Lab2 Delay Simulation of Full Adder

Hong-Jia Yang
Southern University of Science and Technology

Abstract—The correct realization of digital systems depends on the accurate modeling and analysis of delay. This report summaries main concepts and delay properties in each digital system design steps based on VIVADO simulation for full-adder in VHDL. Based on logic circuit, the logic expression of full-adder will be obtained and achieved in VHDL codes. The circuit and simulation waveform results will be discussed together to analyse the systems from different aspects. Some detailed delay principle in digital systems will also be explained.

Index Terms—full-adder, delay, VHDL.

1 Introduction

THE circuit that uses digital signal to complete arithmetic and logic operation is called digital circuit, or digital system.

The development of digital circuits, like analog circuits, has experienced several times from electronic tubes, semiconductor discrete devices to integrated circuits. But its development is faster than that of analog circuits. Since the 1960s, digital integrated devices have been made into small-scale logic devices. In the late 1970s, the emergence of microprocessor produce a qualitative leap in the performance of digital integrated circuits. In recent years, the rapid progress of programmable logic device PLD, especially field programmable gate array FPGA, has brought out a new situation in digital electronic technology. It not only has a large scale, but also combines hardware and software to make the function of the device more perfect and flexible.

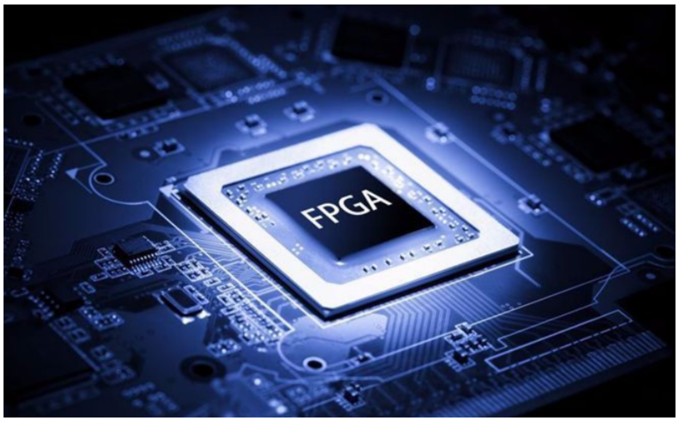


Fig. 1. Field Programmable Gate Array

Digital circuit or digital integrated circuit is a complex circuit composed of many logic gates. Compared with analog circuit, it mainly processes digital signals, so it has strong anti-interference ability.

In this course, we mainly study the digital system design through VHDL language, training the thinking of hardware circuit design, and use the modular method to realize the requirements. It is worth noting that correct hardware design thinking and implementation need to be based on long-term and rigorous learning and analysis of hardware language and software.

In this experiment, we will mainly learn the basic operation of VIVADO software, the main steps and main contents of VHDL language in hardware implementation. Based on the VHDL code of 1-bit full-adder, the simulation results will be obtained, the delay characteristics of each step will be discussed in detail, and the experimental results will be predicted and analyzed.

2 Lab Result

2.1 Pre-lab Preparation

We are asked to simulate the full-adder with the VHDL code and design a test bench with stimulus of A, B and Cin as given. We will firstly introduce the full-adder and its common sense.

Full-adder is a combined circuit to add two binary numbers and calculate the sum. One bit full adder can handle low carry and output standard addition carry. The logic circuit for full-adder is as follow: Cin is the low carry and Cout is the output carry.

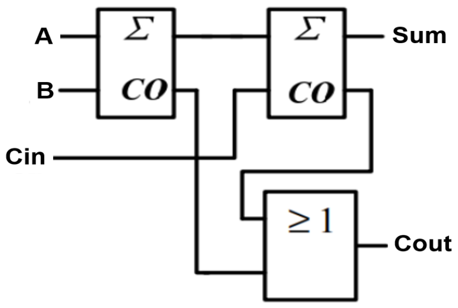


Fig. 2. Logic circuit of full-adder

According to the definition of full-adder, we can write the truth table of full-adder as shown in Fig 3, which has 8 conditions. From the truth table we can derive the logic expressions for Sum and Cout:

*Thanks

- *This work was supported in Department of Electrical and Electronic Engineering in Southern University of Science and Technology with Professor Yajun Yu, and the tutorial assistant and all classmates.*

Input			Output	
A	B	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig. 3. Truth table of full-adder

$$
\begin{aligned}
Sum &= (A'B'Cin' + A'BCin + AB'Cin + ABCin')' \\
&= A \oplus B \oplus Cin
\end{aligned}
$$

$$
\begin{aligned}
Cout &= (A'B' + B'Cin' + A'Cin')' \\
&= AB + Cin(A \oplus B) \cdot
\end{aligned}
$$

So according to the logic expression, we expect the waveform for the result is as follow:

	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80
A	1	0	1	0	1	0	1	0
B	0	0	1	1	0	0	1	1
Cin	0	1	0	0	0	1	0	0
s1	1	0	0	1	1	0	0	1
s2	0	0	0	0	0	0	0	0
s3	0	0	1	0	0	0	1	0
Sum	1	1	0	1	1	1	0	1
Cout	0	0	1	0	0	0	1	0

Fig. 4. Manually derived table

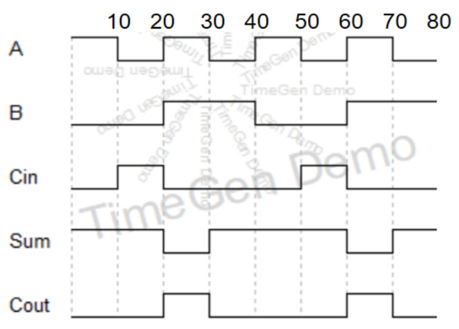


Fig. 5. Manually derived waveform

And the gate circuit diagram according to the logic expression is

2.2 VHDL code and behavioral simulation

Firstly we will write the VHDL code for the full-adder according to the logic expression and the delay, the main part of the code is the signals declaration and assignment.

In the entity declaration part, we declare the input signals A,B and Cin and the output signal Sum and Cout. In the architecture

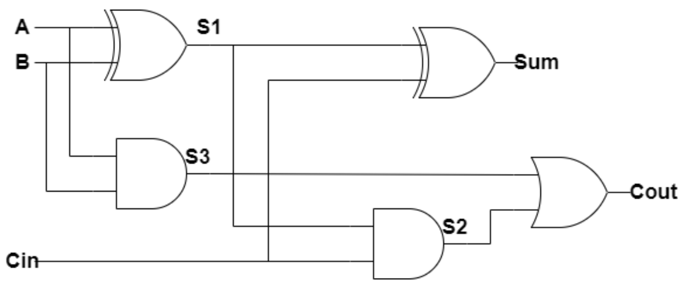


Fig. 6. Gate circuit diagram

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity full_adder is
    Port ( A,B,Cin : in STD_LOGIC;
           Sum,Cout : out STD_LOGIC);
end full_adder;

architecture dataflow of full_adder is
    signal s1, s2, s3: STD_LOGIC;
    constant gate_delay: time := 10ns;
begin
    L1: s1 <= (A xor B) after gate_delay;
    L2: s2 <= (Cin and s1) after gate_delay;
    L3: s3 <= (A and B) after gate_delay;
    L4: Sum <= (s1 xor Cin) after gate_delay;
    L5: Cout <= (s2 or s3) after gate_delay;
end dataflow;
```

part, we declare three signals s1, s2 and s3 as intermediate quantity to model the gate in the gate circuit result and assignment these signals according to the logic expression.

We define a constant gate_delay 10ns to simulate the gate delay in the circuit.

For the testbench code, the top level entity of the test bench has no external ports. In the architecture part, we declare the full-adder component under test. Then, we define stimulus signals and observed signals. Finally, we create an instance of the half_adder circuit and generate stimulus values.

Some important points of behavioral simulation should be noticed:

1. Behavioral simulation sends the VHDL design source program directly to the VHDL simulator, and make full use of the statements, predefined functions and library files in VHDL codes.

2. Behavioral simulation is for high-level system simulation and it can only evaluate and test the feasibility of VHDL system, not for any hardware system.

3. Many statements in the behavioral simulation level cannot be accepted by the synthesizer, that is, they cannot be implemented in the hardware system.

4. Only behavioral simulation can consider the delay in the signal assignment statement. The purpose of VHDL providing this statement is to realistically simulate the delay characteristics of the circuit in behavior simulation.

We get the result of behavioral result:

Obviously, **the behavioral simulation result is consistent with those shown in lecture notes but not consistent with common sense as shown in Fig 4.** This can be easily explained

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tb_full_adder is
--Port();
end tb_full_adder;

architecture test of tb_full_adder is
    component full_adder is
        port (A, B ,Cin: in std_logic;
              Sum, Cout: out std_logic);
    end component full_adder;

signal a_tb, b_tb, cin_tb:std_logic;
signal sum_tb, cout_tb:std_logic;

begin
    UUT:full_adder port map(A=>a_tb,B=>b_tb,
        Cin=>cin_tb,Sum=>sum_tb,Cout=>cout_tb);
    a_tb<='1','0'after 10ns,'1' after 20ns,
              '0'after 30ns,'1' after 40ns,
              '0'after 50ns,'1' after 60ns,
              '0'after 70ns;
    b_tb<='0','1'after 20ns,'0' after 40ns,
              '1'after 60ns;
    cin_tb<='0','1' after 10ns,'0' after 20ns,
                '1' after 50ns,'0' after 60ns;
end test;
```

