# Lab Report2

## Zhenlei Song 826009388

## J-K Flip Flop

a)structural Verilog

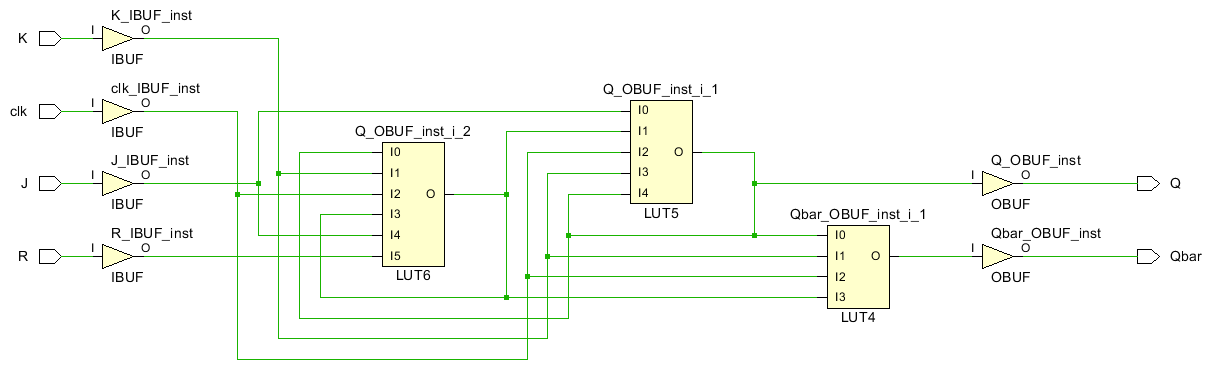


Fig1.1 J-K Flip Flop Structure Schematic

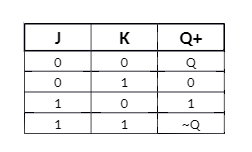


Fig 1.2 Truth Table

Output:

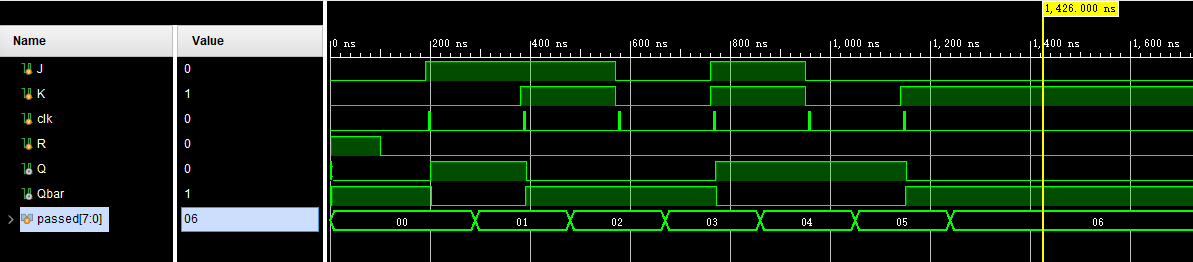


Fig 1.3 Waveform

# run 1000ns

0 Reset=1, J=0, K=0, Q=x

4 Reset=1, J=0, K=0, Q=0

100 Reset=0, J=0, K=0, Q=0

190 Reset=0, J=1, K=0, Q=0

201 Reset=0, J=1, K=0, Q=1

Set passed

380 Reset=0, J=1, K=1, Q=1

393 Reset=0, J=1, K=1, Q=0

Toggle 1 passed

570 Reset=0, J=0, K=0, Q=0

Hold 1 passed

760 Reset=0, J=1, K=1, Q=0

771 Reset=0, J=1, K=1, Q=1

Toggle 2 passed

950 Reset=0, J=0, K=0, Q=1

INFO: [USF-XSim-96] XSim completed. Design snapshot 'JK\_FF\_tb\_behav' loaded.

