Flip flops, Counters and Registers

Lab Report No. 4 + 5

MV407: IC Design Lab-1 (Digital)

Submitted by:

Full Name	**Enrollment No.**
Abhinav M	23PMMT11(@uohyd.ac.in)

Lab Instructor: Dr. Bhawna Gomber

Teaching Assistant: Ms. Bhargavi

Center for Advanced Studies in Electronics Science and Technology
University of Hyderabad
Hyderabad, India
November 22, 2023

Contents

1 | Asynchronous D Flip flop

1.1 | Objective

Design a circuit for Asynchronous D Flip flop. Verify its functionality using testbench.

1.2 | Block diagram

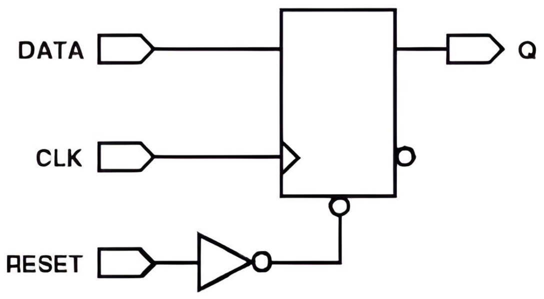

Figure 1.1: Block diagram async DFF

1.3 | Code

1.3.1 | Design

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 10/25/2023 02:52:37 PM
// Design Name:
// Module Name: CB_DFF_async
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////



module CB_DFF_async(input D,clk,rst,en,output reg Q);


always @(posedge clk or posedge rst)
begin
if(~rst & en)
 Q <= D;
else
 Q <= 1'b0;
end

endmodule
```

1.3.2 | Testbench

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 10/25/2023 03:02:13 PM
// Design Name:
// Module Name: tb_CB_DFF_sync
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_CB_DFF_async();
reg D;
reg clk;
wire Q;
reg en;
reg rst;

CB_DFF_async uut(D,clk,rst,en,Q);

initial
    begin
        clk=0;
        rst=0;
        en=1;
        forever #10 clk = ~clk;
        end

initial
    begin
        forever #100 rst = ~rst;
end


initial
    begin

        D <= 0;
        #50;

        D <= 1;
        #50;


        D <= 0;
        #50

        en=0;
        D<= 1;

        #50 $finish;
    end
endmodule
```

1.4 | Elaborated design

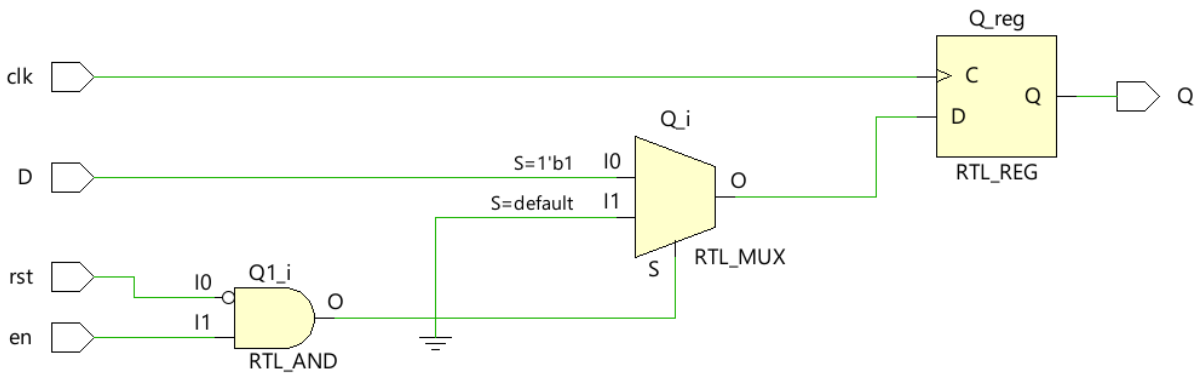

Figure 1.2: Elaborated design

1.5 | Timing diagram

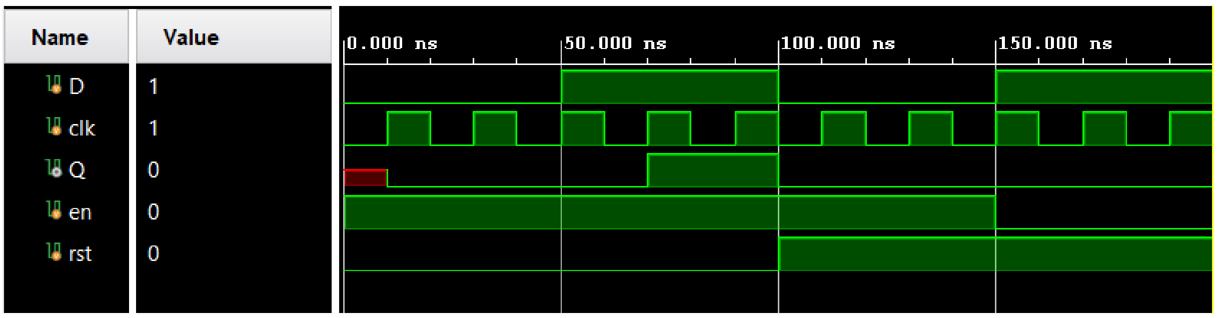

Figure 1.3: Timing diagram

1.6 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

2 | Synchronous D flip flop

2.1 | Objective

Design a circuit for Synchronous D flip flop. Verify its functionality using testbench.

