Verilog code for JK Flipflop:

```
module JK_flipflop (
  input clk, rst_n,
  input j,k,
  output reg q,
  output q_bar
  );

  always@(posedge clk) begin
    if(!rst_n) q <= 0;
    else begin
      case({j,k})
        2'b00: q <= q;
        2'b01: q <= 1'b0;
        2'b10: q <= 1'b1;
        2'b11: q <= ~q;
      endcase
    end
  end
  assign q_bar = ~q;
endmodule
```

Test Bench for JK flipflop:

```
module tb;
  reg clk, rst_n;
  reg j, k;
  wire q, q_bar;
  JK_flipflop dff(clk, rst_n, j, k, q, q_bar);
  always #2 clk = ~clk;
  initial begin
```

```
    clk = 0; rst_n = 0;
    $display("Reset=%b --> q=%b, q_bar=%b", rst_n, q, q_bar);
    #3 rst_n = 1;
    $display("Reset=%b --> q=%b, q_bar=%b", rst_n, q, q_bar);
    drive(2'b00);
    drive(2'b01);
    drive(2'b10);
    drive(2'b11);
    drive(2'b11);
    #5;
    $finish;
  end
  task drive(bit [1:0] ip);
    @(posedge clk);
    {j,k} = ip;
    #1 $display("j=%b, k=%b --> q=%b, q_bar=%b",j, k, q, q_bar);
  endtask
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
endmodule
```

OUTPUT:

```
Reset=0 --> q=x, q_bar=x
Reset=1 --> q=0, q_bar=1
j=0, k=0 --> q=0, q_bar=1
j=0, k=1 --> q=0, q_bar=1
j=1, k=0 --> q=1, q_bar=0
j=1, k=1 --> q=0, q_bar=1
j=1, k=1 --> q=1, q_bar=0
```