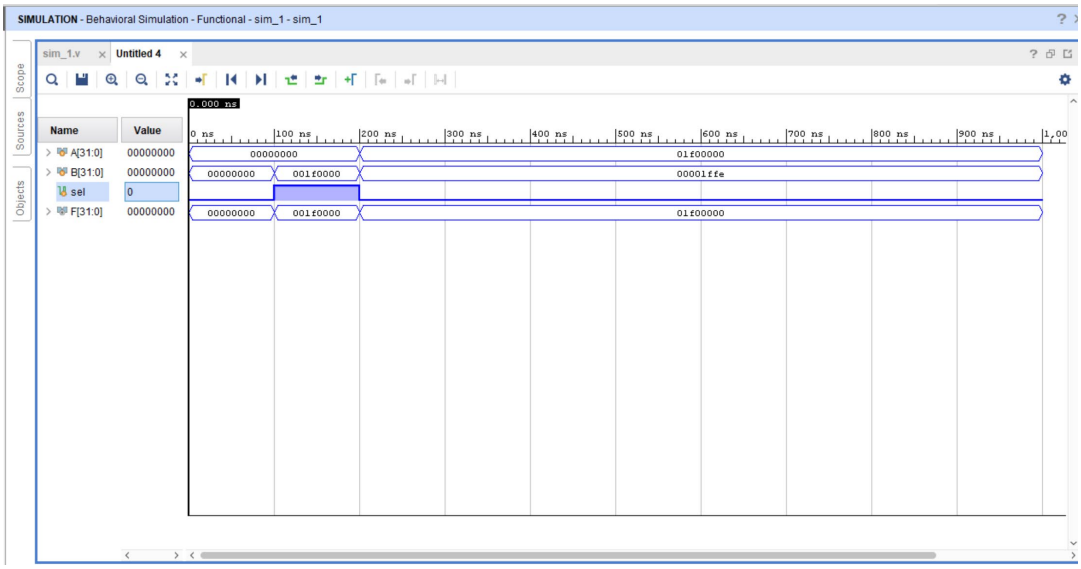
1.

```
module MUX_2_1_32(F, A, B, sel);
    input [31:0] A, B;
    input sel;
    output [31:0] F;
    reg [31:0] F;

    always @(A,B,sel) begin
        case (sel)
        1'b0: F=A;
        1'b1: F=B;
        default F=0;
        endcase
    end
endmodule
```

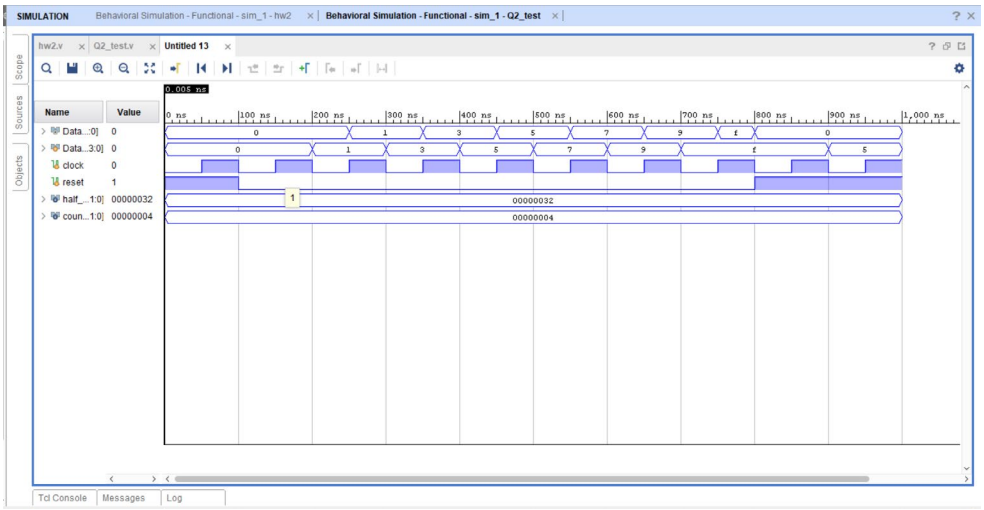
```
module sim_1;
    //inputs
    reg [31:0] A;
    reg [31:0] B;
    reg sel;

    //output
    wire [31:0] F;

    //Instantiate the Unit Under Test(UUT)
    MUX_2_1_32 uut(F,A,B,sel);
        initial begin
        #0
        A=0;
        B=0;
        sel=0;


        #100
        A=32'b00000000000000000000000000000000;
        B=32'b00000000000011111000000000000000;
        sel=1;
        #100
        A=32'b00000000111110000000000000000000;
        B=32'b00000000000000000000111111111110;
        sel=0;
    end
endmodule
```



