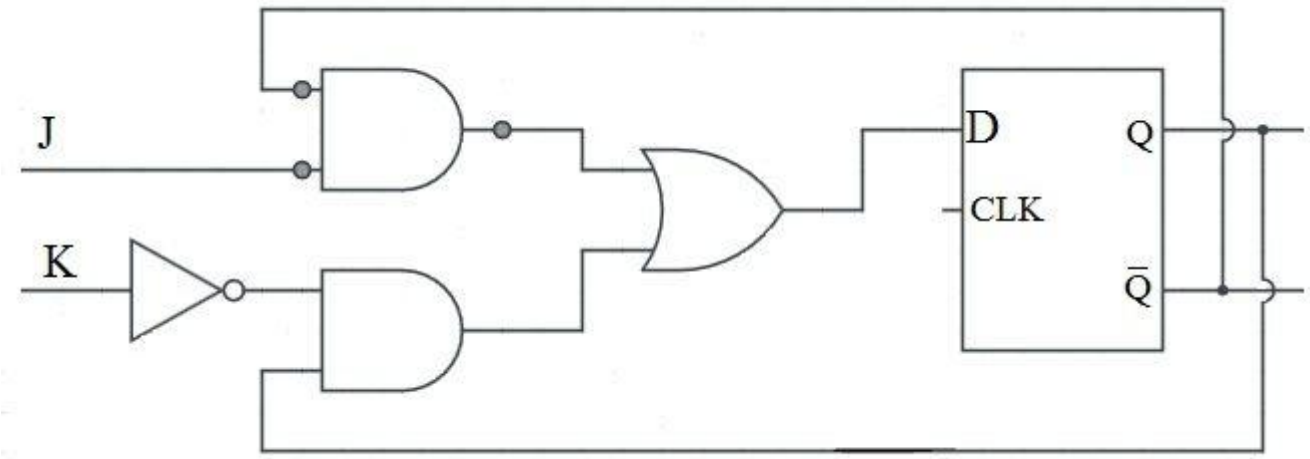
8. Implement a JK Flip Flop using D Flip Flops.

J	K	Qn	Qn+1	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Using K-maps, D = J.Qnbar + Kbar.Qn



CODE:

```
module jk_d (
   input clk,
   input rst,   // asynchronous reset
   input j, k,
   output q,
   output qbar
);
   wire d;
   wire q_int, qbar_int;

   // JK to D conversion: D = J·Q̄ + K̄·Q
   assign d = (j & ~q_int) | (~k & q_int);

   // Instantiate D Flip-Flop
   dff dff_inst (
```

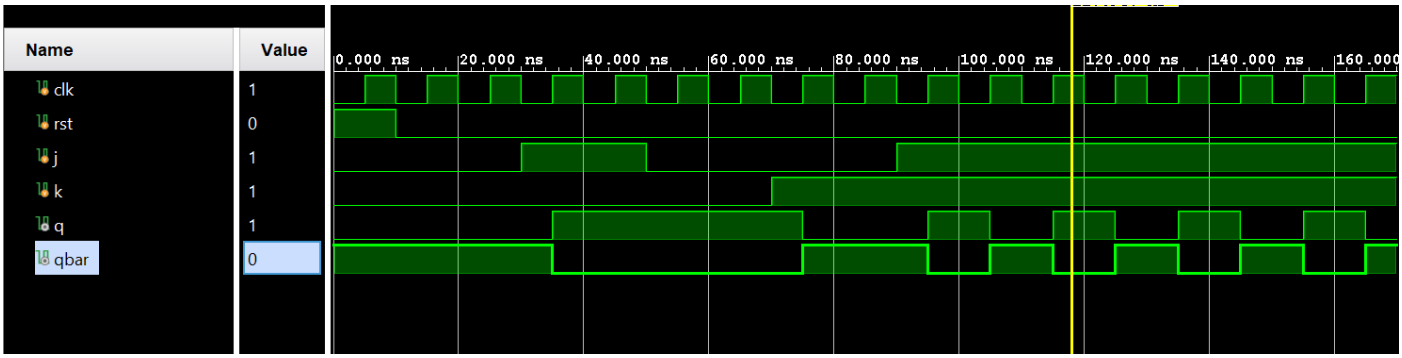
```
        .clk(clk),
        .rst(rst),
        .d(d),
        .q(q_int),
        .qbar(qbar_int)
    );

    assign q     = q_int;
    assign qbar  = qbar_int;
endmodule

module dff (
    input clk,
    input rst,   // asynchronous reset
    input d,
    output reg q,
    output qbar
);
    always @(posedge clk or posedge rst) begin
        if (rst)
            q <= 1'b0;
        else
            q <= d;
    end

    assign qbar = ~q;
endmodule
```

//testbench

```
`timescale 1ns/1ps

module jktb;
    reg clk;
    reg rst;
    reg j, k;
    wire q, qbar;

    // Instantiate JK FF using D FF
    jk_d jk(
        .clk(clk),
        .rst(rst),
        .j(j),
        .k(k),
        .q(q),
```

```
        .qbar(qbar)
    );

    // Clock generation: 10ns period
    initial begin
        clk = 0;
        forever #5 clk = ~clk; // Toggle every 5ns
    end

    // Stimulus
    initial begin
        $dumpfile("jk_ff_using_d_tb.vcd"); // For waveform
        $dumpvars(0, jktb);

        // Initial values
        rst = 1; j = 0; k = 0;
        #10 rst = 0;

        // Test sequence
        // Hold (J=0, K=0)
        j = 0; k = 0; #20;
        // Set (J=1, K=0)
        j = 1; k = 0; #20;
        // Hold again
        j = 0; k = 0; #20;
        // Reset (J=0, K=1)
        j = 0; k = 1; #20;
        // Toggle (J=1, K=1)
        j = 1; k = 1; #40;
        // Multiple toggles
        j = 1; k = 1; #40;

        // End simulation
        $finish;
    end

    // Monitor output
    initial begin
        $monitor("Time=%0t | J=%b K=%b | Q=%b Qbar=%b", $time, j, k, q, qbar);
    end
endmodule
```

Simulation Result:



KEY NOTES:

Had to make separate wires: wire q_int, qbar_int;