گزارشکار آزمایش ۳

D فلیپ فلاپ :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity DFF is
port(
	Clk : in std_logic;
	reset : in std_logic;
	D : in std_logic;
	Q : out std_logic
	);
end DFF;

architecture Behavioral of DFF is
begin
process(Clk, reset)
		begin
		if (reset = '0') then Q <= '0';
		elsif (rising_edge(Clk)) then Q <= D;
		end if;
	end process;
end Behavioral;
```

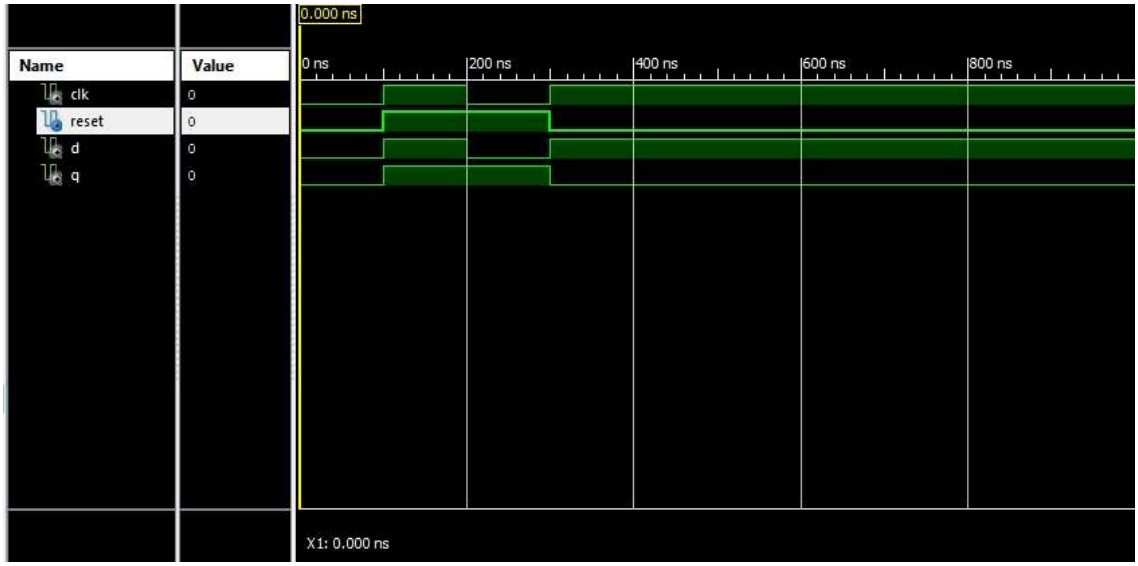
: DFF_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY DFF_tb IS
END DFF_tb;

ARCHITECTURE test OF DFF_tb IS
```

```
  -- Component Declaration for the Unit Under Test (UUT)

  COMPONENT DFF
  PORT(
      Clk : IN  std_logic;
      reset : IN  std_logic;
      D : IN  std_logic;
      Q : OUT  std_logic
     );
  END COMPONENT;


  --Inputs
  signal Clk : std_logic;
  signal reset : std_logic;
  signal D : std_logic;

          --Outputs
  signal Q : std_logic;

BEGIN

          -- Instantiate the Unit Under Test (UUT)
  uut: DFF PORT MAP (
        Clk => Clk,
        reset => reset,
        D => D,
        Q => Q
```

```
    );
        Clk <= '0' ,'1'after 100 ns , '0' after 200 ns,'1' after 300 ns;
        reset <='0','1'after 100 ns ,'0' after 300 ns ;
        D <= '0' ,'1' after 100 ns , '0' after 200 ns ,'1' after 300 ns;
END;
```

خروجی شبیه سازی DFF_tb :



T فليپ فلاپ :