2.

```
module hw2(Data_in, clock, reset, Data_out);
    input [3:0] Data_in;
    input       clock, reset;
    output [3:0] Data_out;
    reg [3:0] Data_out;

    always @ (posedge reset or posedge clock)
        if (reset==1'b1) Data_out <=4'b0;
        else Data_out<=Data_in;
endmodule
```

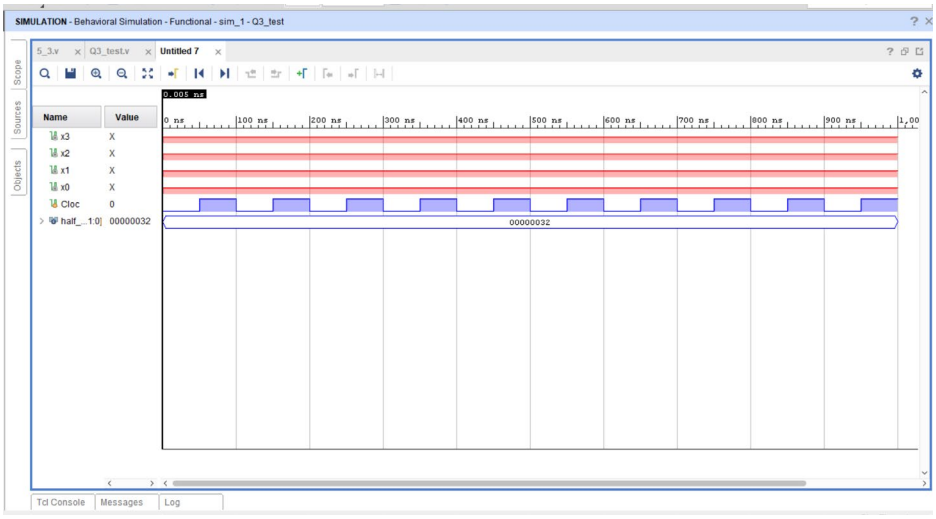
```
module Q2_test;
    parameter half_period=50;
    parameter counter_size=4;

    wire [counter_size-1:0] Data_out;
    reg [counter_size-1:0] Data_in;
    reg clock,reset;

    hw2 #(counter_size) UUT (Data_in, clock, reset, Data_out);

    initial begin
    #0 clock=0;Data_in=0;reset=1;
    #100 reset=0;
    #100 Data_in=4'b0001;
    #100 Data_in=4'b0011;
    #100 Data_in=4'b0101;
    #100 Data_in=4'b0111;
    #100 Data_in=4'b1001;
    #100 Data_in=4'b1111;
    #100 reset=1;
    #100 Data_in=4'b0101;
    end

    always #half_period clock=~clock;


endmodule
```



3.

```
module Hw3 (x3,x2,x1,x0, Cloc);
    input Cloc;
    output x3,x2,x1,x0;

    reg x3,x2,x1,x0,f,a,b,c,d;

    always @ (posedge Cloc) begin
    f=(~a)&d|a&(~d);
    a<=f;
    b<=a;
    c<=b;
    d<=c;
    x3=f;
    x2=a;
    x1=b;
    x0=c;
    end
endmodule
```

