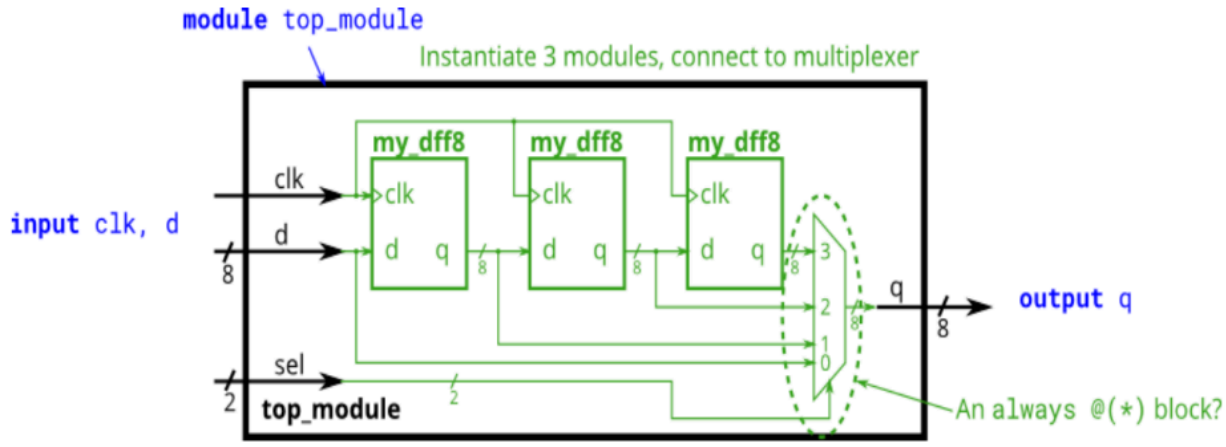
KARTHICK RAJA B
126004121

Given:

You have to design a module my_dff8 with two inputs and one output (that implements a set of 8 D flip-flops). Instantiate three of them, then chain them together to make an 8-bit wide shift register of length 3. In addition, create a 4-to-1 multiplexer that chooses what to output depending on sel[1:0]: The value at the input d, after the first, after the second, or after the third D flip-flop. (Essentially, sel selects how many cycles to delay the input, from zero to three clock cycles.)



Verilog Code:

```
module my_diff8(clk,d,q);
input clk;
input [7:0] d;
output reg [7:0] q;
always @(posedge clk)
begin
q<=d;
end
endmodule


module RTL_Day_14(clk,d,sel,q);
input clk;
input [7:0]d;
input [1:0]sel;
output reg [7:0] q;
```

```
wire [7:0]w1,w2,w3;
my_diff8 d1 (clk,d,w1);
my_diff8 d2 (clk,w1,w2);
my_diff8 d3 (clk,w2,w3);
always @(*)
begin
    case(sel)
    2'b00: q=d;
    2'b01: q=w1;
    2'b10: q=w2;
    2'b11: q=w3;
    default:q=8'b00000000;
    endcase
end
endmodule
```

Testbench code:

```
module test_bench_tb14();
reg clk;
reg [1:0]sel;
reg[7:0] d;
wire[7:0] q;
RTL_Day_14 b1(clk,d,sel,q);
always #50 clk=~clk;
initial
begin
clk=1; d=8'b00000000; sel=2'b00; #100;
d=8'b11111111; sel=2'b01; #100;
sel=2'b10; #100;
sel=2'b11;
end
endmodule
```

Functional Simulation: