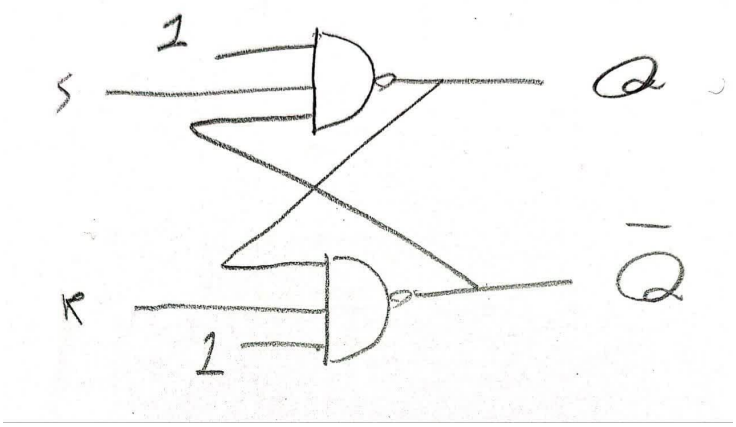
# Question 1)

## Part a)

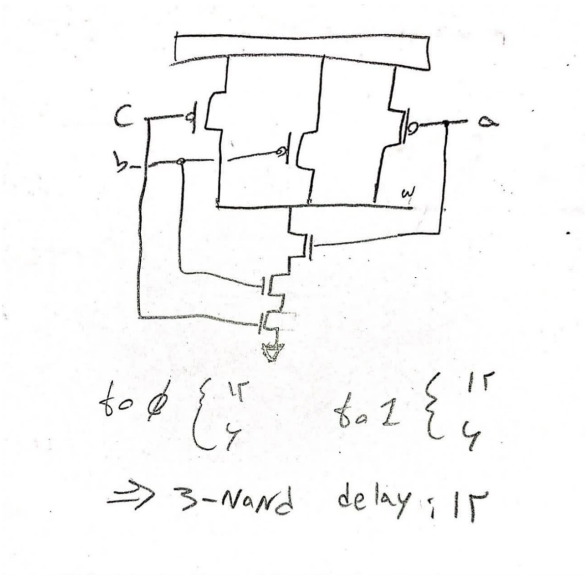
At first take look at the shape of circuit:




And the 1 inputs of this circuit can be replaced with any inputs, to make it a circuit with four inputs. It's four inputs will be used further in question 3. The Verilog is in q1>a>a.sv.

