

SECTION V

QS 7 ABSOLUTE LISTINGS



COLECOVISION PROGRAMMERS' MANUAL  
 Rev. 5  
 ©Coleco Industries, Inc. 1982  
 CONFIDENTIAL DOCUMENT - DO NOT COPY

APPENDIX E  
 JUMP TABLE

1	
2	
3	
4	
5	PLAY_SOUND 1F61
6	ACTIVATE 1F64
7	PUTOBJ 1F67
8	REFLECT_VERTICAL 1F6A
9	REFLECT_HORIZONTAL 1F6D
10	ROTATE_90 1F70
11	ENLARGE 1F73
12	CONTROLLER_SCAN 1F76
13	DECODE 1F79
14	CAME_OUT 1F7C
15	LOAD_ASCII 1F7F
16	FILL_VRAM 1F82
17	MODE_1 1F85
18	UPDATE_SPINNER 1F88
19	INIT_TABLE 1F8B
20	GET_VRAM 1F8E
21	PUT_VRAM 1F91
22	INIT_SPR_ORDER 1F94
23	WR_SPR_H_TBL 1F97
24	INIT_TIMER 1F9A
25	FREE_SIGNAL 1F9D
26	REQUEST_SIGNAL 1FA0
	TEST_SIGNAL 1FA3
	WRITE_REGISTER 1FA6
	WRITE_VRAM 1FA9
	READ_VRAM 1FAC
	INIT_WRITER 1FAF
	SOUND_INIT 1FB2
	PLAY_IT 1FB5
	INIT_TABLE 1FB8
	GET_VRAM 1FBB
	PUT_VRAM 1FBE
	INIT_SPR_ORDER 1FC1
	WR_SPR_H_TBL 1FC4
	INIT_TIMER 1FC7
	FREE_SIGNAL 1FCA
	REQUEST_SIGNAL 1FCB
	TEST_SIGNAL 1FCE
	TIME_POS 1FD3
	TURN_OFF_SOUND 1FD6
	WRITE_REGISTER 1FD9
	READ_REGISTER 1FDC
	WRITE_VRAM 1FDF
	READ_VRAM 1FE2
	INIT_WRITER 1FE5
	WRITER 1FE8
	POLLER 1FEB
	SOUND_INIT 1FEE
	PLAY_IT 1FF1
	SOUND_RAM 1FF4
	ACTIVATE 1FF7
	PUTOBJ 1FFA
	RAND_GEN 1FFD



## APPENDIX F

## OS SYMBOLS

1				
2				
3				
4		ACTIVATE	EQU 01FF7H	OS:OS
		ACTIVATEP	EQU 01F64H	OS:OS
		ADD816	EQU 001B1H	OS:OS
5		AMERICA	EQU 00069H	OS:OS
		ASCII_TABLE	EQU 0006AH	OS:OS
6		ATN_SWEEP	EQU 0012FH	OS:OS
		CARTRIDGE	EQU 08000H	OS:OS
7		CONTROLLER_MAP	EQU 08008H	OS:OS
		CTRL_PORT_PTR	EQU 01D43H	
8		DATA_PORT_PTR	EQU 01D47H	
		DECLSN	EQU 00190H	OS:OS
9		DECM5N	EQU 0019BH	OS:OS
		DECODER	EQU 01F79H	OS:OS
10		DEFER_WRITES	EQU 073C6H	OS:OS
		EFXOVER	EQU 002EEH	OS:OS
11		ENLARGE	EQU 01F73H	OS:OS
		ENLRG	EQU 01D6CH	OS:OS
12		FILL_VRAM	EQU 01F82H	OS:OS
		FREE_SIGNAL	EQU 01FCAH	OS:OS
13		FREE_SIGNALP	EQU 01F9DH	OS:OS
		FREQ_SWEEP	EQU 000FCH	OS:OS
14		GAME_NAME	EQU 08024H	OS:OS
		GAME_OPT	EQU 01F7CH	OS:OS
15		GET_VRAM	EQU 01F8BH	OS:OS
		GET_VRAMP	EQU 01F8EH	OS:OS
16		INIT_SPR_ORDER	EQU 01FC1H	OS:OS
		INIT_SPR_ORDERP	EQU 01F94H	OS:OS
17		INIT_TABLE	EQU 01F88H	OS:OS
		INIT_TABLEP	EQU 01F8BH	OS:OS
18		INIT_TIMER	EQU 01FC7H	OS:OS
		INIT_TIMERP	EQU 01F9AH	OS:OS
19		INIT_WRITER	EQU 01FE5H	OS:OS
		INIT_WRITERP	EQU 01FAFH	OS:OS
20		IRQ_INT_VECT	EQU 0801EH	OS:OS
		LEAVE_EFFECT	EQU 001D5H	OS:OS
21		LOAD_ASCII	EQU 01F7FH	OS:OS
		LOCAL_SPR_TBL	EQU 08002H	OS:OS
22		MODE_1	EQU 01F85H	OS:OS
		MSNTOLSN	EQU 001A6H	OS:OS
23		MUX_SPRITES	EQU 073C7H	OS:OS
		NMI_INT_VECT	EQU 08021H	OS:OS
24		NUMBER_TABLE	EQU 0006CH	OS:OS
		PLAY_IT	EQU 01FF1H	OS:OS
25		PLAY_ITP	EQU 01FB5H	OS:OS
26		PLAY_SONGS	EQU 01F61H	OS:OS

