



گزارش کار اول

عنوان گزارش :

پیاده سازی t -flip flop ، jk -flip flop ، d -flip flop

استاد:

مریم محبتی

گروه کلاس :

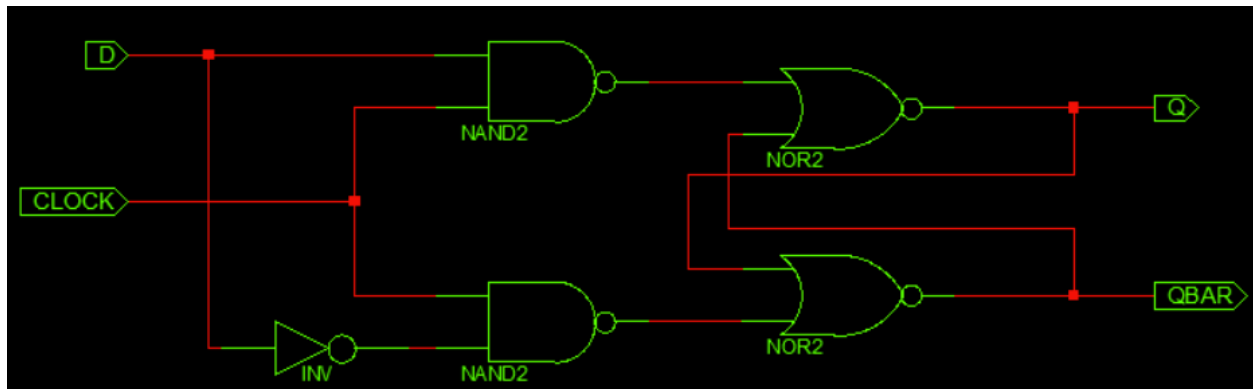
08

گروه انجام دهنده :

علیرضا اسلامی خواه ، فرزانه رحمانی ، ریحانه هاشم زاده

بررسی D FlipFlop :

در این فلیپ فلاپ d که با استفاده از کلاک پالس داده میشود اگر 1 باشد فلیپ فلاپ روی set و اگر 0 باشد روی clear تغییر حالت میدهد. ابتدا به بررسی مداری آن میپردازیم:



سپس جدول صحت آن را نگاه میکنیم :

D Flipflop truth table

Q	D	Q(T+1)
0	0	0
0	1	1
1	0	0
1	1	1

و در انتها هم به بررسی کد آن میپردازیم:

```

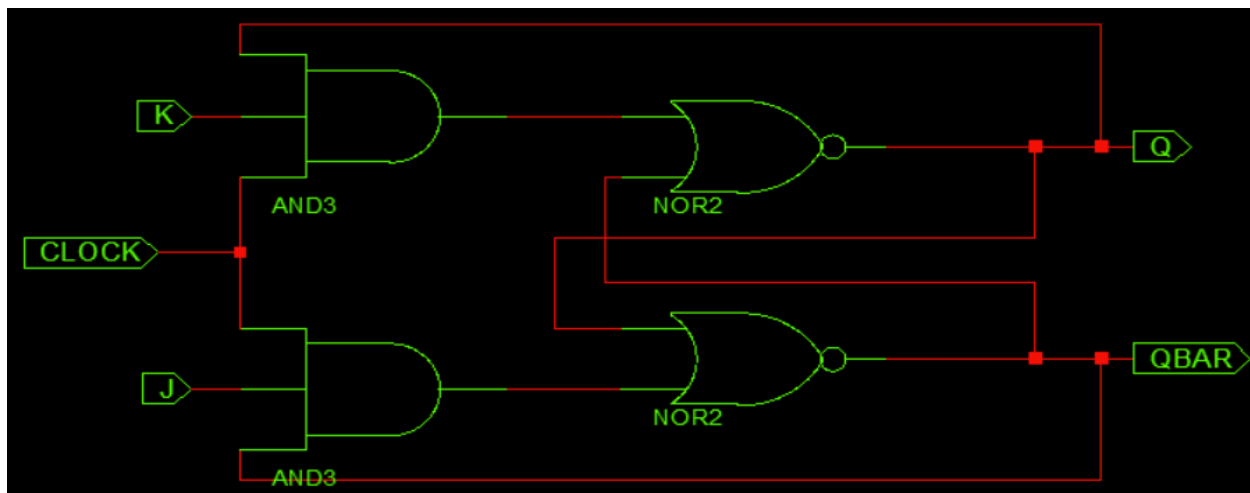
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4  entity D_flip_flop is
5      Port ( d : in  STD_LOGIC;
6            reset : in  STD_LOGIC;
7            clk : in  STD_LOGIC;
8            q : out  STD_LOGIC);
9  end D_flip_flop;
10 architecture Behavioral of D_flip_flop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

```

بررسی JK Flipflop :

در این فلیپ فلاپ ورودی های J و k مانند دکمه های set و reset در فلیپ فلاپ عمل میکنند.

