#50DAYS CHALLENGE

DAY 12

D FLIPFLOP

a)Design Code

```
module d_flipflop(d,clk,rst,q,qbar);
input d,clk,rst;
output reg q;
output qbar;

always@(posedge clk)
begin
if(rst)
q<=0;
else
q<=d;
end
assign qbar=~q;
endmodule
```

b)Testbench

```
module d_flipflop_tb();
wire q,qbar;
reg d,clk,rst;

d_flipflop dut(.d(d),.rst(rst),.clk(clk),.q(q),.qbar(qbar));

initial
begin
clk=1'b0;
forever #10 clk=~clk;
end

task rst_dut();
begin
rst=1'b1;
#10;
rst=1'b0;
end
endtask

task din(input i);
begin
@(negedge clk);
d=i;
end
endtask
```

```
initial
begin
rst_dut;
din(0);
din(1);
din(0);
din(1);
rst_dut;
din(0);
din(1);
#10;
$finish();
end

initial
begin
$monitor("Time=%t |  clk=%b  |rst=%b  |d=%b  |q=%b  |qbar=%b ",$time,clk,rst,d,q,qbar);
end
endmodule
```

c)Waveform

d)Transcript

```
VSIM 4> run -all
# Time=                    0 |  clk=0  |rst=1   |d=x  |q=x  |qbar=x
# Time=                   10 |  clk=1  |rst=0   |d=x  |q=x  |qbar=x
# Time=                   20 |  clk=0  |rst=0   |d=0  |q=x  |qbar=x
# Time=                   30 |  clk=1  |rst=0   |d=0  |q=0  |qbar=1
# Time=                   40 |  clk=0  |rst=0   |d=1  |q=0  |qbar=1
# Time=                   50 |  clk=1  |rst=0   |d=1  |q=1  |qbar=0
# Time=                   60 |  clk=0  |rst=0   |d=0  |q=1  |qbar=0
# Time=                   70 |  clk=1  |rst=0   |d=0  |q=0  |qbar=1
# Time=                   80 |  clk=0  |rst=1   |d=1  |q=0  |qbar=1
# Time=                   90 |  clk=1  |rst=0   |d=1  |q=1  |qbar=0
# Time=                  100 |  clk=0  |rst=0   |d=0  |q=1  |qbar=0
# Time=                  110 |  clk=1  |rst=0   |d=0  |q=0  |qbar=1
# Time=                  120 |  clk=0  |rst=0   |d=1  |q=0  |qbar=1
```

ASYNCHRONOUS D FLIPFLOP

a)Design Code

```
//To design asynchronous rst &  active low set
module async_d_ff(d,en,clk,set,rst,q,qbar);
input d,clk,rst,en,set;
output reg q;
output qbar;


always@(posedge clk or posedge rst or negedge set)
begin

if(rst)
q<=0;
else if(!set)
q<=1;
else if(en)
q<=d;
end


assign qbar=~q;

endmodule
```

b)Testbench

```
module async_d_ff_tb();
reg d,clk,rst,en,set;
wire q,qbar;

async_d_ff dut(.d(d),.clk(clk),.en(en),.rst(rst),.set(set),.q(q),.qbar(qbar));

initial
begin
clk=1'b0;
forever #10 clk=~clk;
end


task rst_dut();
begin
rst=1'b1;
#10;
rst=1'b0;
end
endtask

task enb_input(input y);
begin
en=y;
end
endtask

task set_dut(input active);
begin
    set = active ? 1'b0 : 1'b1;
        #15;
        set = 1'b1;
    end
    endtask
```

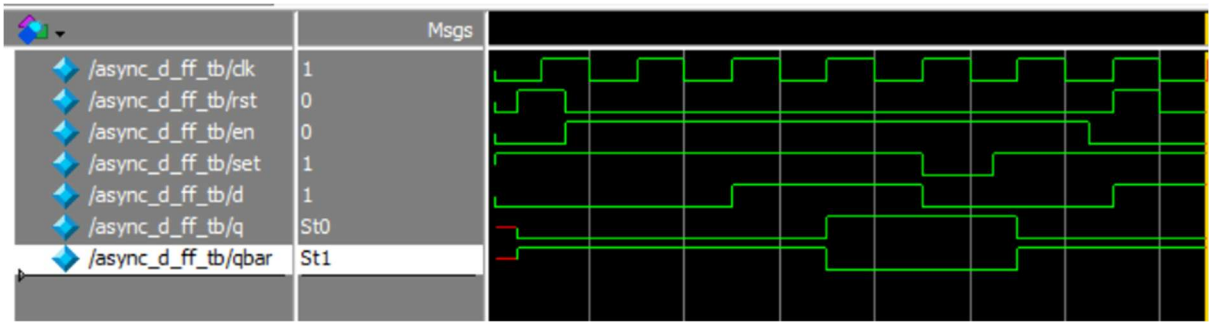
```
task inputs_dut(input i);
begin
 @(posedge clk);
 d=i;
end
 endtask

 initial
begin
  d = 0; en = 0; rst = 0; set = 1;
         #5;
         rst_dut();
         enb_input(1);
         inputs_dut(0);
         inputs_dut(1);
         inputs_dut(1);
         inputs_dut(0);
         set_dut(1);
         #20;


         enb_input(0);
         inputs_dut(1);


         rst_dut();
 #10;
 $finish();
end
 initial
begin
 $monitor("clk=%b,en=%b,set=%b,rst=%b,d=%b,q=%b,qbar=%b",clk,en,set,rst,d,q,qbar);
end
 endmodule
```

