2-1. Example: Top-down design

Use two half adders to implement a full adder. This full adder contains

- 3 buttons as 3 individual inputs:
    - Use KEY1 in the development board as the 1st input, which is the addend.
    - Use KEY2 in the development board as the 2nd input, which is the augend.
    - Use KEY3 in the development board as the 3rd input, which is the carry-on bit from the lower bit.
- Use LED1 in the development board as the carry-on bit output.
- Use LED0 in the development board as the sum bit output.

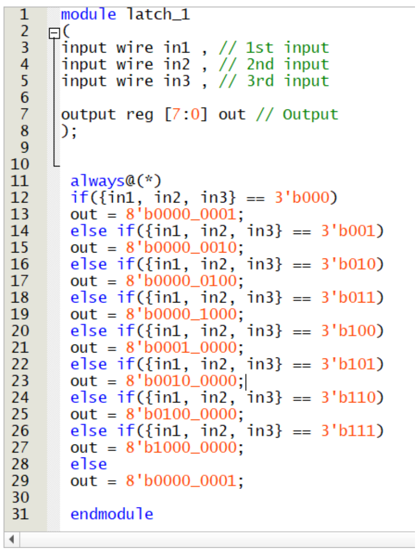
Requirements:

1. Write the Verilog codes to build the required circuit.
2. Do the simulation to verify whether your designs and codes satisfy the required functions.
3. Test your designs in the development board.
4. You do NOT need to include this example in your report.

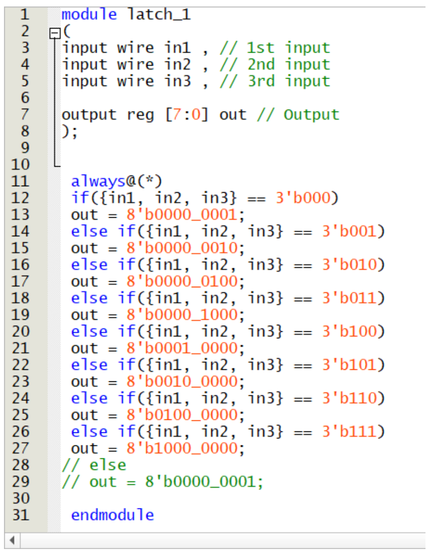
2-2. Avoid Latch

In this experiment, we several methods to implement 3-8 line decoder. The implementations are separated into following a, b, and c conditions. Each condition shows one possible reason of introducing latches. Compare different implementations to find the problems of using latches.

a) Use “if” condition with and without “else”:
Implement the 3-8 line decoder by using the following two codes respectively. Compare their differences of designs and simulation results.

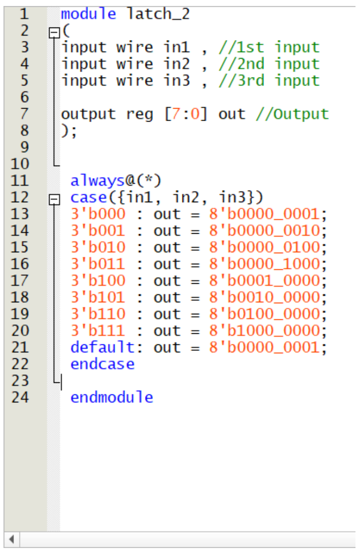


```
module latch_1
(
input wire in1 , // 1st input
input wire in2 , // 2nd input
input wire in3 , // 3rd input

output reg [7:0] out // Output
);

 always@(*)
 if({in1, in2, in3} == 3'b000)
 out = 8'b0000_0001;
 else if({in1, in2, in3} == 3'b001)
 out = 8'b0000_0010;
 else if({in1, in2, in3} == 3'b010)
 out = 8'b0000_0100;
 else if({in1, in2, in3} == 3'b011)
 out = 8'b0000_1000;
 else if({in1, in2, in3} == 3'b100)
 out = 8'b0001_0000;
 else if({in1, in2, in3} == 3'b101)
 out = 8'b0010_0000;
 else if({in1, in2, in3} == 3'b110)
 out = 8'b0100_0000;
 else if({in1, in2, in3} == 3'b111)
 out = 8'b1000_0000;
 else
 out = 8'b0000_0001;

 endmodule
```

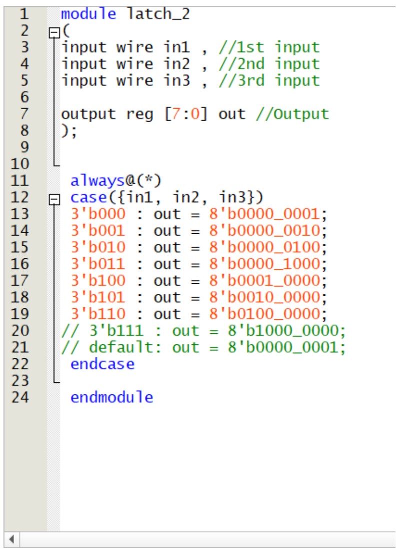


```
module latch_1
(
input wire in1 , // 1st input
input wire in2 , // 2nd input
input wire in3 , // 3rd input

output reg [7:0] out // Output
);

 always@(*)
 if({in1, in2, in3} == 3'b000)
 out = 8'b0000_0001;
 else if({in1, in2, in3} == 3'b001)
 out = 8'b0000_0010;
 else if({in1, in2, in3} == 3'b010)
 out = 8'b0000_0100;
 else if({in1, in2, in3} == 3'b011)
 out = 8'b0000_1000;
 else if({in1, in2, in3} == 3'b100)
 out = 8'b0001_0000;
 else if({in1, in2, in3} == 3'b101)
 out = 8'b0010_0000;
 else if({in1, in2, in3} == 3'b110)
 out = 8'b0100_0000;
 else if({in1, in2, in3} == 3'b111)
 out = 8'b1000_0000;
// else
// out = 8'b0000_0001;

 endmodule
```

b) Use “case” condition with and without full conditions:
Implement the 3-8 line decoder by using the following two codes respectively. Compare their differences of designs and simulation results.



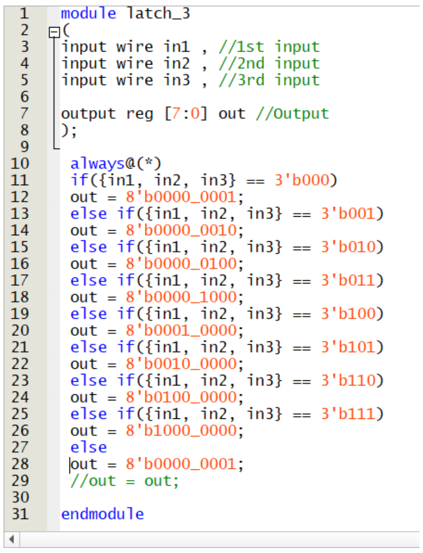
```
module latch_2
(
input wire in1 , //1st input
input wire in2 , //2nd input
input wire in3 , //3rd input

output reg [7:0] out //Output
);

 always@(*)
 case({in1, in2, in3})
 3'b000 : out = 8'b0000_0001;
 3'b001 : out = 8'b0000_0010;
 3'b010 : out = 8'b0000_0100;
 3'b011 : out = 8'b0000_1000;
 3'b100 : out = 8'b0001_0000;
 3'b101 : out = 8'b0010_0000;
 3'b110 : out = 8'b0100_0000;
 3'b111 : out = 8'b1000_0000;
 default: out = 8'b0000_0001;
 endcase

 endmodule
```



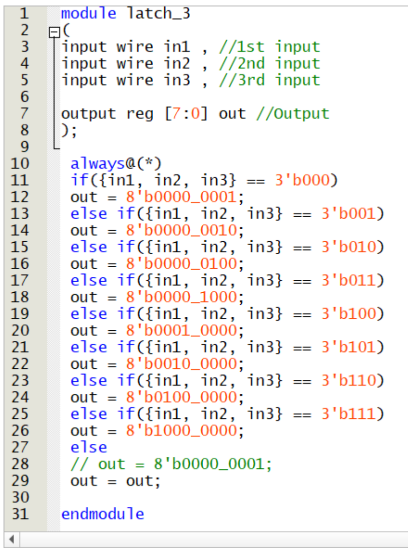
```
module latch_2
(
input wire in1 , //1st input
input wire in2 , //2nd input
input wire in3 , //3rd input

output reg [7:0] out //Output
);

 always@(*)
 case({in1, in2, in3})
 3'b000 : out = 8'b0000_0001;
 3'b001 : out = 8'b0000_0010;
 3'b010 : out = 8'b0000_0100;
 3'b011 : out = 8'b0000_1000;
 3'b100 : out = 8'b0001_0000;
 3'b101 : out = 8'b0010_0000;
 3'b110 : out = 8'b0100_0000;
// 3'b111 : out = 8'b1000_0000;
// default: out = 8'b0000_0001;
 endcase

 endmodule
```

c) Assign output variables to themselves in combinational logic:

Implement the 3-8 line decoder by using the following two codes respectively. Compare their differences of designs and simulation results.



```verilog
module latch_3
(
input wire in1 , //1st input
input wire in2 , //2nd input
input wire in3 , //3rd input

output reg [7:0] out //Output
);

 always@(*)
 if({in1, in2, in3} == 3'b000)
 out = 8'b0000_0001;
 else if({in1, in2, in3} == 3'b001)
 out = 8'b0000_0010;
 else if({in1, in2, in3} == 3'b010)
 out = 8'b0000_0100;
 else if({in1, in2, in3} == 3'b011)
 out = 8'b0000_1000;
 else if({in1, in2, in3} == 3'b100)
 out = 8'b0001_0000;
 else if({in1, in2, in3} == 3'b101)
 out = 8'b0010_0000;
 else if({in1, in2, in3} == 3'b110)
 out = 8'b0100_0000;
 else if({in1, in2, in3} == 3'b111)
 out = 8'b1000_0000;
 else
 out = 8'b0000_0001;
 //out = out;

endmodule
```



```verilog
module latch_3
(
input wire in1 , //1st input
input wire in2 , //2nd input
input wire in3 , //3rd input

output reg [7:0] out //Output
);

 always@(*)
 if({in1, in2, in3} == 3'b000)
 out = 8'b0000_0001;
 else if({in1, in2, in3} == 3'b001)
 out = 8'b0000_0010;
 else if({in1, in2, in3} == 3'b010)
 out = 8'b0000_0100;
 else if({in1, in2, in3} == 3'b011)
 out = 8'b0000_1000;
 else if({in1, in2, in3} == 3'b100)
 out = 8'b0001_0000;
 else if({in1, in2, in3} == 3'b101)
 out = 8'b0010_0000;
 else if({in1, in2, in3} == 3'b110)
 out = 8'b0100_0000;
 else if({in1, in2, in3} == 3'b111)
 out = 8'b1000_0000;
 else
 // out = 8'b0000_0001;
 out = out;

endmodule
```

Requirements:

1. Use the given Verilog codes to build the required circuit respectively. For each condition, compare the differences between 2 designs, especially their RTL views.
2. You need to include your comparisons and the discussions of latch in your report. In your report, you need to include:
   a) Introduction:
      i. Introduce the background. For example, you can introduce
         - What is latch,
         - Drawbacks of including latch in designs,
         - …
      ii. Purpose of the experiment
      iii. Contributions of your group members
      iv. …
   b) Materials and Methods: Briefly explain 2 designs in each condition.
   c) Results and Discussions: You need to point out
      i. Differences of 2 designs in each condition; especially their RTL views
      ii. Risks or problems of including latches;
      iii. …
   d) Conclusion. You need to

   i. Summarize all your comparisons,
   ii. Provide some ways of avoiding latches according to this experiment
   iii. …

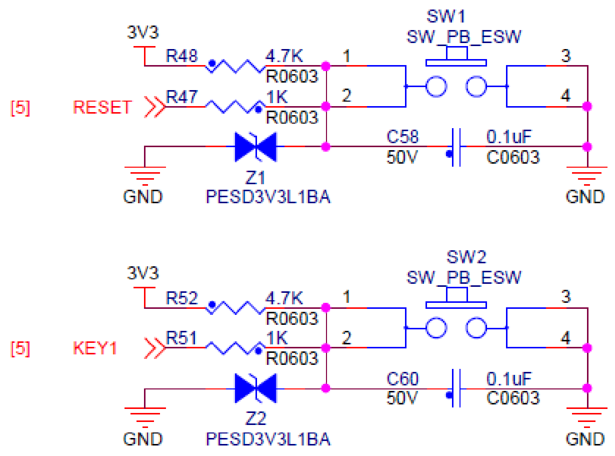
e) References. This is a list of all the sources cited in the report, formatted according to a specific citation style. You need to use the IEEE reference format (http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf).

f) Appendices. This optional section can include codes, raw data, calculations, additional graphs, and other supplementary material that is relevant but not essential to the main report.

