Final Report: Digital Design Lab: Simulation of Combinational Logic using ModelSim

Author: Lixuan Yang

Institution: University College London

Report Module: Digital Design

Tutor: Dr. Georgios Zervas

Date:28th, March 2020

Abstract

The objectives of the experiment were determining the correlation between the Oscillation Period and the number of inverters and measuring the delay of a single LUTs in combinational logic circuit using software such as Quartus Prime Standard Edition and ModelSim ASE for modification and simulation. The number of inverters which were simulated in this experiment is 201, 301, 401 and 501. The result shows the oscillation period has a positive correlation with the number of inverters, and the delay of a single LUTs in combinational logic circuit is reasonable.

1.Introduction

The purpose of this experiment was to apply a Quartus project by using SystemVerilog rather than in schematics and investigated how submodule are instantiated. Also, it was aimed to understand the simulation of combinational logic. Moreover, it was aimed to determine the delay of combinational logic blocks, compare with the delay of an application specific integrated circuit (ASIC) process and indicate some conclusions on the relative performance.
Digital Logic Circuits are mainly separated into two categories, combinational logic and sequential logic which both proceed digital signal in binary system. The output indicates the difference between two logics, as the output of combinational logic depends on the current state of input while the output of sequential logic depends on both current state of input and previous state of input which stored in memory unit as shown in Figure.1,2. *['Combinational and Sequential logic circuits', 2015]* In this report, combinational logic is used and will be discussed more.

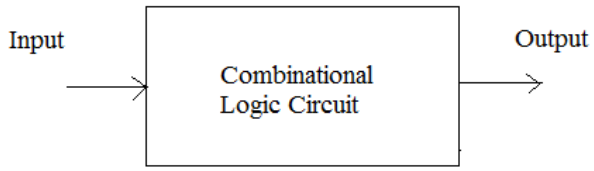

Figure.1 Combinational Logic

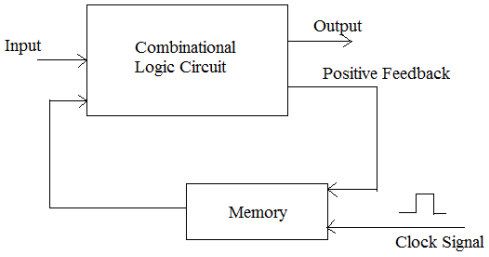

Figure.2 Sequential Logic

Field Programmable Gate Array (FPGA) contains varies combination of combinational logics and sequential logics as it consists of an array of configurable logic blocks with interconnection between the blocks. For combinational logics, Look-up Table (LUT) is used to implement in FPGA as it determines their outputs. During the process of the FPGA, the SystemVerilog code commands desired outputs of the FPGA, then LUTs separated output into 1 bit and set the output for each LUT into 1s or 0s. Next, the combinational logics are arranged to produce output set by each LUT and finally implement on FPGA.

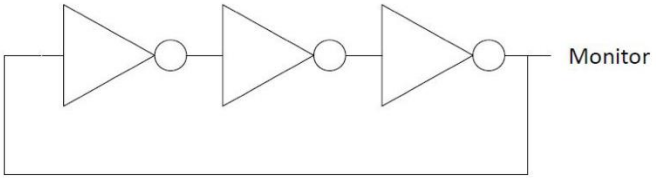

Figure.3 A ring oscillator with 3 inverters ['Lab Script',2020]

Figure.3 shows a ring oscillator with 3 inverters, which is used to characterise the delay of combinational logic. As it could be identified that the ring oscillator is constructed with an odd number of inverter gates connected in series in order to keep the original output from combinational logics, and with the output signal returning as the input signal which arise oscillation. The oscillation period can be used to observe the delay after LUT because of the oscillation properties. Also, due to the structure, the more inverter it contains, the longer the oscillation period it will obtain and the higher the combinational logic delay there is, therefore, with one inverter per LUT, the average delay could be calculated and determined.
As it is shown in Equation 1:

$$T = 2N(\frac{t_{rise}+t_{fall}}{2}) \quad (1)$$

Where T is the oscillation period, N is the number of inverters, t_{rise} and t_{fall} are the propagation delays of the rising and falling edges respectively. Other than the above factors, temperature could also affect the combinational logic delay, as temperature shows a positive correlation with oscillation period and combinational logic delay.
In reality, initial value should be obtained before the chain of inverters as ring oscillator, therefore a logical AND gate is added known as enable, as shown in Figure.4

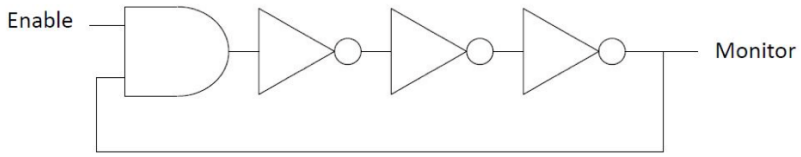

Figure.4 A 3 inverter ring oscillator with an enable

2.Methodology

The lab experiment was done remotely, therefore only software were required in this lab experiment which were Quartus Prime Standard Edition and ModelSim ASE.
At the beginning, a new project should be made in Quartus Prime Standard Edition with both ring_osc.sv and ring_osc_top.sv files. For ring_osc.sv file, line 28 was changed slightly to initialise the ring_signals. For ring_osc_top.sv file, 'enable', 'select' were identified as input, 'monitor' and 'led' were identified as output. Also, one of the ring oscillators with 201 inverters had been instantiated, with similar method, ring oscillators with 301, 401 and 501 inverters were instantiated as well. Then a 4:1 multiplexer was constructed to choose the type of ring oscillators. After that, 'Analysis and Synthesis' and 'EDA Netlist Writer was used debug and construct the code. 'Technology Map Viewer' was used to check the Top-level ring oscillator design which is shown in Figure.5 and the detailed construction was observed to ensure the structure of ring oscillator. Finally, 'Gate Level Simulation' was applied to create a 'Simulation' folder for later simulation in ModelSim ASE.

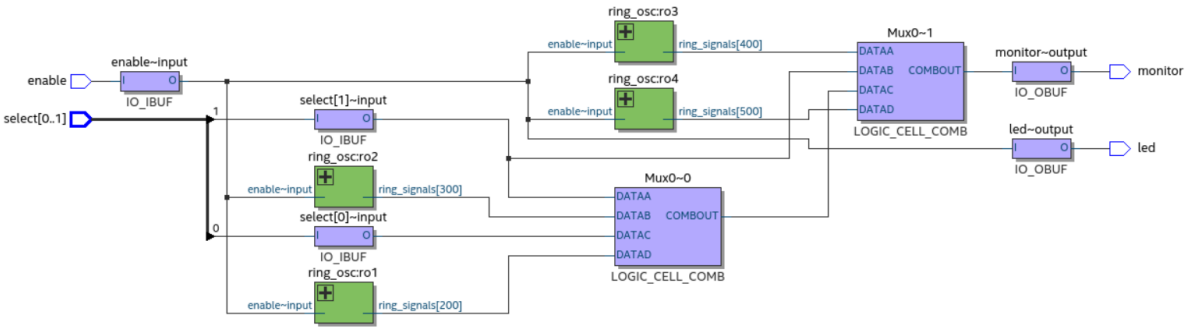

Figure.5 Top level ring oscillator design

After finishing the preparation in Quartus Prime Standard Edition, simulation of the ring oscillator was done in ModelSim ASE. The simulation started with loading the 'Simulation' folder from previous preparation. Outputting 'enable', 'select', 'monitor' and 'led' on the wave monitor. Then the simulation run length was adjusted from 100 ps to 500 ns for more accurate and clearer observation of the waves. The commands were entered one by one as shown in Figure.6, then oscillation period was measured for each ring oscillator by measuring the time of a complete wave. Finally graph of oscillation period versus number of inverters was plotted, the gradient and intercept were calculated and delay of a single LUT was determined.

a. force -freeze sim:/ring_osc_top/enable 0 0
b. force -freeze sim:/ring_osc_top/select 00 0
c. run
d. force -freeze sim:/ring_osc_top/enable 1 0
e. run
f. force -freeze sim:/ring_osc_top/select 01 0
g. run
h. force -freeze sim:/ring_osc_top/select 10 0
i. run
j. force -freeze sim:/ring_osc_top/select 11 0
k. run
l. run
m. run

Figure.6 Command for Simulation

3.Results

The written code for the ring oscillators and multiplexer is shown in Appendix A. The observation in 'Technology Map Viewer' is shown in Figure.5,7,8, which proved the structure shown in Figure.4 with the first LUT represents a AND gate at the beginning as enable and the rest LUTs represent the NOT gates till the end of the ring oscillator.

