

---

## Screenshots of Testbench output waveforms

In this section, I'll show you the procedure that I used to test my solution to show it works properly.

Firstly, we declare and generate the clock signal and initialize input signals for the project simulation as follow. The partial code for testbench is given below. The complete code for this part please check the file "Stimulus.v" in other folder.

```
//Declare clock cycle
parameter CYCLE = 2;
.....

//Generate and initialize the clock signal
initial begin
    clk = 0;
    forever
        #(CYCLE/2)
        clk=~clk;
end

//Initialize input signals for the simulation
initial begin
    #1    u2=1;
    #1    u2=0;
    #3    u3=1;
    #1    u3=0;
    #3    u1=1;
        d2=1;
```

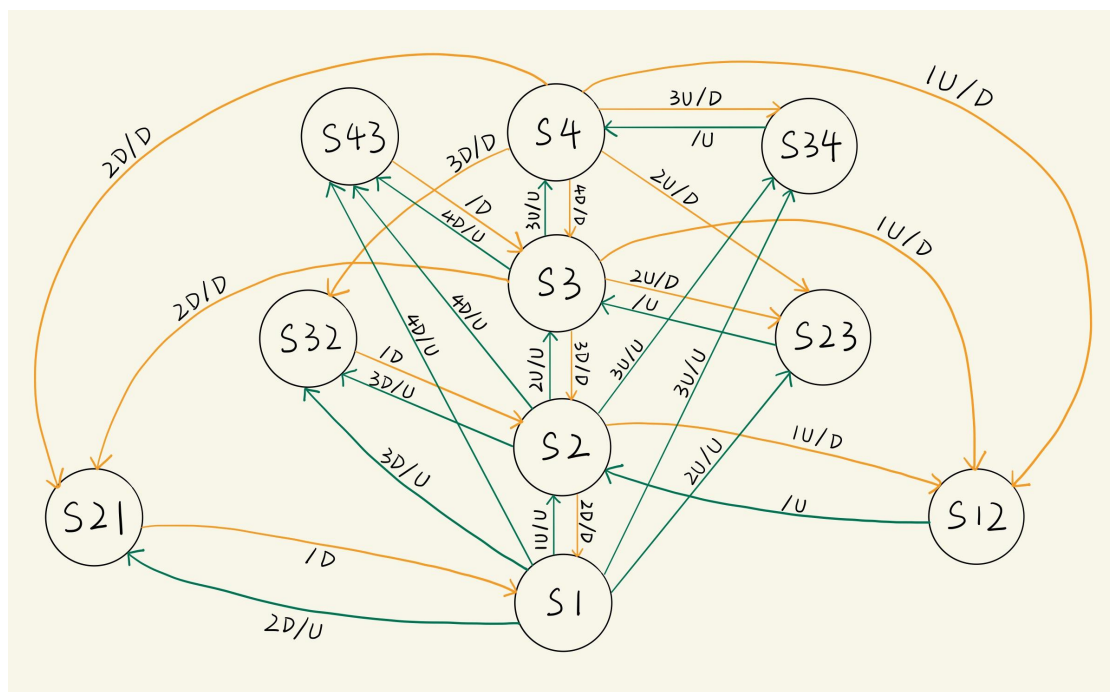
```

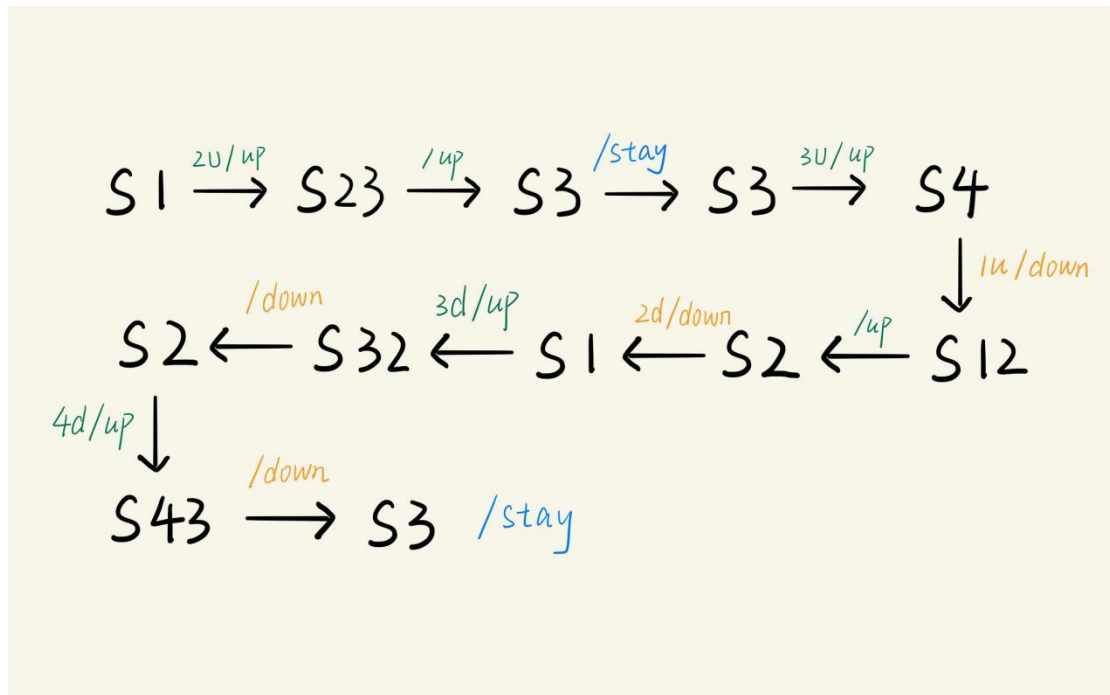
        d3=1;
#1      u1=0;
        d2=0;
        d3=0;
#1      d3=1;
#1      d3=0;
#3      d4=1;
#1      d4=0;

$display("Stimulus has already finished.");
end

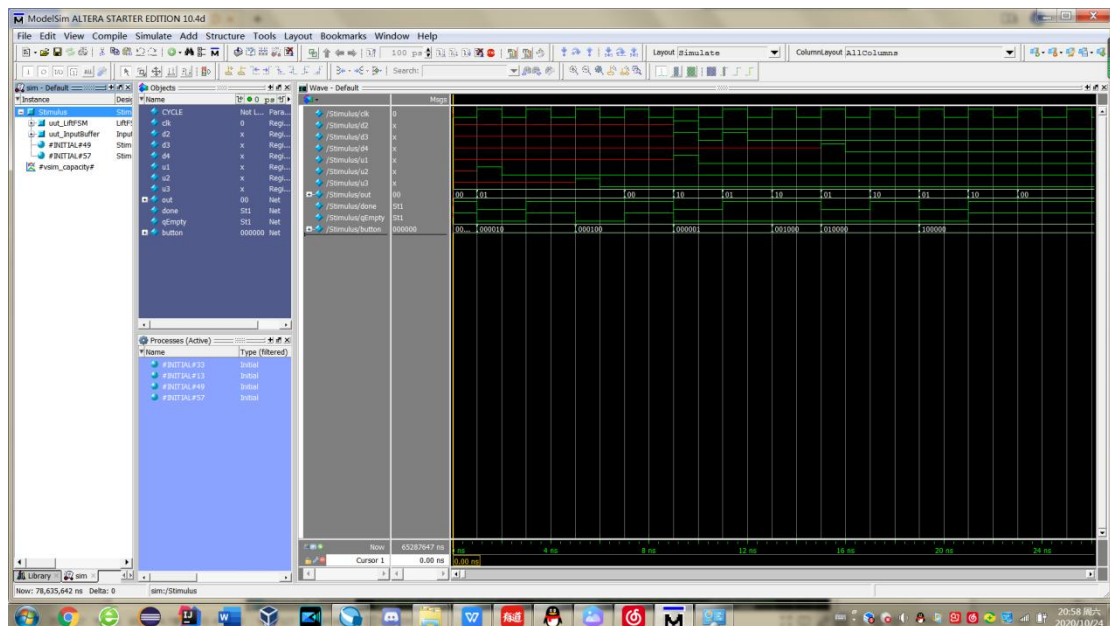
```

In this case, we can transfer the state according to the state diagram. The state transfer behavior of this FSM in every positive clock edge is shown as follow.