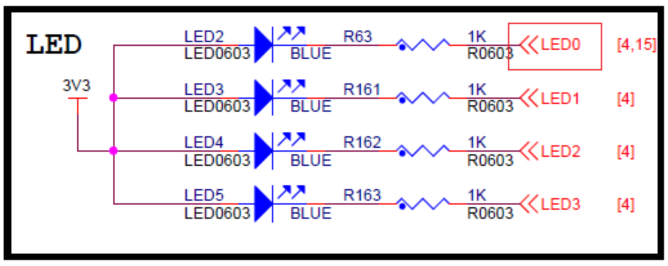
2-3. Compare flip-flops with synchronized and asynchronized reset

Implement a synchronized flip-flop and an asynchronized flip-flop. These flip-flops both contain

- Use REST KEY as the reset button
- Use KEY1 as the input button



- Use LED0 as the output



Note: Check page 27 and page 28 of "Lecture Slide 5: Hardware Description Language (HDL) - 1 Part 2" to find how to implement synchronized reset and asynchronized reset.

Requirements:

1. Write the Verilog codes to build the required circuit.
2. Do the simulation to verify whether your designs and codes satisfy the required functions.
3. Test your designs in the development board.
4. You need to include this experiment in your report. In your report, you need to include:
   a) Introduction:

i. Introduce the background. For example, you can introduce
   - What is the flip-flop,
   - Key functions of the flip-flop,
   - Importance of the flip-flop,
   - Key differences between the synchronized and asynchronized designs
   - ···

ii. Purpose of the experiment

iii. Brief introduce your designs and key results

iv. Contributions of your group members

v. ···

b) Materials and Methods: Show your FPGA solution, especially the RTL views. You need to show

i. Your designed structure block diagram,

ii. Your designed signal waveforms,

iii. Which devices or resources in this development board you used,

iv. ···

c) Results: The results should show the function of the flip-flop and the differences between the synchronized and asynchronized flip-flops. Show your simulation and corresponding results. You need to show

i. How you build the testbench,

ii. Testing inputs,

iii. Expected outputs,

iv. Actual outputs,

v. Differences between the expected outputs and the actual outputs,

vi. Expected values, actual values, and their differences of the important internal variables

vii. Test results in the development board

viii. ···

d) Discussions. This part is essential in this experiment. You need to discuss

i. Differences between synchronized and asynchronized flip-flops based on your RTL views and simulation results,

ii. ···

e) Conclusion. You need to

i. Summarize all your designs and simulation results,

ii. Summarize the key discussions

iii. ···

f) References. This is a list of all the sources cited in the report, formatted according to a specific citation style. You need to use the IEEE reference format (http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf).

g) Appendices. This optional section can include codes, raw data, calculations, additional graphs, and other supplementary material that is relevant but not essential to the main report.

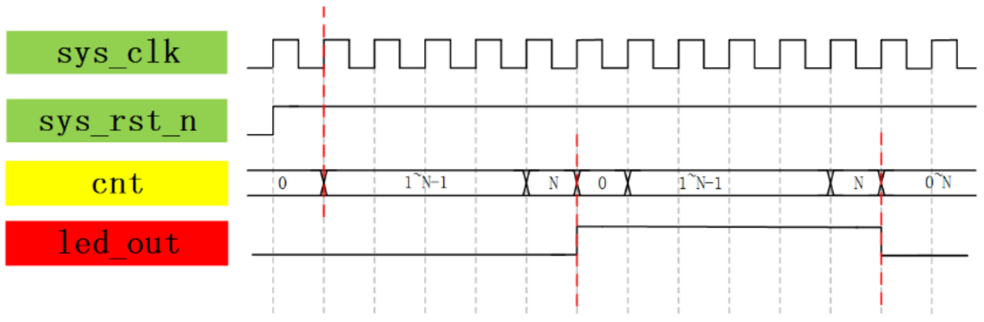
2-4. Counter

Implement a counter that controls a flashing LED light. This LED should flash at 1 Hz, which means that

a) At the beginning, the LED is on.
b) After 0.5 seconds, the LED is turned off
c) After 0.5 seconds, the LED is turned on
d) Repeat b) and c) steps.

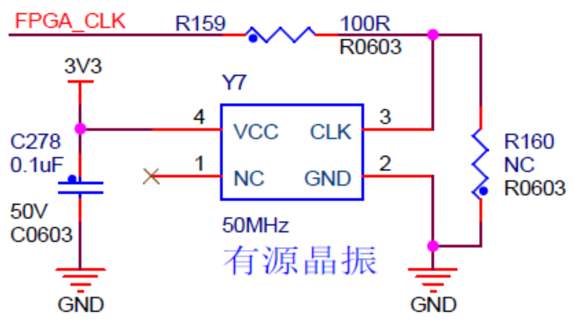
The timing diagram should be the following figure, where

- "sys_clk" is the system clock signal (you can directly use 50MHz active crystal oscillator for the system clock signal);
- "sys_rst_n" is the reset signal;
- "led_out" is the LED output signal;
- "cnt" is the essential internal variable, and you may add other internal variables in your report if you need them.

