SVBoy
Game Boy Specs

**CPU:** Custom 8-bit Sharp LR35902 at 4.19 MHz. This processor is similar to an Intel 8080 in that none of the registers introduced in the Z80 are present. However, some of the Z80's instruction set enhancements over the 8080, particularly bit manipulation, are present. Features removed from the Intel 8080 instruction set include the parity flag, half of the conditional jumps, and I/O instructions. I/O is instead performed through memory load/store instructions. Still, several features are added relative to both the 8080 and the Z80, most notably new load/store instructions to optimize access to memory-mapped registers. The IC also contains integrated sound generation.

**RAM:** 8 KiB internal S-RAM

**Video RAM:** 8 KiB internal

**ROM:** On-CPU-Die 256-byte bootstrap; 32 KiB cartridges (Without MBC, 64 MiB Max with MBC5)

**Sound:** 2 pulse wave generators, 1 PCM 4-bit wave sample (64 4-bit samples played in 1×64 bank or 2×32 bank) channel, 1 noise generator, and one audio input from the cartridge. The unit only has one speaker, but the headphone port outputs stereo sound.

**Display:** Reflective STN LCD 160 × 144 pixels

**Frame rate:** Approximately 59.7 frames per second

**Vertical blank duration:** Approx 1.1 ms

**Screen size:** 66 mm (2.6 in) diagonal

**Color palette:** 2-bit

**Communication:** 2 Game Boys can be linked together via built-in serial ports, up to 4 with a DMG-07 4-player adapter. And 16 in maximum.

**Power:** 6 V, 0.7 W (4 AA batteries provide approximately 15 hours of gameplay)[28]

**Dimensions:** 90 mm (W) × 148 mm (H) × 32 mm (D) / 3.5” × 5.8” × 1.3”[28]

**Weight:** 220 g[28]

[NA Release Date: July 31, 1989](https://en.wikipedia.org/wiki/Game_Boy)
Game Boy Memory Map

http://gameboy.mongenel.com/dmg/asmmemmap.html
GB-Z80 Specs

- 8-bit DATA, 16-bit ADDR, Support 16-bit data operations
- CISC, Similar to the Z-80 Processor
- 4.194304 MHz \(2^{22}\) Hz clock frequency (1 T-Cycle = \(1/2^{22}\) second)
- One Instruction takes 1-5 M-Cycle to execute (1 M-Cycle = 4 T-Cycle)
- 512 Possible Instructions
- 5 Interrupt Service Routines
- 127 x 8 bits built-in RAM (Stack)
RISC Approach

CALL nn
Unconditional function call to the absolute address specified by the operand nn.

Opcode + data: 0b11001101 LSBOf nn + MSBOf nn
Length: 3 bytes
Duration: 6 machine cycles
Flags: 
Timing:
Purpose: Decode | MSBOf nn | LSB of nn | Psr | Flags
Memory: Read PC | Read PC+1 | Read PC+2 | Write SP | Write SP | Read nn

Pseudocode:
```
#define DECODER_CALL_a16

begin
  RISC_OPCODE[2] = LD_XPC;
  RISC_OPCODE[3] = LD_TPC;
  RISC_OPCODE[5] = DEC_SP;
  RISC_OPCODE[6] = LD_SPPCH;
  RISC_OPCODE[7] = DEC_SP;
  RISC_OPCODE[8] = LD_SPPCL;
  RISC_OPCODE[9] = JP_TX;
  NUM_Tcnt = 6'd24;
end
```

Game Boy: Complete Technical Reference, gekkio https://gekkio.fi
Interrupt Handling

Interrupt = IME && (FF0F & FFFF) != 0
Single Port RAMs

• Work RAM / Video RAM : 8192 Bytes

• OAM : 160 Bytes

• Quartus Single Port RAM Template

• Data available on the second half of the same clock cycle
Video Specs

• Screen: 160x144 px
• Background: 256x256 px or 32x32 tiles (8x8 px each), scrollable
• Window: 160x144 px Max, non-scrollable
• Sprite: 8x8 px or 8x16 px
  Up to 40 in OAM
  Up to 10 per line

