FLIP FLOPS USING ALTERA FPGA

TASK:

To create flipflops (D,JK,T) using Altera FPGA

VERILOG CODE :

```
`timescale 1ns/1ps

// D FLIPFLOP
module D_FF (
    input clk,
    input rst,
    input D,
    output reg Q
);
    always @(posedge clk) begin
        if (rst)
            Q <= 0;
        else
            Q <= D;
    end
endmodule

// J K FLIPFLOP
module JK_FF (
    input clk,
    input rst,
    input J,
    input K,
    output reg Q
);
    always @(posedge clk) begin
        if (rst)
```

```
            Q <= 0;
        else begin
            case ({J,K})
                2'b00: Q <= Q;       // No change
                2'b01: Q <= 0;       // Reset
                2'b10: Q <= 1;       // Set
                2'b11: Q <= ~Q;      // Toggle
            endcase
        end
    end
endmodule
```

# T FLIPFLOP

```
module T_FF (
    input clk,
    input rst,
    input T,
    output reg Q
);
    always @(posedge clk) begin
        if (rst)
            Q <= 0;
        else if (T)
            Q <= ~Q;
    end
endmodule
```

TESTBENCH :

```
module tb_FlipFlops;
    reg clk, rst;
    reg D, J, K, T;
    wire Q_D, Q_JK, Q_T;

    D_FF dff (.clk(clk), .rst(rst), .D(D), .Q(Q_D));
    JK_FF jkff (.clk(clk), .rst(rst), .J(J), .K(K), .Q(Q_JK));
    T_FF tff (.clk(clk), .rst(rst), .T(T), .Q(Q_T));

    initial clk = 0;
    always #5 clk = ~clk;
    initial begin
        rst = 1; D = 0; J = 0; K = 0; T = 0;

        #10 rst = 0;
        D = 1; J = 1; K = 0; T = 1;
        #10 D = 0; J = 0; K = 1; T = 0;
        #10 D = 1; J = 1; K = 1; T = 1;
        #10 D = 0; J = 0; K = 0; T = 1;
        #20 $finish;
    end

Endmodule
```

PROCEDURE:

Create a New Project

- File → New Project Wizard → Name your project → Choose directory → Add your Verilog file.

Create Verilog Design File

- File → New → Verilog HDL File → Write your flip-flop code → Save (e.g., `D_FF.v`).

Create Testbench File

- File → New → Verilog HDL File → Write testbench → Save (e.g., `tb_D_FF.v`).

Add Files to Project

- Project → Add/Remove Files in Project → Add your design and testbench files.

Compile the Project

- Processing → Start Compilation → Wait until it finishes.