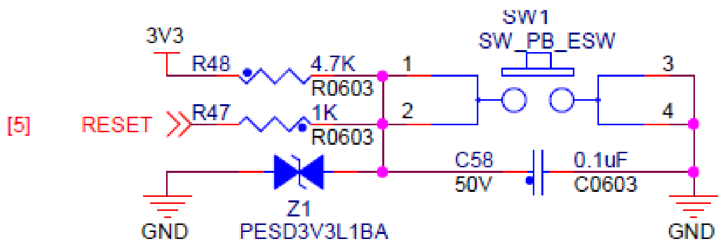


You need to

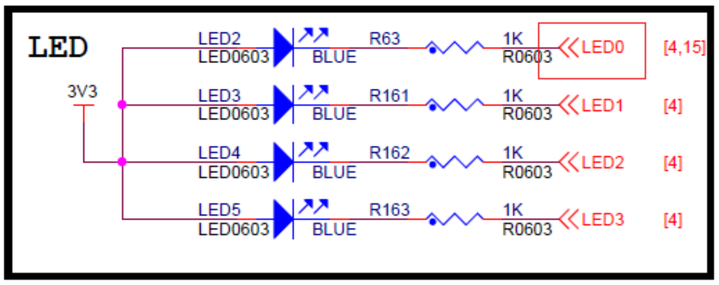
- Use 50MHz active crystal oscillator to generate the system clock and count the time



- Use RESET KEY in the development board as the reset button. When the reset button is pressed, the counter will be reset immediately.



- Use LED0 as the controlled LED.



Note:

- When you declare a variable. You need to declare its bit width. Otherwise, the compiler will use the default bit width (1 bit).

Requirements:

1. Write the Verilog codes to build the required circuit.
2. Do the simulation to verify whether your designs and codes satisfy the required functions.
3. Test your designs in the development board.
4. You need to include this experiment in your report. In your report, you need to include:
   a) Introduction:
      i. Purpose of the experiment
      ii. Brief introduce your designs and key results
      iii. Contributions of your group members
      iv. …
   b) Materials and Methods: Show your FPGA solution, especially the RTL views. You need to show
      i. Your designed structure block diagram,

ii. Your designed signal waveforms,

iii. Which devices or resources in this development board you used,

iv. …

c) Results: Show your simulation and corresponding testing results. You need to show

i. How you build the testbench,

ii. Testing inputs,

iii. Expected outputs,

iv. Actual outputs,

v. Differences between the expected outputs and the actual outputs,

vi. Expected values, actual values, and their differences of the important internal variables

vii. Test results in the development board

viii. …

a) Discussions. This part is not essential in this experiment, but you can show:

i. If you find any bugs or problems that you can not solve in the given time, you can discuss them and show your possible solutions.

ii. If you find any interesting results, you can discuss them.

iii. …

b) Conclusion. You need to

i. Summarize all your designs and simulation results,

ii. Summarize the key discussion results if you have,

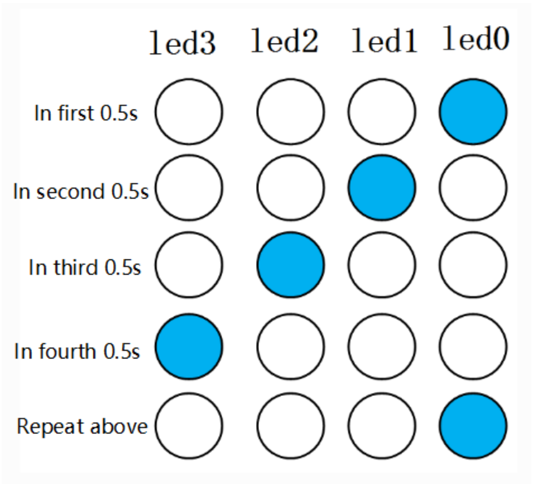
iii. …

d) References. This is a list of all the sources cited in the report, formatted according to a specific citation style. You need to use the IEEE reference format (http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf).

e) Appendices. This optional section can include codes, raw data, calculations, additional graphs, and other supplementary material that is relevant but not essential to the main report.

