AUDC2       =      POKEY+3
670 D204      AUDF3       =      POKEY+4
671 D205      AUDC3       =      POKEY+5

```

```

672 D206 AUDF4 = POKEY+6
673 D207 AUDC4 = POKEY+7
674 D208 AUDCTL = POKEY+B ;NONE AUDCTL<--(SIO)
675 D209 STIMER = POKEY+9
676 D20A SKRES = POKEY+10 ;NONE SKRES<--(SIO)
677 D20D SEROUT = POKEY+13 ;NONE SEROUT<--CSI07
678 D20E IRQEN = POKEY+14 ;POKMSK-->IRQEN (AFFECTED BY OPEN S: OR E:)
679 D20F SKCTL = POKEY+15 ;SSKCTL-->SKCTL SSKCTL<--CSI07
680 ;
681 D000 CTIA = $D000 ;VBLANK ACTION: DESCRIPTION:
682 D000 HPOSP0 = CTIA+0
683 D001 HPOSP1 = CTIA+1
684 D002 HPOSP2 = CTIA+2
685 D003 HPOSP3 = CTIA+3
686 D004 HPOSM0 = CTIA+4
687 D005 HPOSM1 = CTIA+5
688 D006 HPOSM2 = CTIA+6
689 D007 HPOSM3 = CTIA+7
690 D008 SIZEP0 = CTIA+8
691 D009 SIZEP1 = CTIA+9
692 D00A SIZEP2 = CTIA+10
693 D00B SIZEP3 = CTIA+11
694 D00C SIZEM = CTIA+12
695 D00D GRAFP0 = CTIA+13
696 D00E GRAFPI = CTIA+14
697 D00F GRAFP2 = CTIA+15
698 D010 GRAFP3 = CTIA+16
699 D011 GRAFM = CTIA+17
700 D012 COLPM0 = CTIA+18 ;PCOLR0-->COLPM0 WITH ATTRACT MODE
701 D013 COLPMI = CTIA+19 ;PCOLR1-->COLPMI WITH ATTRACT MODE
702 D014 COLPM2 = CTIA+20 ;PCOLR2-->COLPM2 WITH ATTRACT MODE
703 D015 COLPM3 = CTIA+21 ;PCOLR3-->COLPM3 WITH ATTRACT MODE
704 D016 COLPF0 = CTIA+22 ;COLOR0-->COLPF0 WITH ATTRACT MODE
705 D017 COLPF1 = CTIA+23 ;COLOR1-->COLPF1 WITH ATTRACT MODE
706 D018 COLPF2 = CTIA+24 ;COLOR2-->COLPF2 WITH ATTRACT MODE
707 D019 COLPF3 = CTIA+25 ;COLOR3-->COLPF3 WITH ATTRACT MODE
708 D01A COLBK = CTIA+26 ;COLOR4-->COLBK WITH ATTRACT MODE
709 D01B PRIOR = CTIA+27 ;(ON OPEN S: OR E :) GPRIOR-->PRIOR
710 D01C VDELAY = CTIA+28

```

```

711 D01D GRCTL = CTIA+29
712 D01E HITCLR = CTIA+30
713 D01F CONSOL = CTIA+31 ; *08-->CONSOL TURN OFF SPEAKER
714 D000 M0PF = CTIA+0
715 D001 M1PF = CTIA+1
716 D002 M2PF = CTIA+2
717 D003 M3PF = CTIA+3
718 D004 P0PF = CTIA+4
719 D005 P1PF = CTIA+5
720 D006 P2PF = CTIA+6
721 D007 P3PF = CTIA+7
722 D008 M0PL = CTIA+8
723 D009 M1PL = CTIA+9
724 D00A M2PL = CTIA+10
725 D00B M3PL = CTIA+11
726 D00C P0PL = CTIA+12
727 R00D P1PL = CTIA+13
728 D00E P2PL = CTIA+14
729 D00F P3PL = CTIA+15
730 D010 TRIG0 = CTIA+16 ; TRIG0-->STRIGO
731 D011 TRIG1 = CTIA+17 ; TRIG1-->STRIG1
732 D012 TRIG2 = CTIA+18 ; TRIG2-->STRIG2
733 D013 TRIG3 = CTIA+19 ; TRIG3-->STRIG3
734 ;
735 D400 ANTIC = $D400 ; VBLANK ACTION DESCRIPTION
736 D400 DMACTL = ANTIC+0 ; DMACTL<--SDMCTL ON OPEN S: OR E:
737 D401 CHACTL = ANTIC+1 ; CHACTL<--CHACT ON OPEN S: OR E:
738 D402 DLISTL = ANTIC+2 ; DLISTL<--SDLSTL ON OPEN S: OR E:
739 D403 DLISTH = ANTIC+3 ; DLISTH<--SDLSTH ON OPEN S: OR E:
740 D404 HSCROL = ANTIC+4
741 D405 VSCROL = ANTIC+5
742 D407 PMBASE = ANTIC+7
743 D409 CHBASE = ANTIC+9 ; CHBASE<--CHBAS ON OPEN S: OR E:
744 D40A WSYNC = ANTIC+10
745 D40B VCOUNT = ANTIC+11
746 D40C PENH = ANTIC+12
747 D40D PENV = ANTIC+13
748 D40E NMIEEN = ANTIC+14 ; NMIEEN<--40 POWER ON AND CSETVBV7
749 D40F NMIRES = ANTIC+15 ; STROBED

```

ERR LINE ADDR B1 B2 B3 B4

6500 ASSEMBLER VER 1.OMR

Page 21

```
750 D40F NMIST = ANTIC+15
751 D300 PIA = $D300 ;VBLANK ACTION DESCRIPTION
752 D300 PORTA = PIA+0 ;PORTA-->STICK0,1 X-Y CONTROLLERS
753 D301 PORTB = PIA+1 ;PORTB-->STICK2.3 X-Y CONTROLLERS
754 D302 PACTL = PIA+2 ;NONE PACTL<--3C CINIT7
755 D303 PBCTL = PIA+3 ;NONE PBCTL<--3C CINIT7
756 ;
757 ;
758 ;
759 ;.PAGE
760 ;.PAGE
761 LIST S
```