1	POLLER	EQU 01FEBH	OS:OS
2	PUTOBJ	EQU 01FFAH	OS:OS
3	PUTOBJP	EQU 01F67H	OS:OS
4	PUT_VRAM	EQU 01FBEH	OS:OS
5	PUT_VRAMP	EQU 01F91H	OS:OS
6	RAND_GEN	EQU 01FFDH	OS:OS
7	RAND_NUM	EQU 073C8H	OS:OS
8	READ_REGISTER	EQU 01FDCH	OS:OS
9	READ_VRAM	EQU 01FE2H	OS:OS
10	READ_VRAMP	EQU 01FACH	OS:OS
11	REFLECT_HORIZON	EQU 01F6DH	OS:OS
12	REFLECT_VERTICAL	EQU 01F6AH	OS:OS
13	REQUEST_SIGNAL	EQU 01FCDH	OS:OS
14	REQUEST_SIGNALP	EQU 01FA0H	OS:OS
15	ROTATE_90	EQU 01F70H	OS:OS
16	RST_10H_RAM	EQU 0800FH	OS:OS
17	RST_18H_RAM	EQU 08012H	OS:OS
18	RST_20H_RAM	EQU 08015H	OS:OS
19	RST_28H_RAM	EQU 08018H	OS:OS
20	RST_30H_RAM	EQU 0801BH	OS:OS
21	RST_8H_RAM	EQU 0800CH	OS:OS
22	SOUND_INIT	EQU 01FEEH	OS:OS
23	SOUND_INITP	EQU 01FB2H	OS:OS
24	SOUND_MAN	EQU 01FF4H	OS:OS
25	SPRITE_ORDER	EQU 08004H	OS:OS
26	STACK	EQU 073B9H	OS:OS
	START_GAME	EQU 0800AH	OS:OS
	TEST_SIGNAL	EQU 01FD0H	OS:OS
	TEST_SIGNALP	EQU 01FA3H	OS:OS
	TIME_MGR	EQU 01FD3H	OS:OS
	TURN_OFF_SOUND	EQU 01FD6H	OS:OS
	UPDATE_SPINNER	EQU 01F88H	OS:OS
	VDP_MODE_WORD	EQU 073C3H	OS:OS
	VDP_STATUS_BYTE	EQU 073C5H	OS:OS
	WORK_BUFFER	EQU 08006H	OS:OS
	WRITER	EQU 01FEBH	OS:OS
	WRITE_REGISTER	EQU 01FD9H	OS:OS
	WRITE_REGISTERP	EQU 01FA6H	OS:OS
	WRITE_VRAM	EQU 01FDFH	OS:OS
	WRITE_VRAMP	EQU 01FA9H	OS:OS
	WR_SPR_NM_TBL	EQU 01FC4H	OS:OS
	WR_SPR_NM_TBLP	EQU 01F97H	OS:OS

