

Foreword
Acknowledgements
History
Overview
1. Specifications
2. Memory Map
I/O Ports
3. Rendering
4. Sound Controller
5. Joypad Input
6. Serial Data Transfer
7. Timer and Divider Registers
8. Interrupts
8.1. Interrupt Sources
8.2. HALT
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

Interrupt Sources

INT 40 - VBlank Interrupt

This interrupt is requested every time the Game Boy enters VBlank ([Mode 1](#)).

The VBlank interrupt occurs ca. 59.7 times a second on a handheld Game Boy (DMG or CGB) or Game Boy Player and ca. 61.1 times a second on a Super Game Boy (SGB). This interrupt occurs at the beginning of the VBlank period (LY=144). During this period video hardware is not using VRAM so it may be freely accessed. This period lasts approximately 1.1 milliseconds.

INT 48 - STAT Interrupt

There are various sources which can trigger this interrupt to occur as described in [STAT register \(\\$FF41\)](#).

The various STAT interrupt sources (modes 0-2 and LYC=LY) have their state (inactive=low and active=high) logically ORed into a shared “STAT interrupt line” if their respective enable bit is turned on.

A STAT interrupt will be triggered by a rising edge (transition from low to high) on the STAT interrupt line.

STAT BLOCKING

If a STAT interrupt source logically ORs the interrupt line high while (or immediately after) it’s already set high by another source, then there will be no low-to-high transition and so no interrupt will occur. This phenomenon is known as “STAT blocking” ([test ROM example](#)).

As mentioned in the description of the [STAT register](#), the PPU cycles through the different modes in a fixed order. So for example, if interrupts are enabled for two consecutive modes such as Mode 0 and Mode 1, then no interrupt will trigger for Mode 1 (since the STAT interrupt line won’t have a chance to go low between them).

Using the STAT interrupt

One very popular use is to indicate to the user when the video hardware is about to redraw a given LCD line. This can be useful for dynamically controlling the SCX/SCY registers (\$FF43/\$FF42) to [perform special video effects](#).

Example application: set LYC to WY, enable LY=LYC interrupt, and have the handler disable sprites. This can be used if you use the window for a text box (at the bottom of the screen) and you want sprites to be hidden by the text box.

INT 50 - Timer Interrupt

Every time that the timer overflows (that is, when [TIMA](#) exceeds \$FF), an interrupt is requested by setting bit 2 in the IF register (\$FF0F). As soon as that interrupt is enabled, the CPU will execute it by calling the timer interrupt vector at \$0050.

INT 58 - Serial Interrupt

XXXXXX...

Transmitting and receiving serial data is done simultaneously. The received data is automatically stored in SB.

The serial I/O port on the Game Boy is a very simple setup and is crude compared to standard RS-232 (IBM-PC) or RS-485 (Macintosh) serial ports. There are no start or stop bits.

During a transfer, a byte is shifted in at the same time that a byte is shifted out. The rate of the shift is determined by whether the clock source is internal or external. The most significant bit is shifted in and out first.

When the internal clock is selected, it drives the clock pin on the game link port and it stays high when not used. During a transfer it will go low eight times to clock in/out each bit.

The state of the last bit shifted out determines the state of the output line until another

transfer takes place.

If a serial transfer with internal clock is performed and no external Game Boy is present, a value of \$FF will be received in the transfer.

The following code initiates the process of shifting \$75 out the serial port and a byte to be shifted into \$FF01:

```
ld a, $75
ld [$FF01], a
ld a, $81
ld [$FF02], a
```

The Game Boy does not support wake-on-LAN. Completion of an externally clocked serial transfer does not exit STOP mode.

INT 60 - Joypad Interrupt

The Joypad interrupt is requested when any of [P1](#) bits 0-3 change from High to Low. This happens when a button is pressed (provided that the action/direction buttons are enabled bit 5/4, respectively), however, due to switch bounce, one or more High to Low transitions are usually produced when pressing a button.

Using the Joypad Interrupt

This interrupt is useful to identify button presses if we have only selected either action (bit 5) or direction (bit 4), but not both. If both are selected and, for example, a bit is already held Low by an action button, pressing the corresponding direction button would make no difference. The only meaningful purpose of the Joypad interrupt would be to terminate the STOP (low power) standby state. GBA SP, because of the different buttons used, seems to not be affected by switch bounce.