Group 27
Loren Lugosch 260404057
Lulan Shen 260449509

Part 1: Tempo Circuit

Building and Testing

The tempo circuit provides a tempo waveform. It does this by dividing the clock by a division factor proportional to the desired number of beats per minute (bpm). One output, beat gate, is a metronome, and one output, tempo enable, controls the music.

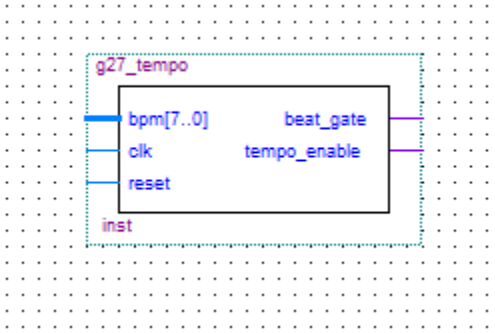

Figure 1- Symbol for Tempo Circuit

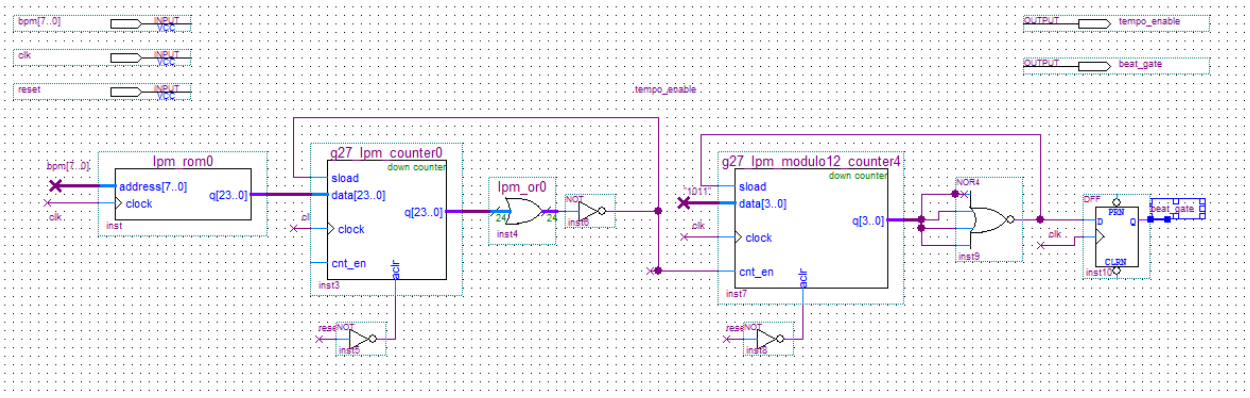

Figure 2- Interior of Tempo Circuit

We made the circuit by altering the design shown in the lab description and tested the circuit by testing each component and adding components to it. We calculated the number of bits of the MIF ROM as ceiling(log 2 (maximum clock division factor)) = 22, where the max factor = 2777777 = 125000000 / 45 (the lowest possible tempo).

First, we make sure the first counter works. We test the counter with the fixed value "00000000000000000000010111" (23) rather than the ROM. This means that sload of the first counter will be high every (data * period) = 480 ns, where data = 24 and period = 20ns (the period of a 500 MHz clock).

We use the following simplified architecture to verify:

```
_____________________________________________

begin

        I3 <= not I2;
        tempo_enable <= I3;
        I0 <= "00000000000000000000010111"; -- 23, fixed for now

        inst1: component g27_lpm_counter0
                        --port map (clock=>clk, aclr=>reset, data=>I0, sload=>I6, cnt_en=>'1', q=>I1);
                        port map (clock=>clk, aclr=>reset, data=>I0, sload=>I3, cnt_en=>'1', q=>I1);

                inst4: component g27_lpm_or0 port
                map(data0=>I1(0),data1=>I1(1),data2=>I1(2),data3=>I1(3),data4=>I1(4),data5=>I1(5),da
                ta6=>I1(6),data7=>I1(7),
                data8=>I1(8),data9=>I1(9),data10=>I1(10),data11=>I1(11),data12=>I1(12),data13=>I1(1
                3),data14=>I1(14),data15=>I1(15),data16=>I1(16),data17=>I1(17),data18=>I1(18),data1
                9=>I1(19),data20=>I1(20),data21=>I1(21),data22=>I1(22),data23=>I1(23),result=>I2);

end inside;
_____________________________________________
```

The first counter's NORed output is high once per 480 ns as we expect, so the tempo_enable generation works.

