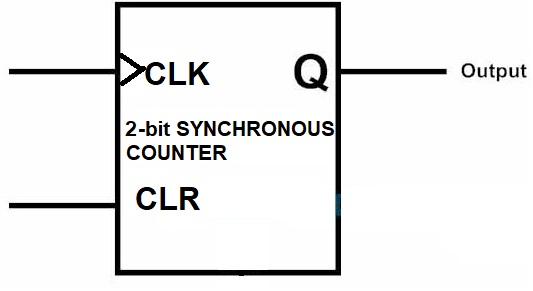
**LAB # 07(Open Ended)**

**Objective:** INTRODUCTION TO BEHAVIOUR LEVEL MODELING.

**Task1:** To design 2-bit synchronous counter in verilog by using Behaviour Level Modeling.

**Circuit diagram:**

**Coding:**

//Design Module

module counter(q,clk,clr);

input clk,clr;

output [1:0]q;

reg [1:0]q;

always @(negedge clk or posedge clr)

begin

if(clr)

q=2'b00;

else

q=q+1;

end

endmodule

//Stimullux Module

module aazib;

reg clk,clr;

wire [1:0]q;

counter c1(q,clk,clr);

initial

clk=1'b0;

always #5

clk=~clk;

initial

begin

clr=1'b1;

#30

clr=1'b0;

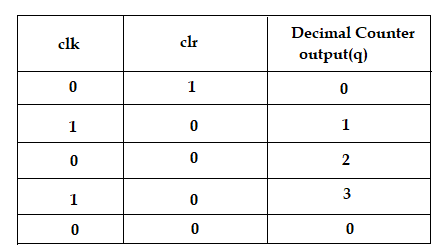
end

initial

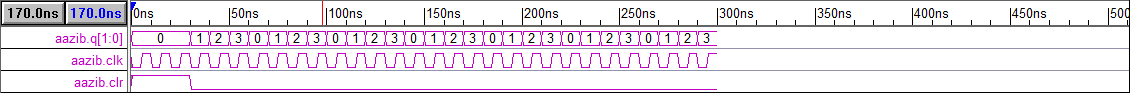
#300

$finish;

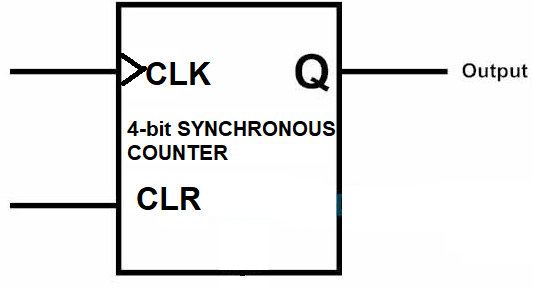
endmodule

**Truth Table:**

**Results (Timing diagram) :**



**Task2:** To design 4-bit synchronous counter in verilog by using Behaviour Level Modeling.

**Circuit diagram:**

**Coding:**

//Design Module

module counter(q,clk,clr);

input clk,clr;

output [3:0]q;

reg [3:0]q;

always @(negedge clk or posedge clr)

begin

if(clr)

q=4'b0000;

else

q=q+1;

end

endmodule

//Stimullux Module

module aazib;

reg clk,clr;

wire [3:0]q;

counter c1(q,clk,clr);

initial

clk=1'b0;

always #5

clk=~clk;

initial

begin

clr=1'b1;

#30

clr=1'b0;

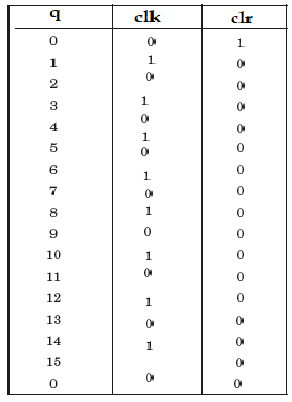
end

initial

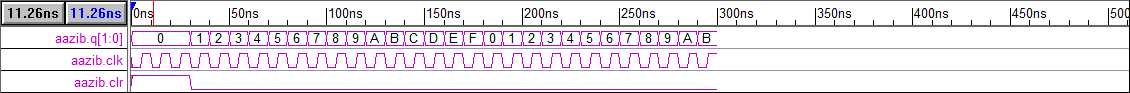
#300

$finish;

