Volt32 GPU

Andrew Clark

March 27, 2021

Table of Contents

General

- The GPU is a 2D one, with sprites and backgrounds.
- Tiles are 8x8.
- The screen resolution of the console is 320x240, or 40 tiles by 30 tiles.
- There are 76,800 bytes, or 320 * 240 bytes, of tile VRAM. This is enough for 1,200 tiles, which is a whole BG's worth of tiles.
- The GPU can also be set to treat tile VRAM as a single framebuffer.

Sprites

- There are 128 sprites.
- Sprites can be of two different sizes: 8x8 and 16x16.
- There can be up to 64 sprites per scanline.
- Sprites have a 3-bit drawing priority field. If two sprites have the same drawing priority and collide with one another, the lower numbered sprite's pixel will be drawn instead of the one with the higher number's pixel. If a sprite has the same or more priority as a background, it will be drawn on top of the background, not counting when a sprite will be drawn on top of that sprite.
- The format of a sprite attribute table (SAT) entry is as follows:

Name	**Bit Range**	**Description**
Reserved[23]	[63:41]	*Reserved for future expansion*
Horizontal Position[9]	[40:32]	Horizontal position on screen (`0x0` means left edge of screen)
Vertical Position[9]	[31:23]	Vertical position on screen (`0x0` means top edge of screen)
Size[0:0]	[22]	`0b0` for 8x8, `0b1` for 16x16
Priority[3]	[21:19]	Drawing priority
Horizontal Flip[1]	[18]	Whether or not to flip the sprite horizontally when it's drawn
Vertical Flip[1]	[17]	Whether or not to flip the sprite vertically when it's drawn
Tile Index[17]	[16:0]	Which tile to draw (8x8), or which top-left tile to draw (16x16)

- 1024 bytes are required to store the SAT.

Backgrounds

- There are 4 backgrounds.
- Backgrounds have a 3-bit drawing priority field. In the case of a tie in drawing priority between two backgrounds, the lower numbered background's pixel will be drawn on top of the higher numbered background's pixel.
- Backgrounds can be scrolled vertically and horizontally.
- Tilemaps are 512x256, or 64 tiles by 32 tiles. There is a grand total of 8 kiB of tilemap space, or 32 bits per every background tile.
- The format of a tilemap entry is as follows:

Name	**Bit Range**	**Description**
Reserved[10]	[31:22]	*Reserved for future expansion*
Priority[3]	[21:19]	Drawing priority
Horizontal Flip[1]	[18]	Whether or not to flip the tile horizontally when it's drawn
Vertical Flip[1]	[17]	Whether or not to flip the tile vertically when it's drawn
Tile Index[17]	[16:0]	Which tile to draw

Palette

- There is a single 256 color palette, though color 0 is the transparent color.
- Colors are 18-bit, with 6-bit channels.
- A structure of arrays is used to store the entire palette. This is due to the CPU's registers being vectors. Each array stores one RGB channel as an 8-bit field. Bits 7 and 6 of each channel table (CHT) entry are ignored by the GPU.
- 768 bytes are needed to store the palette.
- The GPU's framebuffer mode still uses the 256-entry palette, such that it is a paletted framebuffer. For this mode, the GPU treats color 0 as a visible color instead of transparent.

Misc. Registers

1. BG0 Horizontal Scroll[32]
2. BG0 Vertical Scroll[32]
3. BG1 Horizontal Scroll[32]
4. BG1 Vertical Scroll[32]
5. BG2 Horizontal Scroll[32]
6. BG2 Vertical Scroll[32]
7. BG3 Horizontal Scroll[32]
8. BG3 Vertical Scroll[32]
9. GPU Status[32]

Name	**Bit Range**	**Description**
Reserved[31]	[31:30]	*Reserved for future expansion*
Framebuffer Mode Enable[0]	[0:0]	`0b1` to enable the GPU's framebuffer mode (which disables sprites and backgrounds and uses tile VRAM as a framebuffer)

IO Space Map

- IO Space is referred to as such because it's IO from the perspective of the CPU. The CPU itself has two address spaces it can access: IO and main memory. The CPU is the only ”bus master” that can access main memory.
- The ”**Offset**” in the ”**Offset**” and ”**Offset + Size**” columns means ”offset from base address”, where ”base address” is the IO space base address of the GPU.
- Here is the table proper.

Offset	**Offset + Size - 1**	**Name**
`0x0`	`0x1_2bff`	Tiles
`0x1_2c00`	`0x1_4bff`	Tilemaps
`0x1_4c00`	`0x1_4fff`	Sprite Attribute Table
`0x1_5000`	`0x1_52ff`	Palette
`0x1_5300`	`0x1_53ff`	Misc. Registers