The Science of the Bit Fiddle

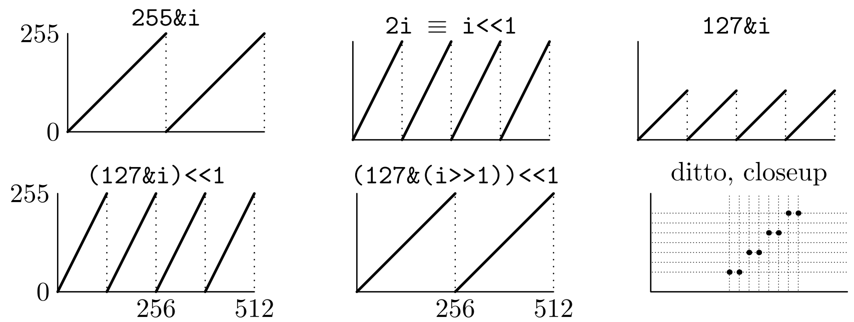
Notes and tables to accompany an analysis of
Rob Miles's 2016 **Bitshift Variations in C Minor**

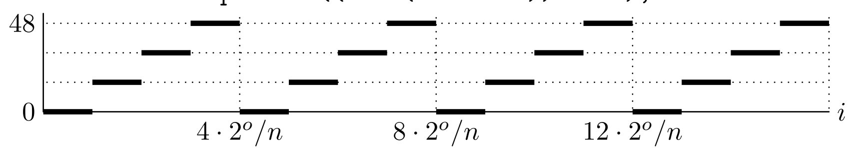
Original code: http://txti.es/bitshiftvariationsincminor
Intro movie: https://www.youtube.com/watch?v=MqZgoNRERY8
Recording: https://soundcloud.com/robertskmiles/bitshift-variations-in-c-minor

Basics

Masking	`i = xxxx xxxx`	Shifting	`i=22 = 0001 0110`	shifting << and >>
	`mask = 0001 1111`		`i<<2 = 0101 1000`	binds tighter than
	`mask&i = 000x xxxx`		`i>>2 = 0000 0101`	masking &

Sawtooth Sound





Sound generator: i = clock, n = pitch indicator, o = octave, 4 = volume

Just Intonation

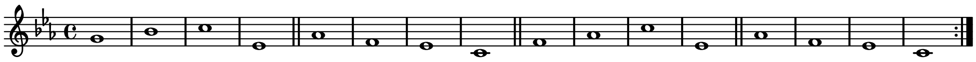
char:	'%'	'6'	'B'	'Q'	'Y'	'j'	'}'	
ASCII:	37	54	66	81	89	106	125	
+51:	88	105	117	132	140	157	176	
ratio:	1	1.19	1.32	1.50	1.59	1.78	2	the ratios of
	1/1	6/5	4/3	3/2	8/5	16/9	2/1	just intonation
note:	C	E♭	F	G	A♭	B♭	C′	C minor

Melodies

Assume playback at 8192 Hz (close to the actual 8000 Hz but gives nice figures). The expression `3&(i>>16)` chooses between two sets of notes. It yields the repeating sequence 0 1 2 3 and progresses every 2^{16} increments of i or about every 8 seconds.

1. `BY}6YB6%` corresponding to F A♭ C′ E♭ A♭ F E♭ C for `3&(i>>16)` in 1, 2, 3
2. `Qj}6jQ6%` corresponding to G B♭ C′ E♭ B♭ G E♭ C for `3&(i>>16)` = 0

First voice: controlled by `n` = `i`/2^{14}, which increments every 2 seconds. The current set of notes is indexed by `n%8`, so we get the 1st half of set 2, the 2nd half of set 1, then the entire set 1, before the melody repeats. Period is thus 32 seconds.



Voices 2, 3, 4: left as an exercise...

Outlook

Omitted here: remaining voices, voice on/off, harmonies, period length. Terseness: exploiting operator precedence, type defaults (int), arguments to main are really just local variables defined using as little code as possible. How was it invented? By genius! But remember the law of downhill synthesis and uphill analysis (V. Braitenberg in his 1986 book *Vehicles*). I found this a *fun* analysis of a *fascinating* synthesis.

dec	*binary*
0	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
4	0000 0100
5	0000 0101
6	0000 0110
7	0000 0111
8	0000 1000
9	0000 1001
10	0000 1010
11	0000 1011
12	0000 1100
13	0000 1101
14	0000 1110
15	0000 1111
16	0001 0000
17	0001 0001
18	0001 0010
19	0001 0011
20	0001 0100
21	0001 0101
22	0001 0110
23	0001 0111
24	0001 1000
25	0001 1001
26	0001 1010
27	0001 1011
28	0001 1100
29	0001 1101
30	0001 1110
31	0001 1111
63	0011 1111
127	0111 1111
255	1111 1111

Timing Divisors at 8192 Hz

`i>>10` ≡ `i/1024`
8 times per sec

`i>>11` ≡ `i/2048`
4 times per sec

`i>>12` ≡ `i/4096`
2 times per sec

`i>>13` ≡ `i/8192`
once per sec

`i>>14` ≡ `i/16384`
two seconds

`i>>17` ≡ `i/131072`
16 seconds

U. J. Rüetschi