ECEN 2350 Spring 2017 Homework 7
Samuel Cuthbertson

---

1. *Book Problems: 5.7, 5.10, 5.16, 5.25*

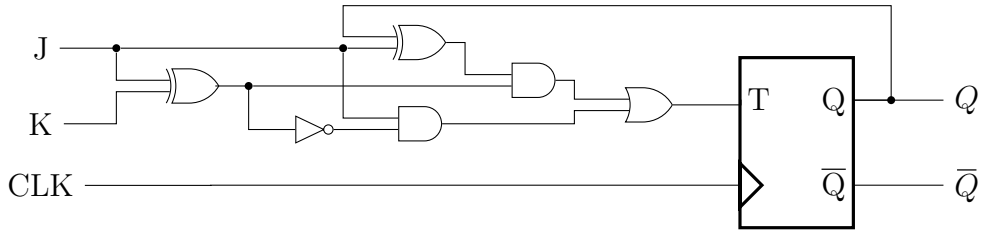
   5.7: *Show how a JK flip-flop can be implemented with a T flip-flop and other gates.*

   Note that the characteristic equations for a T flip-flip and for a JK flip-flop are both shown below.

T	$Q(t+1)$
0	$Q(t)$
1	$\overline{Q(t)}$

J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q(t)}$

   Therefor, for $J = K$ or $J \otimes K$, we can use J as T. For $J \neq K$, we need to set $T = J \oplus Q(t)$, as that will set $T = 1$ when $Q(t) \neq J$ and will toggle $Q(t+1)$. This is implemented below.



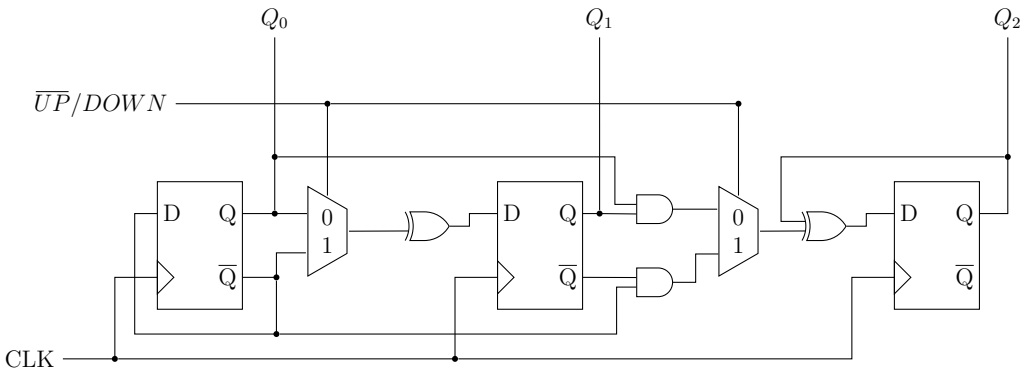
   5.10: *Write the behavioral Verilog code for a JK flip-flop.*

```
module jk_flip_flop(J, K, CLK, Q, nQ)
  input reg J, K, CLK;
  output reg Q, nQ;

  always@(posedge CLK)
  begin
    case({J, K})
      {1'b0, 1'b1}: Q <= 1'b0;
      {1'b1, 1'b0}: Q <= 1'b1;
      {1'b1, 1'b1}: Q <= ~Q;
    endcase
    nQ = ~Q
  end
endmodule
```

5.16: *Design a three bit up/down counter using D flip-flops, using a control signal of* $\overline{UP}/DOWN$.

For an up counter, we can use the equations $D_0 = Q_0 \oplus 1$, $D_1 = Q_1 \oplus Q_0$, $D_2 = Q_2 \oplus Q_1 Q_0$. For a down counter, the equations are similar: $D_0 = Q_0 \oplus 1$, $D_1 = \overline{Q_0} \oplus Q_1$, $D_2 = \overline{Q_1} \oplus Q_1 Q_2$. As the only diffrence between these is whether to use Q_{n-1} or $\overline{Q_{n-1}}$ in the `xor` with the previous `and`, we can use a `mux` at each stage with our control signal as a select.



5.25: *Using the circuit shown in the book, complete the below timing diagram.*

Placeholder answer

2. *Draw a 4-bit Universal shift register that can parallel load, shift left, shift right, and synchronously clear. Provide a characteristic table for the control operations.*

   Placeholder answer

3. *Implement a 3-bit up-counter using only one D flip-flop, one T flip-flop, one JK flip-flop and one AND gate. Assume all flip-flops are positive edge triggered. Show your circuit and a timing diagram.*

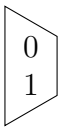
   You don't have a power source or a clock, so all of this is moot, but if you could find both of those then you could implement an up counter as shown below.

   As the first bit Q_0 changes with every posedge of clock, we can simply attach $\overline{Q_0}$ to the input of the D and use that output for Q_0.

   Similarly, as Q_1 toggles between 1 and 0 every time $\overline{Q_0}$ changes from 0 to 1 (posedge) we can attach 1 as the input to the T flip-flop and $\overline{Q_0}$ to the clock, and then use the output from the T flip-flop as Q_1.

   Finally, as Q_2 toggles everytime that $\overline{Q_0}\overline{Q_1}$ goes 0 to 1 (posedge), we can attach the output from that `and` gate to both J and K, making the JK flip-flop behave like a T flip-flop. Then, we can use the output from the JK flip-flop as Q_2.

   All this put together is shown below:



4. *Write the behavioural Verilog for 5.16, calling the module* ***up_down_counter****.*

```
module up_down_counter(clk, ctrl, value);
  input clk;
  input ctrl;
  output reg [2:0] value;

  always@(posedge clk)
  begin
    case(ctrl)
      1'b0: value <= value - 1;
      1'b1: value <= value + 1;
    endcase
  end
endmodule
```