**Code:**

* **Design Module**
* **FlipFlop Module**

module flipflop( input clk,

input d,

output q\_ext,

output qbar\_ext

);

reg q,qbar;

always@(clk)

begin

q <= d ;

qbar <= ~d;

end

assign q\_ext = q;

assign qbar\_ext = qbar;

endmodule

* **Main Module**

// Code your design here

`include"flipflop.sv"

module shift\_register(input clk,

input d,

output out,

output out\_bar);

reg f1\_q,f1\_qbar;

reg f2\_q,f2\_qbar;

reg f3\_q,f3\_qbar;

reg f4\_q,f4\_qbar;

flipflop f1(clk,d,f1\_q,f1\_qbar);

flipflop f2(clk,f1\_q,f2\_q,f2\_qbar);

flipflop f3(clk,f2\_q,f3\_q,f3\_qbar);

flipflop f4(clk,f3\_q,f4\_q,f4\_qbar);

assign out = f4\_q;

assign out\_bar = f4\_qbar;

endmodule

* **TestBench**

// Code your testbench here

// or browse Examples

module tb();

bit clk,d;

wire out,out\_bar;

shift\_register dut(clk,d,out,out\_bar);

always #5 clk = ~clk;

always #5 d = ~d;

// for eda playground waveform

// initial

// begin

// $dumpfile("dump.vcd");

// $dumpvars();

// #500

// $finish;

// end

endmodule

**Output:**

A picture containing graphical user interface

Description automatically generated