The screenshots of wave diagram is shown as follow:

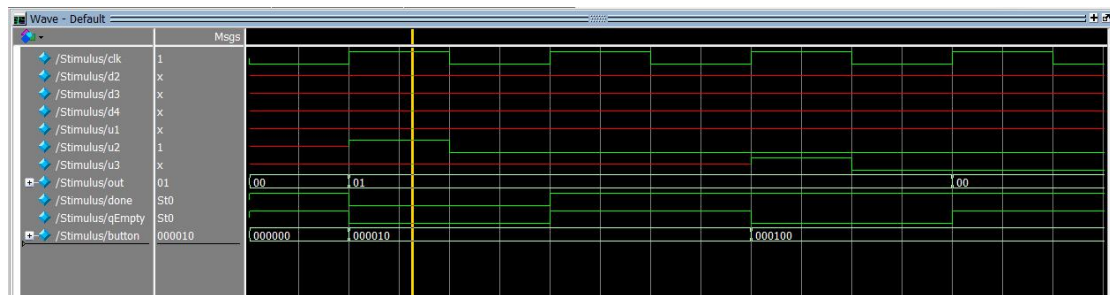


## At the 1st clk cycle: S1 to S23

Only button 2U sends input request.

Out indicates up and qEmpty is 0.

Since it moves to S23, done is 0.

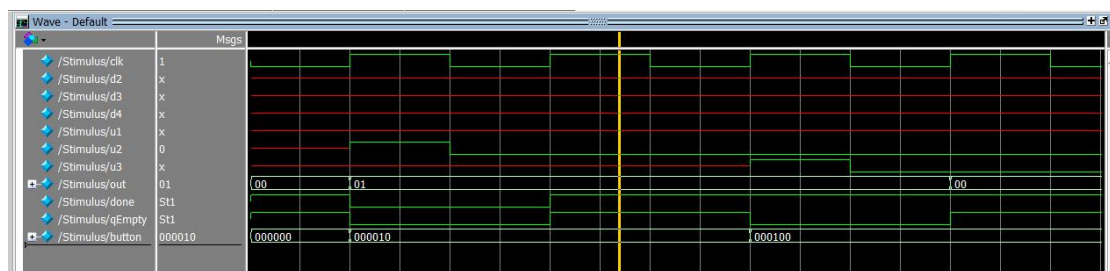


## At the 2nd clk cycle: S23 to S2

No button sends input request.

Out indicates up and qEmpty is 1.

Since it moves to S2, done is 1.



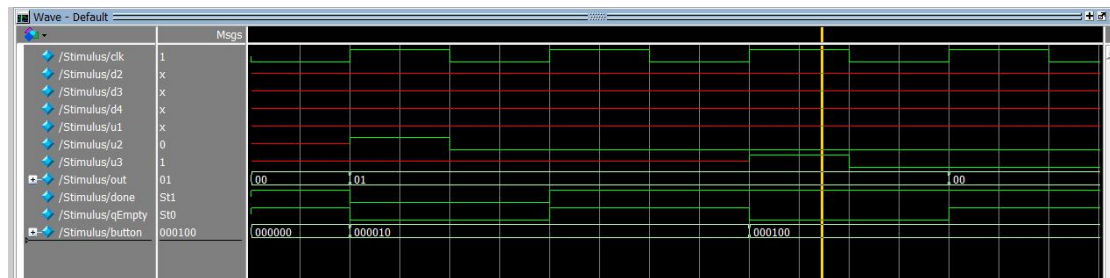
---

### At the 3rd clk cycle: S2 to S4

Only button 3U sends input request.

Out indicates up and qEmpty is 0.

Since it moves to S4, done is 1.

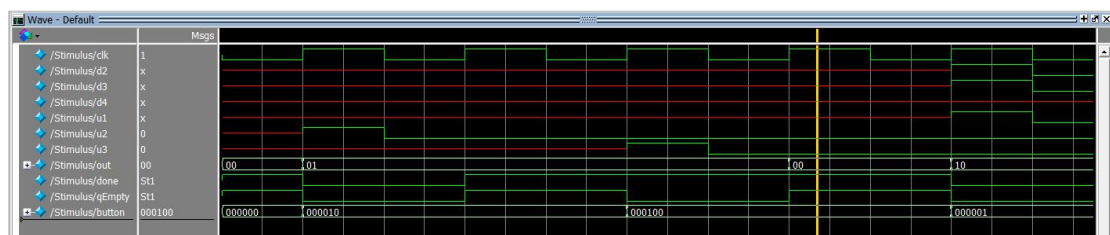


### At the 4th clk cycle: stay at S4

No button sends input request.

Out indicates stay and qEmpty is 1.

Since it stays at S4, done is 1.



