EX. NO: 05	**SHIFT REGISTER**
DATE : 04/10/2024	

AIM: To implement flipflops in Verilog HDL and verify their functionalities through behavioral simulation.

SOFTWARE USED: Xilinx Vivado

HARDWARE USED: Basys3 FPGA Board

SHIFT REGISTER:

VERILOG HDL CODE:

Shift register in behavioral:

```
module shiftregister_behav(I,s,A, clk, clear, SIL, SIR);
input [3:0]I;
input [1:0]s;
input clk, clear, SIL, SIR;
output reg [3:0]A;
always@(posedge clk)
begin
if (~clear)
   A = 4'b0000;
else
   case(s)
   2'b00:A = A;
   2'b01:A = {SIR, A[3:1]};
   2'b10:A = {A[2:0], SIL};
   2'b11:A=I;
   endcase
end
endmodule
```

Shift register in structural:

```
module shiftregister_struct(I,A,sel,clear,clock,SL,SR);
input [3:0]I;
input [1:0]sel;
input clear,clock,SL,SR;
output [3:0]A;
wire [3:0]w;
four_mux M1({I[0],SL,A[1],A[0]},sel,w[0]);
four_mux M2({I[1],A[0],A[2],A[1]},sel,w[1]);
four_mux M3({I[2],A[1],A[3],A[2]},sel,w[2]);
four_mux M4({I[3],A[2],SR,A[3]},sel,w[3]);
```

```
d_flipflop D1(A[0],w[0],clock,clear);
d_flipflop D2(A[1],w[1],clock,clear);
d_flipflop D3(A[2],w[2],clock,clear);
d_flipflop D4(A[3],w[3],clock,clear);
endmodule
module d_flipflop(Q,D,clk,clear);
input D,clk,clear;
output reg Q;
always @(posedge clk)
begin
   if (clear == 0 )
   begin
   Q = 1'b0;
   //Qbar=~Q;
   end
   else
   begin
      Q = D;
    //Qbar=~Q;
   end
end
endmodule
module  four_mux(data_in,sel,data_out);
   input [3:0] data_in;
   input [1:0] sel;
   output data_out;
   wire not_sel0, not_sel1;
   wire x, y, z,w;
   not (not_sel0, sel[0]);
   not (not_sel1, sel[1]);
   and(x, data_in[0], not_sel1, not_sel0);
   and (y, data_in[1], not_sel1, sel[0]);
   and (z, data_in[2], sel[1], not_sel0);
   and (w,data_in[3], sel[1], sel[0]);
   or (data_out, x, y, z,w);
endmodule
```

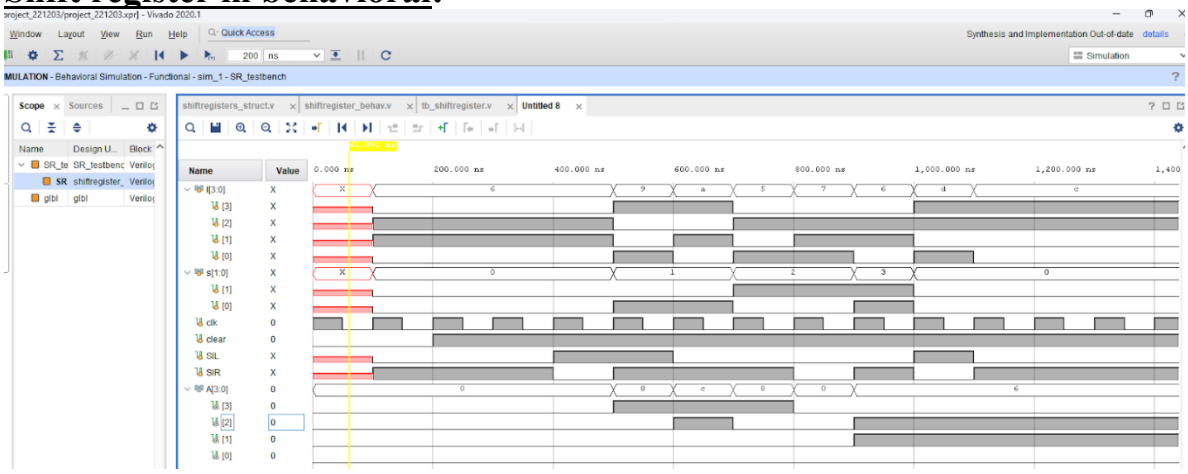
Test bench for shift register:

```
module SR_testbench;
reg [3:0]I;
reg [1:0]s;
reg clk, clear, SIL, SIR;
wire [3:0]A;
shiftregister_struct SR1(I,A,s,clear,clk,SIL, SIR);
//shiftregister_behav SR2(I,s,A, clk, clear, SIL, SIR);
always #50 clk = ~clk;
initial
```

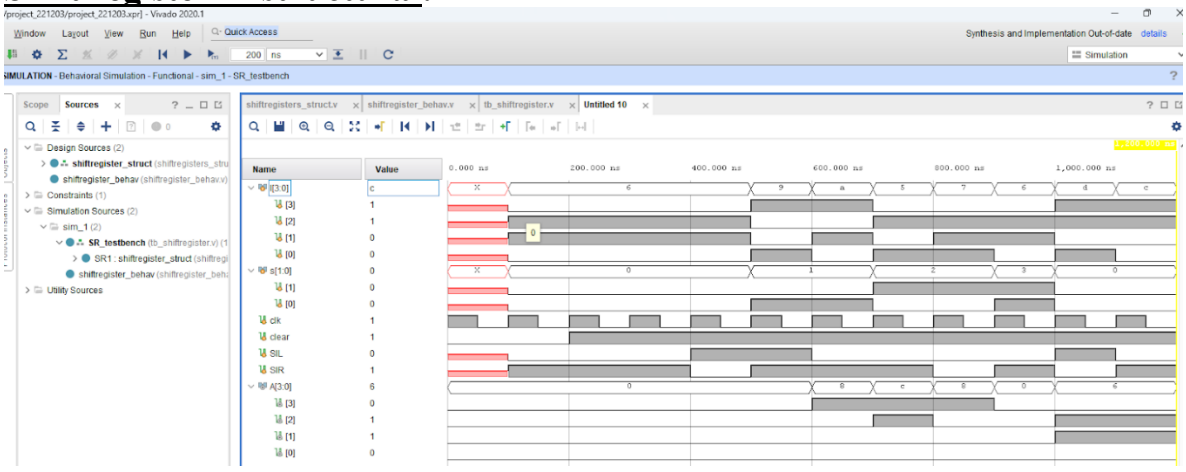
```
begin
    clear = 0; clk=1;
    #100 s=2'b00; I=4'b0110; SIL=0; SIR=1;
    #100 clear = 1; s=2'b00; I=4'b0110; SIL=0; SIR=1;
    #100 s=2'b00; I=4'b0110; SIL=0; SIR=1;
    #100 s=2'b00; I=4'b0110; SIL=1; SIR=0;
    #100 s=2'b01; I=4'b1001; SIL=1; SIR=1;
    #100 s=2'b01; I=4'b1010; SIL=0; SIR=1;
    #100 s=2'b10; I=4'b0101; SIL=0; SIR=1;
    #100 s=2'b10; I=4'b0111; SIL=0; SIR=0;
    #100 s=2'b11; I=4'b0110; SIL=0; SIR=1;
    #100 s=2'b00; I=4'b1101; SIL=1; SIR=0;
    #100 s=2'b00; I=4'b1100; SIL=0; SIR=1;
    #100 $stop;
end
initial begin
$monitor("I=%b,s=%b,A=%b, clk=%b, clear=$b, SIL=%b, SIR=%b",I,s,A, clk,
clear, SIL, SIR);
end
endmodule
```

SIMULATION WAVEFORMS:

Shift register in behavioral:



Shift register in structural:

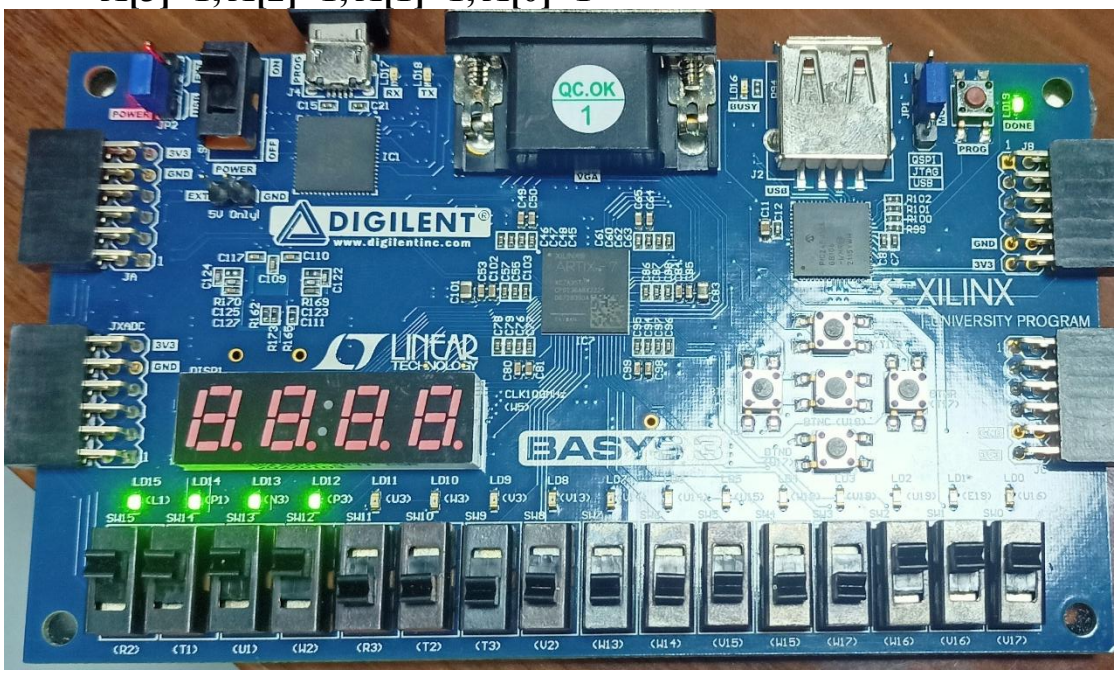


HARDWARE OUTPUT:
I/O ports:

Inputs:		**Outputs:**
I[3]: R2	clear: W16	A[3]: L1
I[2]: T1	clk: W5	A[2]: P1
I[1]: U1	SIL: T2	A[1]: N3
I[0]: W2	SIR: T3	A[0]: P3
s[1]: V16		
s[0]: V17		

For **I[3]=1, I[2]=1, I[1]:1, I[0]=1**
s[1]=1, s[0]=1, clear=1, clk=1
SIL=0, SIR=0,

Output:
A[3]=1, A[2]=1, A[1]=1, A[0]=1



FUNCTION TABLE:

Mode Control		
s_1	s_0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

CIRCUIT DAIGRAM:

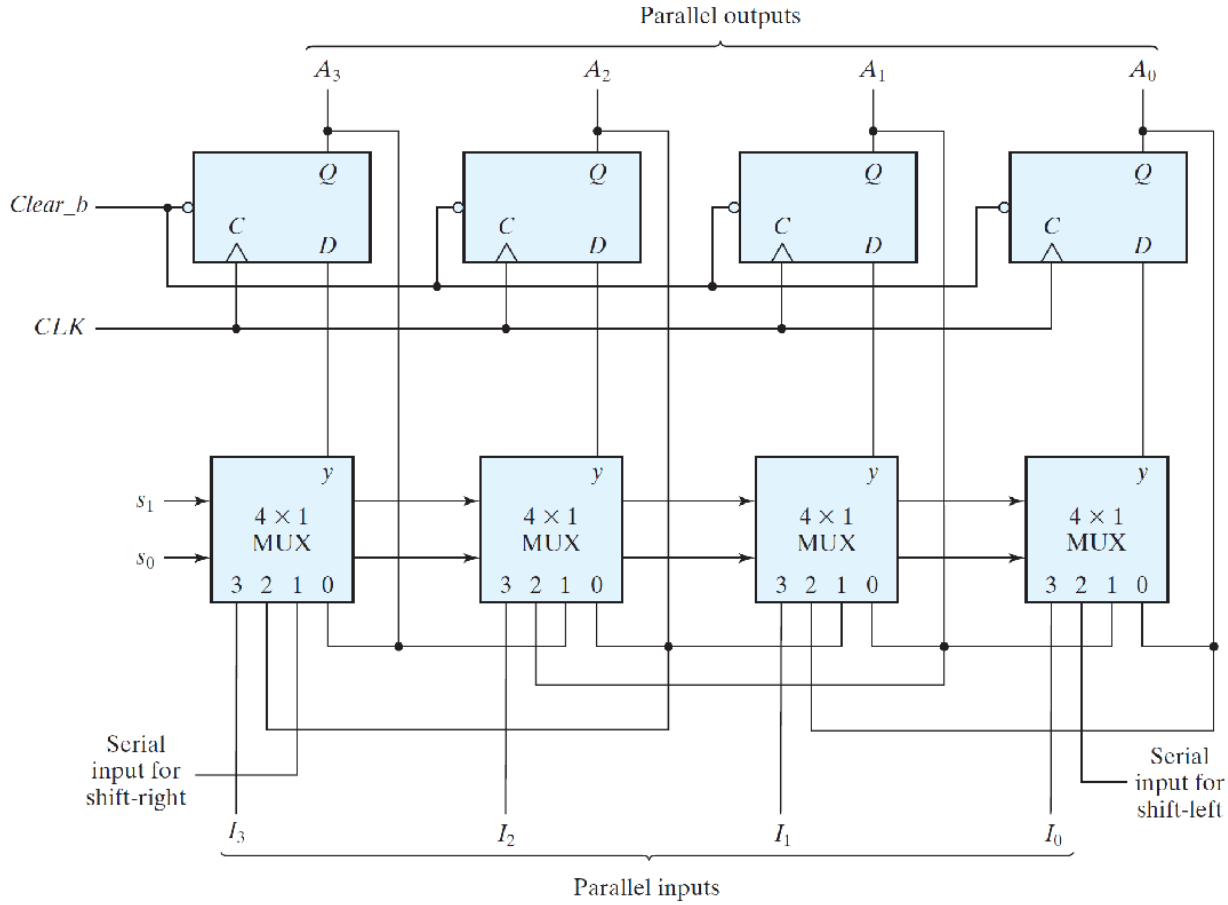


Fig: 4-bit universal shift register

RESULT:

Thus, shift registers were successfully designed and implemented in Verilog HDL using behavioral and structural modeling techniques. Their functionalities were verified through simulation using Xilinx Vivado software on the Basys3 FPGA board.