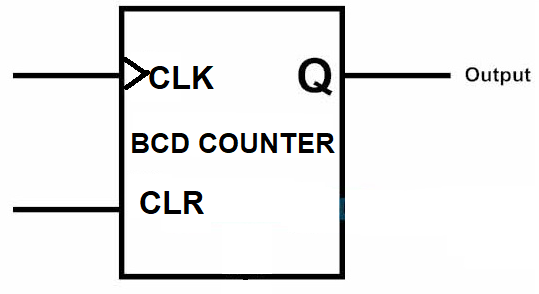
Endmodule

**Truth Table:**

**Results (Timing diagram) :**



**Task3:** To design BCD counter in verilog by using Behaviour Level Modeling.

**Circuit diagram:**

**Coding:**

//Design Module

module counter(q,clk,clr);

input clk,clr;

output [3:0]q;

reg [3:0]q;

always @(negedge clk or posedge clr)

begin

if(clr)

q=4'b0000;

else

q=(q+1)%10;

end

endmodule

//Stimullux Module

module aazib;

reg clk,clr;

wire [3:0]q;

counter c1(q,clk,clr);

initial

clk=1'b0;

always #5

clk=~clk;

initial

begin

clr=1'b1;

#30

clr=1'b0;

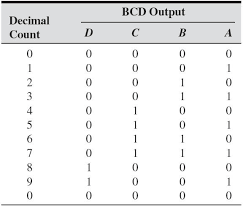
end

initial

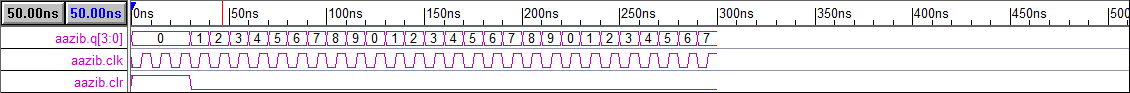
#300

$finish;

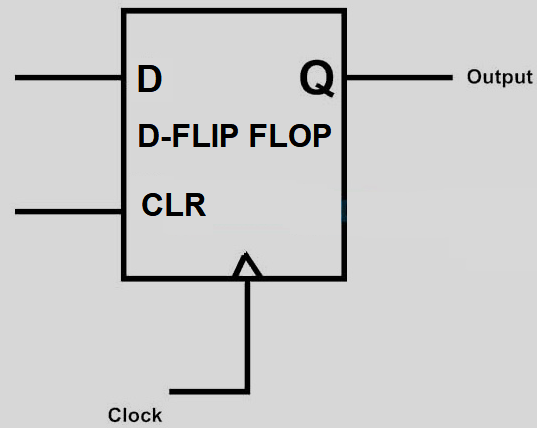
endmodule

**Truth Table:**

**Results (Timing diagram) :**



**Task4:** To design D-flipflop counter in verilog by using Behaviour Level Modeling.

**Circuit diagram:**

**Coding:**

//Design Module

module dff(q,clk,clr,d);

input clk,clr,d;

output q;

reg q;

always @(negedge clk or posedge clr)

begin

if(clr)

q=1'b0;

else

q=d;

end

endmodule

//Stimullux Module

module aazib;

reg clk,clr,d;

wire q;

dff d1(q,clk,clr);

initial

clk=1'b0;

always #5

clk=~clk;

initial

begin

d=1'b1; clr = 1'b1;

#30

d=1'b0; clr = 1'b0;

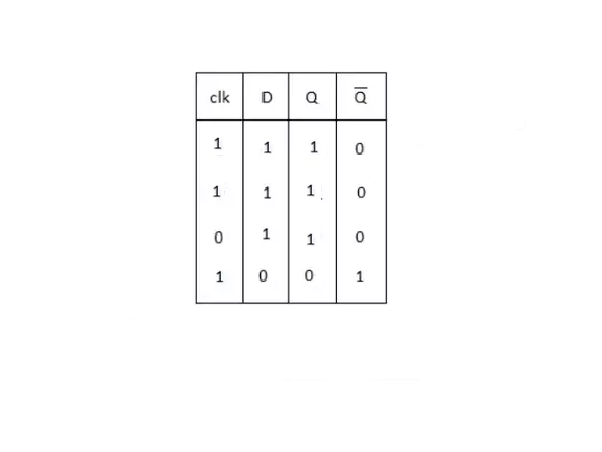
end

initial

#300

$finish;

endmodule

**Truth Table:**

**Results (Timing diagram) :** 