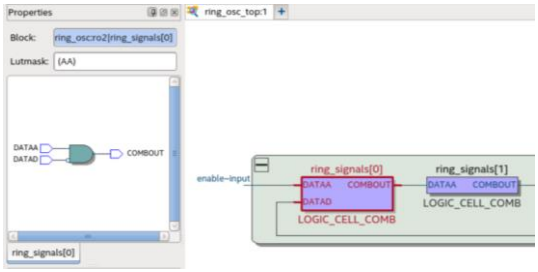

Figure.7 AND Gate

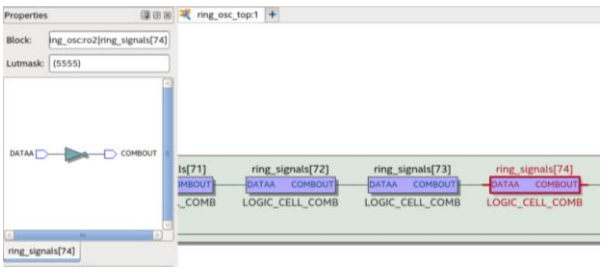

Figure.8 NOT Gate

For the simulation part in ModelSim ASE, observation was record down after commands e, g, i, m as shown in Figure. 9, 10, 11 and 11 respectively.

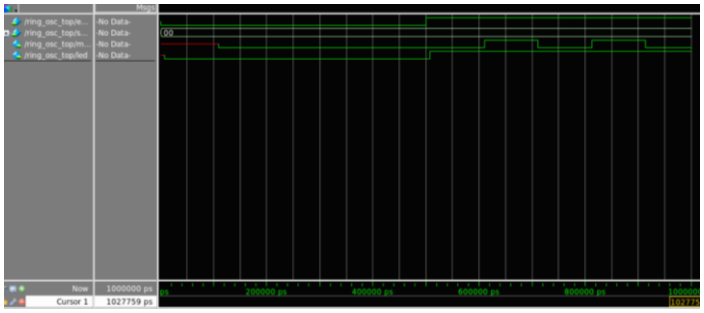

Figure.9 Simulation After Command e

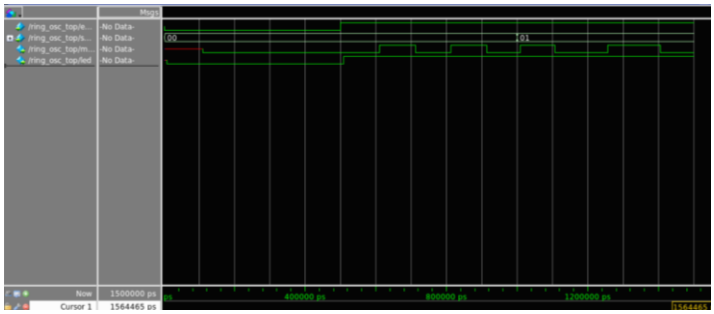

Figure.10 Simulation After Command g

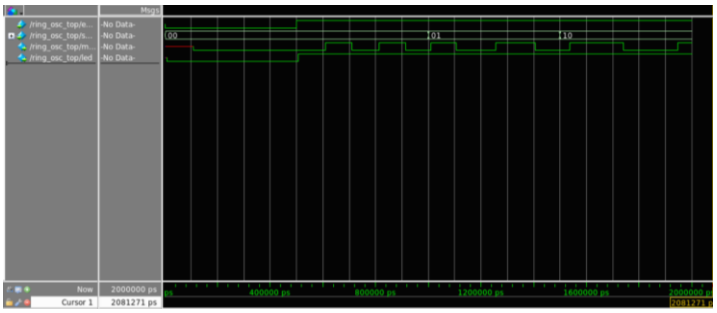

Figure.11 Simulation After Command i

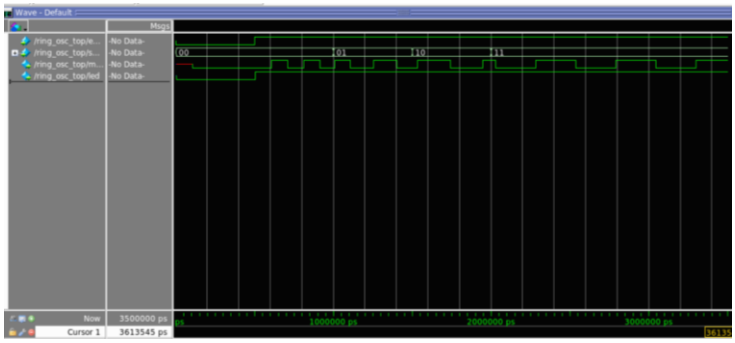

Figure.12 Simulation After Command m

As after command e, enable was set to 1, which triger the ring oscillator and output appeared in monitor and led channels. Comand b, f, h, j changed the ring oscillators with 201 inverters, 301 inverters, 401 inverters and 501 inverters respectively. The oscillation period was mesured with the time period of a compelete wave for each number of inverters, the data is shown in Appendix B. The the data was plotted versus number of LUTs/inverters as shown in Figure.13

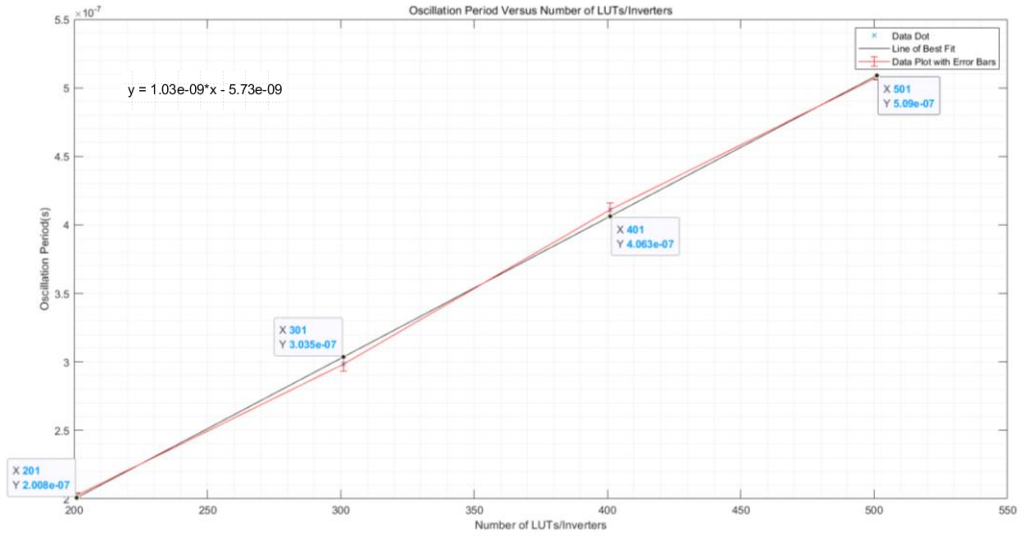

Figure.13 Graph of Oscillation Period Versus Number of LUTs/Inverters