---

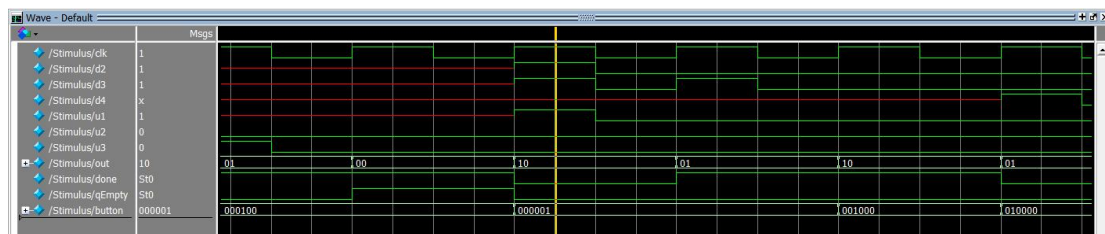
**At the 5th clk cycle: S4 to S12 (Succeed in implementing the order of priority)**

2D, 3D and 1U send input requests at the same time.

Considering the following order of priority: 1U, 2U, 3U, 2D, 3D, 4D, the input request from 1U is being processed at first, and input requests from 2D and 3D is waiting in the buffer in order.

Thus, out indicates down and qEmpty is 0.

Since it moves to at S12, done is 0.



**At the 6th clk cycle: S12 to S2 (Succeed in implementing the order of priority and ignoring the same request which is already in the buffer)**

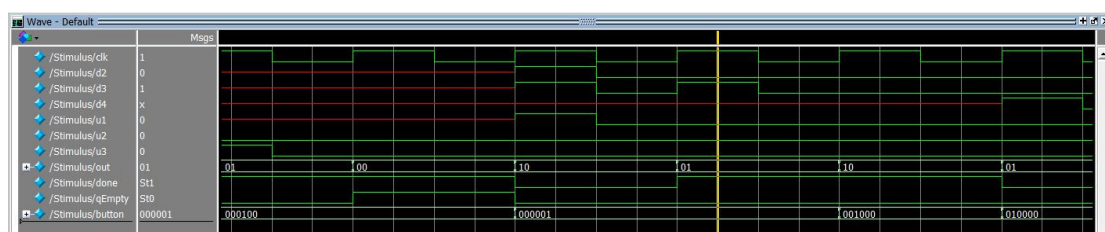
3D button sends input requests again.

Since we design a mechanism ignoring an input if the same request is already present in the buffer, this 3D input is invalid.

Input requests from 2D and 3D is waiting in the buffer in order.

Thus, out indicates down and qEmpty is 0.

Since it moves to at S2, done is 1.



---

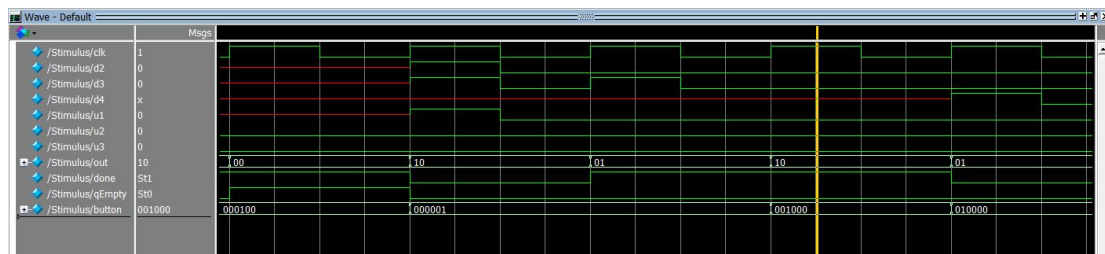
### At the 7th clk cycle: S2 to S1

No button sends input requests.

Considering the following order of priority: 1U, 2U, 3U, 2D, 3D, 4D, the input request from 2D is being processed at first, and input request from 3D is still waiting in the buffer.

Thus, out indicates down and qEmpty is 0.

Since it moves to at S1, done is 1.



