Problem 5.10

```
module jk_flip_flop (
  input wire J,
  input wire K,
  input wire clk,
  input wire rst,
  output reg Q,
  output reg Qn
);
  reg toggle;

  always @(posedge clk or posedge rst) begin
    if (rst) begin
      Q <= 1'b0;
      Qn <= 1'b1;
    end else begin
      if (J && K) begin
      end else if (J) begin
        Q <= 1'b1;
        Qn <= 1'b0;
      end else if (K) begin
        Q <= 1'b0;
        Qn <= 1'b1;
      end else begin
        Q <= toggle;
        Qn <= ~toggle;
      end
    end
  end

  always @(posedge clk) begin
    toggle <= J ^ Q;
  end

endmodule
```

Problem 5.14

```
module T_FlipFlop (
  input wire clk,
  input wire rst,
  input wire T,
  output reg Q
);
  reg D;

  always @(posedge clk or posedge rst) begin
    if (rst) begin
      Q <= 4'b0;
    end else begin
      D <= T ^ Q[0];
      Q <= {Q[2:0], D};
    end
  end
endmodule

module FourBitCounter (
  input wire clk,
  input wire rst,
  output reg [3:0] Q
);
  wire [3:0] Q_int;

  T_FlipFlop num_1 (.clk(clk), .rst(rst), .T(1'b1), .Q(Q_int[0]));
  T_FlipFlop num_2 (.clk(clk), .rst(rst), .T(Q_int[0]), .Q(Q_int[1]));
  T_FlipFlop num_3 (.clk(clk), .rst(rst), .T(Q_int[1]), .Q(Q_int[2]));
  T_FlipFlop num_4 (.clk(clk), .rst(rst), .T(Q_int[2]), .Q(Q_int[3]));

  assign Q = Q_int;
endmodule
```

Problem 5.16

```
module UpDownCounter(
  input wire clk,
  input wire rst,
  input wire down,
  output reg [2:0] count
);
  reg [2:0] next_count;

  always @(posedge clk or posedge rst) begin
    if (rst) begin
      count <= 3'b0;
    end else begin
      if (down) begin
        next_count = count - 1;
      end else begin
        next_count = count + 1;
      end
      count <= next_count;
    end
  end
endmodule
```

Problem 5.20

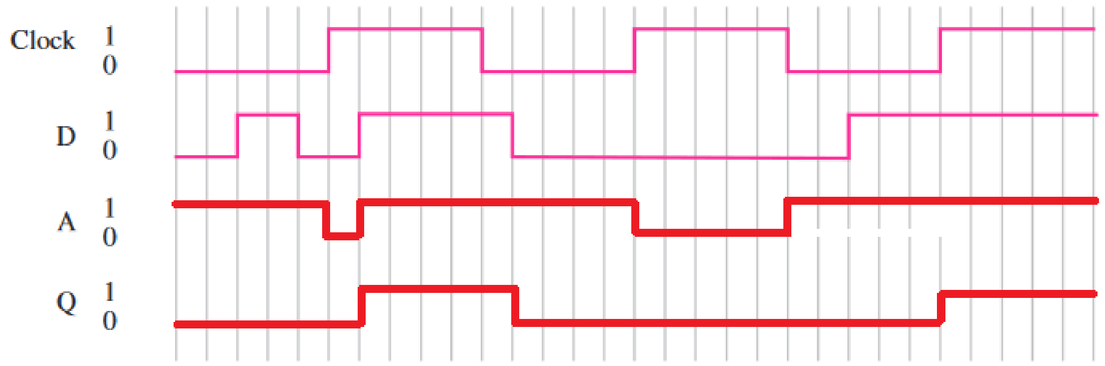
```
module Mod12Counter (
  input wire clk,
  input wire rst,
  output reg [3:0] count
);

  always @(posedge clk) begin
    if (rst) begin
      count <= 4'b0000;
    end else begin
      if (count == 4'b1011) begin
```

```
        count <= 4'b0000;
      end else begin
        count <= count + 1;
      end
    end
  end

endmodule
```

Problem 5.25


Clock
D
A
Q