The Matlab Code for the digram is shown in Appendix B. The graph shows a linear relationship between Oscillation Period and Number of Inverters, the gradient of the line of the best fit is $1.03*10^{-9}$ s, and the intercept point to x = 0, when there is no inverter, is $-5.73*10^{-9}$ s. Rearrange Equation 1, the gradient is also Oscillation Period over Number of Inverters:

$$\frac{T}{N} = t_{rise} + t_{fall} \tag{2}$$

Therefore, it can be deduced that the gradient is the total propagation delay, therefore, delay for a single LUT is gradient diveded by 2, which is $5.14*10^{-10}$ s.

4.Discussion

The number of inverters in the ring oscillator was set in high values during the lab experiment in order to measure the delay more accurate. Also, as larger number of inverters leads to longer oscillation time, the frequency of the oscillation become smaller. Thus, with small value in number of inverters, the frequency of the oscillation become larger, which could be unavailable to output signal during the simulation and cause puzzles in the measurement. Using the expression induced from the graph in Figure 13,

$$y = 1.03 * 10^{-9}x - 5.73 * 10^{-9} \tag{3}$$

Where x is the number of inverters, and y is the oscillation period. If number of inverters is 5, the oscillation period will be $-5.92*10^{-9}$ s based on the simulation model, therefore, when the number of inverters is 5, the simulation could not process.

