Gameboy sound hardware

From GbdevWiki

This documents the behavior of Game Boy sound; details which aren't relevant to the observable behavior have been omitted unless they clarify understanding. It is aimed at answering all questions about exact operation, rather than describing how to use sound effectively in Game Boy programs. Values in hexadecimal (base 16) are generally written with a $ prefix. Bits are numbered from 0 to 7, where bit N has a weight of 2^N. A nibble is 4 bits, half a byte. Obscure behavior is described separately to increase clarity elsewhere.

Contents

Overview

The Game Boy has four sound channels: two square waves with adjustable duty, a programmable wave table, and a noise generator. Each has some kind of frequency (pitch) control. The first square channel also has an automatic frequency sweep unit to help with sound effects. The squares and noise each have a volume envelope unit to help with fading notes and sound effects, while the wave channel has only limited manual volume control. Each channel has a length counter that can silence the channel after a preset time, to handle note durations. Each channel can be individually panned to the far left, center, or far right. The master volume of the left and right outputs can also be adjusted.

Different versions of the Game Boy sound hardware have slightly different behavior. The following models have been tested:

- DMG-CPU-03 original Game Boy
- DMG-CPU-05
- DMG-CPU-06
- MGB-LCPU-01 Game Boy Pocket
- CGB-CPU-02 Game Boy Color
- CGB-CPU-04
- CGB-CPU-05

Registers

Sound registers are mapped to $FF10-$FF3F in memory. Each channel has five logical registers, NRx0-NRx4, though some don't use NRx0. The value written to bits marked with '-' has no effect. Reference to the value in a register means the last value written to it.

```
Name Addr 7654 3210 Function
-----------------------------------------------------------------
       Square 1
NR10 FF10 -PPP NSSS Sweep period, negate, shift
NR11 FF11 DDLL LLLL Duty, Length load (64-L)
NR12 FF12 VVVV APPP Starting volume, Envelope add mode, period
NR13 FF13 FFFF FFFF Frequency LSB
NR14 FF14 TL-- -FFF Trigger, Length enable, Frequency MSB

       Square 2
     FF15 ---- ---- Not used
NR21 FF16 DDLL LLLL Duty, Length load (64-L)
NR22 FF17 VVVV APPP Starting volume, Envelope add mode, period
NR23 FF18 FFFF FFFF Frequency LSB
NR24 FF19 TL-- -FFF Trigger, Length enable, Frequency MSB

       Wave
NR30 FF1A E--- ---- DAC power
NR31 FF1B LLLL LLLL Length load (256-L)
NR32 FF1C -VV- ---- Volume code (00=0%, 01=100%, 10=50%, 11=25%)
NR33 FF1D FFFF FFFF Frequency LSB
NR34 FF1E TL-- -FFF Trigger, Length enable, Frequency MSB

       Noise
     FF1F ---- ---- Not used
NR41 FF20 --LL LLLL Length load (64-L)
NR42 FF21 VVVV APPP Starting volume, Envelope add mode, period
NR43 FF22 SSSS WDDD Clock shift, Width mode of LFSR, Divisor code
NR44 FF23 TL-- ---- Trigger, Length enable

       Control/Status
NR50 FF24 ALLL BRRR Vin L enable, Left vol, Vin R enable, Right vol
NR51 FF25 NW21 NW21 Left enables, Right enables
```

```
NR52 FF26 P--- NW21 Power control/status, Channel length statuses

       Not used
     FF27 ---- ----
     .... ---- ----
     FF2F ---- ----

       Wave Table
     FF30 0000 1111 Samples 0 and 1
     ....
     FF3F 0000 1111 Samples 30 and 31
```

Channels

Each channel has a frequency timer which clocks a waveform generator. The waveform's volume is adjusted and fed to the mixer. The mixer converts each channel's waveform into an electrical signal and outputs this to the left and/or right channels. Finally, a master volume control adjusts the left and right outputs. The channels have the following units that are connected from left to right:

```
Square 1: Sweep -> Timer -> Duty -> Length Counter -> Envelope -> Mixer

Square 2:          Timer -> Duty -> Length Counter -> Envelope -> Mixer

Wave:              Timer -> Wave -> Length Counter -> Volume   -> Mixer

Noise:             Timer -> LFSR -> Length Counter -> Envelope -> Mixer
```

The mixer has a separate DAC for each channel, followed by on/off controls for left and right outputs. The left/right outputs from each channel are then added together and fed to the left/right master volume controls.