INFO: [USF-XSim-97] XSim simulation ran for 1000ns

run 20 us

Hold 2 passed

1140 Reset=0, J=0, K=1, Q=1

1153 Reset=0, J=0, K=1, Q=0

Reset passed

All tests passed

b)behavioral Verilog

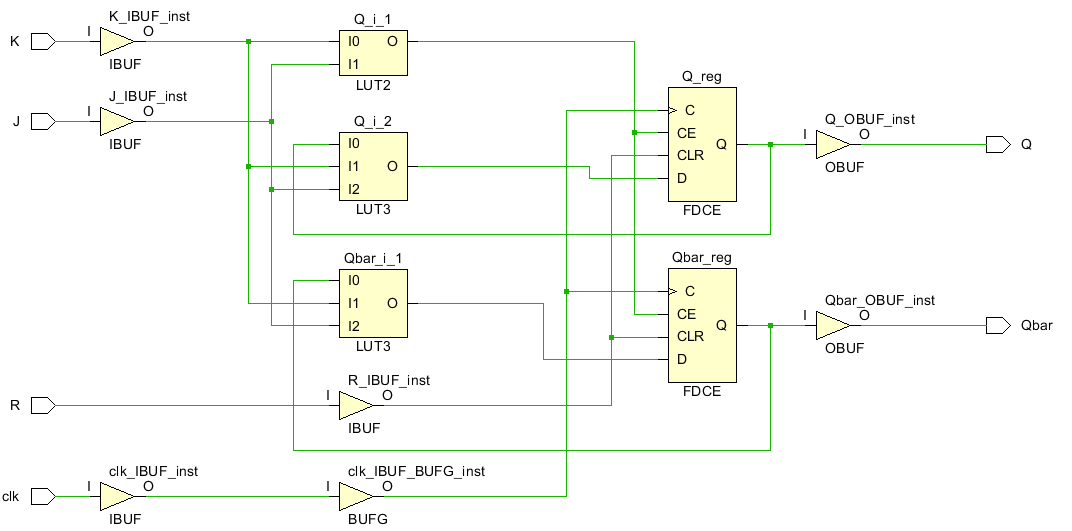


Fig 1.4 J-K Flip Flop Behavioral Schematic

Output:

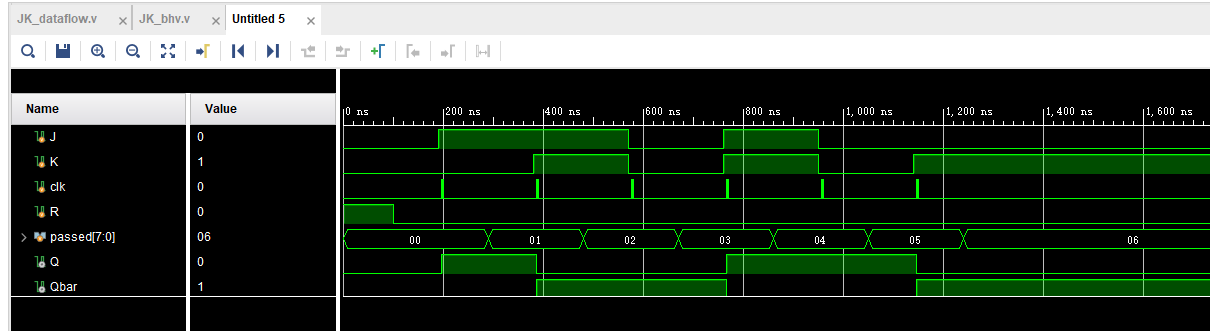


Fig 1.5 J-K Flip Flop Waveform

# run 1000ns

0 Reset=1, J=0, K=0, Q=0

100 Reset=0, J=0, K=0, Q=0

190 Reset=0, J=1, K=0, Q=0

197 Reset=0, J=1, K=0, Q=1

Set passed

380 Reset=0, J=1, K=1, Q=1

387 Reset=0, J=1, K=1, Q=0

Toggle 1 passed

570 Reset=0, J=0, K=0, Q=0

Hold 1 passed

760 Reset=0, J=1, K=1, Q=0

767 Reset=0, J=1, K=1, Q=1

Toggle 2 passed

950 Reset=0, J=0, K=0, Q=1

INFO: [USF-XSim-96] XSim completed. Design snapshot 'JK\_FF\_tb\_behav' loaded.