## Part b)

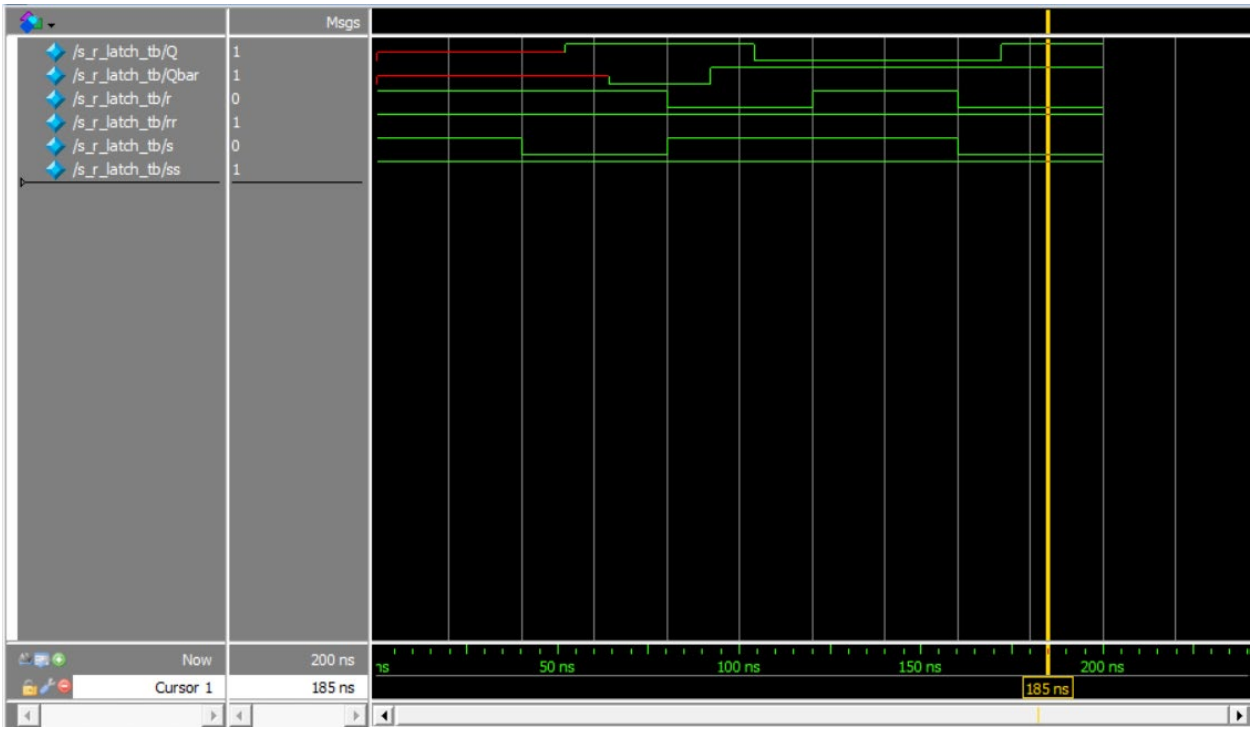
Here is the shape of nand gate and its timing:



The worst case delay of s r latch will be two 3 input nands which will be 24 ns. The Verilog code is in q1>b>b.sv

Part c)

Here are the wave forms. As you can see the q and qbar won't take a true value after s and r are simultaneously 0 and the memory will be lost (as the inputs are active low a 1 and 1 will hold the previous value). As you can see the worst case delay is 24 ns:




Another thing to note is that ss and rr are 1 for now because we didn't need them, but when we do, they will take value.

Question 2)

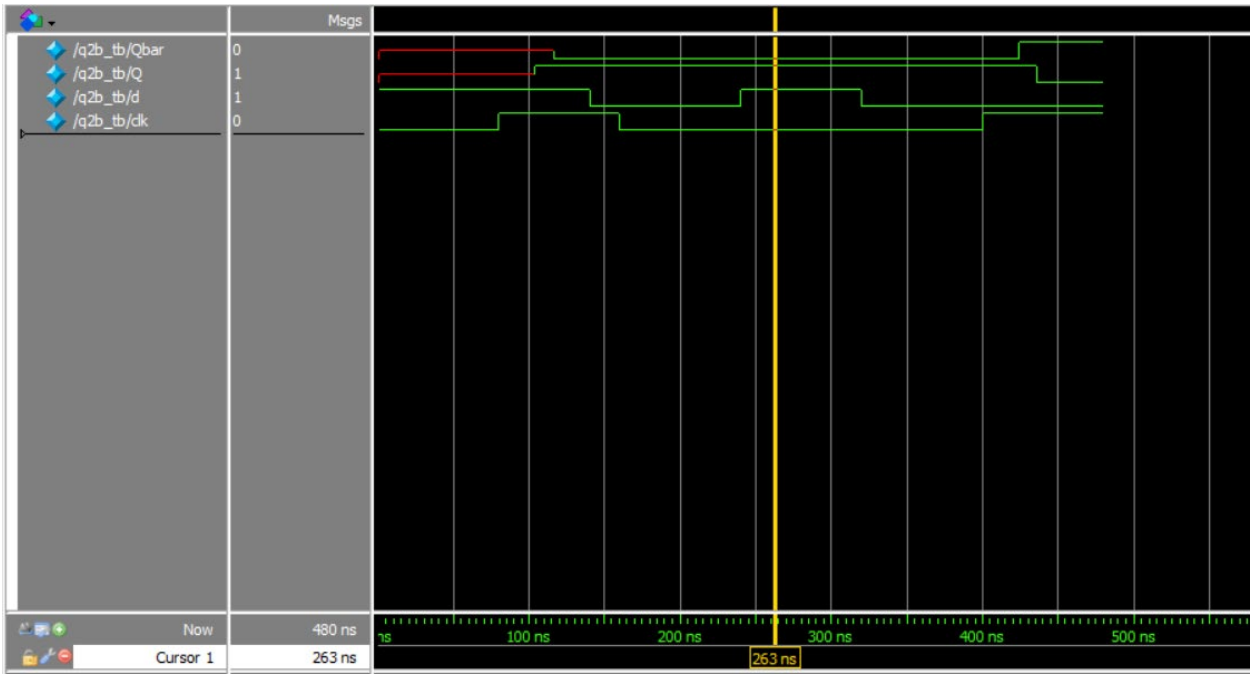
Shapes of these two questions are as homework.