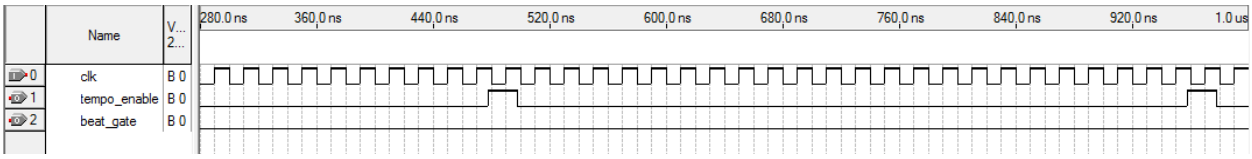

Figure 3- First counter, timing simulation

Next, we test the second counter, at first without the flipflop. Again, the input data to the first counter is 23, so we expect the 12-counter's NORed output to be low every 12 * 480 = 5760 ns. We choose an end time that will allow us to see this output go high at least twice, 12 us.

Using the following simplified architecture, we simulate again:

```
begin

        I3 <= not I2;
        I6 <= not I5;
```

```
	tempo_enable <= I3;
	beat_gate <= I6; -- incorrect: beat_gate is half of this
	I0 <= "00000000000000000010111"; -- (24-1) = 23, fixed for now

	--inst0: lpm_rom
		--	generic map (lpm_widthad=>8,
			--			lpm_numwords=>256,
			--			lpm_outdata=>"UNREGISTERED",
			--		lpm_address_control=>"REGISTERED",
			--			lpm_file => "g27_bpm.mif", --or g27_bpm_low.mif if 50
kHz
			--			lpm_width => 24)
		--	port map (inclock=>clk, address=>bpm, q=>I0);

	inst1: component g27_lpm_counter0
			--port map (clock=>clk, aclr=>reset, data=>I0, sload=>I6, cnt_en=>'1', q=>I1); --
PROBLEM! sload?
			port map (clock=>clk, aclr=>reset, data=>I0, sload=>I3, cnt_en=>'1', q=>I1);

	inst3: component g27_lpm_modulo12_counter4
			--generic map (--lpm_direction => "DOWN", -- KEEP COMMENTED
					--lpm_port_updown => "PORT_UNUSED",
					--lpm_type => "LPM_COUNTER",
					--lpm_width => 4)
			port map (clock=>clk, aclr=>reset, data=>"1100", sload=>I6, cnt_en=>I3,q=>I4);

	inst4: component g27_lpm_or0 port
map(data0=>I1(0),data1=>I1(1),data2=>I1(2),data3=>I1(3),data4=>I1(4),data5=>I1(5),data6=>I1(6),data7
=>I1(7),

	data8=>I1(8),data9=>I1(9),data10=>I1(10),data11=>I1(11),data12=>I1(12),data13=>I1(13),data1
4=>I1(14),data15=>I1(15),

	data16=>I1(16),data17=>I1(17),data18=>I1(18),data19=>I1(19),data20=>I1(20),data21=>I1(21),d
ata22=>I1(22),

	data23=>I1(23),result=>I2);

	inst5: component g27_lpm_or1 port
map(data0=>I4(0),data1=>I4(1),data2=>I4(2),data3=>I4(3),result=>I5);

	--inst2: component g27_lpm_ff2 port map (clock=>clk, aclr=>reset, enable=>I6, q=>beat_gate);

end inside;
```

The simulation shows the 12-counter's sload high at roughly 5775 ns. (Note: output is called "beat_gate", but really this is just sload to the mod-12 counter, since we're measuring the input to the flipflop and not the output.)

