
SECTION X

DEFINED REFERENCE LOCATIONS

10.1 OS ROM

In the OS ROM area, it is IMPORTANT to know that the application programs should only use the OS entry points listed in the OS_SYMBOLS file. Accessing to the OS otherwise is illegal and may cause program malfunction when hardware configuration changes or OS routines relocated due to update. The jump table starts from location JUMP_TABLE through the end of OS ROM. It contains all the subroutine entry points released to the user.

At the beginning of the cartridge, there are eight programmable restarts at addresses 0008H, 0010H, 0018H, 0020H, 0028H and 0030H. Each of the restarts jump to a location in Cartridge ROM where a vector can be provided to access an OS entry point. The Z80A-CPU hardware also designates location 0038H to service maskable interrupt

(MI) and location 0066H to service non-maskable interrupt (NMI). Jump instructions are provided for these two reference locations for the user to implement interrupt vectors in Cartridge ROM. Starting at location 0069H is the OS ROM data area which contains the AMERICA byte, ASCII table address and numeric table address. Figure 10-1 is the OS ROM map showing all the reference locations mentioned above. Appendix E lists all entry points of the Jump Table.

1	0000Н	
2	0008Н	8 Restarts
3		o Restarts
4	0033H	A
5	0038Н	MI VECTOR
6	003BH	
7	0066Н	NAT INCHOR
8		NMI VECTOR
9	0069Н	OS ROM DATA
10		AREAS
11	006EH	
	ا	os ROUTINES $\stackrel{\downarrow}{\approx}$
12		03 100111123
13	1F61H	
14	1FFFH	JUMP TABLE
15		Figure 10-1
16		OS ROM MAP
17		O2 ROW PAY

10.1.1 Europe/America Byte:

18

19

20

21

22

23

24

25

26

The European TV uses PAL system (625-line format) which requires interrupt at the end of each active-display scan every 1/50 second, as opposed to every 1/60 second for the US model (NTSC, 525-line format).

ColecoVision cartridges must be interchangeable between both systems, the Europe/America byte at AMERICA in OS ROM, has been established to detect which version of the unit is in use. If a real-time display (such as a clock) must be implemented, the program will have to access the Europe/America byte to determine the current line frequency. For America-based units, this location will contain 60 (3CH) and for European-based units, it will contain 50 (32H).

10.1.2 Restart Vectors

Figure 10-2 shows the eight programmable restarts their addresses and corresonding locations in Cartridge ROM.

OS ADDRESS	JUMP TO CART. ROM ADDR.
0008Н	800CH
0010H	800FH
0018Н	8012H
0020Н	8015H
0028H	8018H
0030Н	801BH

Figure 10-2
OS RESTARTS

For each of the restart locations above, there should be a vector in Cartridge ROM provided by the user. To use a restart, the user must place a jump instruction to the address of the routine which he or she wishes to access through the Cartridge ROM vector; for example,

JP WRITE_ VRAM at 800CH. These routines are usually the ones most frequently used in order to save application program space.

10.1.3 Graphics Tables

There are two graphics tables in the OS available to the user. The pointers for the ASCII table and Number table are defined in the locations of ASCII-TABLE and NUMBER_TABLE.

The ASCII table contains pattern generators for all 26 upper and psuedo-lower (half-size upper) case letters plus eleven special characters in 5x7 dot matrix form. The number table contains pattern generators for the numbers from 0 to 9 plus seven special characters.

10.2 Cartridge ROM

At the beginning of Cartridge ROM, locations are reserved for testing cartridge presence (Section 8-3), plus a number of pointers which point to tables, buffers and start of the game. On top of the pointers there are

1	spaces allocated for restart (Ref. Figure 10-2) and
2	interrupt vectors. There are up to 60 bytes available
3	to the user starting at location GAME_NAME, to name the
4	cartridge, their format has been described in the title
5	screen in section 8.2. Figure 10-3 shows the cartridge
6	ROM map.
7	
8	8000Н
9	Cartridge Identifier
10	8002H Pointers
11	800СН
12	Restart Vectors
13	801EH Interrupt Vectors
14	8024H
15	Cartridge Title Name (variable size up to 60 bytes)
16	8060Н
17	Applications Program
18	(variable size)
19	FFFFH

