Computer-based project in VLSI Design
Second Interim Report

Tina Wen
College: Trinity
Email: tw311@cam.ac.uk

June 4, 2007

Introduction

After designing the frequency synthesizer from the behavior point of view, the design is further developed from the schematics point of view. The frequency synthesizer is designed in 4 blocks, the ring oscillator, the counter, the comparator, and the control. Each block is built from primitive cells and simulated to make sure that all the outputs behave as expected in *QuickSimII* given a set of input signals. Then these components are brought together to form top level.

Behavior of NOR gate

A very simple NOR gate is connected in the top level, and simulated using *QuickSimII*. The simulation result is shown in Graph 1 in the Appendix. The `NOR` output is the result of A OR B. The `N$460` signal, on the other hand, is the result of A NOR B. This is due to the inverter added right before the output pin. Some kind of buffer, usually an inverter, is needed before any output pin because outputs need to drive enough current. Caution needs to be taken when observing simulation results because all outputs are inverted.

Schematic design of the ring oscillator

The ring oscillator is designed by using 29 NOR gates. The odd number of inverters cause the oscillation to happen. `ring_out1` is taken after 7 NOR gates from the `ring_enable` signal, and `ring_out2` is taken after 25 NOR gates from the `ring_enable` signal. The schematic overview can be seen from Graph 10 in the Appendix.

Schematic design of the 6-bit counter

A good point to start is from a 4-bit counter. The state table of the 4-bit counter is shown in Table 1. By observing the table, we can write logic functions for flip-flops for every bit.
bit 0: $Q_0^+ = \overline{Q_0}$
bit 1: $Q_1^+ = \overline{Q_0}Q_1 + Q_0\overline{Q_1}$
bit 2: $Q_2^+ = \overline{Q_2}Q_1Q_0 + Q_2\overline{Q_0} + Q_2\overline{Q_1}$
bit 3: $Q_3^+ = \overline{Q_3}Q_2Q_1Q_0 + Q_3\overline{Q_2} + Q_3\overline{Q_1} + Q_3\overline{Q_0}$
Based on the trend, bit 4 and bit 5 can be determined as:
bit 4: $Q_4^+ = \overline{Q_4}Q_3Q_2Q_1Q_0 + Q_4\overline{Q_3} + Q_4\overline{Q_2} + Q_4\overline{Q_1} + Q_4\overline{Q_0}$
bit 5: $Q_5^+ = \overline{Q_5}Q_4Q_3Q_2Q_1Q_0 + Q_5\overline{Q_4} + Q_5\overline{Q_3} + Q_5\overline{Q_2} + Q_5\overline{Q_1} + Q_5\overline{Q_0}$
The schematics overview of the 6-bit counter can be found in Graph 9 in the Appendix.

Simulated behavior of the ring oscillator/6-bit counter block

The ring oscillator and the 6-bit counter are combined into one block with inputs `ENB_RING_OSC` and `RESET_COUNTER`, and outputs `ring_out1`, `ring_out2`, and 6 `DATA_OUT` lines. The schematics overview of this block can be found in Graph 12 in the Appendix. Simulation is carried out under unit time mode, in which all gates are assumed to have a delay of $0.1ns$. Delay introduced by paracitic capacitance and manufacturing gate delays are ignored. This mode is good for verification of truth tables, but not practical. The simulation waveforms are shown in Graph 2 in the Appendix. Timing analysis is shown in Table 2 in the Appendix. T is the period of oscillation. t_1 is the delay from `ENB_RING_OSC` to `ring_out1`. t_2 is the delay from `ring_out1` to `ring_out2`. As shown, all the delays are obviously too low due to unpractical modeling.

The timing mode is changed to use the delay model, which allows more realistic timing modeling of each gate. Simulation is run again, and the waveforms can be found in Graph 3 in the Appendix. Timing analysis is shown in Table 2. All the delays are within the specified range. The ring oscillator is working and the NOR gates behave fully consistent with my expectations.

Another oscillator/6-bit counter simulation can be found in Graph 4 in the Appendix. A pulse appears in `RESET_COUNTER`. As it goes low, it sets all the data lines to be low, as expected. A full cycle of counter operation can be seen in Graph 4.

Schematic design of the comparator

The 6-bit comparator is implemented by using 6 Exclusive NOR gates to compare each bit, and a 6-input AND gate to combine the results. The schematics overview can be found in Graph 11 in the Appendix.

Schematic design of the control unit

The truth table for the control unit is shown in Table 3. The design overview is shown in Graph 8 in the Appendix. `rst_c` is simply an OR of `comp` and `reset`. `clk_out`, on the other hand, is not simply a combinational logic. It is synchronous, and clocked by `comp`. $\overline{\texttt{Q}}$ is connected to `D` so a toggle bistable is built. `reset` is resetting the bistable. When `reset` is high, the bistable is reset, and `Q`, thus `rst_c` is always high. When `reset` is low, the flip-flop toggles based on the clock from `comp`. Simulation shows that this control unit is working as expected.

Top level schematic design and simulation result

The top level schematic overview can be found in Graph 7 in the Appendix. The ring oscillator and the 6-bit counter are combined into one block. This frequency synthesizer is working properly and the simulation result can be found in Graph 5 and 6 in the Appendix. Graph 5 shows that the `ring_enb` is controlling whether `ring_out1` and `ring_out2` are oscillating, providing the clock. The divider has value 3, and `clk_out` successfully divides `ring_out` by 3. In Graph 6, the divider is initially set to be 7, and later set to be 4. `reset` is set to be high and then low during the period when divider takes in a new value. After `reset` goes low, the counter starts counting again, and `comp_out` gives the correct output. `clk_out` divides `ring_out1` by the correct amount specified by the divider. It works for both even and odd numbers. `comp_out`, being the output from the comparator, has a high pulse every time compare is true. (The signal is inverted on the graph.) It shows that the comparator is working as expected. `clk_out` outputs the satisfactory signal and can be fed to the phase detector.

Conclusion

The frequency synthesizer is built successfully from primitive cells. This hierarchy approach is efficient in designing VLSI circuits. When building a large top level frequency synthesizer, it is important to simulate each small parts and make sure everything works as expected before putting everything together. This divide and conquer approach is key to success.

Appendix

Q	Q^+	Q	Q^+
0000	0001	1000	1001
0001	0010	1001	1010
0010	0011	1010	1011
0011	0100	1011	1100
0100	0101	1100	1101
0101	0110	1101	1110
0110	0111	1110	1111
0111	1000	1111	0000

Table 1: State table for the 4-bit counter

Timing mode	$T(ns)$	$t_1(ns)$	$t_2(ns)$
UNIT	$56.5 - 50 = 6.5$	$50.7 - 50 = 0.7$	$52.4 - 50.7 = 1.7$
DELAY	$491 - 321 = 170$	$321 - 300 = 21$	$373 - 323 = 50$
specified range	$[145, 185]$	$[17, 21]$	$[40, 60]$

Table 2: Timing analysis for the oscillator/6-bit counter block

comp	reset	clk_out	rst_c
0	0	previous `clk_out`	0
0	1	0	1
1	1	0	1
1	0	$\overline{\text{previous } \texttt{clk\_out}}$	1

Table 3: Truth table of the control unit