In general, all units in the channels are always running. For example, even if a channel is silent, several units will still be calculating values even though they aren't used.

Timer

A timer generates an output clock every N input clocks, where N is the timer's period. If a timer's rate is given as a frequency, its period is 4194304/frequency in Hz. Each timer has an internal counter that is decremented on each input clock. When the counter becomes zero, it is reloaded with the period and an output clock is generated.

Frame Sequencer

The frame sequencer generates low frequency clocks for the modulation units. It is clocked by a 512 Hz timer.

```
Step   Length Ctr  Vol Env     Sweep
---------------------------------------
0      Clock       -           -
1      -           -           -
2      Clock       -           Clock
3      -           -           -
4      Clock       -           -
5      -           -           -
6      Clock       -           Clock
7      -           Clock       -
---------------------------------------
Rate   256 Hz      64 Hz       128 Hz
```

Length Counter

A length counter disables a channel when it decrements to zero. It contains an internal counter and enabled flag. Writing a byte to NRx1 loads the counter with 64-data (256-data for wave channel). The counter can be reloaded at any time.

A channel is said to be disabled when the internal enabled flag is clear. When a channel is disabled, its volume unit receives 0, otherwise its volume unit receives the output of the waveform generator. Other units besides the length counter can enable/disable the channel as well.

Each length counter is clocked at 256 Hz by the frame sequencer. When clocked while enabled by NRx4 and the counter is not zero, it is decremented. If it becomes zero, the channel is disabled.

Volume Envelope

A volume envelope has a volume counter and an internal timer clocked at 64 Hz by the frame sequencer. When the timer generates a clock and the envelope period is not zero, a new volume is calculated by adding or subtracting (as set by NRx2) one from the current volume. If this new volume within the 0 to 15 range, the volume is updated, otherwise it is left unchanged and no further automatic increments/decrements are made to the volume until the channel is triggered again.

When the waveform input is zero the envelope outputs zero, otherwise it outputs the current volume.

Writing to NRx2 causes obscure effects on the volume that differ on different Game Boy models (see obscure behavior).

Square Wave

A square channel's frequency timer period is set to (2048-frequency)*4. Four duty cycles are available, each waveform taking 8 frequency timer clocks to cycle through:

```
Duty   Waveform    Ratio
-------------------------
0      00000001    12.5%
1      10000001    25%
2      10000111    50%
3      01111110    75%
```

Frequency Sweep

The first square channel has a frequency sweep unit, controlled by NR10. This has a timer, internal enabled flag, and frequency shadow register. It can periodically adjust square 1's frequency up or down.

During a trigger event, several things occur:

- Square 1's frequency is copied to the shadow register.
- The sweep timer is reloaded.
- The internal enabled flag is set if either the sweep period or shift are non-zero, cleared otherwise.
- If the sweep shift is non-zero, frequency calculation and the overflow check are performed immediately.

Frequency calculation consists of taking the value in the frequency shadow register, shifting it right by sweep shift, optionally negating the value, and summing this with the frequency shadow register to produce a new frequency. What is done with this new frequency depends on the context.

The overflow check simply calculates the new frequency and if this is greater than 2047, square 1 is disabled.

The sweep timer is clocked at 128 Hz by the frame sequencer. When it generates a clock and the sweep's internal enabled flag is set and the sweep period is not zero, a new frequency is calculated and the overflow check is performed. If the new frequency is 2047 or less and the sweep shift is not zero, this new frequency is written back to the shadow frequency and square 1's frequency in NR13 and NR14, then frequency calculation and overflow check are run AGAIN immediately using this new value, but this second new frequency is not written back.