It could be found that the delay of the ASICs with 45nm standard cell depends on its gate capacitance. The gate capacitance has a maximum value of 25 fF, and the maximum delay is 474.4 ps. It could be identified that FPGA combinational logic has a longer delay than the ASICs. This proves that FPGA works slower than ASIC, as ASICs are fully custom and non-reprogrammable while the FPGAs are flexible and reprogrammable. Therefore, for engineers who are doing digital technology, FPGAs are recommended to use during design and simulation which require modification during process, ASICs are recommended to use for final design output as they are denser and have lower cost.

The results that were obtained from Group have a gradient of $1.02 * 10^{-9}$s and intercept of $-5.84 * 10^{-9}$s. The possible factor responsible for the difference could be estimation and

reading error when the data was collected. Also, it could be caused by the code difference for multiplexer as the length of the code could vary the time delay during simulation. This time delay could be irrelevant to the number of LUTs/inverters.
The significance would be the negative value being obtained for the intercept, as the value should be zero or more likely to be positive theoretically due to the property of combinational logic circuit. Also, the negative value shows that when ring oscillator with small number of inverters will not be appropriate in this case, as the negative value of the intercept will cause systematic errors.

5.Conclusion

During the lab experiment, an example of a Quartus project defined using SystemVerilog had been done, combinational logic was simulated and the delay of combinational logic blocks had been measured which was 1.03 ns and it is longer than the delay of ASIC, 474.4 ps. Also, it could be concluded that oscillation period shows a positive correlation to the number of LUTs/inverters. Moreover, the simulation proves that the ring oscillator with large number of inverters (more than 200) could improve the accuracy of the delay measurement.
Overall, recalling all the process had been done and the result from the lab experiment, all the objectives had been achieved. Therefore, the lab experiment was successful.

Reference

[1]"Combinational and Sequential logic circuits.", *VLSIFacts*, 2015. [Online]. Available: https://www.vlsifacts.com/difference-combinational-sequential-logic-circuits/. [Accessed: 29- Mar- 2020].
[2] "Lab Script", Moodle.ucl.ac.uk, 2020. [Online]. Available: https://moodle.ucl.ac.uk/pluginfile.php/2749501/mod_resource/content/1/ELEC0010%20Lab%20script%20supplement%20for%20remote%20simulation-only%20experiment.pdf [Accessed: 26- Mar- 2020].

Appendix

Appendix A

Code for Ring Oscillators with number of inverters 301, 401 and 501:

```
ring_osc #(301) ro2(
    .enable  (enable),
    .monitor(m[1]));

ring_osc #(401) ro3(
    .enable  (enable),
    .monitor(m[2]));

ring_osc #(501) ro4(
    .enable  (enable),
```

```
        .monitor(m[3]));
```

Code for Multiplexer:

```
    always_comb
        case(select)
            2'b00       :monitor = m[0];
            2'b01       :monitor = m[1];
            2'b10       :monitor = m[2];
            2'b11       :monitor = m[3];
    default             :monitor = 0;
    endcase
```

Appendix B

Data collected for oscillation period

Number of Inverters	Initial Time of a Wave (ps)	End Time of a Wave (ps)	Oscillation Period (s)
201	610383	812921	$2.025*10^{-6}$
301	1106416	1404831	$2.984*10^{-6}$
401	1536231	1947388	$4.112*10^{-6}$
501	2285001	2792417	$5.074*10^{-6}$

Matlab Code:

```
Ta = [610383 1106416 1536231 2285001]*10^-12;
Tb = [812921 1404831 1947388 2792417]*10^-12;
Tp = Tb - Ta;
num_inv = [201 301 401 501];
plot(num_inv, Tp,'x');
p = polyfit(num_inv,Tp,1);
f = polyval(p,num_inv);
hold on;
plot(num_inv,f,'k');
errorbar(num_inv, Tp, Tp-f,'r');
grid minor;
xlabel('Number of LUTs/Inverters');
ylabel('Oscillation Period(s)');
title('Oscillation Period Versus Number of LUTs/Inverters');
legend;
```