# Tile Rendering

<table>
<thead>
<tr>
<th>Region</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>8000-87FF</td>
<td>Tile set #1: tiles 0-127</td>
</tr>
<tr>
<td>8800-8FFF</td>
<td>Tile set #1: tiles 128-255</td>
</tr>
<tr>
<td></td>
<td>Tile set #0: tiles -1 to -128</td>
</tr>
<tr>
<td>9000-97FF</td>
<td>Tile set #0: tiles 0-127</td>
</tr>
<tr>
<td>9800-9BFF</td>
<td>Tile map #0 (1024 entries)</td>
</tr>
<tr>
<td>9C00-9FFFF</td>
<td>Tile map #1 (1024 entries)</td>
</tr>
</tbody>
</table>

**Video RAM layout**

**Background Mapping**

Color Rendering

<table>
<thead>
<tr>
<th>Value</th>
<th>Pixel</th>
<th>Mapped color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Off</td>
<td>[226, 243, 228]</td>
</tr>
<tr>
<td>1</td>
<td>33% on</td>
<td>[148, 227, 68]</td>
</tr>
<tr>
<td>2</td>
<td>66% on</td>
<td>[70, 135, 143]</td>
</tr>
<tr>
<td>3</td>
<td>On</td>
<td>[51, 44, 80]</td>
</tr>
</tbody>
</table>

palette register

Tile data bitmap structure

Video Timing

- **OAM Search**
  - 80 dots

- **Rendering**
  - 174 – 291 dots

- **H-Blank**
  - 85 to 202 dots

- **V-Blank**
  - 10 lines (4560 dots)

Total: 456 dots
Frame Buffer

• 160 x 144 x 2 bits SRAM
• 2-Port, 2-Clock
• Write Clock: GameBoy Clock @ 4.19MHz
• Read Clock: VGA Clock @ 108MHz
• No Vertical Sync
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**Name**  
LCDC (value $91 at reset)

**Contents**  
LCD Control (R/W)

- **Bit 6** - Window Tile Map Display Select  
  0: $9800-$9BFF  
  1: $9C00-$9FFF

- **Bit 4** - BG & Window Tile Data Select  
  0: $8800-$8FFF  
  1: $8000-$8FFF < Same area as OBJ

- **Bit 3** - BG Tile Map Display Select  
  0: $9800-$9BFF  
  1: $9C00-$9FFF

**FF40**

- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

**Frame Buffer**

- **PX SHIFT REG A**
- **PX SHIFT REG B**
- **LX = 0**
Background/Window Rendering

Background, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$87FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Flash Shift Register A
Flash Shift Register B

LX = 1
Frame Buffer
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

<table>
<thead>
<tr>
<th>PX SHIFT REG A</th>
<th>LX = 2</th>
</tr>
</thead>
</table>

| PX SHIFT REG B |

FG40
- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$87FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Frame Buffer
Background/Window Rendering

 BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

PX SHIFT REG A

PX SHIFT REG B

LX = 3

Frame Buffer

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Frame Buffer

PX SHIFT REG A
LX = 4

PX SHIFT REG B
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

PPX SHIFT REG A
PPX SHIFT REG B

FF40
Name: LCDC (value $91 at reset)
Contents: LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$8F7F
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Frame Buffer
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**PX SHIFT REG A**

**PX SHIFT REG B**

**FF40**
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

**Frame Buffer**
### Background/Window Rendering

**FF40**
- Name: LCDC (value $91$ at reset)
- Contents: LCD Control (R/W)

<table>
<thead>
<tr>
<th>Bit 6</th>
<th>Window Tile Map Display Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$9800$-$9BFF</td>
</tr>
<tr>
<td>1</td>
<td>$9C00$-$9FFF</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>BG &amp; Window Tile Data Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$8800$-$97FF</td>
</tr>
<tr>
<td>1</td>
<td>$8000$-$8FFF &lt;- Same area as OBJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>BG Tile Map Display Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$9800$-$9BFF</td>
</tr>
<tr>
<td>1</td>
<td>$9C00$-$9FFF</td>
</tr>
</tbody>
</table>