1	GLB ACTIVATE	OS:OS
2	GLB ACTIVATED	OS:OS
	GLB ADD816	OS:OS
3	GLB AMERICA	OS:OS
	GLB ASCII_TABLE	OS:OS
4	GLB ATN_SWEEP	OS:OS
	GLB CARTRIDGE	OS:OS
5	GLB CONTROLLER_MAP	OS:OS
	GLB CTRL_PORT_PTR	
6	GLB DATA_PORT_PTR	
	GLB DECLSN	OS:OS
7	GLB DECM5N	OS:OS
	GLB DECODER	OS:OS
	GLB DEFER_WRITES	OS:OS
8	GLB EFXOVER	OS:OS
	GLB ENLARGE	OS:OS
9	GLB ENLRC	OS:OS
	GLB FILL_VRAM	OS:OS
10	GLB FREE_SIGNAL	OS:OS
	GLB FREE_SIGNALP	OS:OS
11	GLB FREQ_SWEEP	OS:OS
	GLB GAME_NAME	OS:OS
12	GLB GAME_OPT	OS:OS
	GLB GET_VRAM	OS:OS
13	GLB GET_VRAMP	OS:OS
	GLB INIT_SPR_ORDER	OS:OS
14	GLB INIT_SPR_ORDERP	OS:OS
	GLB INIT_TABLE	OS:OS
15	GLB INIT_TABLEP	OS:OS
	GLB INIT_TIMER	OS:OS
16	GLB INIT_TIMERP	OS:OS
	GLB INIT_WRITER	OS:OS
17	GLB INIT_WRITERP	OS:OS
	GLB IRQ_INT_VECT	OS:OS
18	GLB LEAVE_EFFECT	OS:OS
	GLB LOAD_ASCII	OS:OS
19	GLB LOCAL_SPR_TBL	OS:OS
	GLB MSNTOLSN	OS:OS
20	GLB MODE_1	OS:OS
	GLB MUX_SPRITES	OS:OS
21	GLB NMI_INT_VECT	OS:OS
	GLB NUMBER_TABLE	OS:OS
22	GLB PLAY_IT	OS:OS
	GLB PLAY_ITP	OS:OS
23	GLB PLAY_SONGS	OS:OS
	GLB POLLER	OS:OS
24	GLB PUTOBJ	OS:OS
25	GLB PUTOBJP	OS:OS
26		

1		
2	GLB PUT_VRAM	OS:OS
3	GLB PUT_VRAMP	OS:OS
4	GLB RAND_GEN	OS:OS
5	GLB RAND_NUM	OS:OS
6	GLB READ_REGISTER	OS:OS
7	GLB READ_VRAM	OS:OS
8	GLB READ_VRAMP	OS:OS
9	GLB REFLECT_HORIZON	OS:OS
10	GLB REFLECT_VERTICAL	OS:OS
11	GLB REQUEST_SIGNAL	OS:OS
12	GLB REQUEST_SIGNALP	OS:OS
13	GLB ROTATE_90	OS:OS
14	GLB RST_10H_RAM	OS:OS
15	GLB RST_18H_RAM	OS:OS
16	GLB RST_20H_RAM	OS:OS
17	GLB RST_28H_RAM	OS:OS
18	GLB RST_30H_RAM	OS:OS
19	GLB RST_8H_RAM	OS:OS
20	GLB SOUND_INIT	OS:OS
21	GLB SOUND_INITP	OS:OS
22	GLB SOUND_MAN	OS:OS
23	GLB SPRITE_ORDER	OS:OS
24	GLB STACK	OS:OS
25	GLB START_GAME	OS:OS
26	GLB TEST_SIGNAL	OS:OS
	GLB TEST_SIGNALP	OS:OS
	GLB TIME_MGR	OS:OS
	GLB TURN_OFF_SOUND	OS:OS
	GLB UPDATE_SPINNER	OS:OS
	GLB VDP_MODE_WORD	OS:OS
	GLB VDP_STATUS_BYTE	OS:OS
	GLB WORK_BUFFER	OS:OS
	GLB WRITER	OS:OS
	GLB WRITE_REGISTER	OS:OS
	GLB WRITE_REGISTERP	OS:OS
	GLB WRITE_VRAM	OS:OS
	GLB WRITE_VRAMP	OS:OS
	GLB WR_SPR_NM_TBL	OS:OS
	GLB WR_SPR_NM_TBLP	OS:OS



