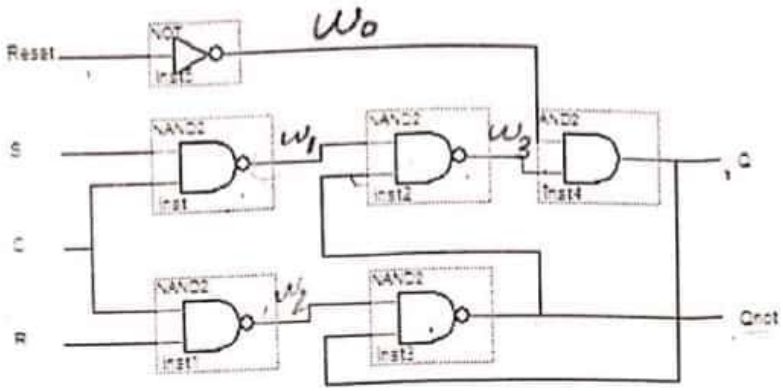
Part 1: Clocked SR-latch Verilog implementation and simulation:



SR-Latch Function table

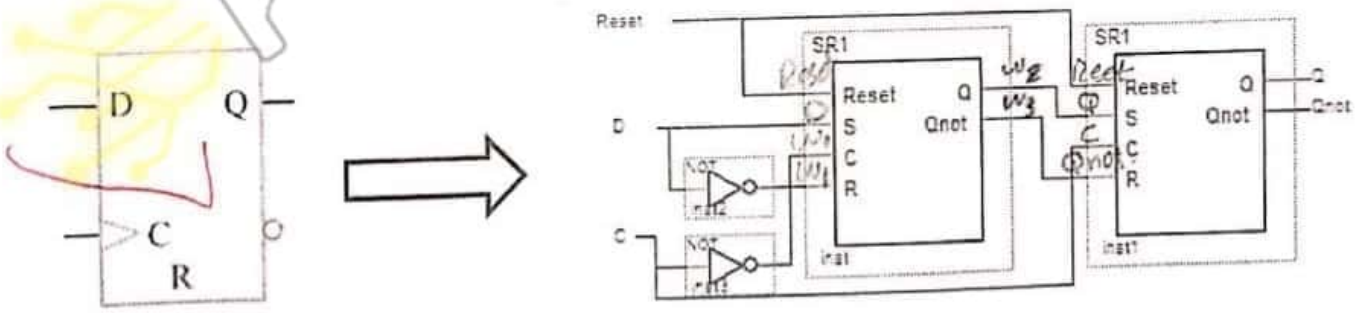
Reset	C	S	R	operation
1	x	x	x	Reset
0	0	x	x	Hold
0	1	0	0	Hold
0	1	0	1	Reset
0	1	1	0	Set
0	1	1	1	Forbidden

1. By hand, complete the below timing diagram for the clocked SR-latch. (Pre lab)

2. Inside SR2.v file, write a structural Verilog module to implement the clocked SR-latch.
   Note: NAND gate modules are already available in lib.v file

3. Use the vector waveform file to perform a functional simulation for your module in SR2.v. Use the same input values as the above timing diagram in your simulation. Also, compare your simulation output to your output in above.

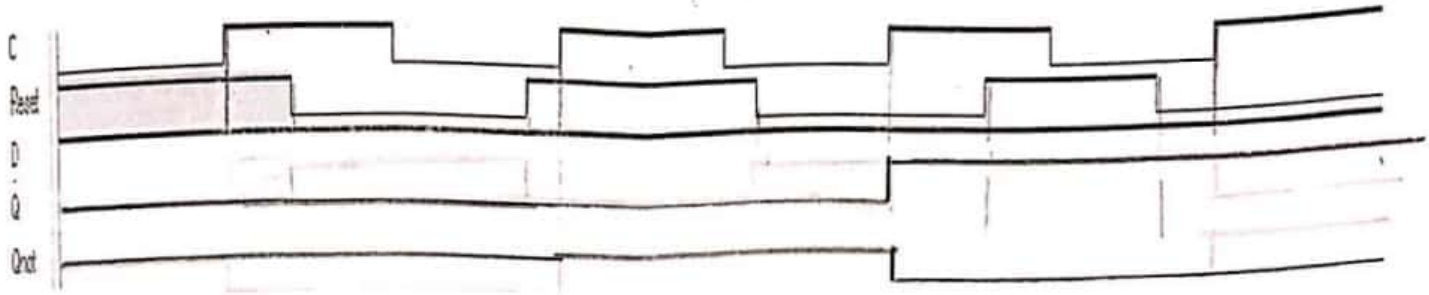
Part 2: D-FF implementation and simulation:



1. Fill the function table in below for the positive-edge triggered D-FF.

R	C	D	operation
1	x	x	reset
0	↑	0	set low (reset)
0	↑	1	set high (set).