Part a)

The Verilog code is in q2>a>a.sv.

Part b)

Here's the timing diagram, as the inputs of two clk nand gates are ready, worst case delay will be 12 + 24 seconds. the change in the output only happens when the clock is in rising edge, and in other times changing d or clock won't cause any changes on the output: (result is as we expected, so the expected diagram is the same)

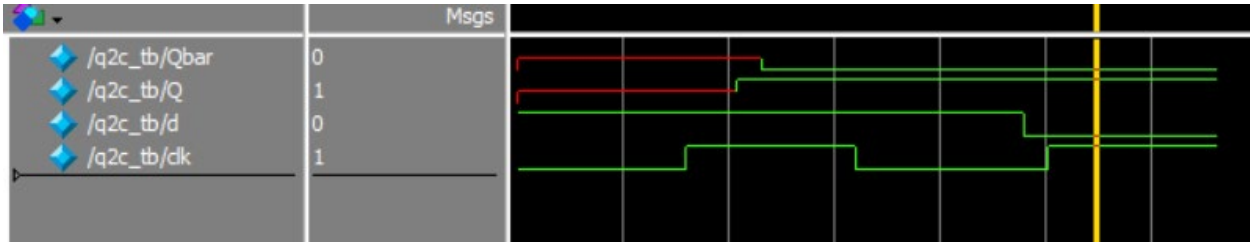



Part c)

As q and q bar are functions of outputs of g2 and g3, the inputs of these two gates are important to us. These two

Gates inputs will completely be ready after g4 and then g1 propegate, which will take two 3-input gates, which will be 24 ns as the worst case delay.

But modelsim does not act as we expect. Gates durations are like this: a gate's inputs change>it waits for the delay time>when the delay time is over, it opens up it's eyes and takes the inputs and does the operation, REGARDLESS of how many times inputs have changed during it's waiting time. So it will reduce one gate's duration for us, which will make the t setup more than 12 for modelsim:

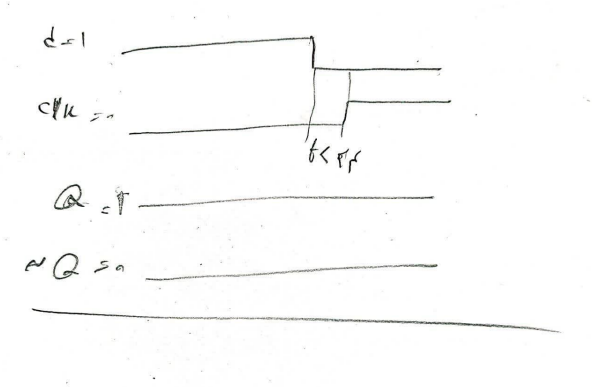



t setup is 11: nothing happens.


t setup is 12: we take place in an unknown state.