2.2 | Block diagram

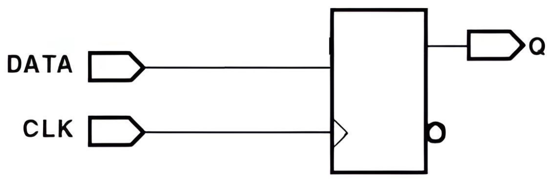

Figure 2.1: Block diagram sync DFF

2.3 | Code

2.3.1 | Design

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 10/25/2023 02:45:45 PM
// Design Name:
// Module Name: CB_DFF_sync
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module CB_DFF_sync(input D,clk,en,output reg Q);


always @(posedge clk & en)
begin
 Q <= D;
end





endmodule
```

2.3.2 | Testbench

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 10/25/2023 03:02:13 PM
// Design Name:
```

```
// Module Name: tb_CB_DFF_sync
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_CB_DFF_sync();
reg D;
reg clk;
wire Q;
reg en;

CB_DFF_sync uut(D,clk,en,Q);

initial
    begin
        clk=0;
        en=1;
        forever #10 clk = ~clk;
    end


initial
    begin

        D <= 0;
        #50;

        D <= 1;
        #50;


        D <= 0;
        #50

        en=0;
        D<= 1;

        #50 $finish;
    end
endmodule
```

2.4 | Elaborated design

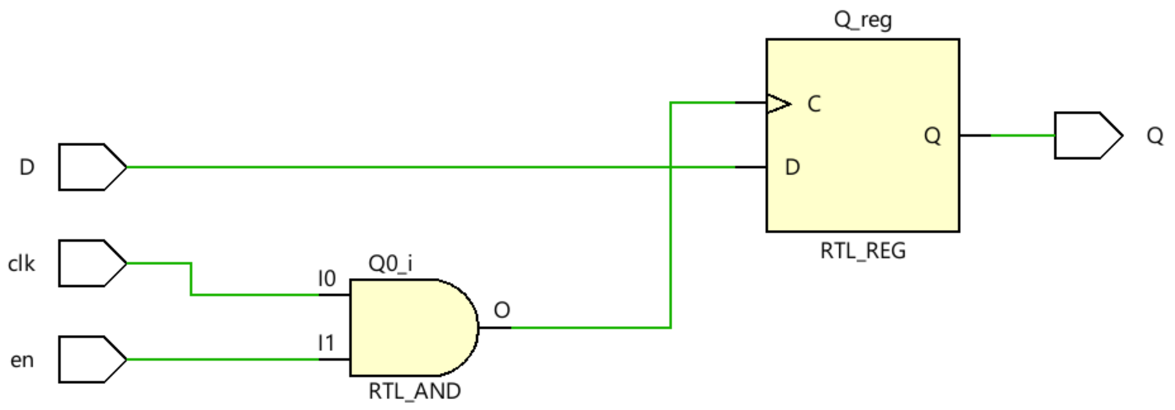

Figure 2.2: Elaborated design

2.5 | Timing diagram

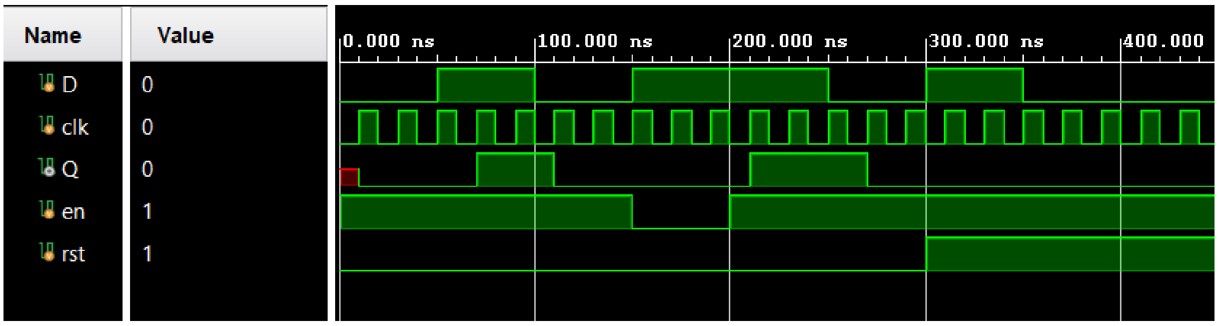

Figure 2.3: Timing diagram

2.6 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

3 | Synchronous D flip flop with reset

3.1 | Objective

Design a circuit for Synchronous D flip flop with reset. Verify its functionality using testbench.

3.2 | Block diagram

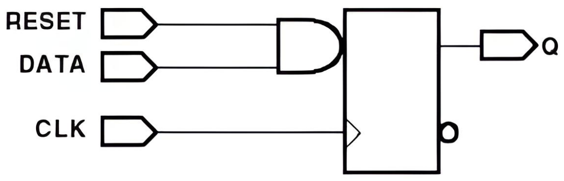

Figure 3.1: Block diagram Sync DFF with rst

3.3 | Code

3.3.1 | Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 10/25/2023 02:49:59 PM
// Design Name:
// Module Name: CB_DFF_syncwrst
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module CB_DFF_syncwrst(input D,clk,rst,en,output reg Q);

always @(posedge clk)
begin
if(~rst & en)
 Q <= D;
else
 Q <= 1'b0;
end

endmodule
```

3.3.2 | Testbench

```

module tb_CB_DFF_syncwrst();
reg D;
reg clk;
wire Q;
reg en;
reg rst;
CB_DFF_syncwrst uut(D,clk,rst,en,Q);
initial
    begin
        clk=0;
        rst=0;
        en=1;
        forever #10 clk = ~clk;
     end
initial
    begin
        D <= 0;
        #50;

        D <= 1;
        #50;

        D <= 0;
        #50

        en=0;

        D <= 1;
        #50;

        en=1;

        D <= 1;
        #50;

        D <= 0;
        #50

        rst=1;

        D <= 1;
        #50;

        D <= 0;
        #50
        #50 $finish;
    end
endmodule
```

3.4 Elaborated design

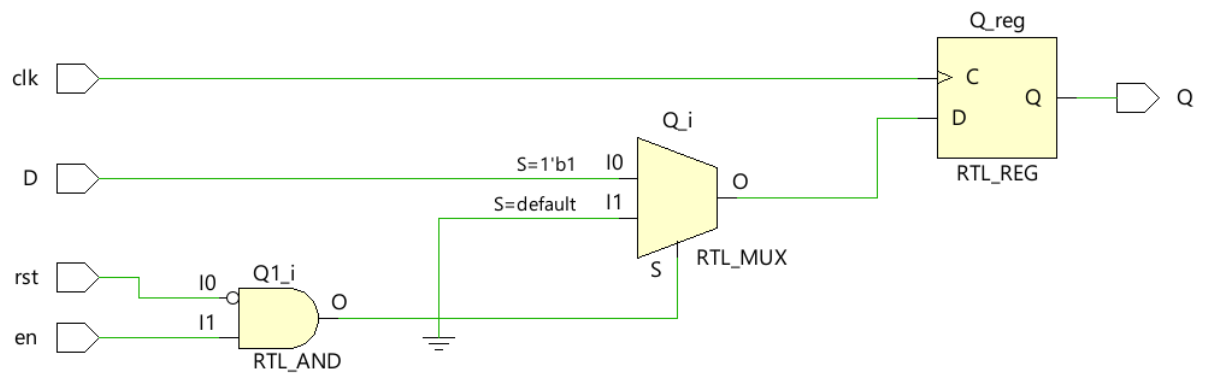

Figure 3.2: Elaborated design

3.5 Timing diagram

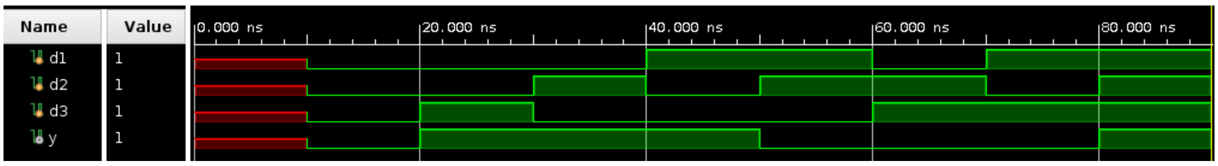

Figure 3.3: Timing diagram.

3.6 Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

4 | 8-bit data register async rst

4.1 | Objective

Design a circuit for 8-bit data register with async rst. Verify its functionality using testbench.

4.2 | Code

4.2.1 | Design

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/01/2023 02:22:58 PM
// Design Name:
// Module Name: DB_8asyncrst
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module DB_8asyncrst (reset, clk, D, Q);
input reset;
input clk;
input [7:0] D;
output [7:0] Q;
reg [7:0] Q;
always @(posedge clk or posedge reset)
if (reset)
Q = 0;
else
Q = D;
endmodule
```

4.2.2 | Testbench

```verilog

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 22.11.2023 21:23:29
// Design Name:
// Module Name: tb_DB_8_asyncrst
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_DB_8_asyncrst();
reg clk,rst;
reg [7:0] D;
wire [7:0] Q;
DB_8asyncrst uut(rst,clk, D, Q);


initial
    begin
        clk=0;
        forever #1 clk = ~clk;
    end
initial
begin
 rst=1; #5

 rst=0; #5

D = 8'b10101010; #5

D = 8'b11111111; #5

D = 8'b01010101; #5

 #150 $finish;
end
endmodule
```

