

Foreword
Acknowledgements
History
Overview
1. Specifications
2. Memory Map
I/O Ports
3. Rendering
3.1. Tile Data
3.2. Tile Maps
3.3. OAM
3.4. LCD Control
3.5. LCD Status
3.6. Scrolling
3.7. Palettes
3.8. Pixel FIFO
4. Sound Controller
5. Joypad Input
6. Serial Data Transfer
7. Timer and Divider Registers
8. Interrupts
9. CGB Registers
10. Infrared Communication
11. SGB Functions
CPU Specifications
12. CPU Registers and Flags
13. CPU Instruction Set
14. CPU Comparison with Z80
Cartridges
15. The Cartridge Header
16. MBCs
Accessories
17. Game Boy Printer
18. Game Boy Camera
19. 4-Player Adapter
20. Game Genie/Shark Cheats
Other
21. Power-Up Sequence
22. Reducing Power Consumption
23. Accessing VRAM and OAM
24. OAM Corruption Bug
25. External Connectors
26. GBC Approval Process
References

LCD Control

FF40 - LCDC (LCD Control) (R/W)

LCDC is the main **LCD** Control register. Its bits toggle what elements are displayed on th screen, and how.

Bit	Name	Usage notes
7	LCD and PPU enable	0=Off, 1=On
6	Window tile map area	0=9800-9BFF, 1=9C00-9FFF
5	Window enable	0=Off, 1=On
4	BG and Window tile data area	0=8800-97FF, 1=8000-8FFF
3	BG tile map area	0=9800-9BFF, 1=9C00-9FFF
2	OBJ size	0=8x8, 1=8x16
1	OBJ enable	0=Off, 1=On
0	BG and Window enable/priority	0=Off, 1=On

LCDC.7 - LCD enable

This bit controls whether the LCD is on and the PPU is active. Setting it to 0 turns both of which grants immediate and full access to VRAM, OAM, etc.

CAUTION

Stopping LCD operation (Bit 7 from 1 to 0) may be performed during VBlank ONLY, disabling the display outside of the VBlank period may damage the hardware by burning in a black horizontal line similar to that which appears when the GB is turned off. This appears to be a serious issue. Nintendo is reported to reject any games not following this rule.

When the display is disabled the screen is blank, which on DMG is displayed as a white “whiter” than color #0.

On SGB, the screen doesn’t turn white, it appears that the previous picture sticks to the screen. (TODO: research this more.)

When re-enabling the LCD, the PPU will immediately start drawing again, but the screen will stay blank during the first frame.

LCDC.6 - Window tile map area

This bit controls which background map the Window uses for rendering. When it’s reset, t \$9800 tilemap is used, otherwise it’s the \$9C00 one.

LCDC.5 - Window enable

This bit controls whether the window shall be displayed or not. This bit is overridden on DMG by [bit 0](#) if that bit is reset.

Changing the value of this register mid-frame triggers a more complex behaviour: [see further below](#).

Note that on CGB models, setting this bit to 0 then back to 1 mid-frame may cause the second write to be ignored. (TODO: test this.)

LCDC.4 - BG and Window tile data area

This bit controls which [addressing mode](#) the BG and Window use to pick tiles.

Sprites aren’t affected by this, and will always use \$8000 addressing mode.

LCDC.3 - BG tile map area

This bit works similarly to [LCDC bit 6](#): if the bit is reset, the BG uses tilemap \$9800, otherwise tilemap \$9C00.

LCDC.2 - OBJ size

This bit controls the sprite size (1 tile or 2 stacked vertically).

Be cautious when changing this mid-frame from 8x8 to 8x16: “remnants” of the sprites intended for 8x8 could “leak” into the 8x16 zone and cause artifacts.

LCDC.1 - OBJ enable

This bit toggles whether sprites are displayed or not.

This can be toggled mid-frame, for example to avoid sprites being displayed on top of a status bar or text box.

(Note: toggling mid-scanline might have funky results on DMG? Investigation needed.)

LCDC.0 - BG and Window enable/priority

LCDC.0 has different meanings depending on Game Boy type and Mode:

Non-CGB Mode (DMG, SGB and CGB in compatibility mode): BG and Window Display

When Bit 0 is cleared, both background and window become blank (white), and the [Window Display Bit](#) is ignored in that case. Only Sprites may still be displayed (if enabled in Bit 1).

CGB Mode: BG and Window Master Priority

When Bit 0 is cleared, the background and window lose their priority - the sprites will be always displayed on top of background and window, independently of the priority flags in OAM and BG Map attributes.

Using LCDC

LCDC is a powerful tool: each bit controls a lot of behavior, and can be modified at any time during the frame.

One of the important aspects of LCDC is that unlike VRAM, the PPU never locks it. It's thus possible to modify it mid-scanline!

Faux-layer textbox/status bar

A problem often seen especially in NES games is sprites rendering on top of the textbox/status bar. It's possible to prevent this using LCDC if the textbox/status bar is “alone” on its scanlines:

- Set LCDC.1 to 1 for gameplay scanlines
- Set LCDC.1 to 0 for textbox/status bar scanlines

Usually, these bars are either at the top or bottom of the screen, so the bit can be set by the VBlank handler.