Square 1's frequency can be modified via NR13 and NR14 while sweep is active, but the shadow frequency won't be affected so the next time the sweep updates the channel's frequency this modification will be lost.

Noise Channel

The noise channel's frequency timer period is set by a base divisor shifted left some number of bits.

```
Divisor code   Divisor
-----------------------
  0              8
  1             16
  2             32
  3             48
  4             64
  5             80
  6             96
  7            112
```

The linear feedback shift register (LFSR) generates a pseudo-random bit sequence. It has a 15-bit shift register with feedback. When clocked by the frequency timer, the low two bits (0 and 1) are XORed, all bits are shifted right by one, and the result of the XOR is put into the now-empty high bit. If width mode is 1 (NR43), the XOR result is ALSO put into bit 6 AFTER the shift, resulting in a 7-bit LFSR. The waveform output is bit 0 of the LFSR, INVERTED.

Wave Channel

The wave channel plays a 32-entry wave table made up of 4-bit samples. Each byte encodes two samples, the first in the high bits. The wave channel has a sample buffer and position counter.

The wave channel's frequency timer period is set to (2048-frequency)*2. When the timer generates a clock, the position counter is advanced one sample in the wave table, looping back to the beginning when it goes past the end, then a sample is read into the sample buffer from this NEW position.

The DAC receives the current value from the upper/lower nibble of the sample buffer, shifted right by the volume control.

```
Code   Shift   Volume
-----------------------
0      4         0% (silent)
1      0       100%
2      1        50%
3      2        25%
```

Wave RAM can only be properly accessed when the channel is disabled (see obscure behavior).

Trigger Event

Writing a value to NRx4 with bit 7 set causes the following things to occur:

- Channel is enabled (see length counter).

- If length counter is zero, it is set to 64 (256 for wave channel).
- Frequency timer is reloaded with period.
- Volume envelope timer is reloaded with period.
- Channel volume is reloaded from NRx2.
- Noise channel's LFSR bits are all set to 1.
- Wave channel's position is set to 0 but sample buffer is NOT refilled.
- Square 1's sweep does several things (see frequency sweep).

Note that if the channel's DAC is off, after the above actions occur the channel will be immediately disabled again.

Channel DAC

Each channel has a 4-bit digital-to-analog convertor (DAC). This converts the input value to a proportional output voltage. An input of 0 generates -1.0 and an input of 15 generates +1.0, using arbitrary voltage units.

DAC power is controlled by the upper 5 bits of NRx2 (top bit of NR30 for wave channel). If these bits are not all clear, the DAC is on, otherwise it's off and outputs 0 volts. Also, any time the DAC is off the channel is kept disabled (but turning the DAC back on does NOT enable the channel).

Mixer

Each channel's DAC output goes to a pair of on/off switches for the left and right channels before they are sent to the left/right mixers. A mixer simply adds the voltages from each channel together. These left/right switches are controlled by NR51. When a switch is off, the mixer receives 0 volts.

The Vin bits of NR50 control mixing of the Vin signal from the cartridge, allowing extra sound hardware.

The mixed left/right signals go to the left/right master volume controls. These multiply the signal by (volume+1). The volume step relative to the channel DAC is such that a single channel enabled via NR51 playing at volume of 2 with a master volume of 7 is about as loud as that channel playing at volume 15 with a master volume of 0.

Power Control

NR52 controls power to the sound hardware. When powered off, all registers (NR10-NR51) are instantly written with zero and any writes to those registers are ignored while power remains off (except on the DMG, where length counters are unaffected by power and can still be written while off). When powered on, the frame sequencer is reset so that the next step will be 0, the square duty units are reset to the first step of the waveform, and the wave channel's sample buffer is reset to 0.

Power state does not affect wave memory, which can always be read/written. It also does not affect the 512 Hz timer that feeds the frame sequencer.