4.3 | Elaborated design

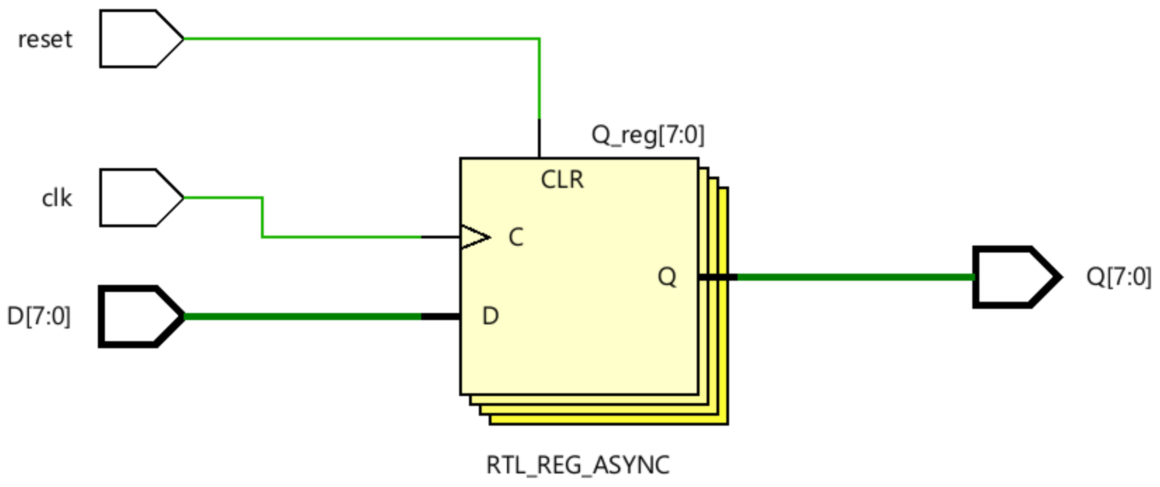

Figure 4.1: Elaborated design

4.4 | Timing diagram

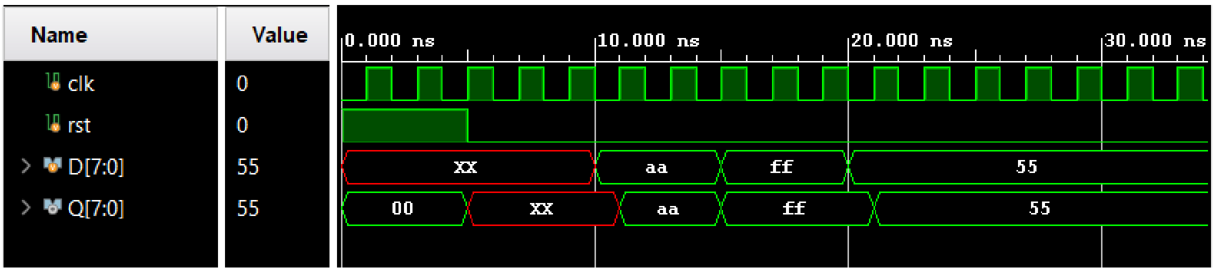

Figure 4.2: Timing diagram

4.5 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

5 | 8-bit data register sync rst

5.1 | Objective

Design a circuit for 8-bit data register sync rst. Verify its functionality using testbench.

5.2 | Code

5.2.1 | Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/01/2023 02:40:10 PM
// Design Name:
// Module Name: DB_8syncrst
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module DB_8syncrst(rst, clk, D, Q);
input rst;
input clk;
input [7:0] D;
output [7:0] Q;
reg [7:0] Q;
always @(posedge clk)
if (rst)
Q = 0;
else
Q = D;
endmodule
```

5.2.2 | Testbench

```

`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 22.11.2023 21:23:29
// Design Name:
// Module Name: tb_DB_8_syncrst
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_DB_8_syncrst();
reg clk,rst;
reg [7:0] D;
wire [7:0] Q;
DB_8syncrst uut(rst,clk, D, Q);


initial
    begin
        clk=0;
        forever #1 clk = ~clk;
    end
initial
begin
 rst=1; #5

 rst=0; #5

D = 8'b10101010; #5

D = 8'b11111111; #5

D = 8'b01010101; #5

 #150 $finish;
end
endmodule
```

5.3 | Elaborated design

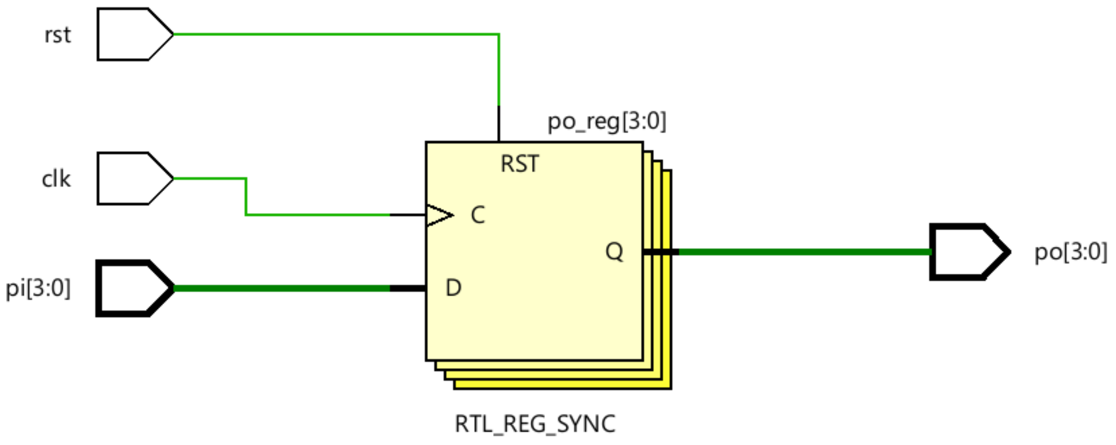

Figure 5.1: Elaborated design

5.4 | Timing diagram

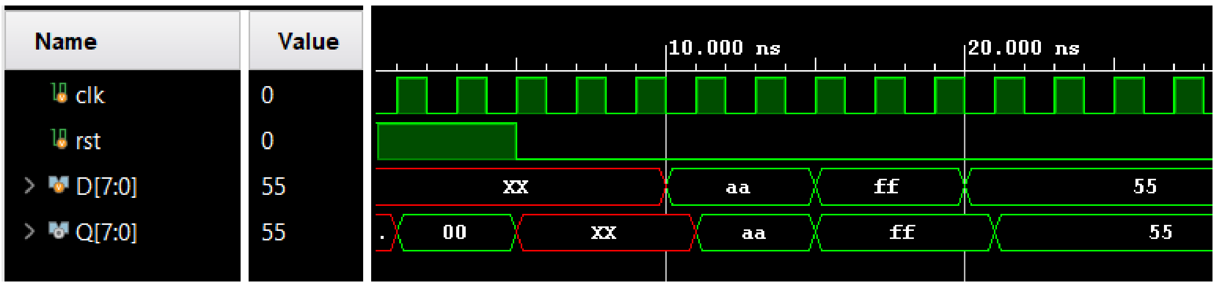

Figure 5.2: Timing diagram

5.5 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

6 | 4-bit PIPO register

6.1 | Objective

Design a circuit for 4-bit PIPO register. Verify its functionality using testbench.

6.2 | Block diagram

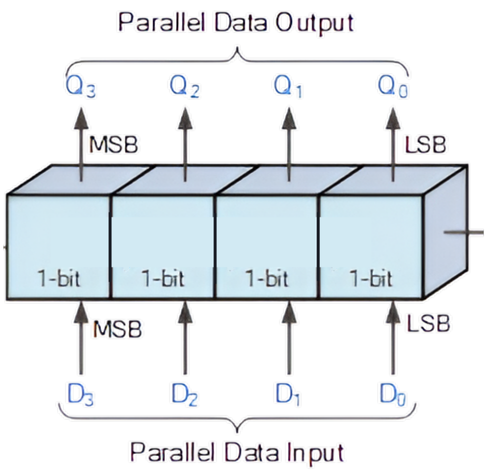

Figure 6.1: 4-bit PIPO register.

6.3 | Code

6.3.1 | Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/08/2023 02:50:43 PM
// Design Name:
// Module Name: E_4_PIPO
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module E_4_PIPO(input clk,rst, [3:0]pi, output reg [3:0]po);

always @(posedge clk)
begin
if (rst)
po<= 4'b0000;
else
po <= pi;
end
endmodule
```

