

در این آزمایش یک شیفت رجیستر را طراحی می کنیم که یک پایه selector و چهار Multiplexer 2x1 و چهار DFF دارد که بر اساس لبه بالارونده کلاکمان در هر بخش شیفت می خورد و بر اساس selector مان در هر مالتیپلکسر بین بیت متناظر Data in یا خروجی های فلیپ فلاپ ها و ورودی ها موازی P in باید یک کدام را انتخاب کرده و در زمان آمدن کلاک بعد آن را تحویل DFF بعدی دهد تا بعد از چهار کلاک خروجی ما مشاهده شود ، همان طور که مشاهده نمی شود در Time Diagram مدار شبیه سازی شده مان چهار کلاک طول کشیده تا اولین ورودی ما به خروجی برسد و در باقی این شبیه ساز هم به همین شکل این مدار عمل می کند .

#### Shift Register Vhdl code:

```

51      P_in : in  STD_LOGIC;
52      Selector : in  STD_LOGIC;
53      Output : out STD_LOGIC;
54  end component multiplexer;
55
56
57  signal DFFOutput : STD_LOGIC_VECTOR(2 downto 0);
58  signal MuxOutput : STD_LOGIC_VECTOR(3 downto 0);
59
60  begin
61
62  multiplexer0 : multiplexer Port map( DataIn => DataIn , P_in => P_in(3) , Selector => Selector , Output => MuxOutput(0) );
63  dff_interface0 : DFLipFlop port map( d => MuxOutput(0) , clock => Clk , reset => '0' , q => DFFOutput(0));
64
65  multiplexer1 : multiplexer Port map( DataIn => DFFOutput(0) , P_in => P_in(2) , Selector => Selector , Output => MuxOutput(1) );
66  dff_interface1 : DFLipFlop port map( d => MuxOutput(1) , clock => Clk , reset => '0' , q => DFFOutput(1));
67
68  multiplexer2 : multiplexer Port map( DataIn => DFFOutput(1) , P_in => P_in(1) , Selector => Selector , Output => MuxOutput(2) );
69  dff_interface2 : DFLipFlop port map( d => MuxOutput(2) , clock => Clk , reset => '0' , q => DFFOutput(2));
70
71  multiplexer3 : multiplexer Port map( DataIn => DFFOutput(2) , P_in => P_in(0) , Selector => Selector , Output => MuxOutput(3) );
72  dff_interface3 : DFLipFlop port map( d => MuxOutput(3) , clock => Clk , reset => '0' , q => Output);
73
74  end structural;
75

```

#### D flipflop Vhdl code:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4
5  entity DFLipFlop is
6      Port ( d : in  STD_LOGIC;
7            clock : in  STD_LOGIC;
8            reset : in  STD_LOGIC;
9            q : out  STD_LOGIC);
10 end DFLipFlop;
11
12 architecture Behavioral of DFLipFlop is
13
14 begin
15 process(clock,reset)
16 begin
17 if(reset ='1')then
18 q <= '0';
19 elsif(CLOCK='1' and CLOCK'EVENT)then
20 q <= d;
21 end if;
22 end process;
23 end Behavioral;

```

Multiplexer Vhdl code:

```
32 entity multiplexer is
33     Port ( DataIn : in  STD_LOGIC;
34           P_in : in  STD_LOGIC;
35           Selector : in  STD_LOGIC;
36           Output : out  STD_LOGIC);
37 end multiplexer;
38
39 architecture Behavioral of multiplexer is
40
41 begin
42
43 process(Selector)
44 begin
45
46     if Selector = '0' then
47         Output <= DataIn ;
48
49     else
50         OutPut <= P_in ;
51
52     end if;
53 end process;
54
55
56 end Behavioral;
```

Circuit Test Bench:

```
78     begin
79         Clk <= '0';
80         wait for Clk_period/2;
81         Clk <= '1';
82         wait for Clk_period/2;
83     end process;
84
85
86     -- Stimulus process
87     stim_proc: process
88     begin
89         -- hold reset state for 100 ns.
90         wait for Clk_period*4;
91         DataIn <= '0' ; P_in <= "1111" ;Selector <= '1' ;
92         wait for Clk_period*4;
93         DataIn <= '1' ; P_in <= "1001" ;Selector <= '0' ;
94         wait for Clk_period*4;
95
96         -- insert stimulus here
97
98         wait;
99     end process;
100
101 END;
102
```

Time Diagram of simulation:

