Lab 3 Report

作者

周延儒

digital logic design

In this lab , we has a progress in the Verilog , and we will acquire the skills of sequential circuits. Sequential circuit is quite different from combinational circuit. In Sequential circuit , the output depends on the not only the current input but also previous inputs, compare with the combinational circuit where the output depends just on the current input . Plus there are 2 types of the sequential circuit . Synchronous sequential and the asynchronous sequential.

In the synchronous sequential , all the signal is synchronized by a clock signal, and on the contrary , the asynchronous sequential, the signals are not synchronized by a clock signal. Fortunately , nearly all the sequential circuit is synchronous. Moreover, in the synchronous sequential , there are also 2 types , which are moore machine , and mealy machine. In moore machine, the output is determined by the current state . On the contrast, the output is determined by the current state and the input too.

In this lab , we are required to implement both of these sequential circuit.

The implement is the ABS system. The more detailed description is specified in the tutorial , so we don't explain the detail. '

So ,to implement the design, we have analyze the state diagram of each of the machine.

Here , we have 2 crucial inputs, timeIn, wheel, and a rst for reset the state. Of course , clk signal is required. And the output is the

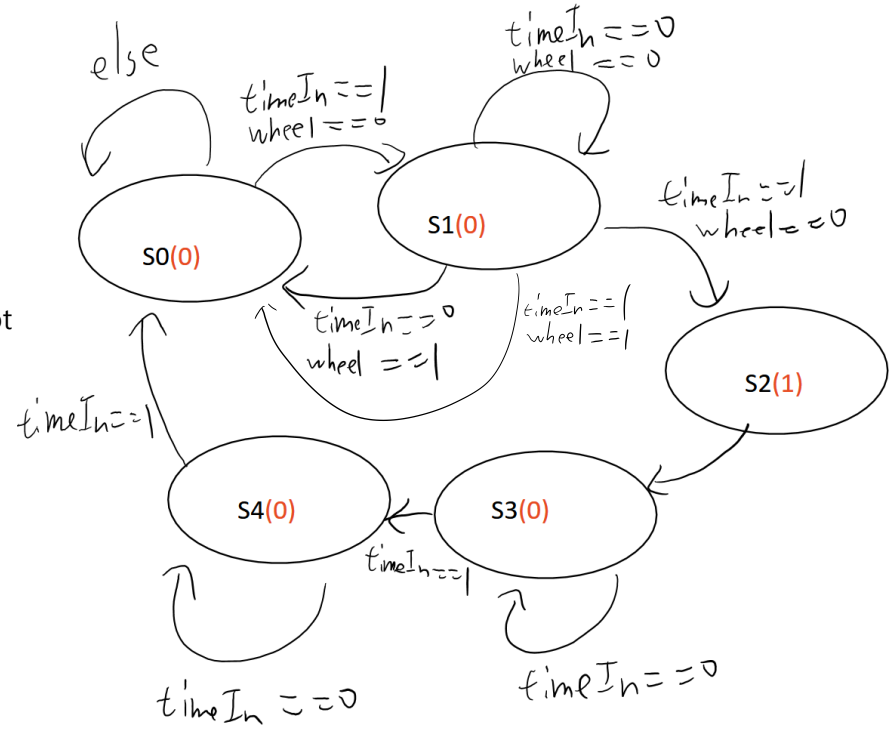
unlock signal. Noticed that we use asynchronous reset here. That is , we can reset the state at any given time. Compared with the asynchronous reset is the synchronous reset where we can only reset the state when the clock happens . The Verilog coding style is distinct ,and we will explain it later.