## LOCATION OBJECT CODE LINE SOURCE LINE

```

1 "Z80"
2 NAME "OS_7PRIME"
3
4
5 DESCRIPTION MACRO
6 .GOTO ENDESCRIPTION
7
8 Author: Coleco Industries Inc.
9 Advanced Research & Development - Software Engineering
10 Userid: OS
11 Starting date: A long long time ago in a galaxy far far away . . .
12
13 Prom release Date: 24 Nov 1982. for internal use only
14 Prom release Rev: 7B
15
16 Prom release Date: December 28, 1982
17 Prom release Rev: 7PRIME
18
19 Header Rev: 2
20
21 *****
22 * ColecoVision Operating System
23 * Absolute Listing ( Rev 7PRIME )
24 * (c) Coleco Industries 1982
25 *
26 *
27 * *** Confidential ***
28 *
29 *****
30
31 This listing has the actual addresses of the start of OS routines
32
33 Rev History (one line note indicating the change)
34
35 Rev. Date Change
36 4 14feb1983 Filler locations changed to 0FFH to
37 reflect OS_7PRIME. Prom release date
38 changed to December 28, 1982 from May
39 1982. Name change to OS_7PRIME to
40 reflect majority of versions in the
41 field at this date.
42
43
44 3 24nov1982 Timing change to shorten LOGO delay
45 Title changes to JMPTABLES and OSSR_EQU
46 Minor comment modifications
47 2 6oct1982
48 1 23sept1982 OS_7 as one absolute file
49 0 May 1982 OS_7 listing by module
50 ENDESCRIPTION:
51 MEND
52 PROG

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
55 ; Operating system sound routine EQUATES
56 ; FILE NAME: OSSR.EQU
57 ; *** Equates ***
58 ; Dedicated Cartridge RAM locations
59 DEDAREA EQU 7020H ;the start of the RAM area dedicated to sound routines
60 PTR TO LIST OF SMD ADDRS EQU DEDAREA+0
61 PTR TO S OM 0 EQU DEDAREA+2
62 PTR TO S OM 1 EQU DEDAREA+4
63 PTR TO S OM 2 EQU DEDAREA+6
64 PTR TO S OM 3 EQU DEDAREA+8
65 SAVE_CTRL EQU DEDAREA+10
66 ; Attenuation level codes
67 OFF EQU 0FH ;OFF [NO SOUND]
68 ; Sound output port
69 SOUND_PORT EQU 0FFH ;data to sound chip thru this port
70 ; Special byte 0 codes
71 INACTIVE EQU 0FFH
72 SEFFECT EQU 62
73 ENDSDATA EQU 0
74 ; Offsets within an SxDATA song data area
75 CH EQU 0
76 SONGNO EQU 0
77 NEXTNOTEPTR EQU 1
78 FREQ EQU 3
79 ATM EQU 4
80 CTRL EQU 4
81 MLEN EQU 5
82 FPS EQU 6
83 FPSV EQU 6
84 FSTEP EQU 7
85 ALEN EQU 8
86 ASTEP EQU 8
87 APS EQU 9
88 APSV EQU 9
89 ; song end codes
90 CH0END EQU 00010000H
91 CH1END EQU 01010000H
92 CH2END EQU 10010000H
93 CH3END EQU 11010000H
94 CH0REP EQU 00011000H
95 CH1REP EQU 01011000H
96 CH2REP EQU 10011000H
97 CH3REP EQU 11011000H
98 ; channel numbers, B7 -B6
99 CH0 EQU 0
100 CH1 EQU 01000000H
101 CH2 EQU 10000000H
102 CH3 EQU 11000000H
103 ; [page]
104 PROG
105

```

LOCATION OBJECT CODE LINE SOURCE LINE

```

107
108 ;
109 ;
110 ;
111 ;
112 ***** EXTERNAL SYMBOLS *****
113
114 * EXTERNAL ROUTINES LINKED INTO OS
115
116 ;EXT REG_WRITE
117 ;EXT REG_READ
118 ;EXT VRAM_WRITE
119 ;EXT VRAM_READ
120 ;EXT INIT_QUEUE
121 ;EXT WRITER
122 ;EXT REG_WRITEQ
123 ;EXT VRAM_WRITEQ
124 ;EXT VRAM_READQ
125 ;EXT INIT_QUEUEQ
126
127 ;EXT POLLER
128 ;EXT UPDATE_SPINNER
129 ;EXT CONT_SCAN
130 ;EXT DECODER
131
132 ;EXT INIT_SOUND
133 ;EXT ALL_OFF
134 ;EXT JUKE_BOX
135 ;EXT SMO_MANAGER
136 ;EXT PLAY_SONGS
137 ;EXT INIT_SOUNDQ
138 ;EXT JUKE_BOXQ
139
140 ;EXT INIT_TIMER
141 ;EXT FREE_SIGNAL
142 ;EXT REQUEST_SIGNAL
143 ;EXT TEST_SIGNAL
144 ;EXT TIME_MGR
145 ;EXT INIT_TIMERQ
146 ;EXT FREE_SIGNALQ
147 ;EXT REQUEST_SIGNALQ
148 ;EXT TEST_SIGNALQ
149
150 ;EXT INIT_TABLE
151 ;EXT GET_VRAM
152 ;EXT PUT_VRAM
153 ;EXT INIT_SPR_ORDER
154 ;EXT WR_SPR_MM_TBL
155 ;EXT INIT_TABLEQ
156 ;EXT GET_VRAMQ
157 ;EXT PUT_VRAMQ
158 ;EXT INIT_SPR_ORDERQ
159 ;EXT WR_SPR_MM_TBLQ
160
161 ;EXT ACTIVATE
162 ;EXT PUTOBJ
163 ;EXT FLCT_VERT

