


Space Invaders - Hardware map

 walkofmind.com/programming/side/hardware.htm

Mar 2004 update: I have added more information and, as a result, SIDE does not support all the features listed in this page. They are supported by Tickle though, which also attempts to directly emulate several sound circuits.

Space Invaders is based on a I8080 CPU and comes equipped with a monochromatic video card and custom analog sound generators.

The game ROM takes 8 Kb from 0000h to 1FFFh. There are 8 Kb of RAM too. Of these, 1 Kb is used by the program as a "scratch" area, and the rest is dedicated to video memory.

Start and End Address		Size	Description
0000h	1FFFh	8K	ROM
2000h	23FFh	1K	RAM
2400h	3FFFh	7K	Video RAM

Table 1 - Video memory layout

The video is a 256x224 monochromatic adapter. The memory layout is quite standard for this kind of hardware, there is one bit per pixel so each byte contains eight consecutive pixels, and the byte sequence is organized by lines, starting at the upper left corner of the screen and going from left to right within each line.

	Bytes			
Line 1	0	1	...	31
Line 2	32	33	...	63
...
Line 224	7136	7137	...	7167

Table 2 - Video memory layout

The physical video screen, however, is rotated ninety degrees (counterclockwise) in the cabinet, and to account for that the game draws all the graphics rotated by ninety degrees too. Since emulators cannot ask users to tilt their monitors, a few lines of code are needed to adjust the image for a "standard" screen layout.

	Bytes			
Line 1	31	63	...	7167
Line 2	30	62	...	7166
...
Line 255	1	33	...	7137
Line 256	0	32	...	7136

Table 3 - Video memory layout (rotated)

The above table shows the memory layout after it has been rotated, that is how the screen appears to players in the original game.

It should be noted that although the video is monochromatic, sometimes bands of transparent colored plastic were attached to the cabinet display in order to improve the graphics. Typically, the bunkers and the player ship appeared green, while the enemy spaceship took a red color.

Port 1 maps all the keys for a standard Space Invaders cabinet. Each bit reflects the current status of the corresponding key: the bit is set when the key is pressed, reset when the key is not pressed. Bit 0 is mapped to the coin slot: the bit is set when a coin is inserted, and it is automatically reset after the port is read.

Bit	Description
0	Coin slot (1=coin inserted)
1	Two players button
2	One player button
3	n/a
4	Player one - Fire button
5	Player one - Left button
6	Player one - Right button
7	n/a

Table 4 - Port 1 (input)

Port 2 maps the controls for player two and a couple of DIP switches that can be used to configure the game. Usually player two uses the same keys of player one, but the game does allow for different controls. Bits 0 and 1 control the number of ships (or lives) that are given at

the beginning of the game, from 3 to 6. For example if bit0=0 and bit1=1 then the value is binary 10, or 2 decimal, corresponding to 2+3=5 ships. Bit 7 controls whether or not the "1 player 1 coin/2 players 2 coins" notice is displayed in the screen after the demo game.

Bit	Description
0, 1	DIP switch: number of ships
2	DIP switch: easy/hard mode (0=hard, 1=easy)
3	n/a
4	Player two - Fire button
5	Player two - Left button
6	Player two - Right button
7	DIP switch: show/hide coin info

Table 5 - Port 2 (input)

Port 2 is also writable, but writing to it does not change the above values. Rather, the value is simply stored and is used when computing the value returned by port 3 (yes, this is a bit strange).

Port 3 input is connected to a special circuit and returns a value computed according to the following formula:

$$(((\text{port4_hi}) \ll 8) \mid \text{port4_lo}) \ll \text{port2} \gg 8$$

In practice, port 4 provides a 16 bit value that is shifted left by the number of bits specified by port 2 and port 3 returns the high order byte of the result.

Output of port 3 is connected to the sound hardware. Setting a bit starts the corresponding sound. The bit goes to zero immediately thereafter, there is no need to reset it. The only exception is bit 0, which corresponds to the sound made by the alien spaceship that appears every now and then on top of the screen. This sound is looped (that is it restarts automatically from the beginning as soon as it ends) and must be manually stopped by resetting its bit to zero.

Bit	Description
0	Spaceship (looped) sound
1	Shot sound
2	Base (your ship) hit sound
3	Invader hit sound

4	Extended play sound
5	n/a
6	n/a
7	n/a

Table 6 - Port 3 (output)

Port 4 is used to specify a 16 bit value used to compute the value returned by reading port 3. Since you can only write an 8 bits value to a port, it takes two steps to specify a 16 bit value: first, the low order byte is written to the port, then the high order byte.

Port 5 is also used to play sounds and it works like port 3.

Bit	Description
0	Invaders walk 1 sound
1	Invaders walk 2 sound
2	Invaders walk 3 sound
3	Invaders walk 4 sound
4	Spaceship hit sound
5	Amplifier enabled/disabled
6	n/a
7	n/a

Table 7 - Port 5 (output)

In Space Invaders, all sounds are generated by analog circuits rather than by microchips, and it would be extremely difficult to emulate that circuitry. Because of this, sound emulation is usually accomplished by playing digital samples of the original sounds.

Copyright (c) 2001-2003,2004 Alessandro Scotti. All rights reserved.