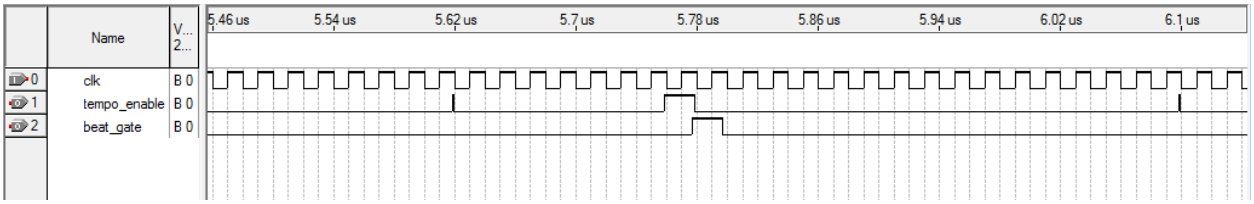

Figure 4- 12 Counter first high, timing simulation

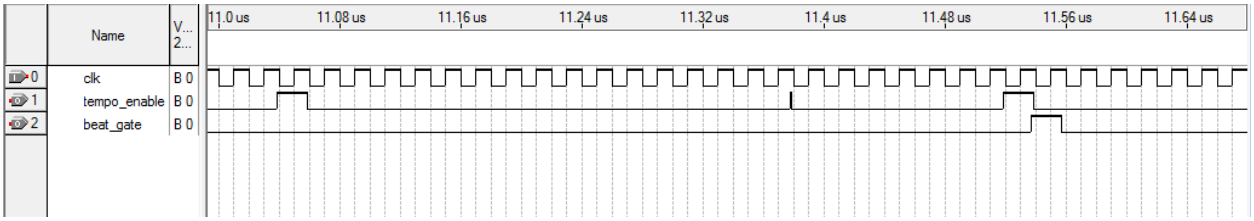

Figure 5- 12 Counter second high, timing simulation

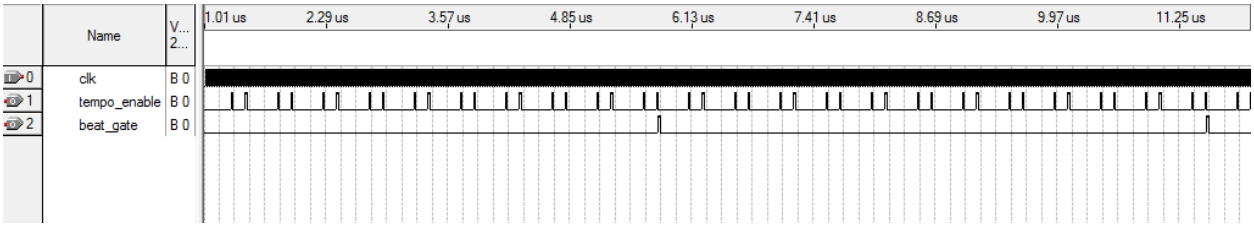

Figure 6- Overview of counters. (Glitches in 12-counter smaller than they appear.)

Finally, we hook up the flipflop. Beat_gate should be a square wave of period 24 * 480 = 11520 ns. The simulation confirms this.

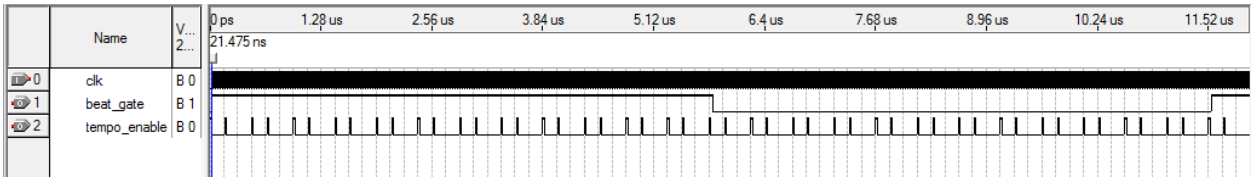

Figure 7- beat_gate and tempo_enable, timing simulation

We got rid of the glitches by replacing the OR gate with a multiplexor.