c)Waveform



d)Transcript

```
VSIM 10> run -all
# clk=0,en=0,set=1,rst=0,d=0,q=x,qbar=x
# clk=0,en=0,set=1,rst=1,d=0,q=0,qbar=1
# clk=1,en=0,set=1,rst=1,d=0,q=0,qbar=1
# clk=1,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=0,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=1,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=0,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=1,en=1,set=1,rst=0,d=1,q=0,qbar=1
# clk=0,en=1,set=1,rst=0,d=1,q=0,qbar=1
# clk=1,en=1,set=1,rst=0,d=1,q=1,qbar=0
# clk=0,en=1,set=1,rst=0,d=1,q=1,qbar=0
# clk=1,en=1,set=0,rst=0,d=0,q=1,qbar=0
# clk=0,en=1,set=0,rst=0,d=0,q=1,qbar=0
# clk=0,en=1,set=1,rst=0,d=0,q=1,qbar=0
# clk=1,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=0,en=1,set=1,rst=0,d=0,q=0,qbar=1
# clk=0,en=0,set=1,rst=0,d=0,q=0,qbar=1
# clk=1,en=0,set=1,rst=1,d=1,q=0,qbar=1
# clk=0,en=0,set=1,rst=0,d=1,q=0,qbar=1
```

SYNCHRONOUS D FLIPFLOP

a)Design Code

```
module sync_d_ff(d,clk,reset,set,en,q);
input d,clk,reset,set,en;
 output reg q;

always@(posedge clk)
begin
if(en)
begin

 if(reset)
q<=0;
else if(set)
q<=1;
else
q<=d;
end
end
endmodule
```

b)Testbench

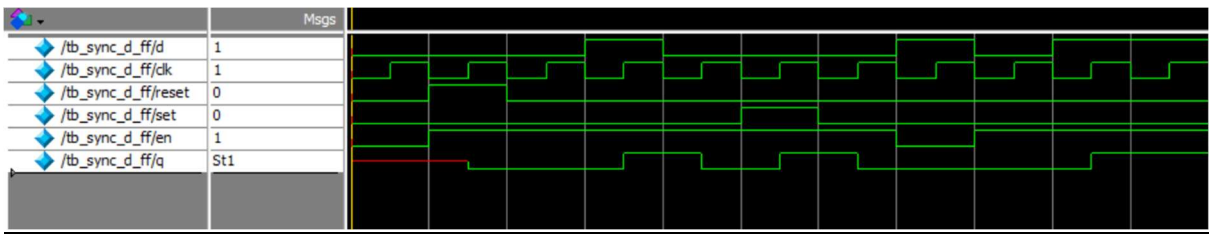
```
`timescale 1ns/1ps

module tb_sync_d_ff;

reg d, clk, reset, set, en;
wire q;
sync_d_ff dut ( .d(d),  .clk(clk),  .reset(reset), .set(set),.en(en), .q(q));
initial begin
    clk = 0;
    forever #5 clk = ~clk;
end
initial
 begin
    d = 0; reset = 0; set = 0; en = 0;
    #10 en = 1; reset = 1;
    #10 reset = 0;
    #10 d = 1;
    #10 d = 0;
    #10 set = 1;
    #10 set = 0;
    #10 en = 0; d = 1;
    #10 en = 1; d = 0;
    #10 d = 1;
    #20 $finish;
end
initial
 begin
    $monitor("Time=%0t | clk=%b | reset=%b | set=%b | en=%b | d=%b | q=%b",
             $time, clk, reset, set, en, d, q);
end

endmodule
```

c)Waveform



d)Transcript

```
# Time=0 | clk=0 | reset=0 | set=0 | en=0 | d=0 | q=x
# Time=5000 | clk=1 | reset=0 | set=0 | en=0 | d=0 | q=x
# Time=10000 | clk=0 | reset=1 | set=0 | en=1 | d=0 | q=x
# Time=15000 | clk=1 | reset=1 | set=0 | en=1 | d=0 | q=0
# Time=20000 | clk=0 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=25000 | clk=1 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=30000 | clk=0 | reset=0 | set=0 | en=1 | d=1 | q=0
# Time=35000 | clk=1 | reset=0 | set=0 | en=1 | d=1 | q=1
# Time=40000 | clk=0 | reset=0 | set=0 | en=1 | d=0 | q=1
# Time=45000 | clk=1 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=50000 | clk=0 | reset=0 | set=1 | en=1 | d=0 | q=0
# Time=55000 | clk=1 | reset=0 | set=1 | en=1 | d=0 | q=1
# Time=60000 | clk=0 | reset=0 | set=0 | en=1 | d=0 | q=1
# Time=65000 | clk=1 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=70000 | clk=0 | reset=0 | set=0 | en=0 | d=1 | q=0
# Time=75000 | clk=1 | reset=0 | set=0 | en=0 | d=1 | q=0
# Time=80000 | clk=0 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=85000 | clk=1 | reset=0 | set=0 | en=1 | d=0 | q=0
# Time=90000 | clk=0 | reset=0 | set=0 | en=1 | d=1 | q=0
# Time=95000 | clk=1 | reset=0 | set=0 | en=1 | d=1 | q=1
# Time=100000 | clk=0 | reset=0 | set=0 | en=1 | d=1 | q=1
# Time=105000 | clk=1 | reset=0 | set=0 | en=1 | d=1 | q=1
```