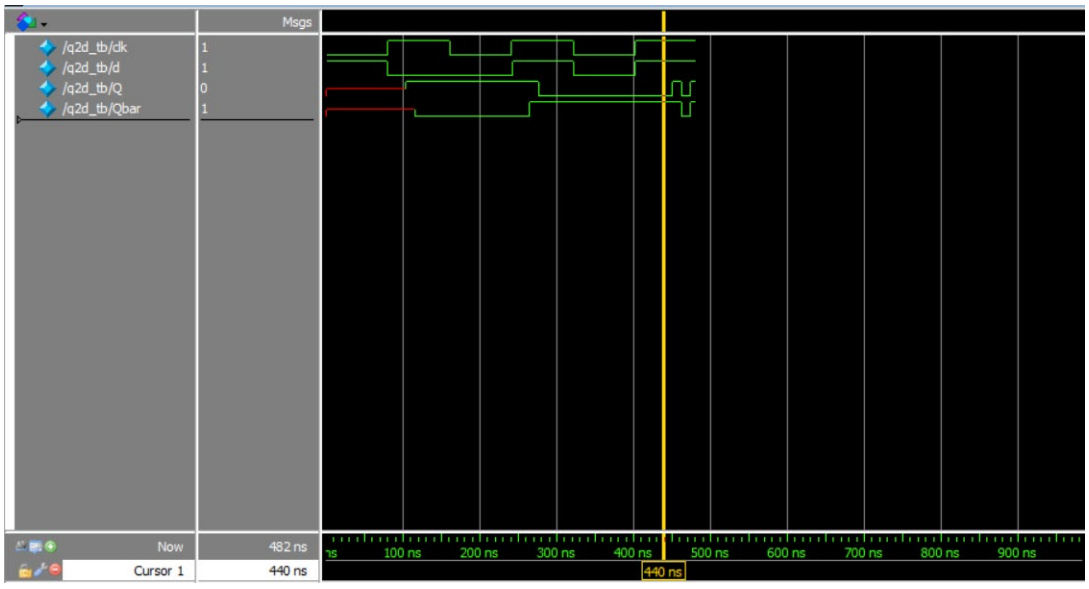

t setup is 13: we successfully see the output change.

Though we expected to see this waveform:



Part d)

When clk goes from 0 to 1, g2 and g3 gonna need time to propagate. When d is 0, g3 must change from 1 to 0. When d is 1, g2 needs to change from 1 to 0. In clk's falling edge, one of g2 or g3 must turn from 0 to 1. So overall we have to wait for only one gate to propagate which will be 12 in this case. Now the real problem is g4 needs 12 ns to propagate, so they are exactly the same, and as being the same might put us in unwanted states, we will choose t hold the minimum possible value, which will be 1 in this case:



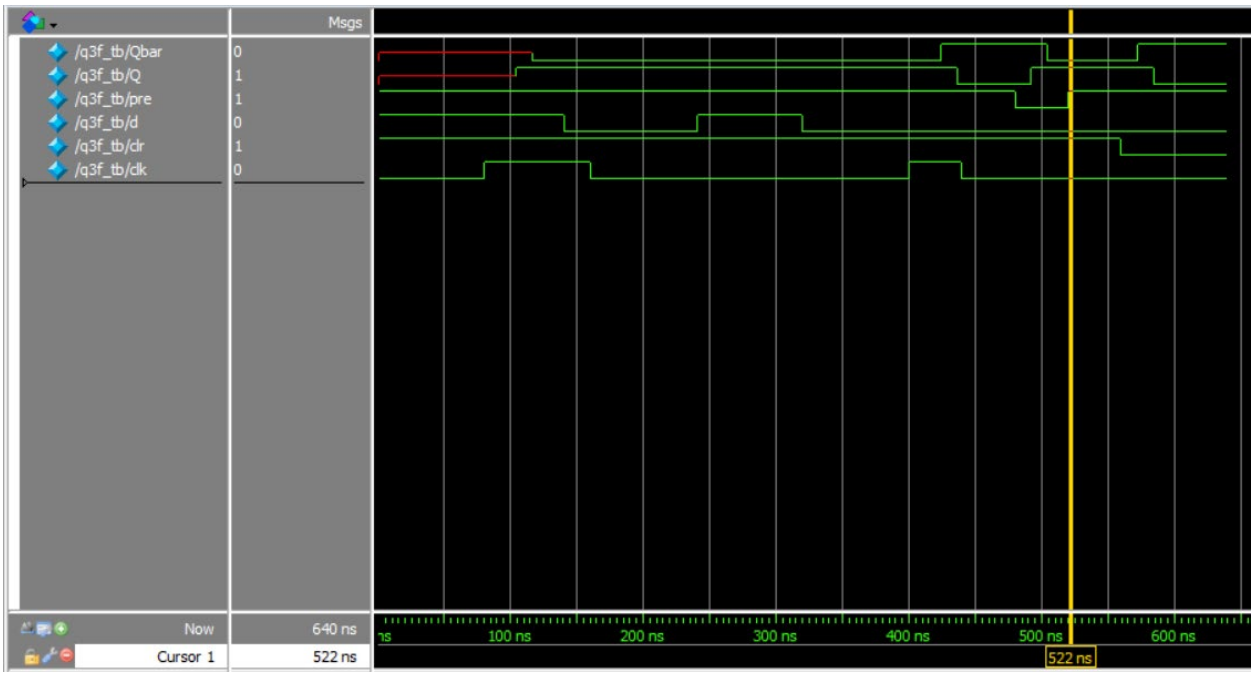
As you can see, as long as we put 1 ns for t hold everything is fine, but when we put it away and change d from 0 to 1, the circuit will take place in an unknown state. Result is as we expected so we don’t show the expected diagram.

Question 3)

Part e) the Verilog code is available in q3>e>e.sv. one thing to note is that pre and clr are active low.

Part f)

Timings and delays are not different for clocking as you can see in the waveforms. For clr and pre, the direct gate of each will take 12 ns and the undirect gate will take 24 ns to propagate, the direct gate for pre is q and for clr is qbar:

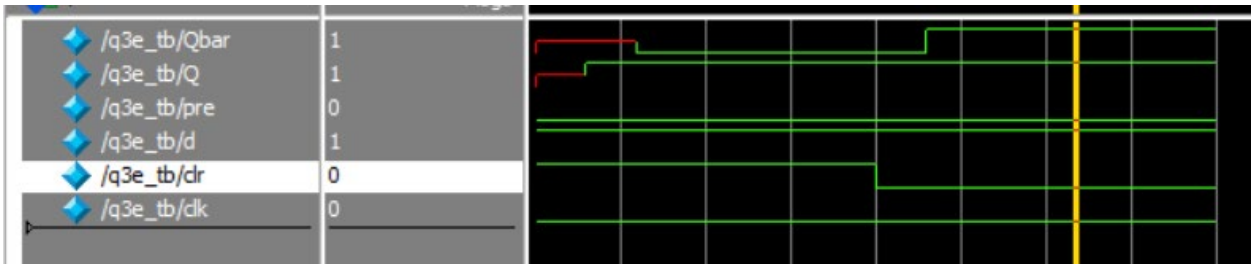



And you can see the result is as we expected for q and

Qbar.

Part g)

Changing clk while pre or clr are active (for example let's say pre is active) will lead us to loss of memory, because the output of q is surly 1(as it's input gate is connected to pre) so we have to check the input of qbar, and as it's following the output of g2, it will take 12 + 12 + 12 seconds to propagate as the worst case delay, then it will become 1 as well as q bar, and our memory will be lost.




The output is as we expected.

Part h)

When they are both active, the outputs of Q and Qbar are surely 1, so without any delays and no matter what, they don't change.