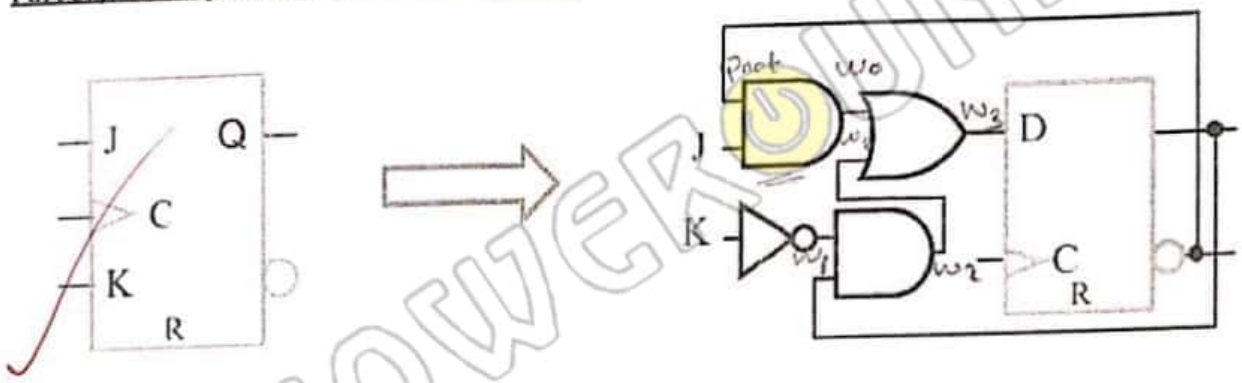
2. By hand, complete the below timing diagram for the D-FF. (Pre lab)



3. Inside dff1.v file, write a structural Verilog module to implement the D-FF circuit.
Note: You need to use the clocked SR-latch module defined in SR2.v and the basic gate modules defined in lib.v.

4. Use the vector waveform file to perform a functional simulation for your module in jkff1.v. Use the same input values as the above timing diagram in your simulation. Also, compare your simulation output to your output in above.

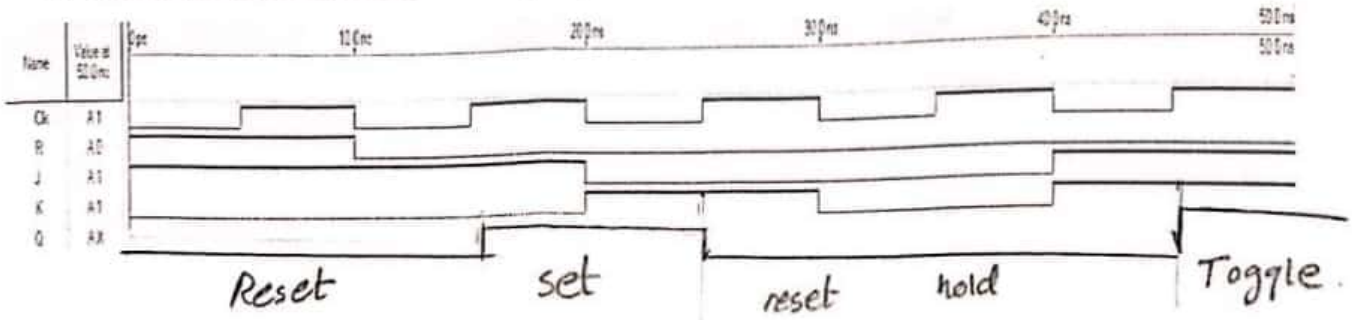
Part 3: JK-FF implementation and simulation:



1. Fill the function table in below for the positive-edge triggered JK-FF.

R	C	J	K	operation
1	x	x	x	reset
0	↑	0	0	hold
0	↑	0	1	reset
0	↑	1	0	set
0	↑	1	1	Toggle

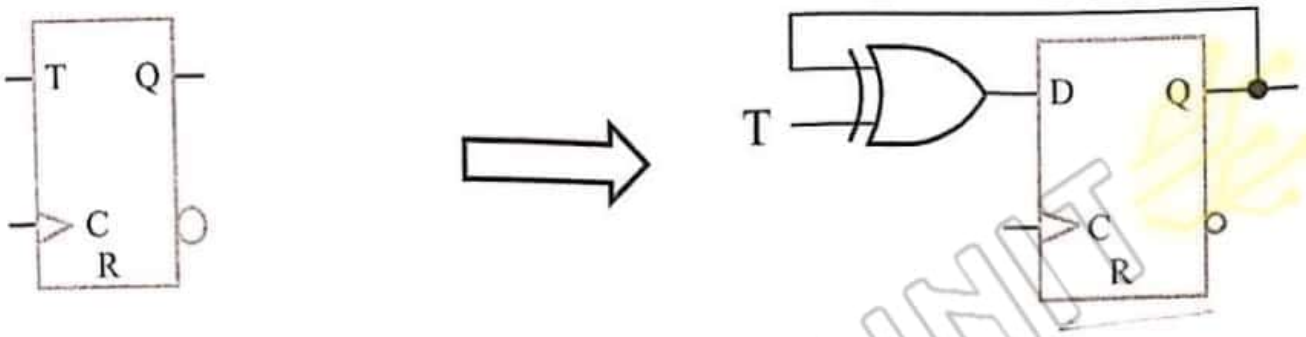
2. By hand, complete the below timing diagram for the JK-FF. (Pre lab)



3. Inside jkff1.v file, write a structural Verilog module to implement the JK-FF circuit.
   Note: You need to use the D-FF module defined in dff1.v and the basic gates modules defined in lib.v.

4. Use the vector waveform file to perform a functional simulation for your module in dff1.v. Use the same input values as the above timing diagram in your simulation. Also, compare your simulation output to your output in above.

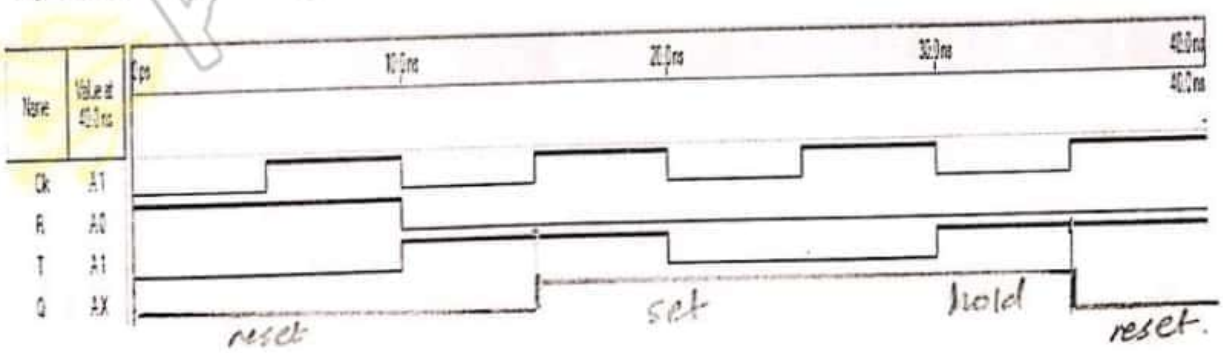
Part 4: T-FF implementation and simulation:



1. Fill the function table in below for the positive-edge triggered T-FF.

R	C	T	operation
1	x	x	Reset
0	↑	0	hold (no change)
0	↑	1	Toggle

2. By hand, complete the below timing diagram for the T-FF. (Pre lab)



3. Inside tff1.v file, write a structural Verilog module to implement the T-FF circuit.
   Note: You need to use the D-FF module defined in dff1.v and the XOR module defined in lib.v.

4. Use the vector waveform file to perform a functional simulation for your module in tff1.v. Use the same input values as the above timing diagram in your simulation. Also, compare your simulation output to your output in above.