```

Modified February 14, 1983. Filler areas were changed to OFFH to reflect OS 7PRIME. Also minor comment changes were made to clarify 055AAH for test cartridge condition.

\*\*\*\*\* EXTERNAL SYMBOLS \*\*\*\*\*

\*\*\*\*\* VIDEO DRIVERS \*\*\*\*\*

\*\*\*\*\* PASCAL CALLS \*\*\*\*\*

\*\*\*\*\* CONTROLLER ROUTINE \*\*\*\*\*

\*\*\*\*\* SOUND ROUTINES \*\*\*\*\*

\*\*\*\*\* PASCAL CALLS \*\*\*\*\*

\*\*\*\*\* TIME MGMT ROUTINES \*\*\*\*\*

\*\*\*\*\* PASCAL CALLS \*\*\*\*\*

\*\*\*\*\* TABLE MA \*\*\*\*\*

\*\*\*\*\* PASCAL CALLS \*\*\*\*\*

\*\*\*\*\* GRAPHICS ROUTINES \*\*\*\*\*

LOCATION	OBJECT CODE	LINE	SOURCE LINE
164			;EXT RFLCT HOR
165			;EXT ROT 90
166			;EXT ENLGR
167			;EXT PUTOBJG
168			;EXT ACTIVATED
169			
170			;EXT GAME OPT
171			;EXT LOAD_ASCII
172			;EXT FILL VRAM
173			;EXT MODE_1
174			
175			
176	*	"HIDDEN EXTERNALS"	
177			
178			;EXT
179			;EXT
180			;EXT
181			;EXT
182			
183			
184			
185			
186	*	ENTRY POINTS TO OS ROUTINES	
187			
188			GLB INIT TABLE
189			GLB GET_VRAM
190			GLB PUT_VRAM
191			GLB INIT_SPR_ORDER
192			GLB WR_SPR_MM_TBL
193			GLB INIT_TABLEP
194			GLB GET_VRAMP
195			GLB PUT_VRAMP
196			GLB INIT_SPR_ORDERP
197			GLB WR_SPR_MM_TBLP
198			
199			GLB WRITE_REGISTER
200			GLB READ_REGISTER
201			GLB WRITE_VRAM
202			GLB READ_VRAM
203			GLB INIT_WRITER
204			GLB WRITER
205			GLB WRITE_REGISTERP
206			GLB WRITE_VRAMP
207			GLB READ_VRAMP
208			GLB INIT_WRITERP
209			
210			GLB POLLER
211			GLB UPDATE_SPINNER
212			GLB CONTROLLER_SCAN
213			GLB DECODER
214			
215			GLB SOUND_INIT
216			GLB TURN_OFF_SOUND
217			GLB PLAY_IT
218			GLB SOUND_MAN
219			GLB PLAY_SONGS
220			GLB SOUND_INITP