When the Game Boy is switched on (before the internal boot ROM executes), the values in the wave table depend on the model. On the DMG, they are somewhat random, though the particular pattern is generally the same for each individual Game Boy unit. The game R-Type doesn't initialize wave RAM and thus relies on these. One set of values is

```
84 40 43 AA 2D 78 92 3C 60 59 59 B0 34 B8 2E DA
```

On the Game Boy Color, the values are consistently

```
00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF
```

Register Reading

Reading NR52 yields the current power status and each channel's enabled status (from the length counter).

Wave RAM reads back as the last value written.

When an NRxx register is read back, the last written value ORed with the following is returned:

```
     NRx0 NRx1 NRx2 NRx3 NRx4
    ---------------------------
NR1x  $80  $3F $00  $FF  $BF
NR2x  $FF  $3F $00  $FF  $BF
NR3x  $7F  $FF $9F  $FF  $BF
NR4x  $FF  $FF $00  $00  $BF
NR5x  $00  $00 $70

$FF27-$FF2F always read back as $FF
```

That is, the channel length counters, frequencies, and unused bits always read back as set to all 1s.

Vin Mixing

The cartridge connector includes a sound input called Vin. When enabled via NR50, it is mixed in before the master volume controls. On the DMG and MGB, 0.847 volts gives equivalent to 0 on a channel DAC, and 3.710 volts is equivalent to 15 on a DAC, with other values linearly distributed between those voltages. On the CGB, the range is 1.920 volts to 2.740 volts, a quarter of the DMG range, thus sound fed to the CGB's Vin is significantly louder. When nothing is connected to Vin, it naturally floats at the middle voltage (silence).

Obscure Behavior

- The volume envelope and sweep timers treat a period of 0 as 8.
- Just after powering on, the first duty step of the square waves after they are triggered for the first time is played as if it were 0. Also, the square duty sequence clocking is disabled until the first trigger.
- When triggering the wave channel, the first sample to play is the previous one still in the high nibble of the sample buffer, and the next sample is the second nibble from the wave table. This is because it doesn't load the first byte on trigger like it "should". The first nibble from the wave table is thus not played until the waveform loops.
- When triggering a square channel, the low two bits of the frequency timer are NOT modified.
- Extra length clocking occurs when writing to NRx4 when the frame sequencer's next step is one that doesn't clock the length counter. In this case, if the length counter was PREVIOUSLY disabled and now enabled and the length counter is not zero, it is decremented. If this decrement makes it zero and trigger is clear, the channel is disabled. On the CGB-02, the length counter only has to have been disabled before; the current length enable state doesn't matter. This breaks at least one game (Prehistorik Man), and was fixed on CGB-04 and CGB-05.
- If a channel is triggered when the frame sequencer's next step is one that doesn't clock the length counter and the length counter is now enabled and length is being set to 64 (256 for wave channel) because it was previously zero, it is set to 63 instead (255 for wave channel).
- If a channel is triggered when the frame sequencer's next step will clock the volume envelope, the envelope's timer is reloaded with one greater than it would have been.
- Using a noise channel clock shift of 14 or 15 results in the LFSR receiving no clocks.
- Clearing the sweep negate mode bit in NR10 after at least one sweep calculation has been made using the negate mode since the last trigger causes the channel to be immediately disabled. This prevents you from having the sweep lower the frequency then raise the frequency without a trigger inbetween.
- If the wave channel is enabled, accessing any byte from $FF30-$FF3F is equivalent to accessing the current byte selected by the waveform

position. Further, on the DMG accesses will only work in this manner if made within a couple of clocks of the wave channel accessing wave RAM; if made at any other time, reads return $FF and writes have no effect.

- Triggering the wave channel on the DMG while it reads a sample byte will alter the first four bytes of wave RAM. If the channel was reading one of the first four bytes, the only first byte will be rewritten with the byte being read. If the channel was reading one of the later 12 bytes, the first FOUR bytes of wave RAM will be rewritten with the four aligned bytes that the read was from (bytes 4-7, 8-11, or 12-15); for example if it were reading byte 9 when it was retriggered, the first four bytes would be rewritten with the contents of bytes 8-11. To avoid this corruption you should stop the wave by writing 0 then $80 to NR30 before triggering it again. The game Duck Tales encounters this issue part way through most songs.

