PES UNIVERSITY

FINAL SEMESTER ASSESSMENT (FSA)
B.TECH. (CSE)
III SEMESTER

UE18CS206 – DIGITAL DESIGN & COMPUTER ORGANIZATION LABORATORY

PROJECT REPORT

ON

Design and implement a 4 bit register with four d flip flops and four 4:1 multiplexers with mode selection input s0 and s1. The register operates according to the select lines: if s1s0=00 then no change, s1s0=01 then complement output, s1s0=10 then shift right, s1s0=11 then shift left.

SUBMITTED BY

TEAM MEMBERS

Akash Yadav
Dhanik Shetty
Naveen Suresh
Saiprakash L Shetty

AUGUST – DECEMBER 2019

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

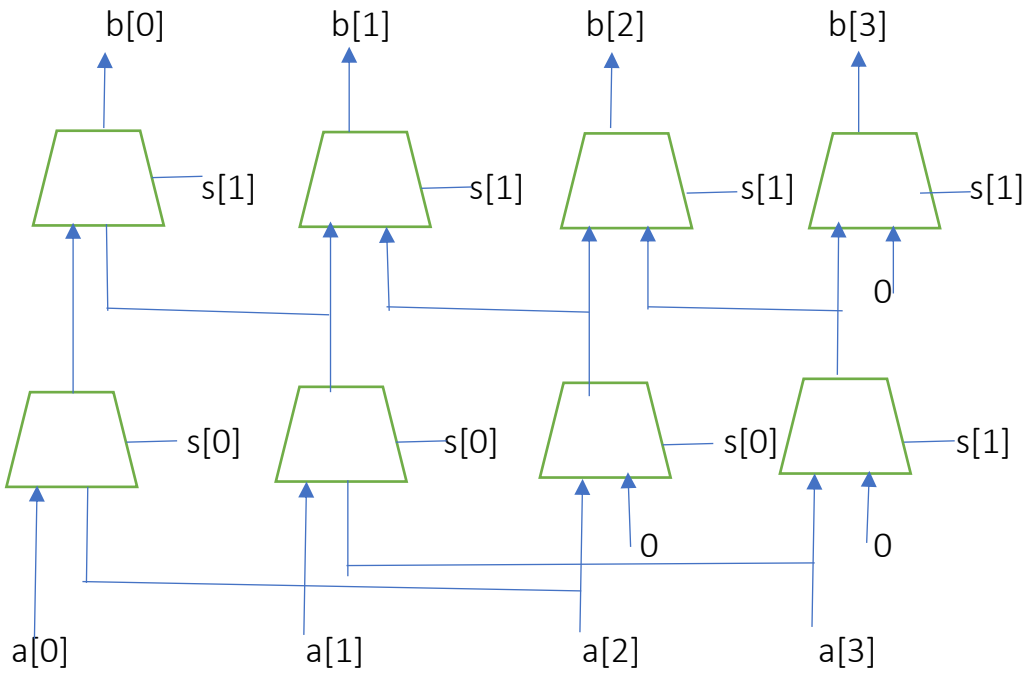
TABLE OF CONTENTS		
SL.No	TOPIC	PAGE No
1.	ABSTRACT OF THE PROJECT	1
2.	CIRCUIT DIAGRAM	2-5
3.	MAIN VERILOG CODE	6-8
4.	TEST BENCH FILE	9
5.	SCREEN SHOTS OF THE OUTPUT	10

1) **ABSTRACT OF THE PROJECT:**

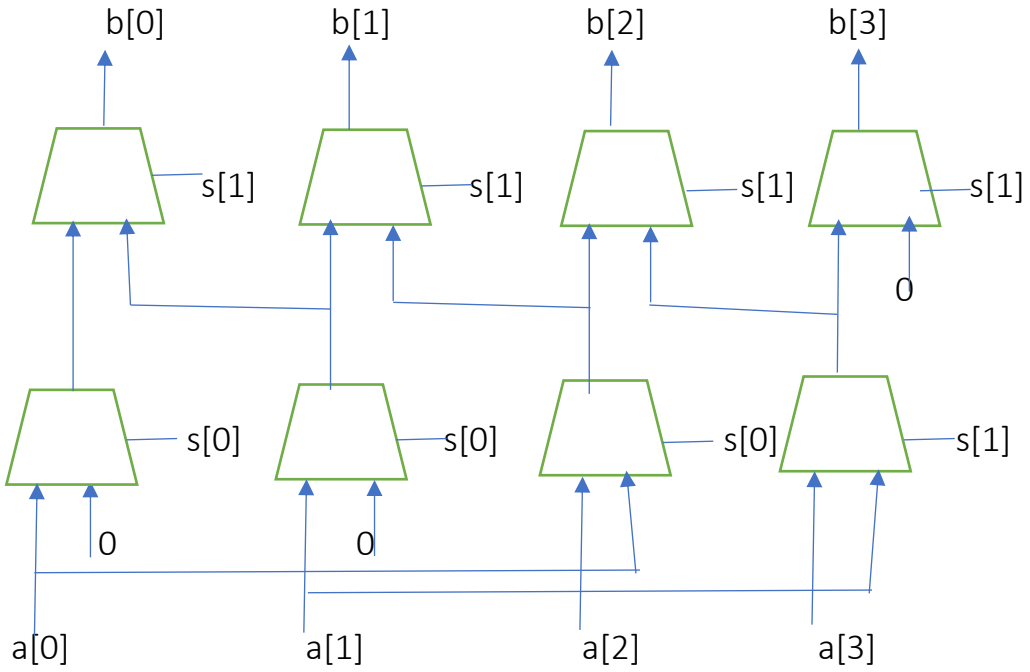
The input for the register is passed in the testbench. According to the selection lines the shifting operations are performed. Initially the inputs are passed to 2:1 multiplexers and later to the D flip flop with reset and load. The output from these D flip flops are inserted to 'process' module where the left shift, right shift and complement output operations are performed. For the left and right shift operations a total of sixteen 2:1 multiplexers are used (8 for left shift, 8 for right shift). The complement operation is implemented using not gates. The 4 bit outputs from these operations are passed to a 4:1 mux whose output will be based on the mode selection inputs.

2) **CIRCUIT DIAGRAM**

(i) left shift

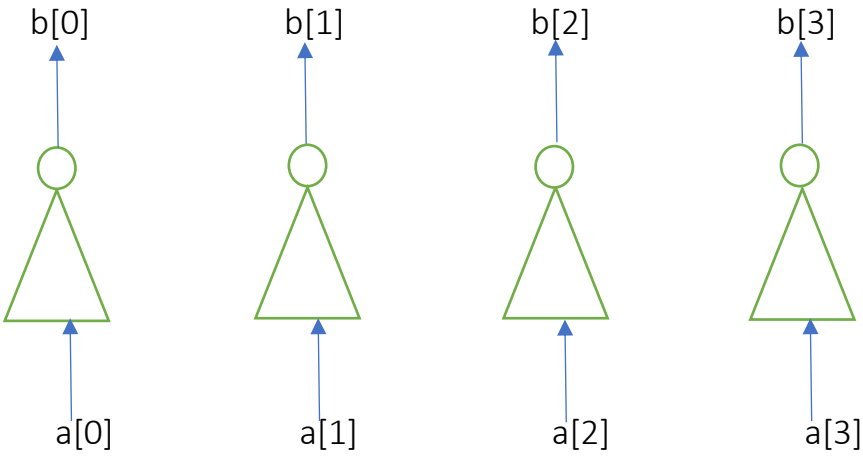


(ii) right shift

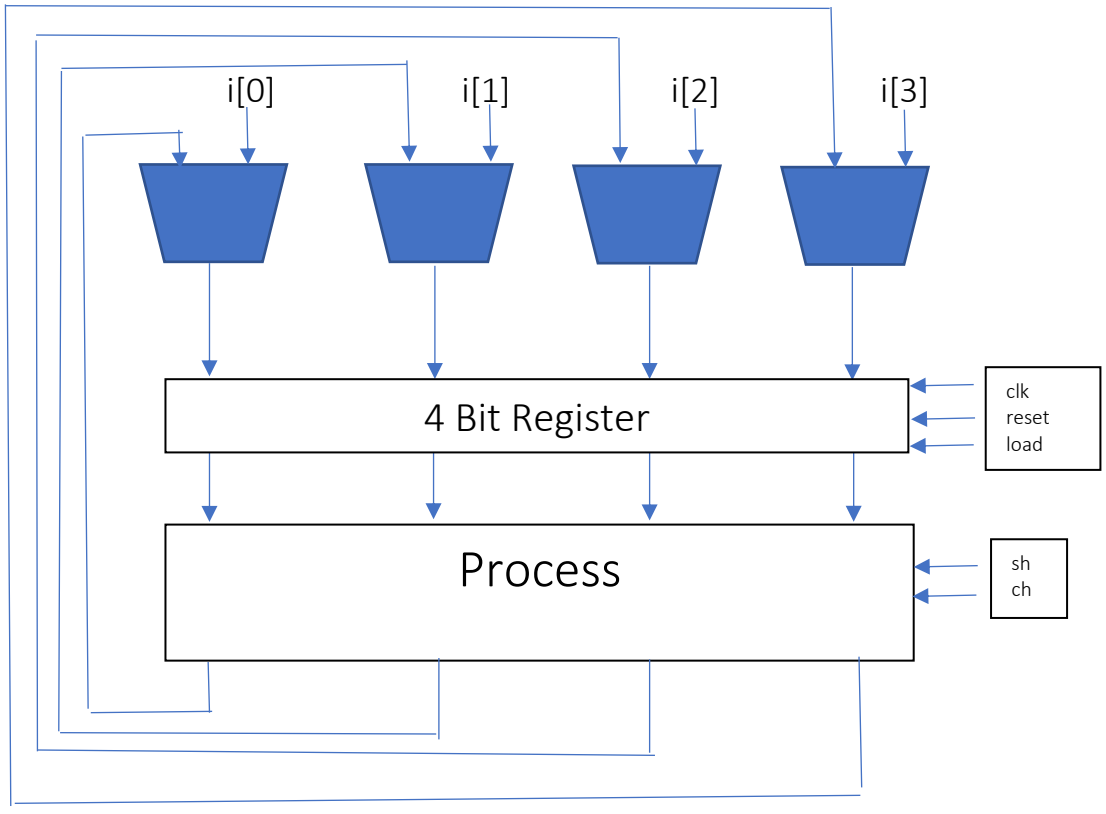


(3)

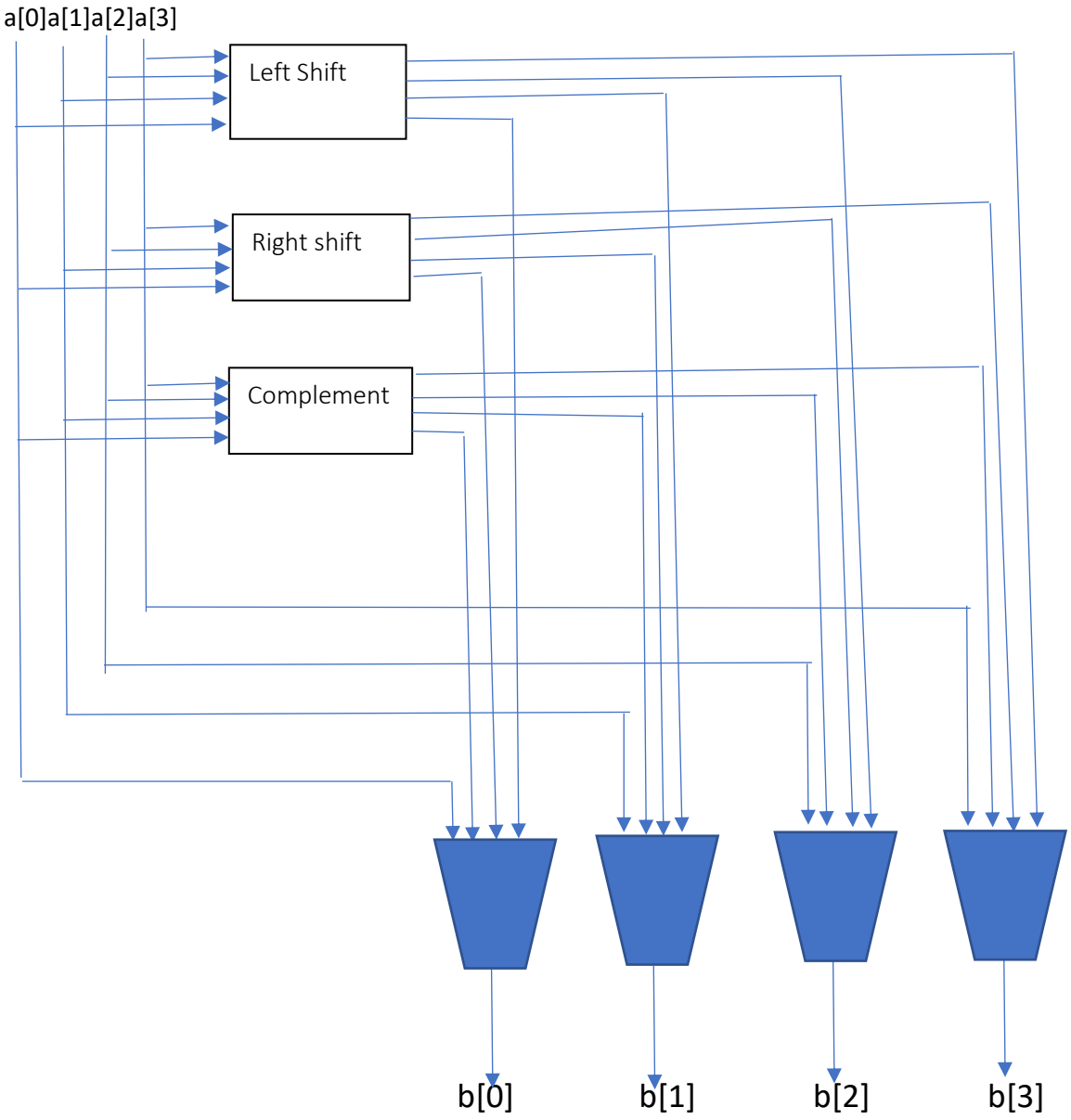
(iii) Complement



(iv) ddco_project module



(v) process module



3) **MAIN VERILOG CODE**

(i) proj.v

```
module mux4x4 (input wire i1,i2,i3,i4, input wire j0, j1, output
wire o);
      wire  t0, t1;
      mux2 mux2_0 (i1, i2, j1, t0);
      mux2 mux2_1 (i3, i4, j1, t1);
      mux2 mux2_2 (t0, t1, j0, o);
endmodule

module left_shifter(input wire [0:3]a,input wire [0:1]s,output wire
[0:3]b);
      wire [0:3]w;
      wire f;
      assign f=1'b0;
      mux2 m1(a[3],f,s[0],w[3]);
      mux2 m2(a[2],f,s[0],w[2]);
      mux2 m3(a[1],a[3],s[0],w[1]);
      mux2 m4(a[0],a[2],s[0],w[0]);
      mux2 m5(w[3],f,s[1],b[3]);
      mux2 m6(w[2],w[3],s[1],b[2]);
      mux2 m7(w[1],w[2],s[1],b[1]);
      mux2 m8(w[0],w[1],s[1],b[0]);
endmodule

module right_shifter(input wire [0:3]a,input wire [0:1]s,output wire
[0:3]b);
      wire w[0:3];
      wire f;
      assign f=1'b0;
      mux2 m1(a[3],a[1],s[0],w[3]);
      mux2 m2(a[2],a[0],s[0],w[2]);
      mux2 m3(a[1],f,s[0],w[1]);
      mux2 m4(a[0],f,s[0],w[0]);
      mux2 m5(w[3],a[2],s[1],b[3]);
      mux2 m6(w[2],a[1],s[1],b[2]);
      mux2 m7(w[1],a[0],s[1],b[1]);
      mux2 m8(w[0],f,s[1],b[0]);
endmodule

module compliment(input wire [0:3]a,output wire [0:3]b);
      not(b[0],a[0]);
      not(b[1],a[1]);
      not(b[2],a[2]);
      not(b[3],a[3]);
```

```
endmodule

module register(input wire clk,reset,load,input wire [0:3]in,output
wire [0:3]out);
      dfrl m1(clk,reset,load,in[0],out[0]);
      dfrl m2(clk,reset,load,in[1],out[1]);
      dfrl m3(clk,reset,load,in[2],out[2]);
      dfrl m4(clk,reset,load,in[3],out[3]);
endmodule

module process(input wire [0:3]a,input wire [0:1]s,input wire
[0:1]s1,output wire [0:3]b);
      wire [0:3]x;wire [0:3]y;wire [0:3]z;
      left_shifter m11(a,s1,x);
      right_shifter m12(a,s1,y);
      compliment m13(a,z);
      mux4x4 m1(a[0],z[0],y[0],x[0],s[0],s[1],b[0]); //here
b0,b1,b2,b3 is the out which comes in the gtkwave
      mux4x4 m2(a[1],z[1],y[1],x[1],s[0],s[1],b[1]);
      mux4x4 m3(a[2],z[2],y[2],x[2],s[0],s[1],b[2]);
      mux4x4 m4(a[3],z[3],y[3],x[3],s[0],s[1],b[3]);
endmodule

module ddco_project(input wire clk,reset,load,input wire
[0:3]in,input wire [0:1]ch,input wire [0:1]sh,input wire rg,output
wire [0:3]out);
      wire [0:3]x;wire [0:3]y;
      mux2 m1(in[0],out[0],rg,x[0]);
      mux2 m2(in[1],out[1],rg,x[1]);
      mux2 m3(in[2],out[2],rg,x[2]);
      mux2 m4(in[3],out[3],rg,x[3]);
      register m6(clk,reset,load,x,y);
      process m5(y,ch,sh,out);
endmodule
```

(ii) lib_main.v

```
module invert (input wire i, output wire o);
   assign o = !i;
endmodule

module and2 (input wire i0, i1, output wire o);
  assign o = i0 & i1;
endmodule

module mux2 (input wire i0, i1, j, output wire o);
  assign o = (j==0)?i0:i1;
endmodule

module df (input wire clk, in, output wire out);
  reg df_out;
  always@(posedge clk) df_out <= in;
  assign out = df_out;
endmodule

module dfr (input wire clk, reset, in, output wire out);
  wire reset_, df_in;
  invert invert_0 (reset, reset_);
  and2 and2_0 (in, reset_, df_in);
  df df_0 (clk, df_in, out);
endmodule

module dfrl (input wire clk, reset, load, in, output wire out);
  wire _in;
  dfr dfr_1(clk, reset, _in, out);
  mux2 mux2_0(out, in, load, _in);
endmodule
```

4) **TEST BENCH FILE**

tb_project.v

```
module tb;
reg clk,reset,load,rg;
reg [1:0]ch;
reg [1:0]sh;
reg [3:0]in;
wire [3:0]out;
ddco_project m1(clk,reset,load,in,ch,sh,rg,out);
initial
begin
$dumpfile("proj.vcd");
$dumpvars(1,tb);
//$monitor(clk,reset,load,in,ch,sh,rg,out);
end
initial begin reset = 1'b0;end
initial begin load = 1'b1;end
initial clk = 1'b1; always #10 clk =~ clk;
initial begin
in=4'b0010;ch=2'b00;sh=2'b00;rg=1'b0; //Input number is 2 (0010)
#20 in=4'b0010;ch=2'b00;sh=2'b00;rg=1'b0;
#20 in=4'bxxxx;ch=2'b01;sh=2'b01;rg=1'b1;
#20 in=4'bxxxx;ch=2'b10;sh=2'b01;rg=1'b1;
#20 in=4'bxxxx;ch=2'b11;sh=2'b01;rg=1'b1;
end
initial #200 $finish;
endmodule
```

5) **SCREEN SHOTS OF THE OUTPUT:**

C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.864]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\DDCO PROJECT>iverilog -o test lib_main.v proj.v tb_project.v

D:\DDCO PROJECT>vvp test
VCD info: dumpfile proj.vcd opened for output.

D:\DDCO PROJECT>gtkwave proj.vcd
```