6.3.2 | Testbench

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 22.11.2023 21:23:29
// Design Name:
// Module Name: tb_E_4_PIPO
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_E_4_PIPO();
reg clk,rst;
reg [3:0] pi;
wire [3:0] po;
E_4_PIPO uut(clk,rst, pi, po);


initial
    begin
        clk=0;
        forever #1 clk = ~clk;
    end
initial
begin
 rst=1; #5

 rst=0; #5

 pi = 4'b1010; #5

 pi = 4'b1111; #5

 pi = 4'b0101; #5

 #150 $finish;
end
endmodule
```

6.4 | Elaborated design

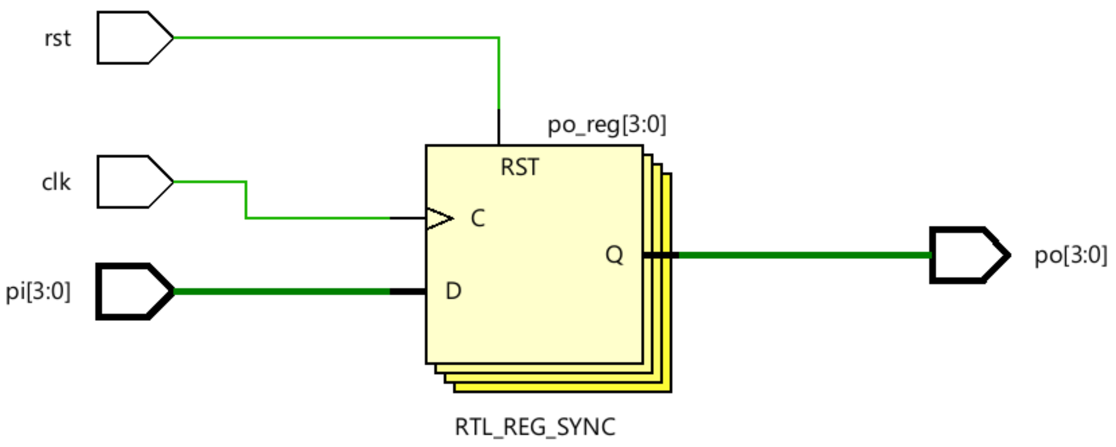

Figure 6.2: Elaborated design

6.5 | Timing diagram

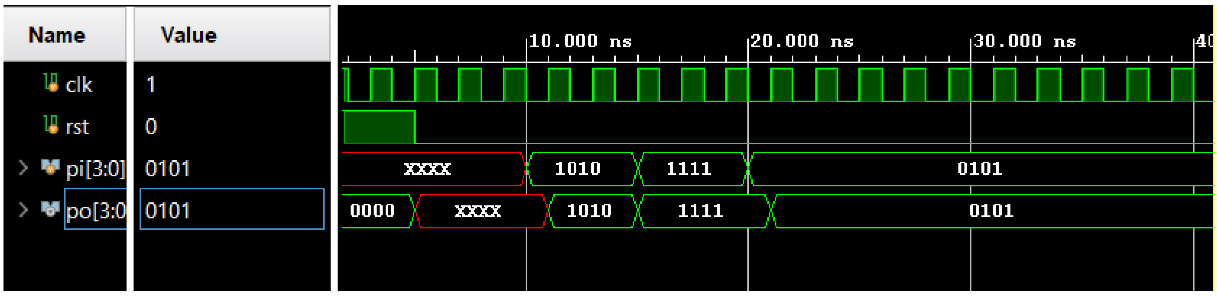

Figure 6.3: Timing diagram

6.6 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

7 | 4-bit SISO register

7.1 | Objective

Design a circuit for 4-bit SISO register. Verify its functionality using testbench.

7.2 | Block diagram

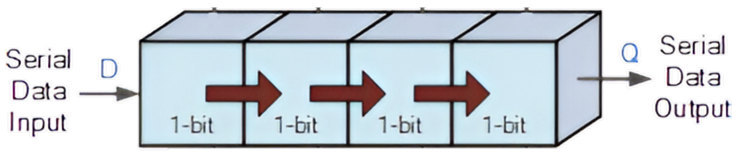

Figure 7.1: 4-bit SISO working (L→ R)

7.3 | Code

7.3.1 | Design

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/08/2023 02:44:22 PM
// Design Name:
// Module Name: E_4_SISO
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module E_4_SISO(input clk,rst,si,output reg so);

reg [3:0] tmp;

always @(posedge clk )
begin

if (rst)
tmp <= 4'b0000;


else
tmp <= tmp << 1;

tmp[0] <= si;
so = tmp[3];

end
endmodule
```