Figure 10-3

20

21

22

94

.5

6

CARTRIDGE ROM MAP

1	10.3 CRAM A	reas
2	7000Н	User RAM
4 5	7020H 702AH	OS Sound Data
6 7	73В9Н	User RAM Stack
9	73BAH 73FFH	OS RAM

Figure 10-4 CRAM MAP

Figure 10-4 is the CRAM Map. Eleven bytes are reserved for OS sound data starting at 7020H; seventy-one bytes at the high end of memory are used by various OS routines. The top of the stack is sitting at address 73B9H which grows in the decrementing direction. Between stack and user buffer there are 942 bytes available for the application program. However, care should be exercised in both size and boundary when using CRAM as scratch pad.

Table 10-1 lists all reserved CRAM areas for user reference.

25

1			
2		able 10-	
3	DETAILED CRA	M REFEREI	NCE LOCATIONS
			(Start of user RAM)
4	PTR_TO_LST_OF_SND_ADDRS	7020H	(OS Sound Data Area)
5	+1	7021H	
6	PTR_TO_S_ON_O	7022H	
7	+1	7023H	
8	PTR_TO_S_ON_1	7024H	
9	+1	7025H	
10	PTR_TO_S_ON_2	7026H	
11	+1	7027H	
12	PTR_TO_S_ON_3	7028H	
13	+1	7029H	
14	SAVE_CTRL	702AH	
15		702BH	(Resume user RAM)
16	STACK	73В9Н	(Top of Stack)
17	PARAM_AREA	73BAH	(Parameter passing area for
18	+1	73BBH	Pascal calls to OS routines)
19	+2	73BCH	
20	+3	73BDH	
21			
22			
23			

	¥		
1	+4	73BEH	
2	+5	73BFH	
3	+6	73COH	
4	+7	73C1H	
5	+8	73C2H	
6	VDP_MODE_WORD	73C3H	
7	+1	73C4H	
8	VDP_STATUS BYTE	73C5H	
9	DEFER_WRITES	73C6H	
10	MUX_SPRITES	73C7H	
11	RAND_NUM	73C8H	
12	+1	73C9H	
13	QUEUE_SIZE	73CAH	
14	QUEUE_HEAD	73CBH	
15	QUEUE_TAFL	73CCH	
16	HEAD_ADDRESS	73CDH	
17	+1	73CEH	
18	TAIL_ADDRESS	73CFH	
19	+1	73DOH	
20	BUFFER	73D1H	
21	+1	73D2H	
22	TIMER_TABLE_BAS	73D3H	
23			
24			

i 1 ,	+1	73D4H	
2	NEXT_TIMER_DATA	73D5H	
5	+1	73D6H	
4	DBNCE_BUFF	73D7H (FIRE_OLD - Player 0)	
5	+1	73D8H (FIRE_STATE - Player 0)	
6	+2	73D9H(JOY_OLD - Player 0)	
7	+3	73DAH (JOY_STATE - Player 0)	
8	+4	73DBH(SPIN_OLD - Player 0)	
9	+5	73DCH(SPIN_STATE - Player 0)	
10	+6	73DDH(ARM_OLD - Player 0)	
11	+7	73DEH(ARM_STATE - Player 0)	
12	+8	73DFH(KBD_OLD - Player 0)	
13	+9	73E0H(KBD_STATE - Player 0)	
14	+10	73E1H(FIRE_OLD - Player 1)	
15	+11	73E2H(FIRE_STATE - Player 1)	
16	+12	73E3H(JOY_OLD - Player 1)	
17	+13	73E4H(JOY_STATE - Player 1)	
18	+14	73E5H(SPIN_OLD - Player 1)	
19	+15	73E6H(SPIN_STATE - Player 1)	
20	+16	73E7H(ARM_OLD - Player 1)	
21	+17	73E8H(ARM_STATE- Player 1)	
22	+18	73E9H(KBD_OLD - Player 1)	
23			
24			
25			