Here is the state digram:

Moore machine:

```
`define S0  5'b00001
`define S1  5'b00010
`define S2  5'b00100
`define S3  5'b01000
`define S4  5'b10000
```

5 states with one hot



Mealy machine:

parameter S0=4'b0001;

parameter S1 =4'b0010;

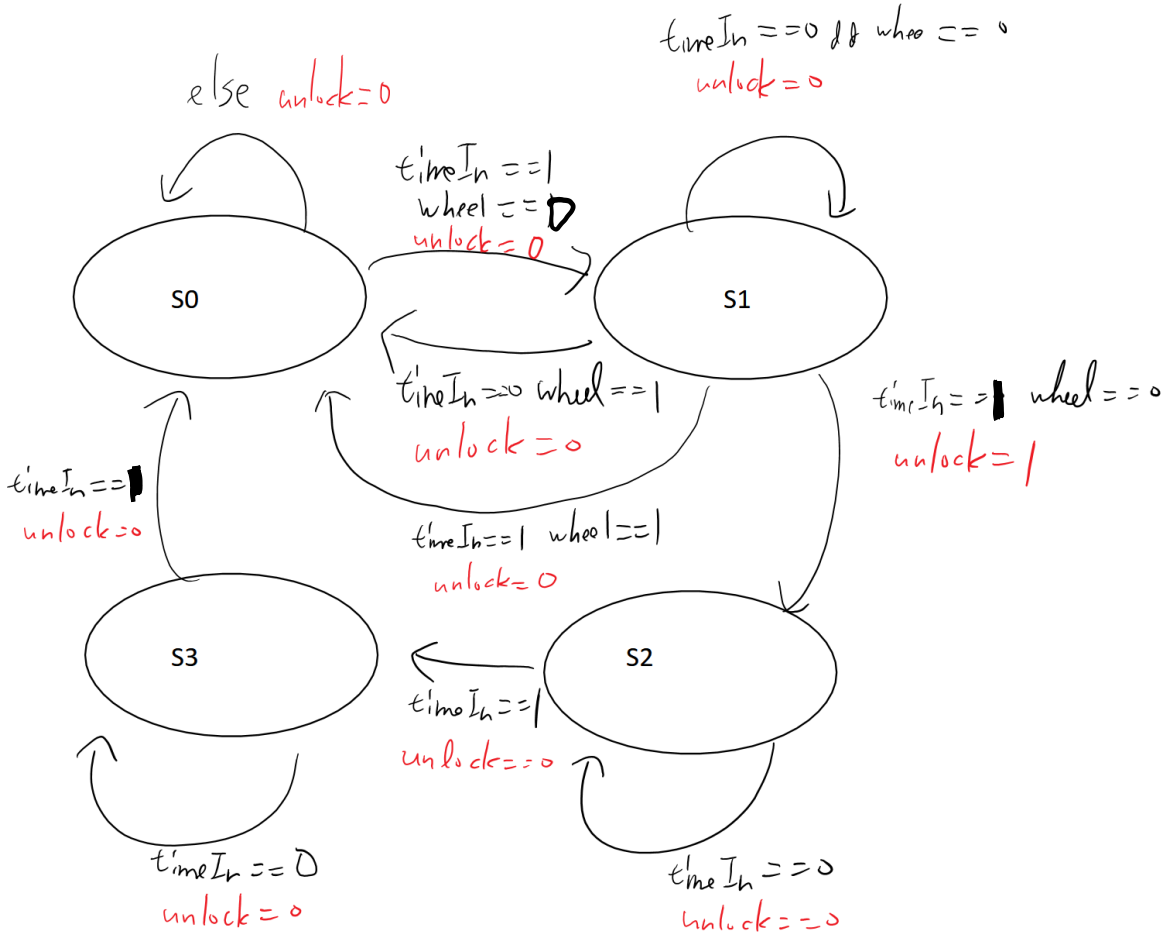
parameter S2 =4'b0011;

parameter S3 =4'b0100;

parameter S4=4'b0101;