## LOCATION OBJECT CODE LINE SOURCE LINE

```

221      GLB PLAY_ITP
222
223      GLB INIT_TIMER
224      GLB FREE_SIGNAL
225      GLB REQUEST_SIGNAL
226      GLB TEST_SIGNAL
227      GLB TIME_MGR
228      GLB INIT_TIMERP
229      GLB FREE_SIGNALP
230      GLB REQUEST_SIGNALP
231      GLB TEST_SIGNALP
232
233      GLB STACK
234      GLB VOP_STATUS_BYTE
235      GLB VOP_MODE_WORD
236      GLB AMERICA
237      GLB MUX_SPRITES
238      GLB DEFER_WRITES
239      GLB RAND_GEN
240
241      GLB PUTOBJ
242      GLB ACTIVATE
243      GLB REFLECT_VERTICAL
244      GLB REFLECT_HORIZONTAL
245      GLB ROTATE_90
246      GLB ENLARGE
247      GLB PUTOBJP
248      GLB ACTIVATEP
249
250      GLB GAME_OPT
251      GLB LOAD_ASCII
252      GLB FILL_VRAM
253      GLB MODE_1
254      GLB ASCII_TABLE
255      GLB NUMBER_TABLE
256
; TIME MGMT ROUTINES
;
; PASCAL CALLS
;
; MISC GLOBALS
;
; Can be called from Pascal
; or assembly language
; GRAPHICS ROUTINES
;
; PASCAL CALLS
;
; GAME OPTIONS DISPLAY
; LOADS ASCII CHARACTER GENERATORS
; FILLS DESIGNATED AREA OF VRAM WITH VALUE
; SETS UP A DEFAULT GRAPHICS MODE 1
; POINTER TO TABLE OF ASCII GENERATORS
; POINTER TO TABLE OF 0-9 PATTERN GENERATORS

```

## LOCATION OBJECT CODE LINE SOURCE LINE

```

258 ***** CARTRIDGE ROM DATA AREA *****
259
260 GLB CARTRIDGE EQU CARTRIDGE
261 <8000> GLB EQU 8000H
262 * THIS IS THE MEMORY LOCATION TESTED TO SEE IF A CARTRIDGE IS PLUGGED
263 * IN. IF IT CONTAINS THE PATTERN AA55H THE OS ASSUMES THAT A GAME
264 * CARTRIDGE IS PRESENT. IF IT CONTAINS THE PATTERN 55AAH, THE OS
265 * ASSUMES THAT A TEST CARTRIDGE IS PRESENT.
266
267 GLB LOCAL_SPR_TBL LOCAL_SPR_TBL
268 <8002> GLB EQU 8002H
269 * THIS IS A POINTER TO THE CPU RAM COPY OF THE SPRITE NAME TABLE. THE
270 * TABLE COPY IS USED WHENEVER ONE LEVEL OF INDIRECTION IS DESIRED IN
271 * ADDRESSING THE VRAM TABLE. FOR EXAMPLE WHEN USING THE OS SPRITE
272 * MULTIPLEXING SOFTWARE.
273
274 GLB SPRITE_ORDER SPRITE_ORDER
275 <8004> GLB EQU 8004H
276 * THIS IS A POINTER TO THE CPU RAM SPRITE ORDER TABLE. THIS TABLE IS
277 * USED TO ORDER THE LOCAL SPRITE NAME TABLE.
278
279 GLB WORK_BUFFER WORK_BUFFER
280 <8006> GLB EQU 8006H
281 * THIS IS A POINTER TO A FREE BUFFER SPACE IN RAM. THE OBJECT ORIENTED
282 * GRAPHICS ROUTINES USED THIS BUFFER FOR TEMPORARY STORAGE.
283
284 GLB CONTROLLER_MAP CONTROLLER_MAP
285 <8008> GLB EQU 8008H
286 * THIS IS A POINTER TO THE CONTROLLER MEMORY MAP THAT IS MAINTAINED BY
287 * THE HIGH-LEVEL CONTROLLER SCANNING AND DEBOUNCE SOFTWARE.
288
289 GLB START_GAME START_GAME
290 <800A> GLB EQU 800AH
291 * THIS IS A POINTER TO THE START OF THE GAME.
292
293 ***** RESTART AND INTERRUPT VECTORS *****
294 * THESE ARE ADDRESSES IN CARTRIDGE ROM OF VECTORS WHICH MUST BE PLACED
295 * THERE BY THE CARTRIDGE PROGRAMMER. WHEN AN INTERRUPT OR RESTART
296 * OCCURS, THE OS VECTORS IT THROUGH THIS AREA. THE CARTRIDGE PROGRAMMER
297 * SHOULD PLACE A JUMP TO HIS OWN INTERRUPT HANDLER IN THE APPROPRIATE
298 * LOCATION.
299
300 GLB RST_8H_RAM RST_8H_RAM
301 <800C> GLB EQU 800CH
302 * THIS IS THE RESTART 8 SOFT VECTOR.
303
304 GLB RST_10H_RAM RST_10H_RAM
305 <800F> GLB EQU 800FH
306 * THIS IS THE RESTART 10 SOFT VECTOR.
307
308 GLB RST_18H_RAM RST_18H_RAM
309 <8012> GLB EQU 8012H
310 * THIS IS THE RESTART 18 SOFT VECTOR.
311
312 GLB RST_20H_RAM RST_20H_RAM
313 <8015> GLB EQU 8015H
314 * THIS IS THE RESTART 20 SOFT VECTOR.

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
315
316 GLB RST_28H_RAM
317 EQU 8018H
<8018> 318 * THIS IS THE RESTART 28 SOFT VECTOR.
319
320 GLB RST_30H_RAM
321 EQU 801BH
<801B> 322 * THIS IS THE RESTART 30 SOFT VECTOR.
323
324 GLB IRQ_INT_VECT
325 EQU 801EH
<801E> 326 * THIS IS THE MASKABLE INTERRUPT SOFT VECTOR
327
328 GLB MMI_INT_VECT
329 EQU 8021H
<8021> 330 * THIS IS THE MMI SOFT VECTOR.
331
332 GLB GAME_NAME
333 EQU 8024H
<8024> 334 * FROM HERE TO START GAME THERE SHOULD BE A STRING OF ASCII CHARACTERS
335 * NAMES THAT HAS THE FOLLOWING FORM:
336 *
337 * NAME_OF_THIS_GAME/MAKER_OF_THIS_GAME/COPYWRITE_YEAR.
338 *
339 * FOR EXAMPLE:
340 *
341 * "DONKEY KONG/NINTEMDO/1982"
342
343 * IMPORTANT NOTE *****
344
345 * ***** IT IS THE RESPONSIBILITY OF THE *****
346 * ***** CARTRIDGE PROGRAMMER TO PLACE *****
347 * ***** THESE CODES IN CARTRIDGE ROM *****
348