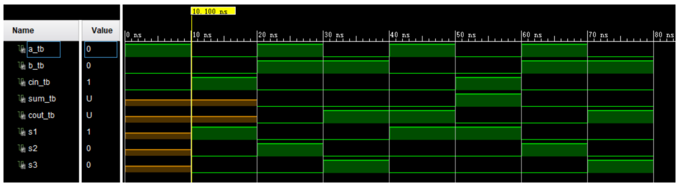


Fig. 7. Result of Behavioral Simulation

according to our important points of behavioral simulation above. That is, behavioral simulation holds our setting time delay 10ns and all the results are derived according to this delay. We write a waveform analysis table to show each step:

	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80
A	1	0	1	0	1	0	1	0
B	0	0	1	1	0	0	1	1
Cin	0	1	0	0	0	1	0	0
s1	U(1)	1(0)	0(0)	0(1)	1(1)	1(0)	0(0)	0(1)
s2	U(0)	0(1)	1(0)	0(0)	0(0)	0(1)	1(0)	0(0)
s3	U(0)	0(0)	0(1)	1(0)	0(0)	0(0)	0(1)	1(0)
Sum	U(U)	U(0)	0(0)	0(0)	0(1)	1(0)	0(0)	0(0)
Cout	U(U)	U(0)	0(1)	1(1)	1(0)	0(0)	0(1)	1(1)

Fig. 8. Waveform analysis current value(predict value)

We derive the value for s1,s2,s3,Sum and Cout. In the beginning, they are all undefined. During time 0-10ns, according to the logic expression, s1 and s3 can be obtained from the value of A and B, and these values will be assigned after 10ns(during 10-20ns). **Although s1 is undefined during 0-10ns, s2 can be derived as ’0’ for Cin is ’0’ and s2 is obtained by ’and’ logic operation.**

In the 10-20ns, the value derived for s1,s2 and s3 during 0-10ns will be assigned to them and new value for these signals will be derived. And during this time period, Sum and Cout will be derived.

In the following time period, similar process will occur and we can derive all values for these signals.

2.3 Post-Synthesis

Also, post-synthesis has some important points must be noticed:

1. **After synthesis, the VHDL synthesizer can generate a VHDL netlist file. That is, post-synthesis converts software into hardware circuit.**

2. **The post-synthesis will omit the delay value set by the assignment statement in the synthesis process**. In the **functional simulation** after post-synthesis, the delay between signals or variables is regarded as 0 delay in theory. However, since both behavioral simulation and functional simulation are software simulation by computer, even in the simulation and execution of parallel statements, there is a sequence. The cause of this false simulation is that the assumption of 0 delay cannot exist in the objective world.