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**Frame Buffer**

**PX SHIFT REG A**

**PX SHIFT REG B**
Background/Window Rendering

BG, SCX = 3, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

Frame Buffer

PX SHIFT REG A

LX = 0

PX SHIFT REG B

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF < Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 3, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**FF40**
- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

**Bit 6 - Window Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

**Bit 4 - BG & Window Tile Data Select**
- 0: $8800-$8FF
- 1: $8000-$8FFF <- Same area as OBJ

**Bit 3 - BG Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

Frame Buffer
### Background/Window Rendering

**BG, SCX = 3, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0**

### FF40

**Name**  
LCDC (value $91$ at reset)

**Contents**  
LCD Control (R/W)

<table>
<thead>
<tr>
<th>Bit 6 - Window Tile Map Display Select</th>
<th>0: $9800$-$9BFF</th>
<th>1: $9C00$-$9FFF</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bit 4 - BG &amp; Window Tile Data Select</th>
<th>0: $8800$-$97FF$</th>
<th>1: $8000$-$8FFF$ &lt;- Same area as OBJ</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bit 3 - BG Tile Map Display Select</th>
<th>0: $9800$-$9BFF$</th>
<th>1: $9C00$-$9FFF$</th>
</tr>
</thead>
</table>

### PX SHIFT REG A

### PX SHIFT REG B

### LX = 0

### Frame Buffer
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**FF40**
- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

**Bit 6** - Window Tile Map Display Select
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

**Bit 4** - BG & Window Tile Data Select
- 0: $8800-$97FF
- 1: $8000-$8FFF <- Same area as OBJ

**Bit 3** - BG Tile Map Display Select
- 0: $9800-$9BFF
- 1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**FF40**
- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

**Bit 6 - Window Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

**Bit 4 - BG & Window Tile Data Select**
- 0: $8800-$97FF
- 1: $8000-$8FFF <- Same area as OBJ

**Bit 3 - BG Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF
### Background/Window Rendering

**FF40**
- **Name**: LCDC (value $91$ at reset)
- **Contents**: LCD Control (R/W)

<table>
<thead>
<tr>
<th>Bit 6</th>
<th>Window Tile Map Display Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$9800$-$9BFF$</td>
</tr>
<tr>
<td>1</td>
<td>$9C00$-$9FFF$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 4</th>
<th>BG &amp; Window Tile Data Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$8800$-$97FF$</td>
</tr>
<tr>
<td>1</td>
<td>$8000$-$8FFF$ &lt;- Same area as OBJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit 3</th>
<th>BG Tile Map Display Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$9800$-$9BFF$</td>
</tr>
<tr>
<td>1</td>
<td>$9C00$-$9FFF$</td>
</tr>
</tbody>
</table>

**BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0**

- **PX SHIFT REG A**: LX = 3
- **PX SHIFT REG B**: Frame Buffer
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**Frame Buffer**

**LX = 4**

**PX SHIFT REG A**

**PX SHIFT REG B**

FF40

- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Frame Buffer

PX SHIFT REG A
PX SHIFT REG B
LX = 5
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

Frame Buffer

PX SHIFT REG A

LX = 6

PX SHIFT REG B

Reflects the following register:

FF40

Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$87FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

FF40
Name - LCDC (value $91 at reset)
Contents - LCD Control (R/W)

Bit 6 - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

Bit 4 - BG & Window Tile Data Select
0: $8800-$97FF
1: $8000-$8FFF <- Same area as OBJ

Bit 3 - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

PX SHIFT REG A
LX = 7
PX SHIFT REG B

Frame Buffer
Background/Window Rendering

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**FF40**
- Name: LCDC (value $91 at reset)
- Contents: LCD Control (R/W)