7.3.2 | Testbench

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 22.11.2023 21:23:29
// Design Name:
// Module Name: tb_E_4_PIPO
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_E_4_SISO();
reg clk,rst;
reg si;
wire so;
E_4_SISO uut(clk,rst, si, so);
initial
    begin
        clk=0;
        forever #5 clk = ~clk;
    end
initial
begin
 rst=1; #10

 rst=0; #10

 si = 1'b1; #5

 si = 1'b1; #5

 si = 1'b0; #5

  si = 1'b0; #5

 si = 1'b1; #5

 si = 1'b1; #5

  si = 1'b0; #5

 si = 1'b0; #5

 si = 1'b1; #5

  si = 1'b1; #5

 si = 1'b1; #5

 si = 1'b1; #5

  si = 1'b1; #5

 si = 1'b1; #5

 si = 1'b1; #5

  si = 1'b1; #5

 si = 1'b1; #5

 si = 1'b1; #5
 #100 $finish;
end
endmodule
```

7.4 | Elaborated design

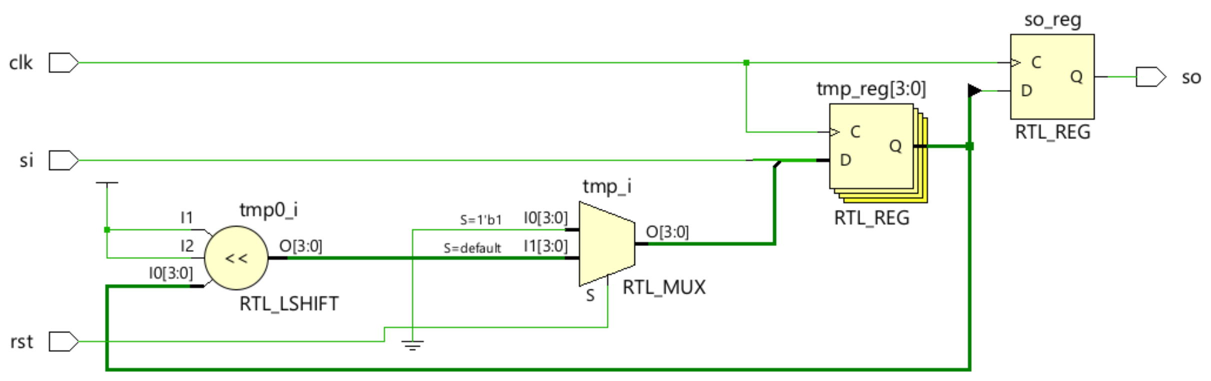

Figure 7.2: Elaborated design

7.5 | Timing diagram

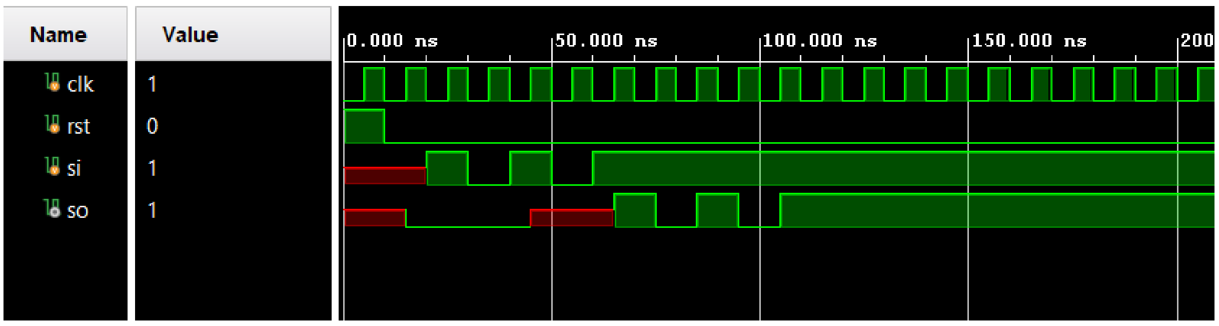

Figure 7.3: Timing diagram

7.6 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

8 | 4-bit Universal Shift Register

8.1 | Objective

Design a circuit for 4-bit Universal Shift Register. Verify its functionality using testbench.

8.2 | Block diagram

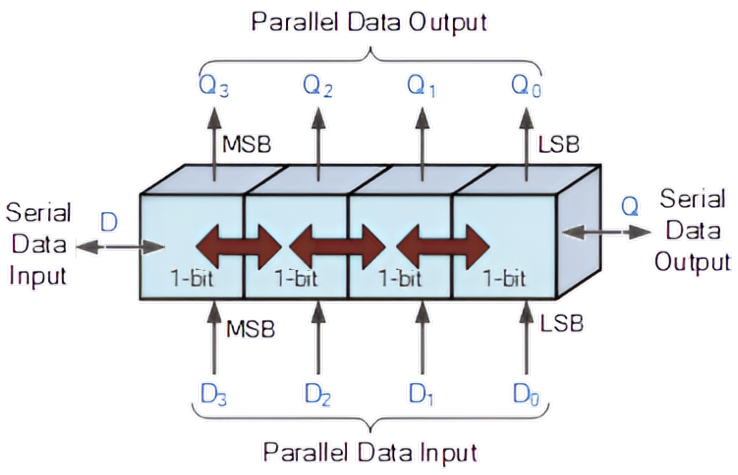

Figure 8.1: 4-bit universal shift register.

8.3 | Code

8.3.1 | Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/08/2023 04:48:39 PM
// Design Name:
// Module Name: E_4_USR
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module E_4_USR(
  input clk, rst,
  input [1:0] sel, // select
  input [3:0] pdi,  // parallel data in
  input sldi,   // serial left data in
  input srdi,  // serial right data in
  output reg [3:0] pdo, //parallel data out
  output reg [3:0] do,
  output sldo, // serial left data out
  output srdo // serial right data out
);
  always@(posedge clk)
  begin
    if(rst)
    begin
    pdo<=0;
    do<=0;
  end
    else begin
```

```verilog
      case(sel)
        2'h1: do <= {srdi,pdo[3:1]}; // Right Shift
        2'h2: do <= {pdo[2:0],sldi};  // Left Shift
        2'h3: pdo <= pdi; // PIPO
        default: do <= do; // Do nothing
      endcase
    end
  end
  assign sldo = do[0];
  assign srdo =do[3];
endmodule
```