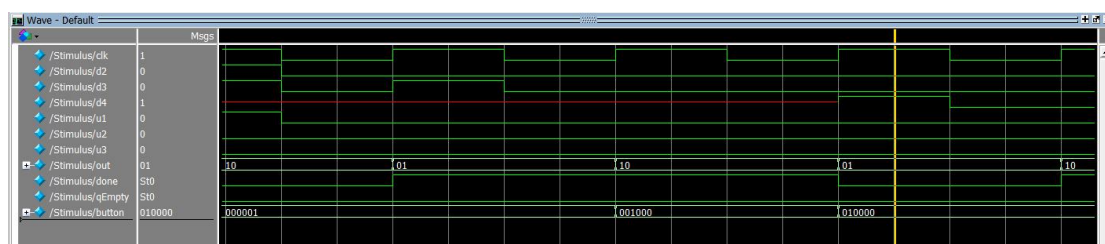
### At the 8th clk cycle: S1 to S32

4D button sends input request which is supposed to be placed in buffer in order.

Input request from 3D in the buffer can be processed in this clk cycle.

Thus, out indicates up and qEmpty is 0.

Since it moves to at S32, done is 0.



---

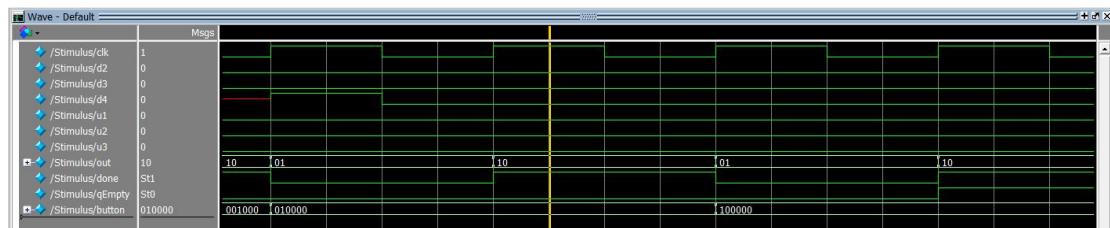
### At the 9th clk cycle: S32 to S2

No button sends input request.

Input request from 4D is still in the buffer.

Thus, out indicates down and qEmpty is 0.

Since it moves to at S2, done is 1.



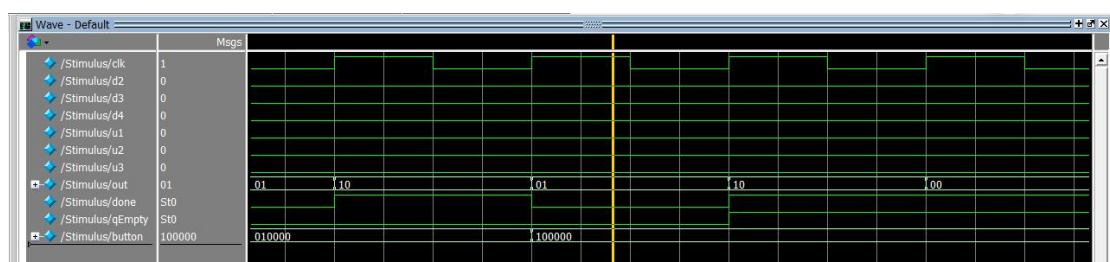
### At the 10th clk cycle: S2 to S43

No button sends input request.

Input request from 4D in the buffer can be processed in this clk cycle.

Thus, out indicates up and qEmpty is 0.

Since it moves to at S43, done is 0.





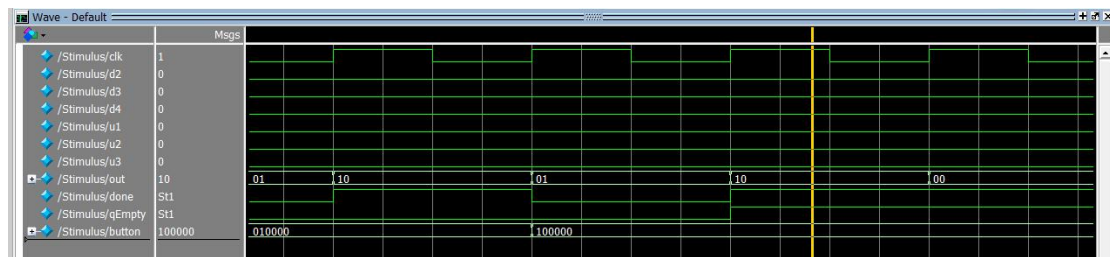
---

### At the 11st clk cycle: S43 to S3

No button sends input request.

Thus, out indicates down and qEmpty is 1.

Since it moves to at S3, done is 1.



### As for the following clk cycles: stay at S3

No button sends input request.

Also, qEmpty and done are both 1.

It keeps stay.