We repeat this for 15 more values of "data". Binary representation and period of tempo_enable and beat_gate are as follows:

data + 1	**Binary repr. of data**	**tempo_enable period**	**beat_gate period**
100	1100100	2000	48000
200	11001000	4000	96000
300	100101100	6000	144000
400	110010000	8000	192000

500	111110100	10000	240000
600	1001011000	12000	288000
700	1010111100	14000	336000
800	1100100000	16000	384000
900	1110000100	18000	432000
1000	1111101000	20000	480000
1100	10001001100	22000	528000
1200	10010110000	24000	576000
1300	10100010100	26000	624000
1400	10101111000	28000	672000
1500	10111011100	30000	720000

The waveforms indicate the correct functioning of the circuit. (Note: all that changes in these images is the duration.)

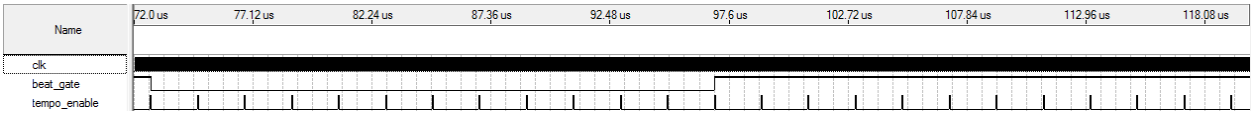

Figure 8- 100

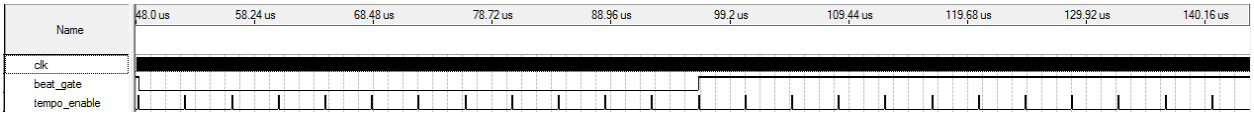

Figure 9- 200

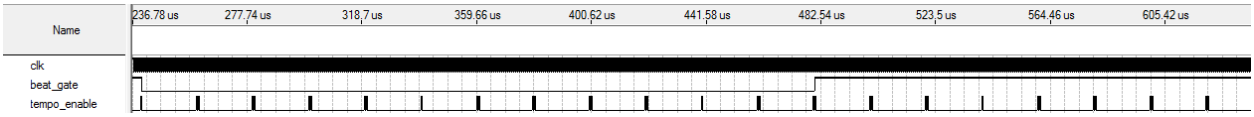

Figure 10- 1000

Timing Analysis

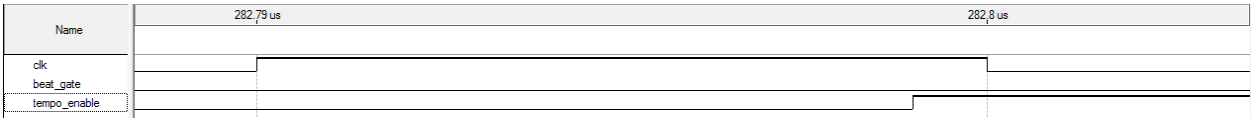

Figure 11- Delay between rising edge of clock and tempo_enable high

There is a 9 ns delay.

FPGA Utilization

Flow Status	Successful - Fri Nov 08 19:21:13 2013
Quartus II 64-Bit Version	9.1 Build 350 03/24/2010 SP 2 SJ Full Version
Revision Name	g27_lab3
Top-level Entity Name	g27_tempo
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	649 / 18,752 (3 %)
Total combinational functions	405 / 18,752 (2 %)
Dedicated logic registers	529 / 18,752 (3 %)
Total registers	529
Total pins	12 / 315 (4 %)
Total virtual pins	0
Total memory bits	8,448 / 239,616 (4 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)