**Bit 6 - Window Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

**Bit 4 - BG & Window Tile Data Select**
- 0: $8800-$97FF
- 1: $8000-$8FFF <- Same area as OBJ

**Bit 3 - BG Tile Map Display Select**
- 0: $9800-$9BFF
- 1: $9C00-$9FFF

Frame Buffer

PX SHIFT REG A

PX SHIFT REG B

LX = 8
BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

**FF40**

- **Name**: LCDC (value $91 at reset)
- **Contents**: LCD Control (R/W)

**Bit 6** - Window Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

**Bit 4** - BG & Window Tile Data Select
0: $8800-$87FF
1: $8000-$8FFF <- Same area as OBJ

**Bit 3** - BG Tile Map Display Select
0: $9800-$9BFF
1: $9C00-$9FFF

**LX = 9**

**Frame Buffer**
OAM Search

1. Iterate through all 40 entries in OAM
2. Read Byte0, to see if it is on the current line
3. If it is, store Byte1 and its position in OAM in a local OAM

<table>
<thead>
<tr>
<th>X pos</th>
<th>Pattern #</th>
<th>OAM pos</th>
<th>Flag</th>
<th>Used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>TBD</td>
<td>0</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>TBD</td>
<td>1</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>30</td>
<td>TBD</td>
<td>2</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>36</td>
<td>TBD</td>
<td>4</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>78</td>
<td>TBD</td>
<td>17</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
</tbody>
</table>

Byte0  Y position on the screen
Byte1  X position on the screen
Byte2  Pattern number 0-255 (Unlike some tile numbers, sprite pattern numbers are unsigned. LSB is ignored (treated as 0) in 8x16 mode.)
Byte3  Flags:

Bit7  Priority
If this bit is set to 0, sprite is displayed on top of background & window. If this bit is set to 1, then sprite will be hidden behind colors 1, 2, and 3 of the background & window. (Sprite only prevails over color 0 of BG & win.)

Bit6  Y flip
Sprite pattern is flipped vertically if this bit is set to 1.

Bit5  X flip
Sprite pattern is flipped horizontally if this bit is set to 1.

Bit4  Palette number
Sprite colors are taken from OBJ1PAL if this bit is set to 1 and from OBJ0PAL otherwise.
## Sprite Rendering

### Table: X pos, Pattern #, OAM pos, Flag, Used?

<table>
<thead>
<tr>
<th>X pos</th>
<th>Pattern #</th>
<th>OAM pos</th>
<th>Flag</th>
<th>Used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>TBD</td>
<td>0</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>TBD</td>
<td>1</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>30</td>
<td>TBD</td>
<td>2</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>36</td>
<td>TBD</td>
<td>4</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>78</td>
<td>TBD</td>
<td>17</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
</tbody>
</table>

### Diagram:
- **BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0**
- **LX = 10**
- **PX SHIFT REG A**
- **PX SHIFT REG B**
- **SP SHIFT REG 0**
- **Frame Buffer**
Sprite Rendering

<table>
<thead>
<tr>
<th>X pos</th>
<th>Pattern #</th>
<th>OAM pos</th>
<th>Flag</th>
<th>Used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>TBD</td>
<td>0</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>30</td>
<td>TBD</td>
<td>2</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>36</td>
<td>TBD</td>
<td>4</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>78</td>
<td>TBD</td>
<td>17</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
</tbody>
</table>

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

PX SHIFT REG A

LX = 10

PX SHIFT REG B

SP SHIFT REG 0

Frame Buffer

Local OAM
Sprite Rendering

<table>
<thead>
<tr>
<th>X pos</th>
<th>Pattern #</th>
<th>OAM pos</th>
<th>Flag</th>
<th>Used?</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>TBD</td>
<td>0</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>1</td>
<td>8’h8F</td>
<td>Yes</td>
</tr>
<tr>
<td>30</td>
<td>TBD</td>
<td>2</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>36</td>
<td>TBD</td>
<td>4</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>78</td>
<td>TBD</td>
<td>17</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>64</td>
<td>TBD</td>
<td>No</td>
</tr>
</tbody>
</table>