```

LOCATION OBJECT CODE LINE SOURCE LINE

```
350 *****
351 *****
352 *
353 * OPERATING SYSTEM ROM CODE
354 *
355 *****
356 *****
357 ***** PAGE ZERO *****
358 * PAGE ZERO CONTAINS THE RESTART VECTORS, INTERRUPT VECTORS, AND
359 * THE INTERRUPT VECTORING SOFTWARE, AS WELL AS THE DEFAULT HANDLERS
360 * FOR INTERRUPTS AND RESTARTS.
361
362 * BOOT-UP ROUTINE
363
364 * THE BOOT-UP ROUTINE HANDLES POWER ON RESETS AND RESTARTS TO 0. IT
365 * INITIALIZES THE STACK AND JUMPS TO THE POWER_UP ROUTINE.
366
367 * BEGIN BOOT-UP
368 BOOT_UP PROG
369
370 * KICK STACK
371 LD SP,STACK
372
373 * JUMP TO POWER_UP
374
375 JP POWER_UP
376 END BOOTUP
377 * END BOOT-UP
```

0000 317389

0003 C3004E  
0006



LOCATION OBJECT CODE LINE SOURCE LINE

```

379
380 * RESTART VECTORS
381
382 * THE FOLLOWING ARE THE 8 PROGRAMMABLE RESTARTS. FOR EACH OF THE
383 * RESTART LOCATIONS BELOW THERE IS A VECTOR IN CARTRIDGE ROM.
384 * TO USE A RESTART, THE PROGRAMMER MUST PLACE THE ADDRESS OF THE
385 * ROUTINE WHICH HE/SHE WISHES TO ACCESS THROUGH THE RESTART AT THE
386 * CORRESPONDING VECTOR. THEREAFTER EVERY TIME THAT RESTART IS
387 * EXECUTED, THE CARTRIDGE PROGRAMMER'S ROUTINE WILL BE CALLED.
388
0006 FFFF          HEX          FF,FF
0008 C3800C        JP RST_8H    ; Filler
389
0008 FFFFFFFF      HEX          FF,FF,FF,FF
0010 C3800F        JP RST_10H   ; Filler
391
0013 FFFFFFFF      HEX          FF,FF,FF,FF
0018 C38012        JP RST_18H   ; Filler
394
0018 FFFFFFFF      HEX          FF,FF,FF,FF
0020 C38015        JP RST_20H   ; Filler
397
0023 FFFFFFFF      HEX          FF,FF,FF,FF
0028 C38018        JP RST_28H   ; Filler
401
0028 FFFFFFFF      HEX          FF,FF,FF,FF
0030 C3801B        JP RST_30H   ; Filler
403
404
405
406