Simulate Using ModelSim

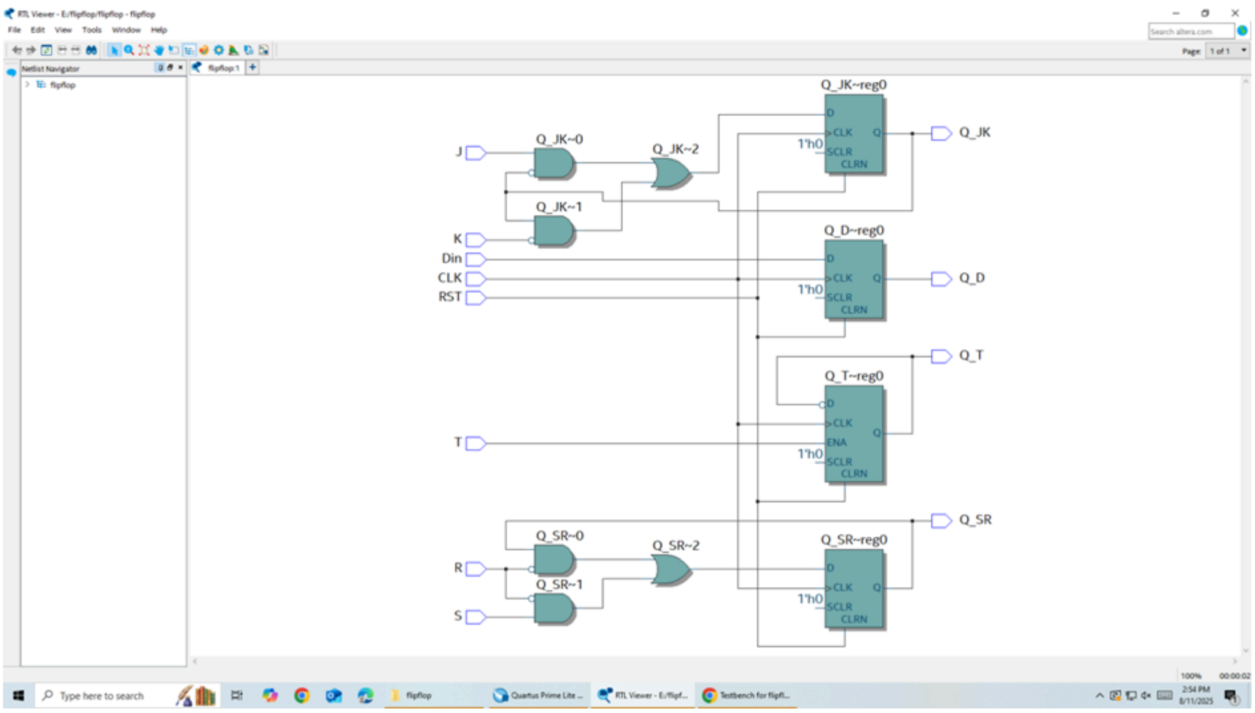
- Tools → Run Simulation Tool → RTL Simulation → Choose your testbench → Run → Observe waveforms.

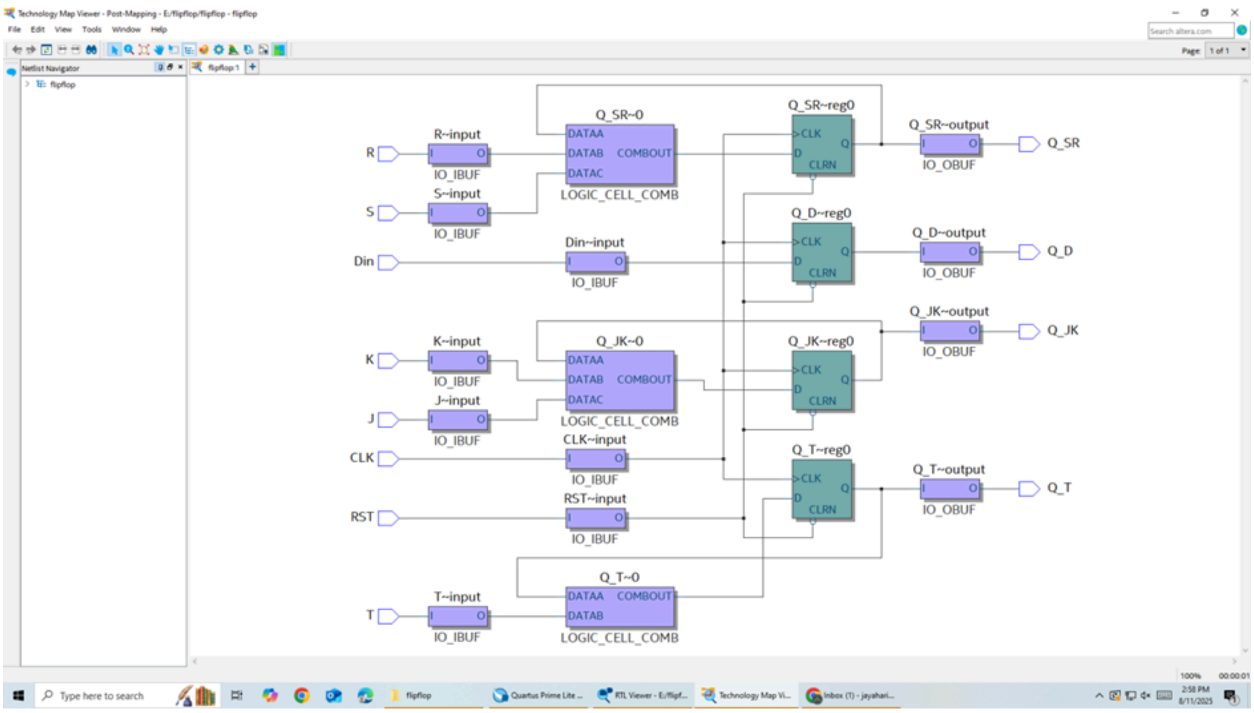
Assign Pins for FPGA (Optional for Hardware)