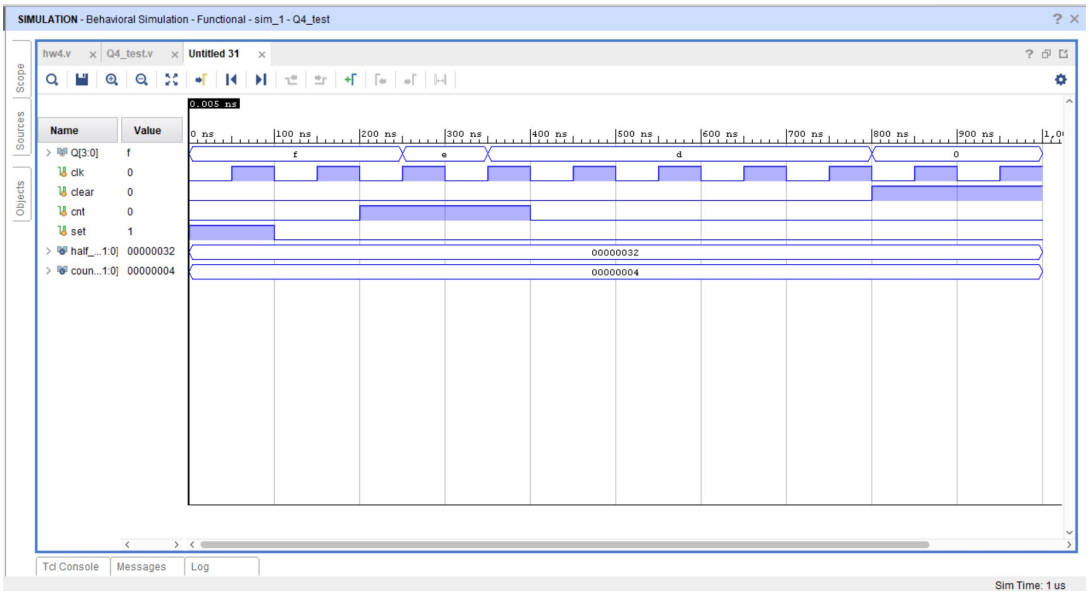
```
module Q3_test;
parameter half_period=50;
wire x3,x2,x1,x0;
reg Cloc;

Hw3 UUT(x3,x2,x1,x0, Cloc);

initial begin
#0 Cloc=0;
end

   always #half_period Cloc=~Cloc;
endmodule
```



4.

```
module hw4(
    clk,clear,cnt,set,Q,
    );
    input clk,clear,cnt,set;

    output [3:0] Q;

    reg [3:0] Q;
    always @(posedge clear or posedge clk or posedge set)
        if (clear==1'b1) Q<=0;
        else if (set==1'b1) Q<=15;
        else if (cnt==1'b1) Q=Q-1;
        else Q=Q;

endmodule
```

```
module Q4_test;
 parameter half_period=50;
   parameter counter_size=4;

   wire [counter_size-1:0] Q;
   reg clk,clear,cnt,set;

   hw4 #(counter_size) UUT (clk,clear,cnt,set,Q,);

   initial begin
   #0 clk=0;set=1;clear=0;cnt=0;
   #100 set=0;
   #100 cnt=1;
   #200 cnt=0;
   #400 clear=1;
   end

   always #half_period clk=~clk;
endmodule
```



5.

```
module hw5(
clk,reset,upper,Q
    );
    input clk,reset;
    output [3:0] Q;
    output upper;
    reg [3:0] Q;
    reg upper;

    always @(posedge reset or posedge clk)
    begin
        if (reset==1'b1) Q<=0;
        else Q<=Q+1;
        if ((Q>6)&(Q<15)) upper=1;
        else upper=0;
    end
endmodule
```

```
22
23  module Q5_test;
24  parameter half_period=25;
25  parameter counter_size=4;
26
27  wire [counter_size-1:0] Q;
28  wire upper;
29
30  reg clk,reset;
31
32  hw5 #(counter_size) UUT(clk,reset,upper,Q);
33
34  initial begin
35  #0 clk=0;reset=1;
36  #50 reset=0;
37  end
38
39  always #half_period clk=~clk;
40
```