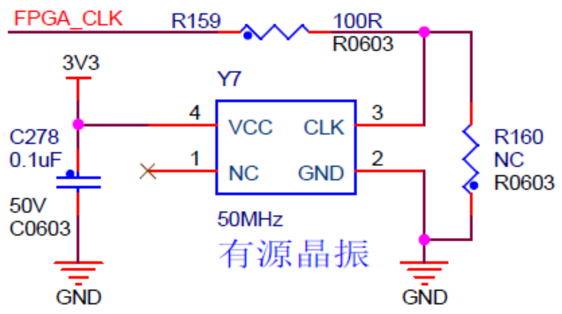
2-5. Running light

Implement a running light. The LED lights arranged in a row flash in turn, like "running water" and it keeps circulating. Our experiment this time is to make the LED lights flash in turn with an interval of 0.5s, that is, only one LED light is on at a time, and each time it is on for 0.5s. These LEDs should flash as following figure.

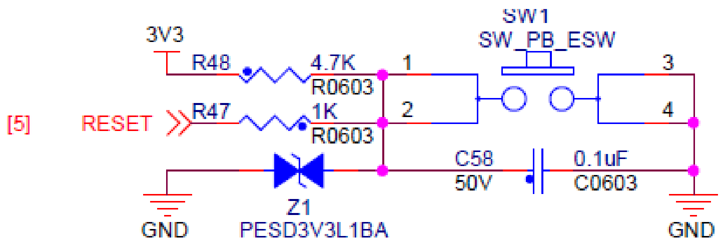


You need to

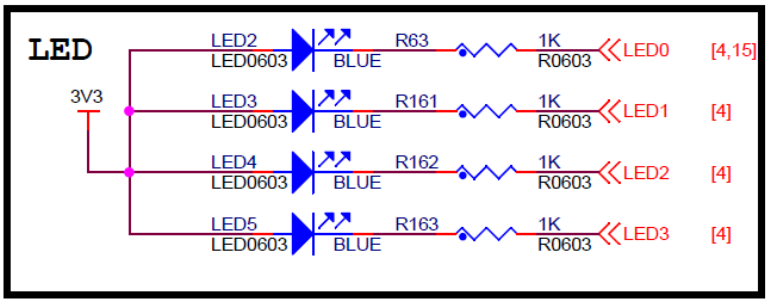
- Use 50MHz active crystal oscillator to generate the system clock and count the time



- Use RESET KEY in the development board as the reset button. When the reset button is pressed, the system will be reset immediately.



- Use LED0, LED1, LED2, and LED3 as the controlled LED lights.



Notes:

- You can use “<<” or “>>” to shift bits.
- This is also an opportunity to test the differences between shift (“<<” and “>>”) and arithmetic shift (“<<<” and “>>>”).

Requirements:

1. Write the Verilog codes to build the required circuit.
2. Do the simulation to verify whether your designs and codes satisfy the required functions.
3. Test your designs in the development board.
4. You need to include this experiment in your report. In your report, you need to include:
    a) Introduction:
        i. Purpose of the experiment
        ii. Brief introduce your designs and key results
        iii. Contributions of your group members
        iv. …
    b) Materials and Methods: Show your FPGA solution, especially the RTL views. You need to show
        i. Your designed structure block diagram,
        ii. Your designed signal waveforms,
        iii. Which devices or resources in this development board you used,
        iv. …
    c) Results: Show your simulation and corresponding testing results. You need to show
        i. How you build the testbench,
        ii. Testing inputs,

    iii. Expected outputs,
    iv. Actual outputs,
    v. Differences between the expected outputs and the actual outputs,
    vi. Expected values, actual values, and their differences of the important internal variables
    vii. Test results in the development board
    viii. …

c) Discussions. This part is not essential in this experiment, but you can show:
    i. If you find any bugs or problems that you can not solve in the given time, you can discuss them and show your possible solutions.
    ii. If you find any interesting results, you can discuss them.
    iii. …

d) Conclusion. You need to
    i. Summarize all your designs and simulation results,
    ii. Summarize the key discussion results if you have,
    iii. …

d) References. This is a list of all the sources cited in the report, formatted according to a specific citation style. You need to use the IEEE reference format (http://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf).

e) Appendices. This optional section can include codes, raw data, calculations, additional graphs, and other supplementary material that is relevant but not essential to the main report.