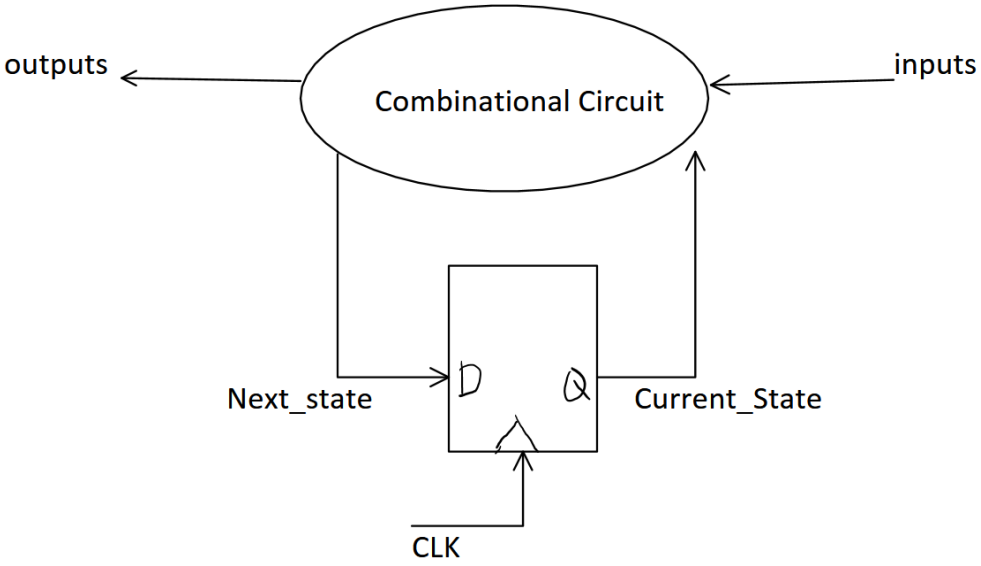
we define 5 states but 4 states is ok

binary form



After drawing the state diagram respectively, we can begin with the Verilog codes .

Since it's a finite state machine , overall analysis is following digram:



So we can divide the module into two parts: combinational and sequential
In sequential circuit ,we can write these in verilog

```
always @(posedge clk or posedge rst) begin
            if(rst)
           state <= `S0;
        else
           state <= next_state;

    end
```

whenever a clk happens, state will get the value of next state .

Moreover , we use the asynchronous reset so we write

```
always @(posedge clk or posedge rst) begin
            if(rst)
```

That means whenever the rst turns 1 , we can reset the state.

Additionally , Reset signal gets the highest priority.

Another way to reset the state is use synchronous rest ,like

```
always @(posedge clk ) begin
        if(rst)
```

These are the most common way to reset the signal.

	Synchronous Reset	Asynchronous Reset
clock	Reset at clock edges	Regardless of clock
Always block	Always @ (posedge clk	Always @ (posedge clk or nededge Reset)
Logic	Additional Mux or AND gate	Additional reset input port in DFF

The advantage of asynchronous reset is fast without any Mux or AND,and can reset anytime without any clock signal.

However, the disadvantages is that reset is vulnerable to glitches. In others words, whenever a glitch occurs, it may reset the signal ,doing damage to the system.

The synchronous reset can avoid the glitches ,but if the clk signal is too short, it is difficult to reset the state Even we can never reset the state in the worst case.

In my opinion , I prefer the synchronous reset to avoid the glitches.

Plus ,in the sequential circuit , the non—blocking assignment is required to complete the circuit. Compared with the blocking assignment in the combinational circuit where it execute in

sequence, it actually execute in parallel with the non-blocking assignment in the sequential circuit. Besides , with combinational circuit as well as the sequential circuit ,we can achieve RTL design ,which describes the behavior of combinational circuits between registers .

In combinational circuit, we can describe the behavior of the moore and mealy machine. And we can use statement. In each case, we determined the next_state . Based on the type of machine, we decide the output in each case. As what we have mentioned ,in the Moore machine , the output depends on the current state, And in the mealy machine ,the output depends on the inputs and current state.

There are 5,4 states in moore and mealy machine respectively.

After finishing the design , I found that we can simplify the design by adding a reg counter to count the time. Because after the unlock is 1, it wait for 2 times of timeIn signal , we can integrate the state which waits for detect 2 times after the unlock signals with the introduction of counter, which counts for just 2 times. Through this method, the state diagram could be simpler.

After implementing sequential and combinational parts of the module. The design is done.

In this lab, we acquire the technique of implementing the sequential circuit. Hope we can learn more in the following weeks.