8.3.2 | Testbench

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 22.11.2023 22:06:39
// Design Name:
// Module Name: tb_E_4_USR
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_E_4_USR;
  reg clk, rst;
  reg [1:0] sel;
  reg [3:0] pdi;
  reg sldi, srdi;
  wire [3:0] pdo; //parallel data out
  wire sldo, srdo;

  E_4_USR uut(clk, rst, sel, pdi, sldi, srdi, pdo, sldo, srdo);

  always #2 clk = ~clk;
  initial begin

    clk = 0; rst = 1;
    #3 rst = 0;
    #5

    pdi = 4'b1001;
    sel = 2'h3; #4;



   #100 $finish;
  end


endmodule
```

8.4 | Elaborated design

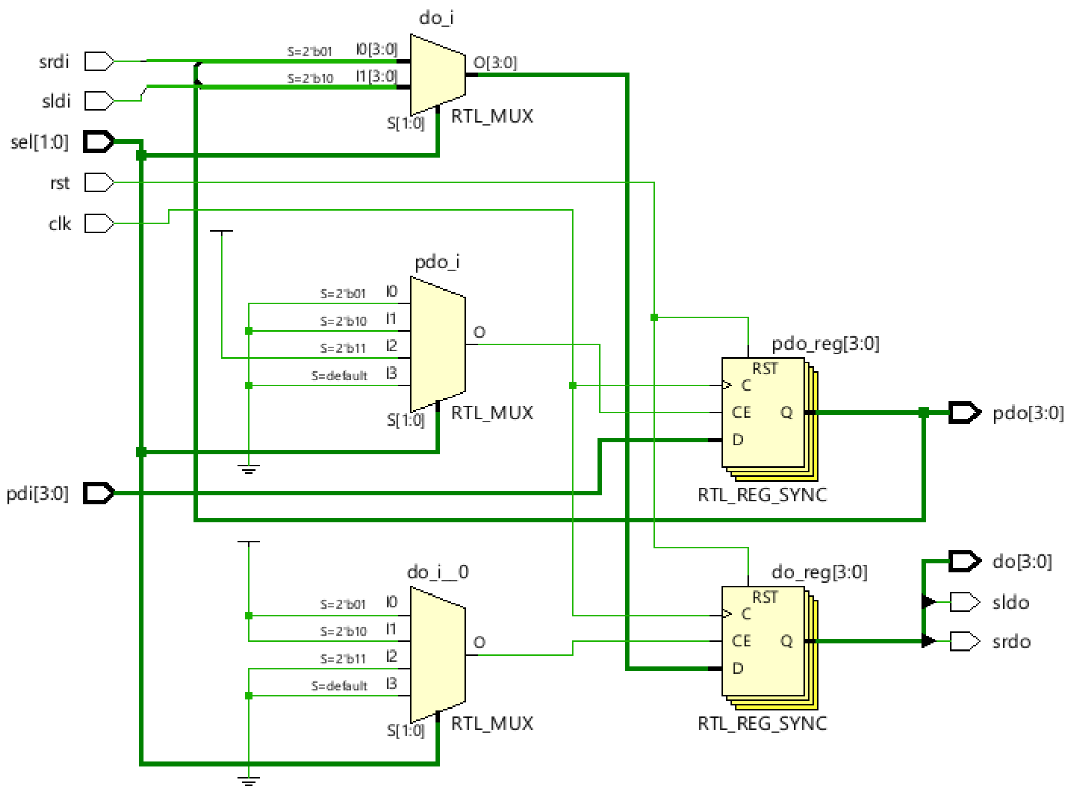

Figure 8.2: Elaborated design

8.5 | Timing diagram

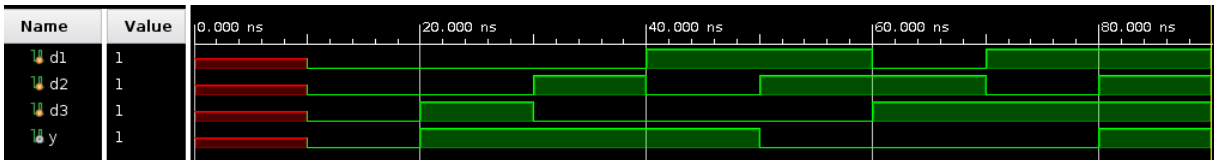

Figure 8.3: Timing diagram.

8.6 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

9 | 8-bit sync up-counter

9.1 | Objective

Design a circuit for 8-bit sync up-counter. Verify its functionality using testbench.

9.2 | Code

9.2.1 | Design

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/08/2023 03:20:20 PM
// Design Name:
// Module Name: E_8_upcount_sync
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module E_8_upcount_sync(input clk,output reg [7:0]count);
initial
count=0;
always @(posedge clk )
begin
if(count<8'hff)
count<= count+8'h01;
else
count=0;
end
endmodule
```