شماتیک jk flipflop :



و جدول صحت آن:

Q	J	K	Q(T+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

بررسی کد پیاده سازی :

که در این کد به ترتیب k , clk , rst ورودی های ما و q , $qbar$ خروجی های ما میباشند.

```

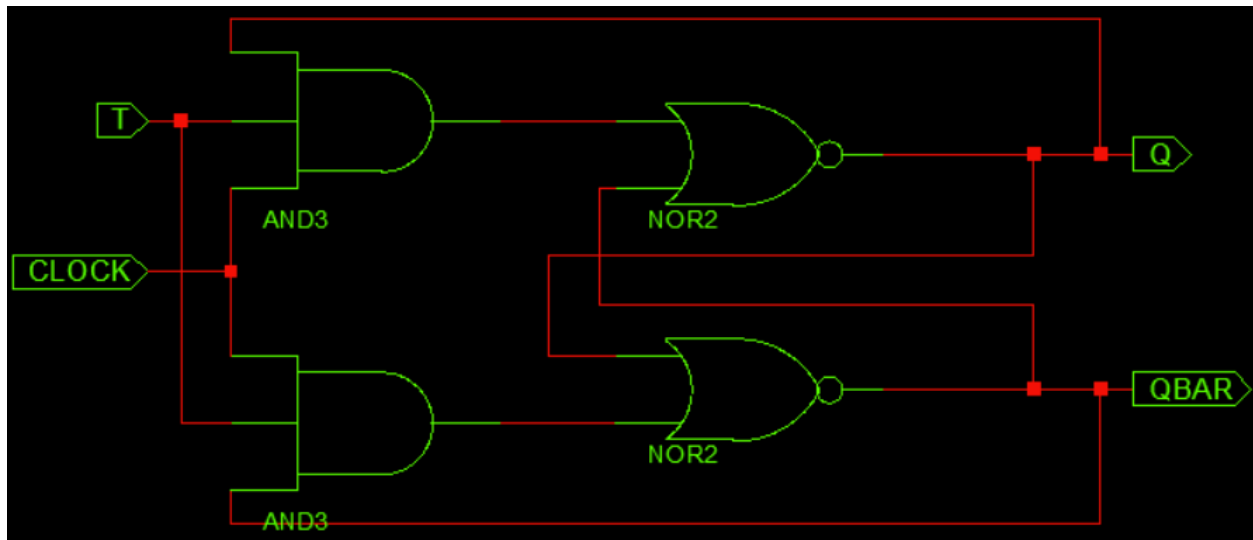
11  architecture Behavioral of JK_flip_flop is
12  begin
13      process(clk, rst)
14          variable qn : std_logic;
15          begin
16              if(rst = '1')then
17                  qn := '0';
18              elsif(clk'event and clk = '1')then
19                  if(J='0' and K='0')then
20                      qn := qn;
21                  elsif(J='0' and K='1')then
22                      qn := '0';
23                  elsif(J='1' and K='0')then
24                      qn := '1';
25                  elsif(J='1' and K='1')then
26                      qn := not qn;
27                  else
28                      null;
29                  end if;
30              else
31                  null;
32              end if;
33              Q <= qn;
34              Qbar <= not qn;
35          end process;
36  end Behavioral;

```

بررسی T FlipFlop :

این فلیپ فلاپ یک نسخه تک ورودی از فلیپ فلاپ JK میباشد. خروجی این فلیپ فلاپ با هر پالس کلاک toggle میکند.

بررسی شماتیک این فلیپ فلاپ :



بررسی جدول صحت:

T Flipflop truth table

Q	T	Q(T+1)
0	0	0
0	1	1
1	0	1
1	1	0

بررسی کد این فلیپ فلاپ :

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity T_FLIPFLOP is
4      Port ( T : in  STD_LOGIC;
5            CLK : in  STD_LOGIC;
6            RES : in  STD_LOGIC;
7            TEMP : in  STD_LOGIC;
8            Q : out  STD_LOGIC;
9            QB : out  STD_LOGIC);
10 end T_FLIPFLOP;
11 architecture Behavioral of T_FLIPFLOP is
12 begin
13     PROCESS(T,CLK,RES)
14     VARIABLE TEMP:STD_LOGIC:='0';
15     BEGIN
16         IF(RES='1')THEN
17             TEMP:='0';
18         ELSIF(RISING_EDGE(CLK))THEN
19             IF(T='1')THEN
20                 TEMP:= NOT TEMP;
21             END IF;
22         END IF;
23         Q<= NOT TEMP;
24         QB<= TEMP;
25     END PROCESS;
26 end Behavioral;

```

همانطور که مشاهده میشود این فلیپ فلاپ حساس به لبه بالا رونده میباشد. حال نوبت به تست کردن مدار های مربوطه میباشد. که به ترتیب برای تست کردن d flipflop کد زیر را با استفاده از نرم افزار زایلینیکس پیاده سازی کردیم.

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  ENTITY D_flip_flop_test IS
4  END D_flip_flop_test;
5
6  ARCHITECTURE behavior OF D_flip_flop_test IS
7      COMPONENT D_flip_flop
8      PORT(
9          d : IN  std_logic;
10         reset : IN  std_logic;
11         clk : IN  std_logic;
12         q : OUT  std_logic
13     );
14     END COMPONENT;
15     --Inputs
16     signal d : std_logic := '0';
17     signal reset : std_logic := '0';
18     signal clk : std_logic := '0';
19     --Outputs
20     signal q : std_logic;
21     -- Clock period definitions
22     constant clk_period : time := 10 ns;
23 BEGIN
24     -- Instantiate the Unit Under Test (UUT)
25     uut: D_flip_flop PORT MAP (
26         d => d,
27         reset => reset,
28         clk => clk,
29         q => q
30     );

