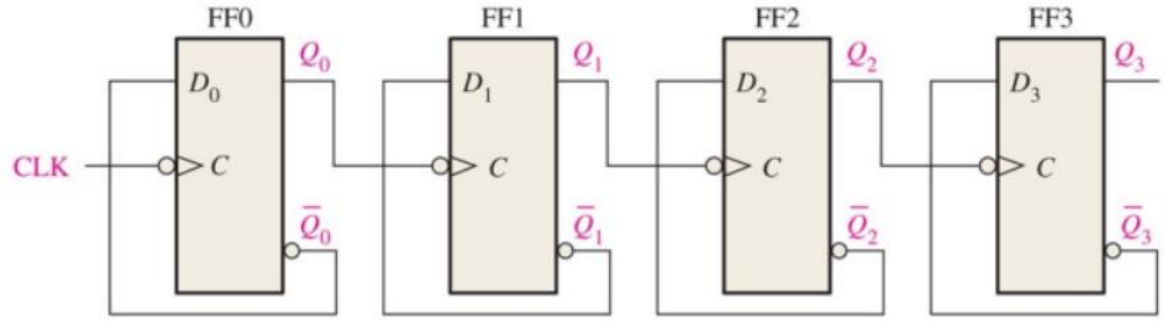
DAY 28

4 BIT ASYNCHRONOUS UP COUNTER:



(a)

RTL CODE:

```
module async_up_counter_with_D(input
clk,reset,d,output reg q,qbar);
  always @(negedge clk)
    begin

      if (reset)
        q <= 0;
      else
        q <= d;
```

```
    end
  assign qbar = ~q;
endmodule

module asyn(inpot clk,reset,[3:0]d,output q,qbar,output
[3:0]cnt);
  wire q0,q0bar,q1,qbbar,q2,q2bar;
  sync_up_counter_with_D a1(clk,reset,d[0],q0,q0bar);
  sync_up_counter_with_D a2(q0,reset,d[1],q1,q1bar);
  sync_up_counter_with_D a3(q1,reset,d[2],q2,q2bar);
  sync_up_counter_with_D a4(q2,reset,d[3],q,qbar);
  assign d[0]=q0bar;
  assign d[1]=q1bar;
  assign d[2]=q2bar;
  assign d[3]=qbar;
  assign cnt={q,q2,q1,q0};
endmodule
```

TEST BENCH:

```
module async_up_counter_tb;
```

```
  reg clk;
  reg reset;
  reg [3:0] d;
  wire  q;
  wire  qbar;
  wire [3:0] cnt;
  async_up_counter_with_D
dut(clk,reset,d,{q,qbar,cnt});
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
  initial begin
    clk=0;
    forever #5 clk=~clk;
  end
  initial begin
    reset = 1;
    #10;
    reset = 0;
  end
```

```
  initial begin
    d = 4'b0000;
    #20;
    d = 4'b0001;
    #20;
    d = 4'b0010;
    #20;
    $stop;
  end
endmodule
```