1 +19				
\$\frac{1}{4}\$ \text{STROBE_FLG}\$ 73ECH \$\frac{1}{5}\$ \text{SO_CG}\$ 73EEH \$\frac{1}{5}\$ \text{SO_CG}\$ 73EH \$\frac{1}{5}\$ \text{SO_CG}\$ 73EH \$\frac{1}{7}\$ \text{S1_CO}\$ 73FOH \$\frac{1}{8}\$ \text{S1_CI}\$ 73F1H \$\frac{1}{7}\$ \text{VRAM_ADDR_TABLE}\$ 73F2H \$\frac{1}{10}\$ \text{SPRITENAMETBL}\$ 73F2H \$\frac{1}{11}\$ +1 73F3H \$\frac{1}{12}\$ \text{SPRITEGENTBL}\$ 73F4H \$\frac{1}{13}\$ +1 73F5H \$\frac{1}{14}\$ PATTRNNAMETBL 73F6H \$\frac{1}{15}\$ +1 73F9H \$\frac{1}{16}\$ COLORTABLE 73FAH \$\frac{1}{1}\$ 73F9H \$\frac{1}{1}\$ 73F9H \$\frac{1}{1}\$ SAVE_TEMP 73FCH \$\frac{1}{1}\$ 73F9H \$\frac{1}{1}\$ 73F9H \$\frac{1}{1}\$ 73FFH \$\frac{1}{1}\$ 73FFH	1	+19	73EAH	(KBD_STATE - Player 1)
### STROBE_FLG	2	SPIN_SWO_CT	73EBH	
5 SO_CO 73EEH 6 SO_C1 73EFH 7 S1_CO 73FOH 8 S1_C1 73F1H 9 VRAM_ADDR_TABLE 73F2H 10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73F9H 20 SAVE_TEMP 73FCH 21 +1 73FPH 22 SAVED_COUNT 73FEH 23 +1 73FFH	3	SPIN_SW1_CT	73ECH	
6 SO_C1 73EFH 7 S1_CO 73FOH 8 S1_C1 73F1H 9 VRAM_ADDR_TABLE 73F2H 10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73F9H 20 SAVE_TEMP 73FCH 21 +1 73FPH 22 SAVED_COUNT 73FEH 23 +1 73FFH	4	STROBE_FLG	73EDH	
7 S1_CO 73F0H 8 S1_C1 73F1H 9 VRAM_ADDR_TABLE 73F2H 10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	5	so_co	73EEH	
8 S1_C1 73F1H 9 VRAM_ADDR_TABLE 73F2H 10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FPH 22 SAVED_COUNT 73FEH 23 +1 73FFH	6	so_c1	73EFH	
9 VRAM_ADDR_TABLE 73F2H 10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73P4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	7	S1_CO	73F0H	
10 SPRITENAMETBL 73F2H 11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	8	S1_C1	73F1H	
11 +1 73F3H 12 SPRITEGENTBL 73F4H 13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	9	VRAM_ADDR_TABLE	73F2H	
SPRITEGENTBL 73F4H	10	SPRITENAMETBL	73F2H	
13 +1 73F5H 14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	11	+1	73F3H	
14 PATTRNNAMETBL 73F6H 15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	12	SPRITEGENTBL	73F4H	
15 +1 73F7H 16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	13	+1	73F5H	
16 PATTRNGENTBL 73F8H 17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	14	PATTRNNAMETBL	73F6H	
17 +1 73F9H 18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	15	+1	73F7H	
18 COLORTABLE 73FAH 19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH	16	PATTRNGENTBL	73F8H	
19 +1 73FBH 20 SAVE_TEMP 73FCH 21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH 24	17	+1	73F9H	
20 SAVE_TEMP 73PCH 21 +1 73PDH 22 SAVED_COUNT 73PEH 23 +1 73PFH 24	18	COLORTABLE	73FAH	
21 +1 73FDH 22 SAVED_COUNT 73FEH 23 +1 73FFH 24	19	+1	73FBH	
22 SAVED_COUNT 73PEH 23 +1 73FFH 24	20	SAVE_TEMP	73PCH	
23 +1 73FFH 24	21	+1	73FDH	
24	22	SAVED_COUNT	73FEH	
	23	+1	73FFH	
25	24			
	25			