9.2.2 | Testbench

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 11/08/2023 03:33:33 PM
// Design Name:
// Module Name: tb_E_8_downcount_async
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_E_8_upcount_async();
wire [7:0]count ;
reg clk;
E_8_upcount_sync uut(clk,count);
initial
    begin
        clk=0;
        forever #1 clk = ~clk;
    end
#500 $finish;
end
endmodule
```

9.3 | Elaborated design

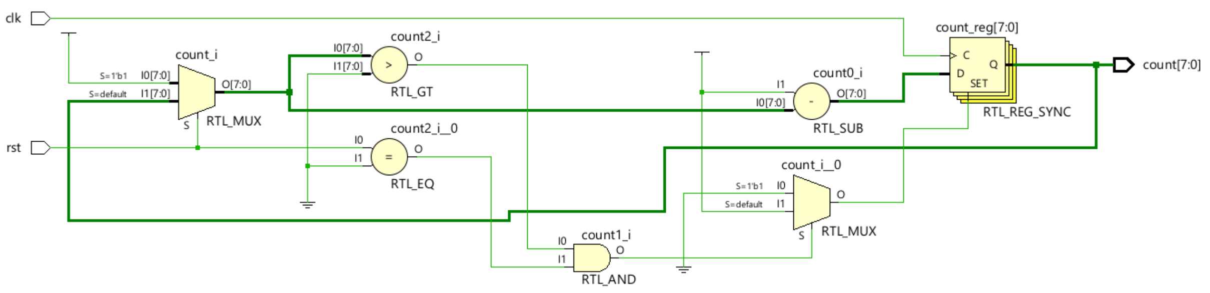

Figure 9.1: Elaborated design

9.4 | Timing diagram

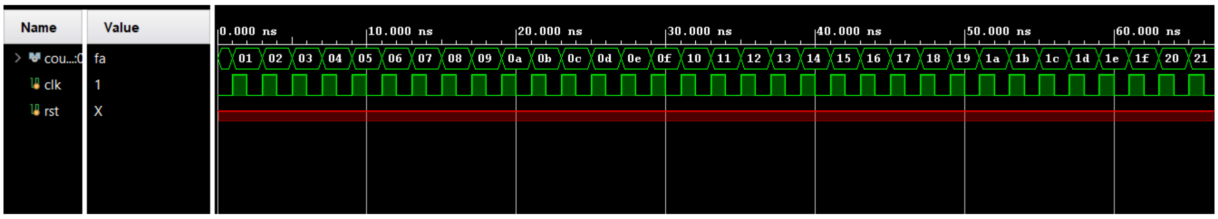
Name
Value
cou...:0
fa
clk
1
rst
X
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21

Figure 9.2: Timing diagram

9.5 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

10 | 8-bit async down-counter

10.1 | Objective

Design a circuit 8-bit async down-counter. Verify its functionality using testbench.

10.2 | Code

10.2.1 | Design

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company : CASEST, University of Hyderabad:
// Author : Abhinav M
//
// Create Date: 11/08/2023 03:31:05 PM
// Design Name:
// Module Name: E_8_downcount_async
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module E_8_downcount_async(input clk,rst, output reg [7:0]count);
initial
count=8'b11111111;

always @(posedge clk or posedge rst)
begin
if(rst)
count=8'b11111111;

if(count>8'b00000000 & rst==0)
count<= count-8'b00000001;

else if(count==8'b00000000);
count=8'b11111111;
end





endmodule
```

10.2.2 | Testbench

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Author: Abhinav M
//
// Create Date: 11/08/2023 03:33:33 PM
// Design Name:
// Module Name: tb_E_8_downcount_async
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module tb_E_8_downcount_async();
wire [7:0]count ;
reg clk,rst;
```

```
E_8_downcount_async uut(clk,rst,count);

initial
    begin
        clk=0;
        forever #1 clk = ~clk;

    end

initial
begin
rst=0;
#20

rst=1;

#20

rst=0;
#20

rst=1;
#20

rst=0;
#20

#500 $finish;

end
endmodule
```

10.3 | Elaborated design

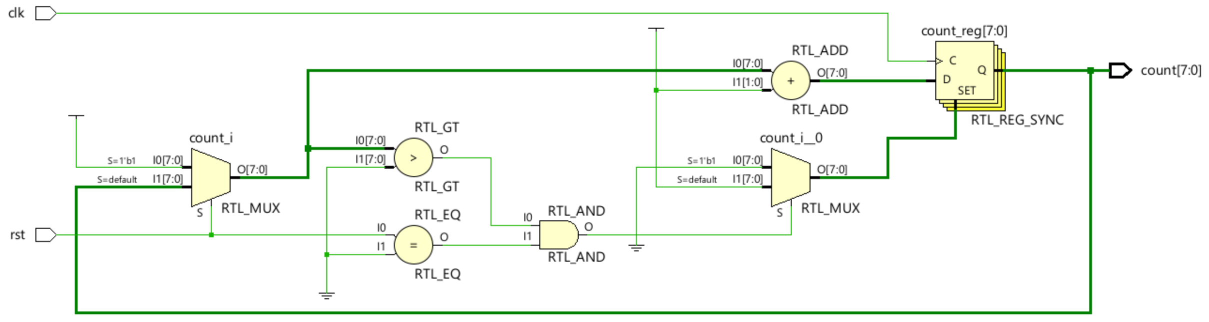

Figure 10.1: Elaborated design

10.4 | Timing diagram

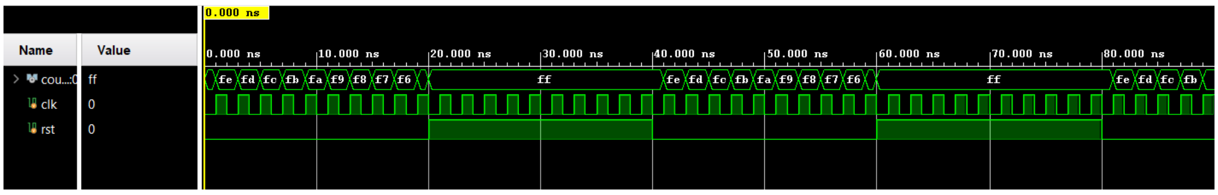

Figure 10.2: Timing diagram

10.5 | Remarks

Circuit was designed and tested successfully, expected functionality was achieved.

11 | References

A | Appendix A: GitHub Repo

The GitHub repo referenced below contains the Verilog code and testbenches mentioned in this report.
https://github.com/AbhinavM2000/MV401D/