In order to make a logical sequence of symbols for information transmission, a small delay——delta, will be automatically set, that is, the minimum resolution time of a VHDL simulator. **The setting of this delta is only for simulation and cannot fully represent the actual inertia delay of the device. In most cases, this inherent delay approximately reflects the behavior of the actual device. However, due to the subsequent wiring, the delay of each gate is certainly not the same, so there is a problem with the setting of this delta.Maybe after the synthesis, because the delta delay setting, there is no problem with the timing waveform, but there is a problem after the place and route.**

From what has been discussed above, we can easily understand the post-synthesis functional simulation waveform as follow:

Only the logic functions described by VHDL are tested and simulated in functional simulation to understand whether the functions realized meet the requirements of the original design. The simulation process does not involve the hardware characteristics of specific devices.

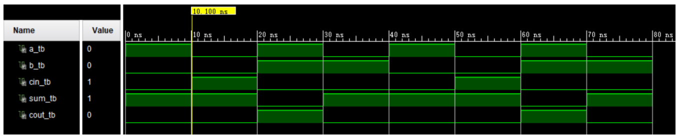


Fig. 9. Result of Post-Synthesis Functional Simulation

This time, the simulation result is not consistent with those shown in lecture notes but consistent with common sense. This is due to our setting time delay 10ns has been omitted and replace with delta delay. So we will not see a long delay in the waveform and the result looks like it’s real-time(Although there exits delta delay).

Then we generate the post-synthesis timing simulation waveform, we can see time delay of 2.576ns after each time period for Sum and Cout waveform. **This timing simulation considers the time delay for different kinds of gates(that is, the inertial delay) but not include the time delay for layout and wiring.** Generally, the result waveform is consistent with common sense except some delay. **Although there is some deviation from this timing simulation to the actual delay, it still has reference value.**

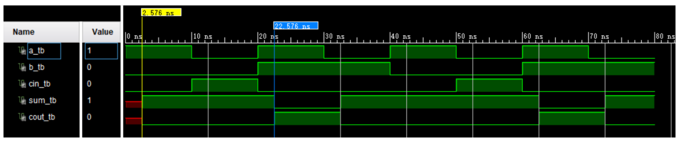


Fig. 10. Result of Post-Synthesis Timing Simulation

2.4 Post-Implementation

For post-implementation, there are also several important points:

1. **After the logic post-synthesis is passed, the adapter must be used to map the synthesized netlist file to a specific target device, including bottom device configuration, logic segmentation, logic optimization, wiring and operation. After the implementation is completed, the simulation file generated by the implementation can be used for accurate timing simulation.**

2. **The timing simulation results after implementation are close to the simulation of real device operation. The hardware characteristics of the device have been taken into account in the simulation process, so the simulation accuracy is much higher.**

It must be pointed out that different constraint files will produce different implementation results, and then generate different timing waveforms. This has been confirmed by my many experiments.

Alter implementation, we get the design and schematic:

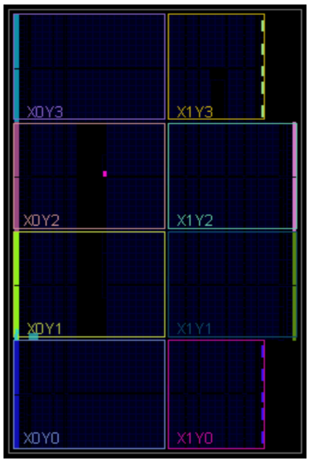


Fig. 11. Design

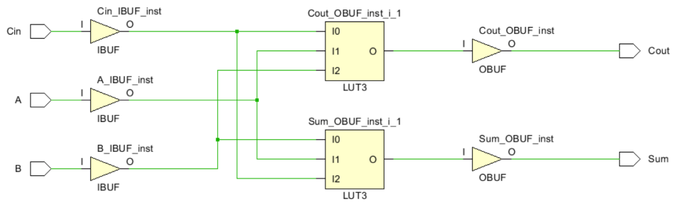
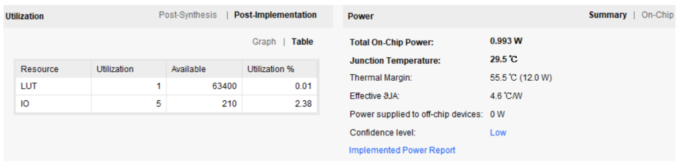


Fig. 12. Schematic

Also, our design summary is as follow:

The result of post-implementation functional simulation is as follow:

Also, the simulation result is not consistent with those shown in lecture notes but consistent with common sense and it is



Utilization | Post-Synthesis | Post-Implementation

Graph | Table

Resource	Utilization	Available	Utilization %
LUT	1	63400	0.01
IO	5	210	2.38

Power | Summary | On-Chip

Total On-Chip Power:	0.993 W
Junction Temperature:	29.5 ℃
Thermal Margin:	55.5 ℃ (12.0 W)
Effective θJA:	4.6 ℃/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Implemented Power Report

Fig. 13. Summary

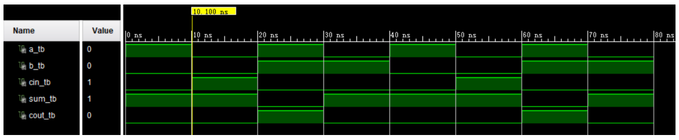


Fig. 14. Result of Post-Implementation Functional Simulation

basically the same as the functional simulation in post-synthesis. This means our design will work well functionally and achieve our full-adder goal.

However, when we get the result of post-implementation timing simulation, we find that it is a little terrible. There are many spikes in the waveform of Sum and Cout.

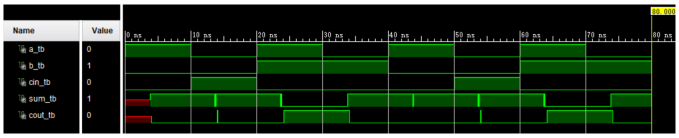


Fig. 15. Result of Post-Implementation Timing Simulation

It's not hard to accept this terrible result according to our discussion above. That is, the time delay considering the gate delay, layout and wiring delay according to our setting produces some conflict in signals values, which is called **Competitive Risk**. Circuit signals arrive at the gate in different order. **We can solve this problem by introducing blocking pulse, strobe pulse, filter capacitor or modifying the logical expression and adding redundancy.**

We enlarge the result and use some markers to record the time. We can simply infer the causes of some spikes:

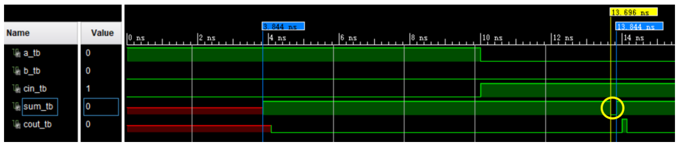


Fig. 16. Result of Post-Implementation Timing Simulation(local)

For the spike of Sum waveform near 14ns, one possible reason for it changes from '1' to '0' and quickly recovers to '1' is that after 10ns, the value of A and Cin are changed. Once the value of Cin changes from '0' to '1', and the value of s1 is still '1' at this time, the expression of $Sum <= (S1 \quad XOR \quad Cin)$ gives Sum the value of '0'. After s1 finishes the expression of $s1 <= (A \quad XOR \quad B)$, s1 changes value to '0' and Sum has the value of '1' again.

3 Conclusion

In this lab, we learned how to use VIVADO simulation for the simple digital system design——full-adder. We analyze the principle of full-adder and derive its truth table, logic expression and waveform.

The main part of this report is the process of VIVADO digital system design. For each step we give a detailed explained for its goal and result. Our discussion is mainly concern about the time delay in waveform and each simulation step has its own solution. We use the principle of each simulation step to explain the waveform result, together with our derived table and expressions.

To get a clear thinking for the later experiments, the first experiment is of great importance for we have a clear and detailed understanding of digital system design process with VIVADO and VHDL.

Some important points need to be emphasized and remembered again:

(1) Behavior simulation test the feasibility of our codes but not for any hardware system. Out setting delay can be used in this process.

(2) Post-synthesis converts software into hardware circuit. Post-implementation considers the place and route.

(3) Function simulation omit the delay value set by the assignment simulation instead with delta delay. Timing simulation for post-synthesis consider the inertial time delay for gate and timing simulation for post-implementation consider time delay in gate, combined with layout and wiring.

(4) For the Competitive Risk in circuit simulation waveform, we should use time constraints analysis and solve the problem before actual use.

All these experiments are to train our hardware design thinking and modular programming.