BG, SCX = 0, SCY = 0, FF40[4] = 1, FF40[3] = 0, LY = 0

Frame Buffer

Local OAM
Cartridge

- Max 64MByte ROM + 1MByte RAM
- On Board SDRAM @ 67.108864 MHz (16x GameBoy Clock)
- Intel SDRAM Controller IP is used
- Emulated SRAM Behavior
Sound

- 4 Channels
  - A square wave ("pulse") channel that perform frequency sweeps
  - A second square wave channel that can only play a constant frequency
  - A noise channel
  - An arbitrary wave channel

- 4 Bit Raw Resolution

- On Chip CODEC @ 16Bit 48KHz

https://github.com/aselker/gameboy-sound-chip
Square Wave Channel

Timer -> Duty -> Length Counter -> Envelope -> Mixer

https://github.com/aselker/gameboy-sound-chip
Square Wave Channel With Sweep

https://github.com/aselker/gameboy-sound-chip

Sweep -> Timer -> Duty -> Length Counter -> Envelope -> Mixer
Noise Channel

7-Stages LFSR implementing a \( x^7 + x + 1 \) binary polynomial counter

Timer -> LFSR -> Length Counter -> Envelope -> Mixer

https://github.com/aselker/gameboy-sound-chip
Wave Channel

32 4-bit Samples in Internal Wave RAM

https://github.com/aselker/gameboy-sound-chip

Timer -> Wave -> Length Counter -> Volume -> Mixer
Joypad - Hardware

http://gbdev.gg8.se/wiki/articles/DMG_Schematics

Nintendo. *Game Boy Programming Manual*
Joypad - Implementation

• Device driver to send joypad register status
• User space program can configure any USB keyboard keys (except ESC and modifiers) as joypad keys
• SPACE key is reserved for double speed
• Sends joypad status to kernel if any configured joypad keys are pressed
Cartridge – ROM and RAM

• ROM files are downloaded online
• ROM contents are loaded to SDRAM on the DE1-SoC via `mmap`
• The real Game Boy saves data in RAM on the cartridge, powered by its own battery (expected lifespan of 10 years)
• Any SAV file of the game is automatically loaded into SDRAM
• Game Boy stops running upon pressing ESC and game data is saved on the PC
Cartridge - Memory Bank Controllers

- MBC1 and MBC5 are the most common

0000-3FFF: ROM Bank 00 (Read Only);
4000-7FFFF - ROM Bank 01-7F (Read Only);
A000-BFFF - RAM Bank 00-03, if any (Read/Write);

0000-1FFF - RAM Enable (Write Only);
2000-3FFF - ROM Bank Number (Write Only);
4000-5FFF - RAM Bank Number or Upper Bits of ROM Bank Number (Write Only);
6000-7FFF - ROM/RAM Mode Select (Write Only)

Nintendo. Game Boy Programming Manual
Serial – I/O Registers

Nintendo. *Game Boy Programming Manual*
Serial – Timing

• Sending and receiving data (8-bits) occur simultaneously
Accuracy Tests

• Mooneye GB ([https://github.com/Gekkio/mooneye-gb](https://github.com/Gekkio/mooneye-gb)) and Blargg’s ([http://gbdev.gg8.se/files/roms/blargg-gb-tests/](http://gbdev.gg8.se/files/roms/blargg-gb-tests/)) test ROMs are developed from running them with real Game Boy devices

• Our results compared to others:
Demo

- oh.gb (ROM+MBC1)
- pocket.gb (ROM+MBC1)
- Kirby’s Dream Land (ROM+MBC1)
- Pokemon Yellow (ROM+MBC5+RAM+BATTERY)
- Tetris (ROM only)