INFO: [USF-XSim-97] XSim simulation ran for 1000ns

launch\_simulation: Time (s): cpu = 00:00:02 ; elapsed = 00:00:06 . Memory (MB): peak = 2091.500 ; gain = 0.000

run 20 us

Hold 2 passed

1140 Reset=0, J=0, K=1, Q=1

1147 Reset=0, J=0, K=1, Q=0

Reset passed

All tests passed

### D Flip Flop

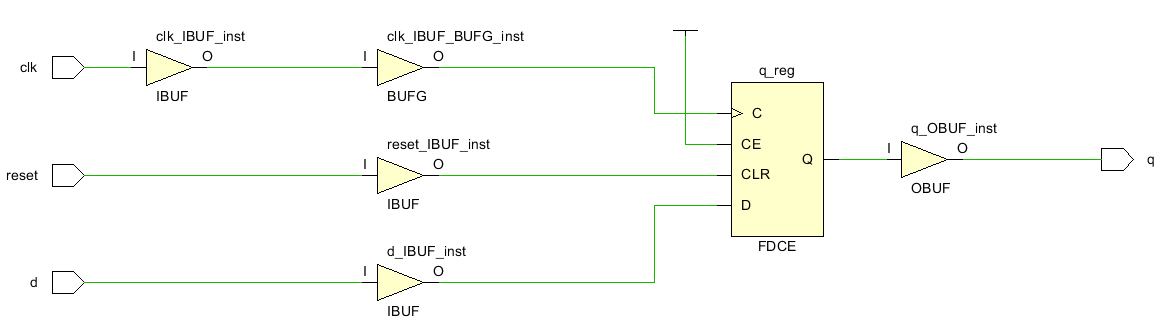


Fig 2.1 D Flip Flop Behavioral Schematic

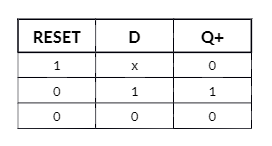


Fig 2.2 D Flip Flop Truth Table

Output:

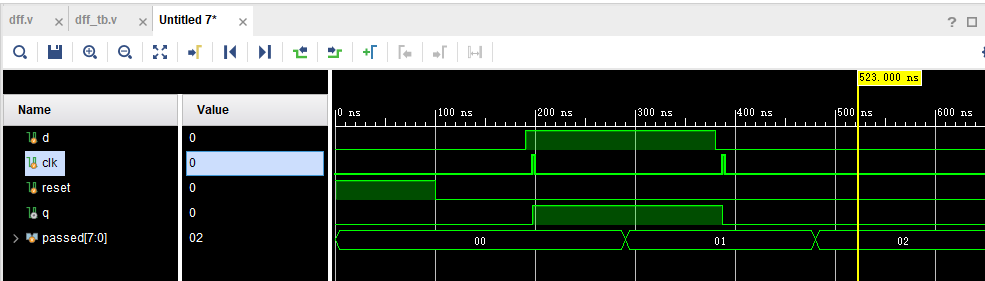


Fig 2.3 D Flip Flop Waveform

# run 1000ns

0 D=0, Reset=1, Q=0

100 D=0, Reset=0, Q=0

190 D=1, Reset=0, Q=0

197 D=1, Reset=0, Q=1

Output = Input passed

380 D=0, Reset=0, Q=1

387 D=0, Reset=0, Q=0

Output = 0 passed

All tests passed

INFO: [USF-XSim-96] XSim completed. Design snapshot 'dff\_tb\_behav' loaded.