- "Zombie" mode: the volume can be manually altered while a channel is playing by writing to NRx2. Behavior depends on the old and new values of NRx2, and whether the envlope has stopped automatic updates. The CGB-02 and CGB-04 are the most consistent:

  - If the old envelope period was zero and the envelope is still doing automatic updates, volume is incremented by 1, otherwise if the envelope was in subtract mode, volume is incremented by 2.
  - If the mode was changed (add to subtract or subtract to add), volume is set to 16-volume.
  - Only the low 4 bits of volume are kept after the above operations.

Other models behave differently, especially the DMG units which have crazy behavior in some cases. The only useful consistent behavior is using add mode with a period of zero in order to increment the volume by 1. That is, write $V8 to NRx2 to set the initial volume to V before triggering the channel, then write $08 to NRx2 to increment the volume as the sound plays (repeat 15 times to decrement the volume by 1). This allows manual volume control on all units tested.

- When all four channel DACs are off, the master volume units are disconnected from the sound output and the output level becomes 0. When any channel DAC is on, a high-pass filter capacitor is connected which slowly removes any DC component from the signal. The following code applied at 4194304 Hz implements these two behaviors for one of the DMG output channels (unoptimized floating point for clarity):

```
static double capacitor = 0.0;

double high_pass( double in, bool dacs_enabled )
{
    double out = 0.0;

```

```
    if ( dacs_enabled )
    {
        out = in - capacitor;

        // capacitor slowly charges to 'in' via their difference
        capacitor = in - out * 0.999958; // use 0.998943 for MGB&CGB
    }
    return out;
}
```

The charge factor can be calculated for any output sampling rate as 0.999958^(4194304/rate). So if you were applying high_pass() at 44100 Hz, you'd use a charge factor of 0.996.

Differences

This summarizes differences I've found among the models tested.

Wave RAM access:

- Possible only when it's doing wave RAM read (DMG-03, DMG-05, DMG-06, MGB-01).
- Can be accessed any time (CGB-02, CGB-04, CGB-05).

Wave channel re-trigger without disabling first (via NR30):

- Re-writes first four bytes of wave RAM (DMG-03, DMG-05, DMG-06, MGB-01).
- Behaves normally (CGB-02, CGB-04, CGB-05).

Length counters and power off:

- Preserved and can be written while off (DMG-03, DMG-05, DMG-06, MGB-01).
- Always zero at power on (CGB-02, CGB-04, CGB-05).

Length clocking on NRx4:

- New length enable doesn't matter (CGB-02).
- Length must now be enabled (DMG-03, DMG-05, DMG-06, CGB-04, CGB-05, MGB-01).

Volume changes on NRx2 write:

- $x0 to $xx and $x7 to $xx are very screwey (DMG-03, DMG-05, DMG-06, MGB-01).
- Behavior as described in obscure behavior (CGB-02, CGB-04).
- If mode isn't being changed, only $x8 to $xx affects volume. Mode change is also a bit different (CGB-05).

To Do

- Using an envelope or sweep period of 0 then switching to another period also causes an extra clock in some cases.

- Frequency sweep has some really intricate behavior when rewriting sweep register

- Noise's frequency timer is more complex than described, resulting in trigger doing something more than simply reloading it. It may have multiple dividers to achieve the documented periods, with only some of them being reset on trigger.

- Behavior when triggering and writing to registers within a few clocks of frame sequencer events has yet to be determined. There will be lots of odd things uncovered for sure.

- Figure out exactly how noise LFSR is implemented with regard to mode changes.

- Document exact timing for DMG wave issues.

Thanks

- Lord Nightmare for GBSOUND.txt, assistance, testing, GBs to test.
- Laguna for the gnuboy emulator.
- Ville Helin for WLA DX GB-Z80 assembler.
- sinamas for feedback about this document and my test ROMs.

- This page was last modified on 8 May 2008, at 19:10.