```

LOCATION OBJECT CODE LINE SOURCE LINE

```

408
409 * MASKABLE INTERRUPT VECTURING SOFTWARE
410
411 * A MASKABLE INTERRUPT OCCURRING IN THE SYSTEM IS EQUIVALENT TO A
412 * RESTART TO 30H. THUS, THE MASKABLE INTERRUPT IS VECTORED IN EXACTLY
413 * THE SAME WAY AS THE VARIOUS RESTARTS GIVEN ABOVE. IN ORDER TO USE
414 * THE INTERRUPT, THE CARTRIDGE MUST PLACE THE ADDRESS OF HIS/HER
415 * INTERRUPT HANDLER IN THE IRQ_INT_VECT LOCATION IN CARTRIDGE ROM.
416
417 * THE CARTRIDGE PROGRAMMER IS RESPONSIBLE FOR SAVING ANY REGISTERS
418 * HIS/HER OWN INTERRUPT HANDLERS MAY USE, AND FOR RE-ENABLING
419 * INTERRUPTS IF HE/SHE NEEDS THEM TO BE RE-ENABLED.
420
421 * MASKABLE INTERRUPT
422     FF,FF,FF,FF,FF      ;Filler
423     IRQ_INTERRUPT      ;30H
424     JP                (IRQ_INT_VECT)
425
426 ***** RANDOM NUMBER GENERATOR *****
427
428 * (PLACED HERE FOR PURPOSES OF CODE COMPACTION)
429
430 * Random number generator (psuedo) for a 16 bit value
431 * This routine 'exclusive or's the 15th and 8th bit
432 * together. It then rotates the entire quantity to the
433 * left and inserts the 'exclusive or'd bit into the rightmost
434 * bit. Upon leaving it stores the random # in a specified
435 * memory location.
436
437 * The random number can be accessed from the global location
438 * RAND_NUM or the HL pair or the Accumulator.
439
440 RAND_GEN:
441     LD HL,[RAND_NUM]
442     BIT 7,H
443     JR Z,NOT_ON
444
445     BIT 0,H
446     JR Z,SET
447     JR RESET
448 NOT_ON:
449     BIT 0,H
450     JR Z,RESET
451
452 SET:
453     SCF
454     JR CARRY_READY
455 RESET:
456     OR A
457 CARRY_READY:
458     RL L
459     RL H
460     LD LD,[RAND_NUM],HL
461     LD A,L
462     RET
463 AFTER_RANDOM
464
0038 FFFFFFFF
0039
003B 2A73C8
003C C87C
0040 2806
0042 CB44
0044 2806
0046 1807
0048
004B CB44
004C 2803
004C
004C 37
004D 1801
004F
004F B7
0050
0050 CB15
0052 CB14
0054 2273C8
0057 7D
0058 C9
0059

```

LOCATION OBJECT CODE LINE      SOURCE LINE

465 .....  
466 .....

## LOCATION OBJECT CODE LINE SOURCE LINE

```

468
469 * THE NMI VECTORING SOFTWARE AND DEFAULT HANDLER
470
471 * WHEN AN NMI IS RAISED BY THE VDP IN THE COLECOVISION SYSTEM, IT
472 * CAUSES THE CPU TO RESTART TO 66H. THE VECTORING SOFTWARE FOR THE
473 * NMI IS IDENTICAL TO THAT FOR THE MASKABLE INTERRUPT EXCEPT THAT
474 * IT GETS ITS VECTOR FROM NMI_INT_VECT INSTEAD OF IRQ_INT_VECT.
475
476 * AGAIN THE CARTRIDGE PROGRAMMER IS RESPONSIBLE, IN HIS/HER OWN
477 * INTERRUPT HANDLERS FOR SAVING AND RESTORING THE PROCESSOR STATE
478 * WHEN NECESSARY, AND FOR CLEARING THE VDP CONDITION BY READING THE
479 * VDP STATUS REGISTER.
480
0059 FFFFFFFF      HEX      FF,FF,FF,FF,FF      ;Filler
005E FFFFFFFF      HEX      FF,FF,FF,FF,FF      ;Filler
0063 FFFFFF        HEX      FF,FF,FF      ;Filler
484 * NON-MASKABLE INTERRUPT
485 NMI_INTERRUPT    JP      (NMI_INT_VECT)
0066 C38021
0066 C38021
487

```