INFO: [USF-XSim-97] XSim simulation ran for 1000ns

### 2-4 Decoder

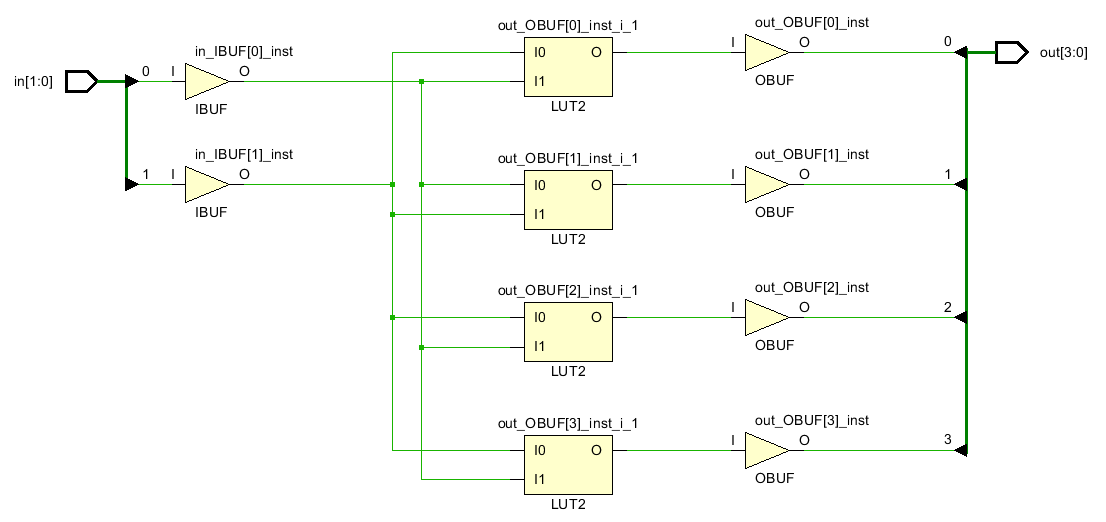


Fig 3.1 2-4 Decoder Schematic

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enable | In[1] | In[0] | Output[3] | Output[2] | Output[1] | Output[0] |
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Fig 3.2 Truth Table

Output:

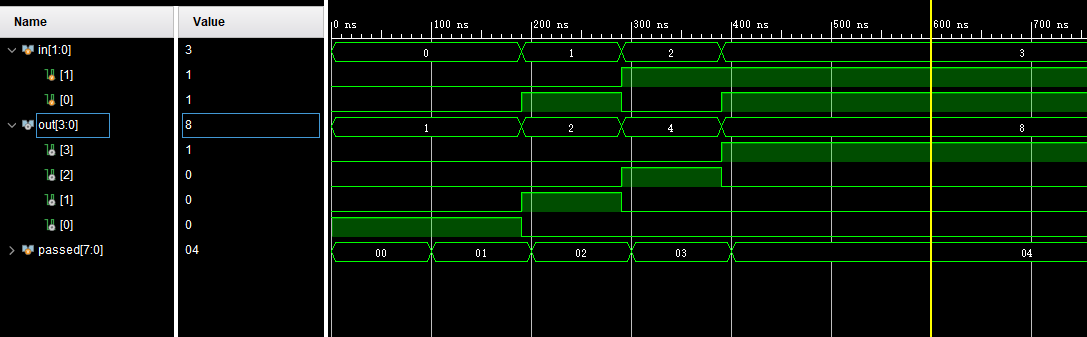


Fig 3.3 2-4 Decoder Waveform

# run 1000ns

0 enable=0, in=00, out=0000

50 enable=1, in=00, out=0001

Input0 passed

240 enable=1, in=01, out=0010

Input1 passed

340 enable=1, in=10, out=0100

Input2 passed

440 enable=1, in=11, out=1000

Input3 passed

All tests passed

INFO: [USF-XSim-96] XSim completed. Design snapshot 'decode2\_4\_tb\_behav' loaded.

INFO: [USF-XSim-97] XSim simulation ran for 1000ns

launch\_simulation: Time (s): cpu = 00:00:02 ; elapsed = 00:00:05 . Memory (MB): peak = 2091.500 ; gain = 0.000

### Questions

1. **In behavioral Verilog, two types of assignment statements exists. What are they, and when would you use one over the other?**

There are blocking assignments and non-blocking assignments. When doing combinational logic circuit, blocking assignments are ideal. In sequential logic circuits, non-blocking assignments are better.

1. **Compare and contrast the structural verses behavioral implementation of the JK flip-flop. Which level of abstraction might you use for a processor design and why?**

Structural implementation consists of basic gate circuits. While in behavioral implementation, higher level logic components are used. For design of a processor, behavioral implementation is better, because lower level hardware has no need to be shown in the design of processors. Plus it’s easier to test functions in behavioral choices.

1. **Based on the gate level diagram you created for the 2:4 decoder, how would the structural implementation compare to the dataflow implementation? Could you use behavioral Verilog to create the 2:4 decoder? If so, provide a snippet of code to do so.**

In dataflow implementation, circuits are designed from truth table, the logic is based on how value is transformed from inputs to outputs. While in structural design, the idea is about how gates are connected.

Yes.Below is the code.

module decode2\_4\_beh(

out, in, enable

);

input [1:0]in;

input enable;

output [3:0]out;

reg [3:0]out;

always @\*

begin

if(~enable)

begin

out <= 0;

end

else if(in == 00)

begin

out <= 1;

end

else if(in == 01)

begin

out <= 2;

end

else if(in == 10)

begin

out <= 4;

end

else if(in == 11)

begin

out <= 8;

end

end

endmodule