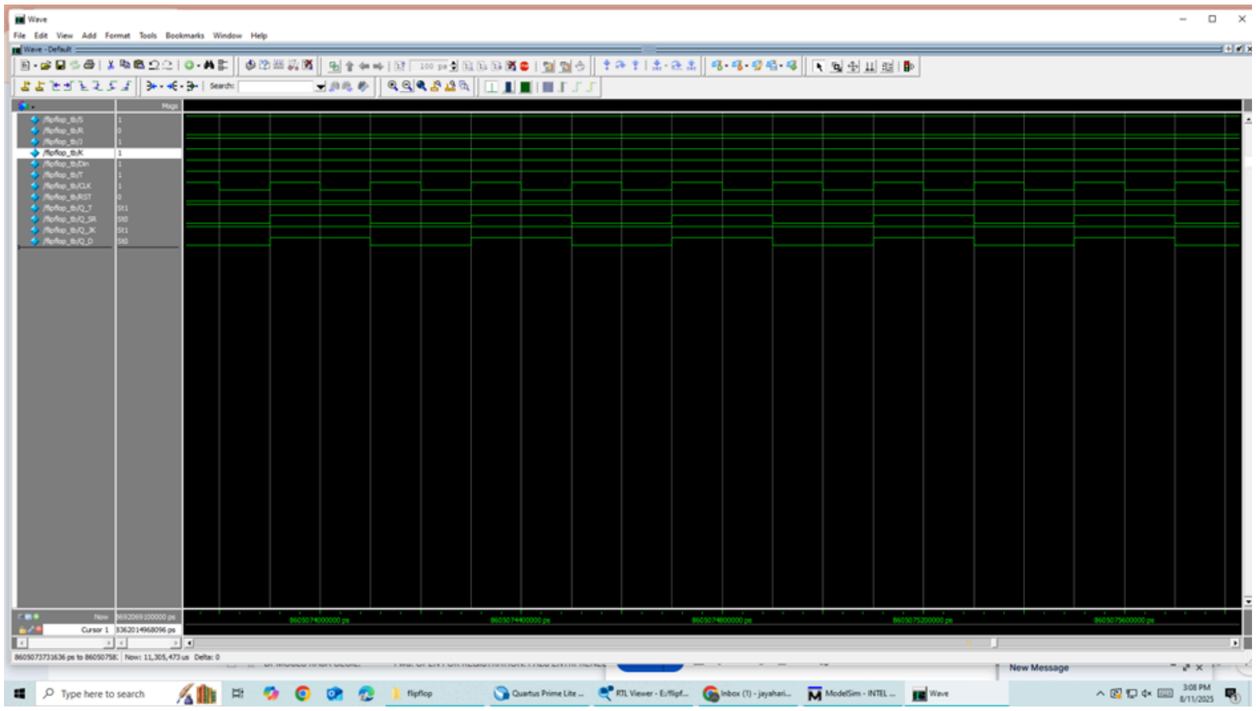
- Assign → Pin Planner → Map `clk`, `rst`, `D`, `J`, `K`, `T` → Compile again → Program FPGA.

PROCEDURAL STEPS :

1. Open Quartus Prime V17 software
2. Click on New Project Wizard
3. Create a new folder in your system and select that directory in the pop uop window
4. Add project name → Empty Project → Next → Next → Board → DE1-SoC Board
5. In EDA Test Settings pop up window select the following options :
   i. Design Entry / Synthesis : Precision Synthesis
   ii. Simulation : ModelSim-Altera
6. Click Next → Finish
7. Create a new file or Ctrl + N
8. In the pop up of different file options, choose Verilog HDL file under Design files and Click OK
9. Type in your verilog code in the new file created and save the file with the same name as your module
10. Type in your testbench code in the new file created and save the file with the same name as your module
11. In the left access window, under compile design and other compiling options click the Edit Settings option
12. Go to Simulation under EDA Tool Settings and change the Tool name to ModelSim-Altera
13. Under NativeLink Settings click Compile Design
14. In the Test bench field area add your testbench file you created then click Run simulation until all vector stimuli are used click Ok → Ok → Apply → Ok
15. In the Toolbar go to Tools → Run Simulation Tool → RTL Simulation
16. Go to Assignments on the Toolbar and then press Pin planner
17. In the pop up window assign the pins
18. Under the Task section in the left access window, Click on Program Device (Open Programmer Option)
19. Click Hardware Setup → DE-SoC → Close
20. Click Add File → output_files → Open → sof file → Open → Start
21. Check output


RTL Viewer - E:/flipflop/flipflop - flipflop
File Edit View Tools Window Help
Netlist Navigator
flipflop
flipflop:1
Page: 1 of 1
Q_JK~reg0
Q_JK~0
Q_JK~2
Q_JK~1
J
K
Din
CLK
RST
T
R
S
1'h0
D
CLK
SCLR
CLRN
ENA
Q
Q_JK
Q_D~reg0
Q_D
Q_T
Q_T~reg0
Q_SR
Q_SR~0
Q_SR~2
Q_SR~1
Q_SR~reg0
100%
00:00:02
Type here to search
Quartus Prime Lite ...
RTL Viewer - E:/flipf...
Testbench for flipfl...
2:54 PM
8/11/2025


Technology Map Viewer - Post-Mapping - E:/flipflop/flipflop - flipflop
File Edit View Tools Window Help
Netlist Navigator
flipflop
flipflop:1
Page: 1 of 1
R
R~input
IO_IBUF
S
S~input
Din
Din~input
K
K~input
J
J~input
CLK
CLK~input
RST
RST~input
T
T~input
Q_SR~0
DATAA
DATAB
DATAC
COMBOUT
LOGIC_CELL_COMB
Q_JK~0
Q_T~0
Q_SR~reg0
Q_D~reg0
Q_JK~reg0
Q_T~reg0
CLK
D
CLRN
Q
Q_SR~output
Q_D~output
Q_JK~output
Q_T~output
IO_OBUF
Q_SR
Q_D
Q_JK
Q_T
100%
00:00:01
Type here to search
Quartus Prime Lite ...
RTL Viewer - E:/flipf...
Technology Map Vi...
Inbox (1) - jayahari...
2:58 PM
8/11/2025



```
/flipflop_tb/S      1
/flipflop_tb/R      0
/flipflop_tb/J      1
/flipflop_tb/K      1
/flipflop_tb/Din    1
/flipflop_tb/T      1
/flipflop_tb/CLK    1
/flipflop_tb/RST    0
/flipflop_tb/Q_T    St1
/flipflop_tb/Q_SR   St0
/flipflop_tb/Q_JK   St1
/flipflop_tb/Q_D    St0

Now       8692069100000 ps
Cursor 1  3362014968096 ps

8605074000000 ps   8605074400000 ps   8605074800000 ps   8605075200000 ps   8605075600000 ps

8605073731636 ps to 86050758:  Now: 11,305,473 us  Delta: 0
```