```

```

31     -- Clock process definitions
32     clk_process :process
33     begin
34         clk <= '0';
35         wait for clk_period/2;
36         clk <= '1';
37         wait for clk_period/2;
38     end process;
39     d <= '1', 'X' after 22 ns, '1' after 24 ns, 'U' after 35ns, '1' after 40 ns;
40     reset <= '1' after 50 ns, '0' after 60 ns;
41 END;

```


تست jk flip flop :

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  ENTITY JK_FF_testbench IS
4  END JK_FF_testbench;
5  ARCHITECTURE behavior OF JK_FF_testbench IS
6      COMPONENT JK_flip_flop
7      PORT(
8          J : IN  std_logic;
9          K : IN  std_logic;
10         clk : IN  std_logic;
11         rst : IN  std_logic;
12         Q : OUT std_logic;
13         Qbar : OUT std_logic
14     );
15     END COMPONENT;
16     --Inputs
17     signal J : std_logic := '0';
18     signal K : std_logic := '0';
19     signal clk : std_logic := '0';
20     signal rst : std_logic := '0';
21     --Outputs
22     signal Q : std_logic;
23     signal Qbar : std_logic;
24
25     -- Clock period definitions
26     constant clk_period : time := 10 ns;
27 BEGIN
28
29     -- Instantiate the Unit Under Test (UUT)
30     uut: JK flip flop PORT MAP (
```

```
30 uut: JK_flip_flop PORT MAP (  
31     J => J,  
32     K => K,  
33     clk => clk,  
34     rst => rst,  
35     Q => Q,  
36     Qbar => Qbar  
37 );  
38  
39 -- Clock process definitions  
40 clk_process :process  
41 begin  
42     clk <= '0';  
43     wait for clk_period/2;  
44     clk <= '1';  
45     wait for clk_period/2;  
46 end process;  
47 Force: process  
48 begin  
49     J <= '0';  
50     K <= '0';  
51     rst <= '0';  
52     wait for 20 ns;  
53  
54     J <= '0';  
55     K <= '1';  
56     rst <= '0';  
57     wait for 20 ns;  
58  
59     J <= '1';
```

```

58
59     J <= '1';
60     K <= '0';
61     rst <= '0';
62     wait for 20 ns;
63
64     J <= '1';
65     K <= '1';
66     rst <= '0';
67     wait for 20 ns;
68
69     J <= '1';
70     K <= '1';
71     rst <= '0';
72     wait for 20 ns;
73
74     J <= '0';
75     K <= '0';
76     rst <= '0';
77     wait for 20 ns;
78
79     J <= '0';
80     K <= '0';
81     rst <= '1';
82     wait for 20 ns;
83     end process;
84
85 END;

```

تست t flipflop :

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  ENTITY TFF_testbench IS
4  END TFF_testbench;
5  ARCHITECTURE behavior OF TFF_testbench IS
6      -- Component Declaration for the Unit Under Test (UUT)
7      COMPONENT T_FLIPFLOP
8      PORT(
9          T : IN  std_logic;
10         CLK : IN  std_logic;
11         RES : IN  std_logic;
12         TEMP : IN  std_logic;
13         Q : OUT  std_logic;
14         QB : OUT  std_logic
15     );
16     END COMPONENT;
17     --Inputs
18     signal T : std_logic := '0';
19     signal CLK : std_logic := '0';
20     signal RES : std_logic := '0';
21     signal TEMP : std_logic := '0';
22
23     --Outputs
24     signal Q : std_logic;
25     signal QB : std_logic;
26
27     -- Clock period definitions
28     constant CLK_period : time := 10 ns;
29 BEGIN
30     -- Instantiate the Unit Under Test (UUT)

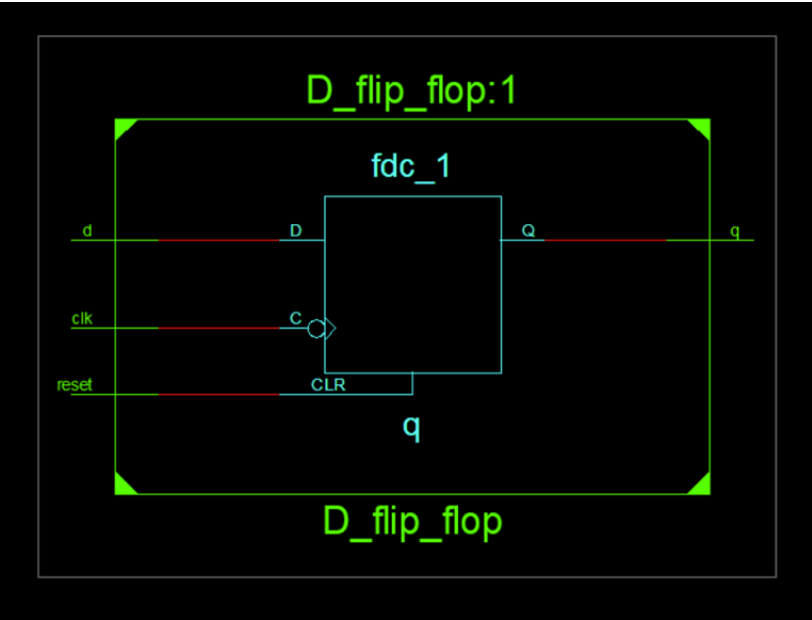
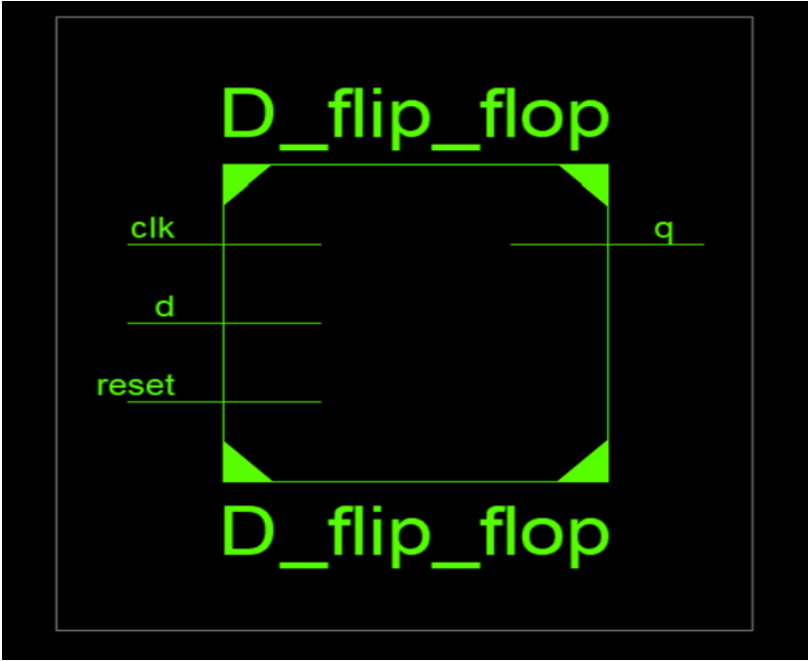
```

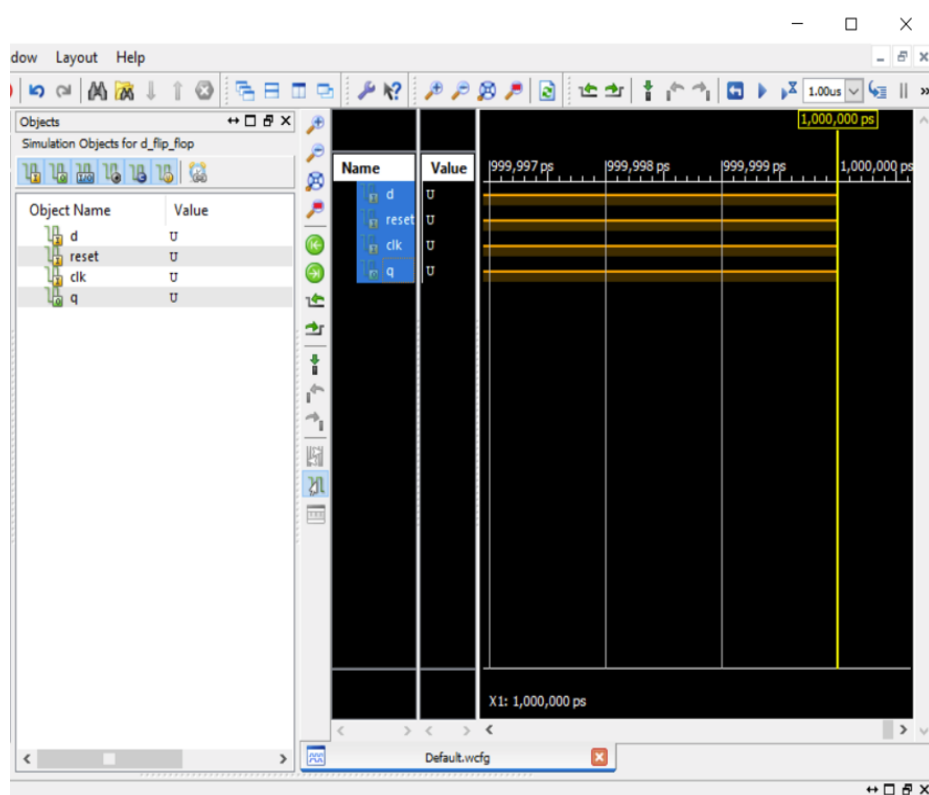
```

29 BEGIN
30     -- Instantiate the Unit Under Test (UUT)
31     uut: T_FLIPFLOP PORT MAP (
32         T => T,
33         CLK => CLK,
34         RES => RES,
35         TEMP => TEMP,
36         Q => Q,
37         QB => QB
38     );
39
40     -- Clock process definitions
41     CLK_process :process
42     begin
43         CLK <= '0';
44         wait for CLK_period/2;
45         CLK <= '1';
46         wait for CLK_period/2;
47     end process;
48     stim: process
49     begin
50         RES <= '0';
51         T <= '0';
52         wait for 20 ns;
53         T <= '1';
54         wait for 20 ns;
55     end process;
56 END;

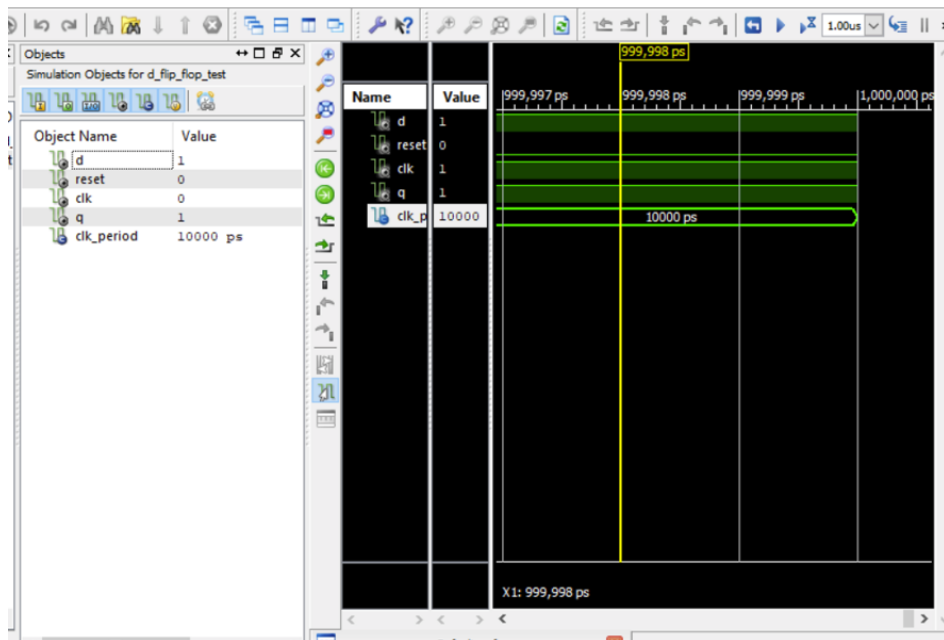
```

در ادامه تصاویری از شماتیک کلی فلیپ فلاپ ها و نتایج ران کردن آنها در زایلینکس ارائه می‌دهیم.

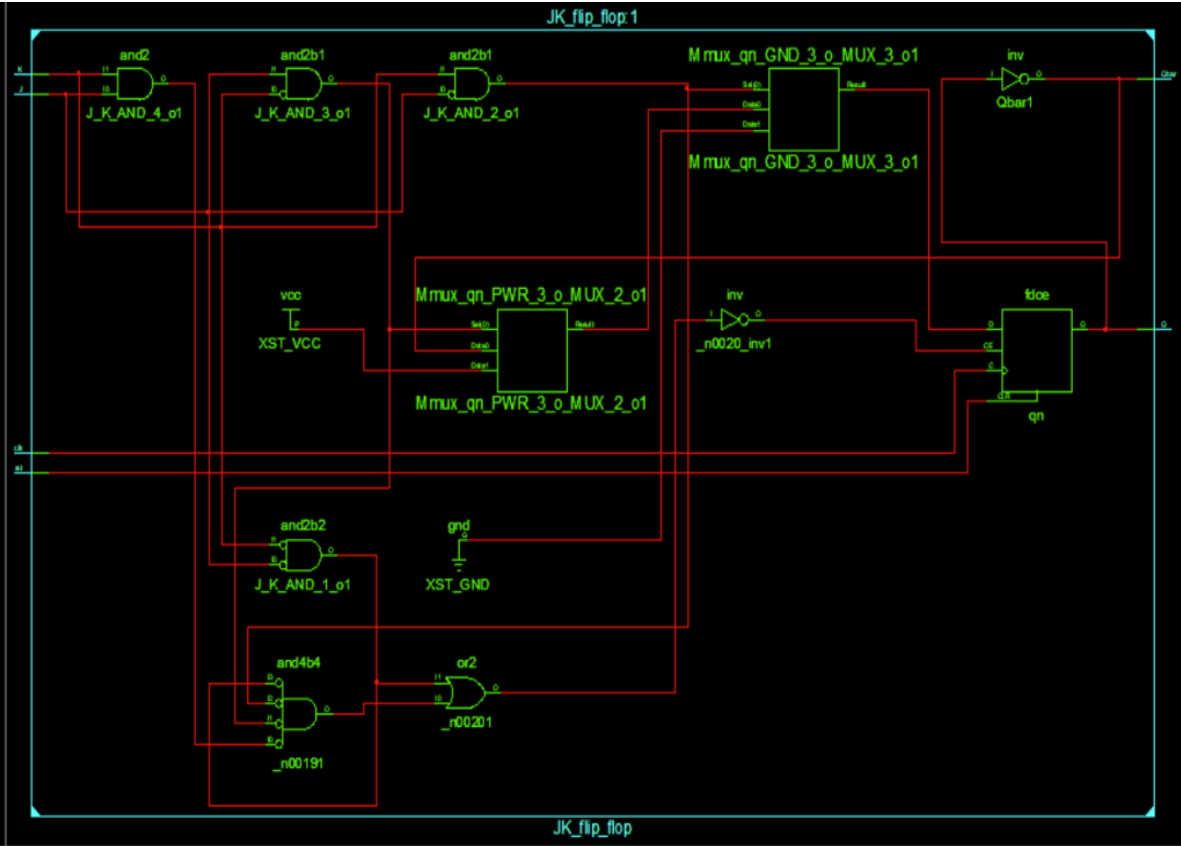
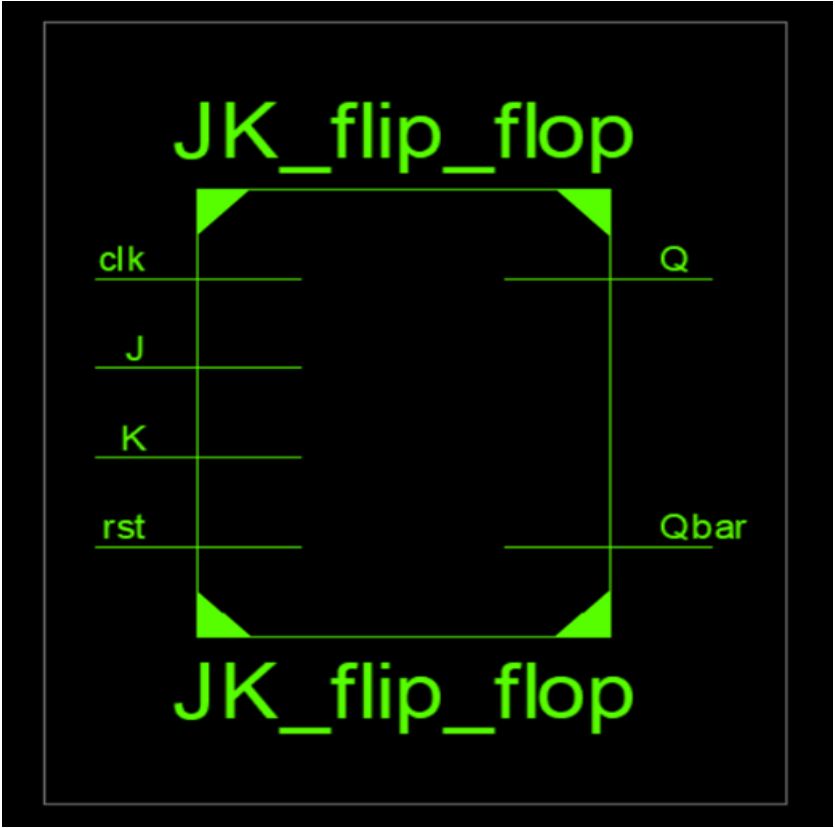


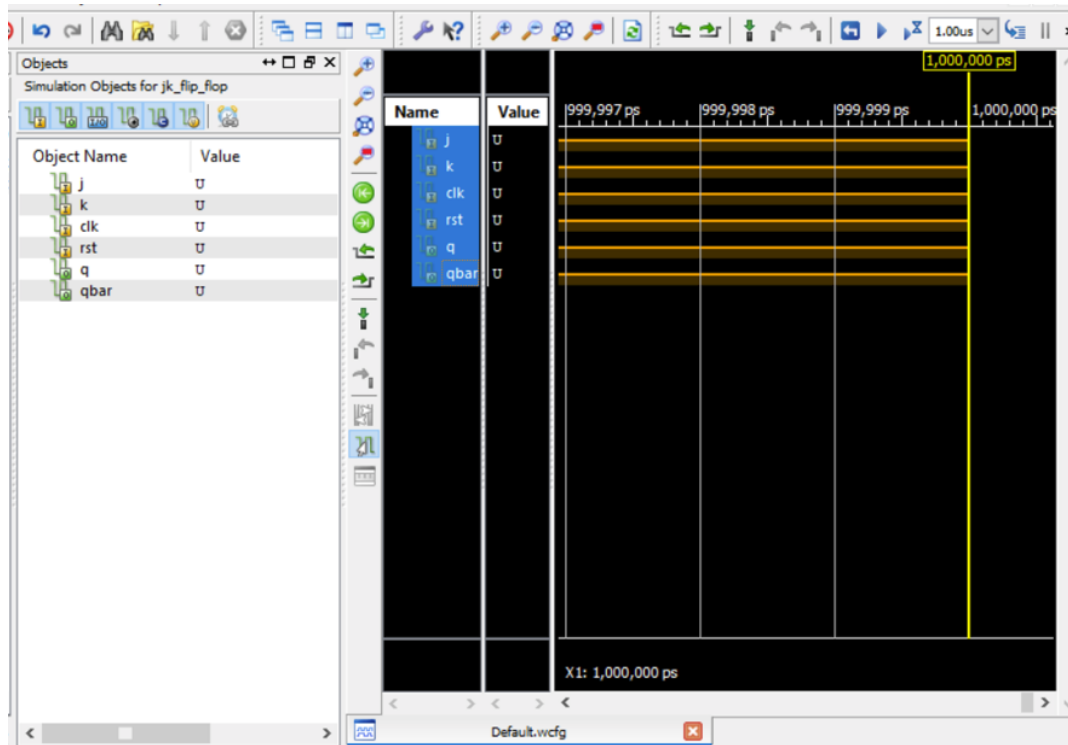


شکل 1-1 برای مدار d flip flop

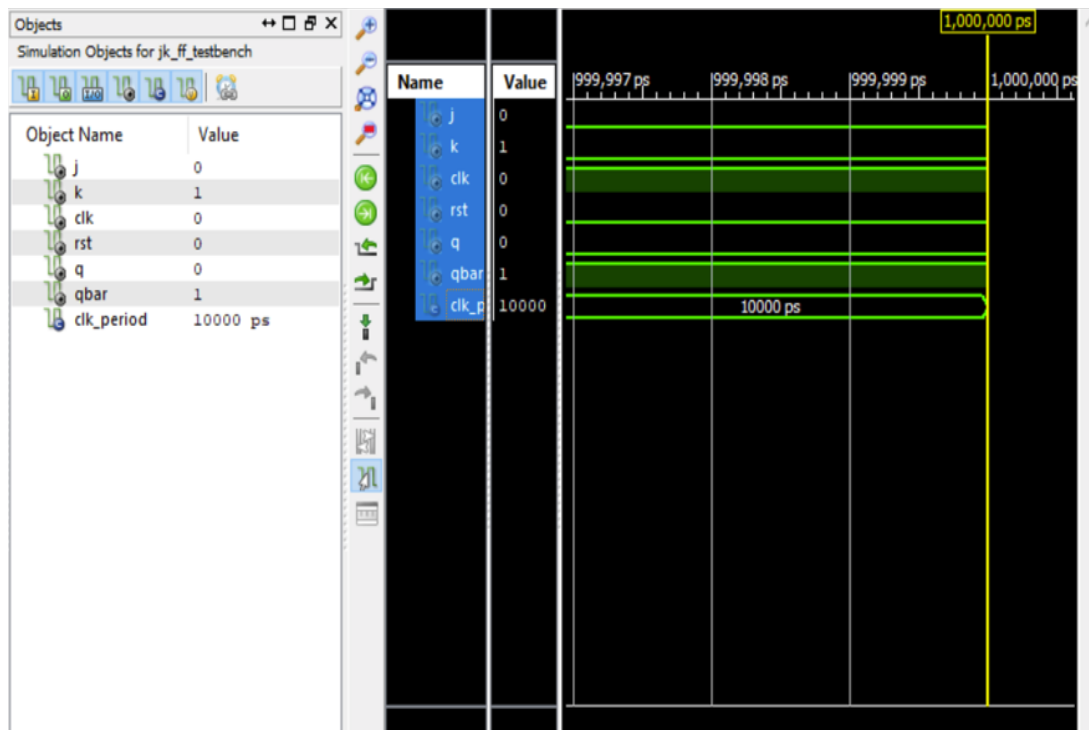


شکل 1-2 تست dflip flop

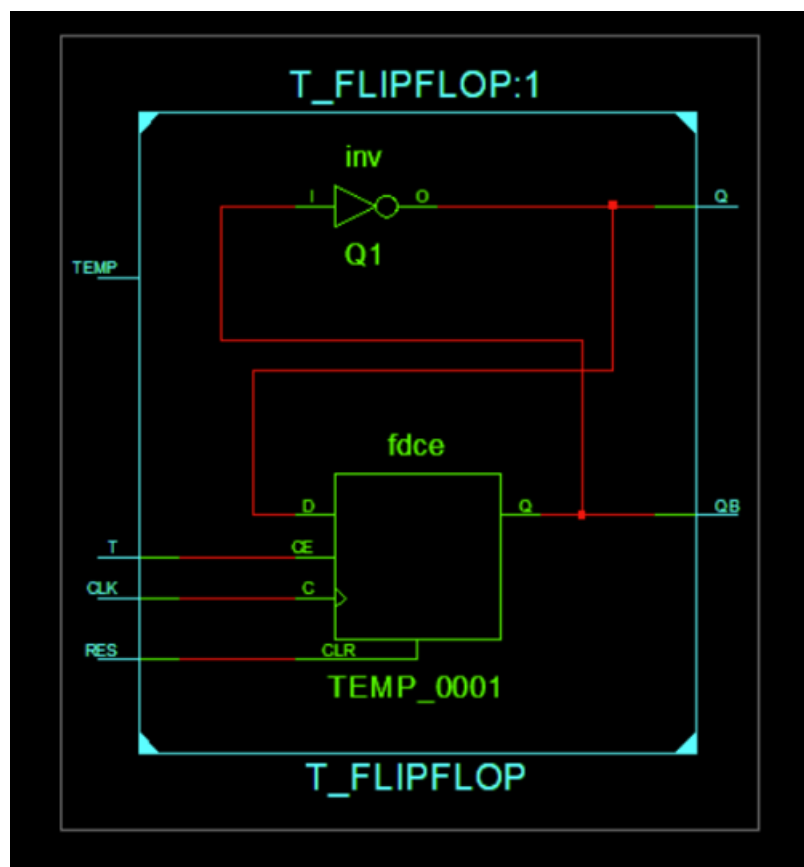
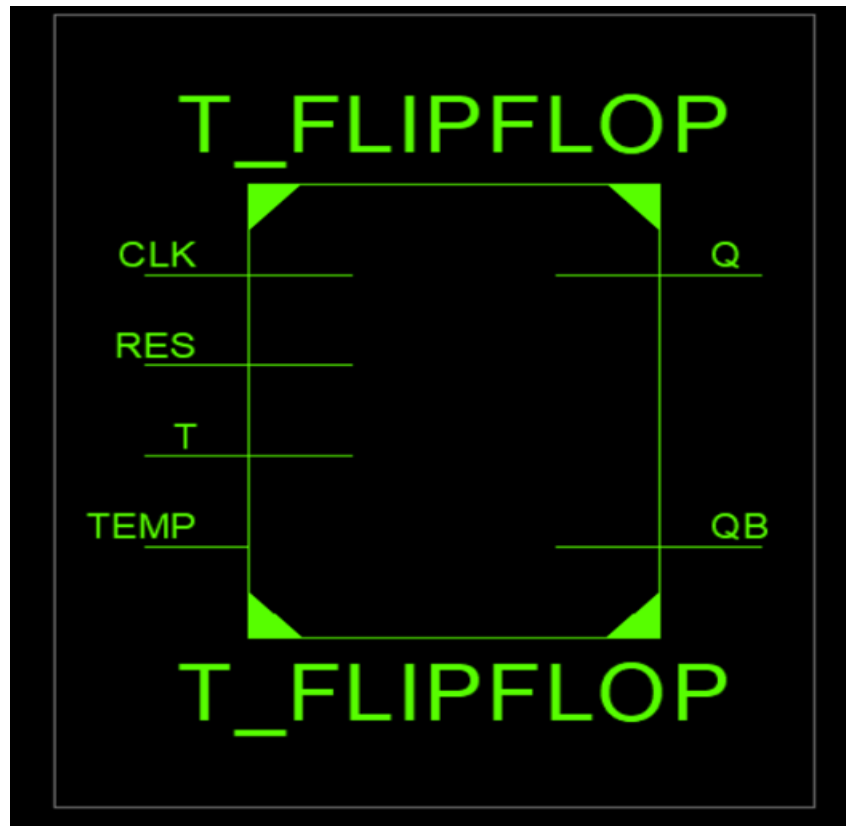


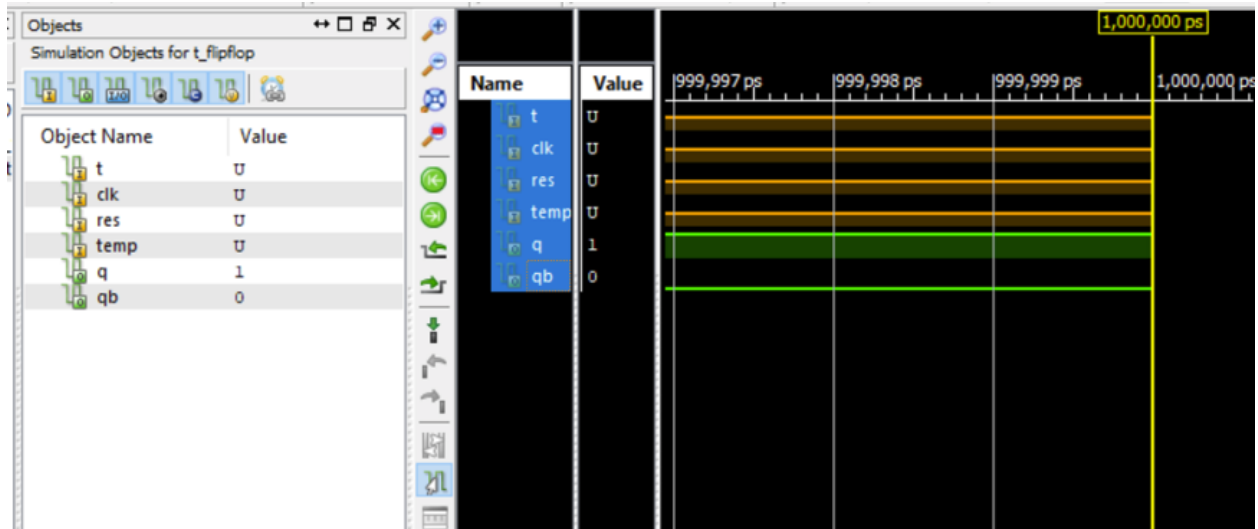


شکل 1 3 برای jk flip flop

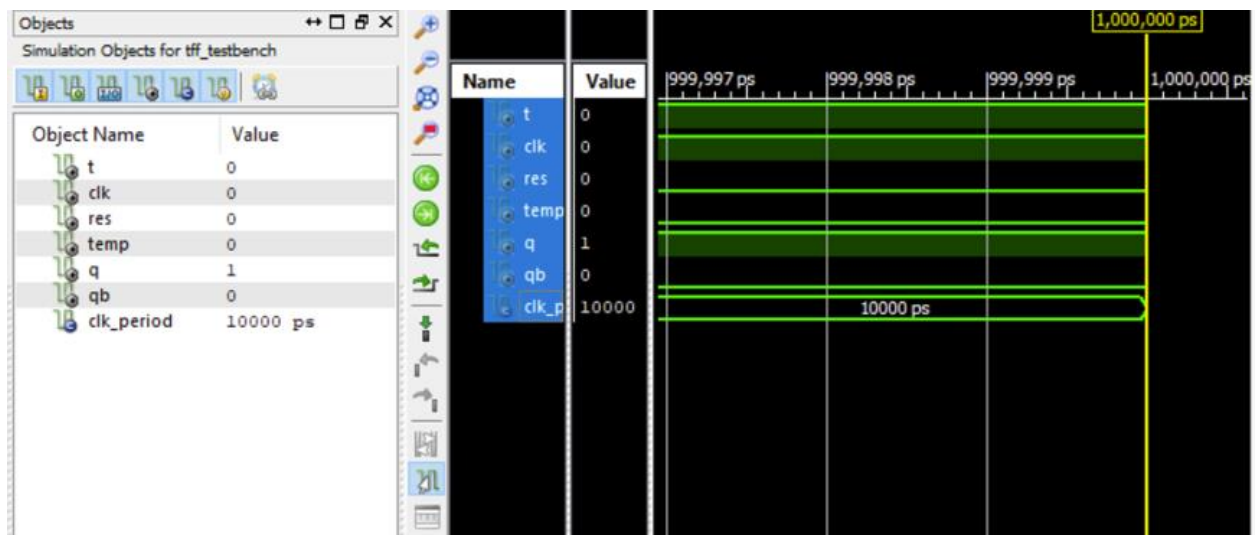


شکل 1 4 تست jk flipflop





شکل 1 5 برای t flip flop



شکل 1 6 تست t flip flop