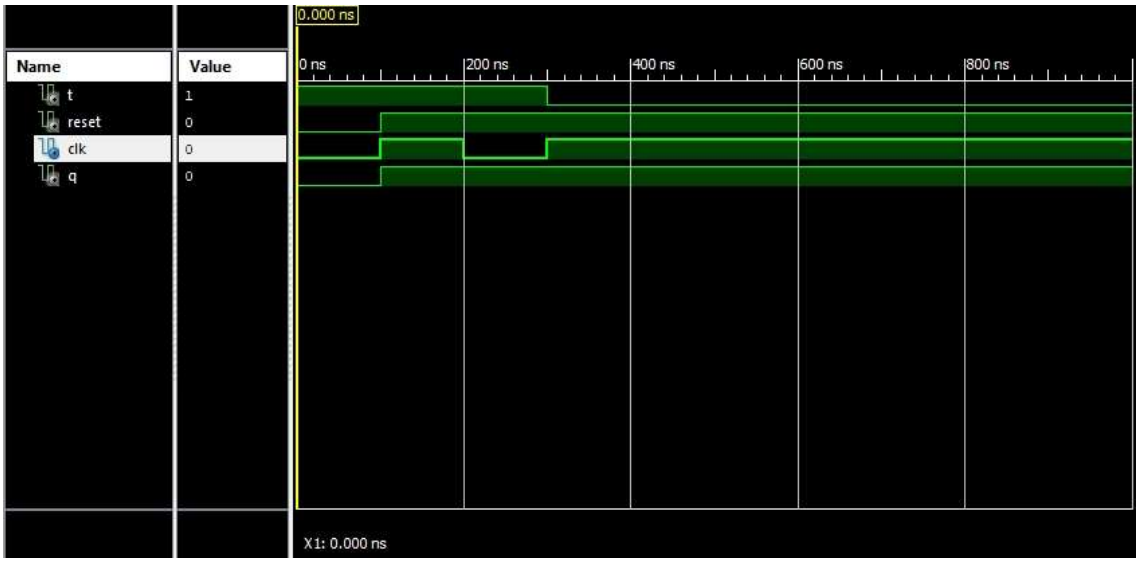
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity TFF is
port ( T,reset,CLK : in std_logic;
                    Q : out std_logic );
end TFF;
architecture Behavioral of TFF is
begin
        process(Clk,reset)
  variable temp: std_logic;
                    begin
                    if (reset = '0') then temp := '0';
                    elsif (rising_edge(Clk)) then
                     if T = '1' then temp  := not temp;
                     end if;
                    end if;
        Q <= temp;
end process;
end Behavioral;
```

: TFF_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY TFF_tb IS
END TFF_tb;

ARCHITECTURE test OF TFF_tb IS
```

```
   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT TFF
   PORT(
        T : IN  std_logic;
        reset : IN  std_logic;
        CLK : IN  std_logic;
        Q : OUT  std_logic
       );
   END COMPONENT;
  --Inputs
  signal T : std_logic;
  signal reset : std_logic;
  signal CLK : std_logic;
          --Outputs
  signal Q : std_logic;
BEGIN
          -- Instantiate the Unit Under Test (UUT)
  uut: TFF PORT MAP (
        T => T,
        reset => reset,
        CLK => CLK,
        Q => Q
       );
  T<='1' ,'0' after 300 ns ;
  Clk <= '0' ,'1' after 100 ns , '0' after 200 ns,'1' after 300 ns;
   reset <='0','1' after 100 ns;
END;
```

خروجی شبیه سازی TFF_tb :



: RippleCounter4bits

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity RippleCounter4bits is
  Port ( CLK : in  std_logic;
       reset : in  std_logic;
       count : out  std_logic_vector (3 downto 0)
         );
end RippleCounter4bits;

architecture Behavioral of RippleCounter4bits is
         component TFF
                  port ( T,reset,CLK : in std_logic;
                   Q : out std_logic );
         end component;
         signal s : std_logic_vector (3 downto 0);
begin
         T0: TFF port map(T=>'1', CLK=>CLK, reset => reset, Q=>s(0));
         T1: TFF port map(T=>'1', CLK=>not s(0), reset => reset,Q=>s(1));
         T2: TFF port map(T=>'1', CLK=>not s(1), reset => reset, Q=>s(2));
         T3: TFF port map(T=>'1', CLK=>not s(2), reset => reset, Q=>s(3));
         count <= s;
end Behavioral;
```

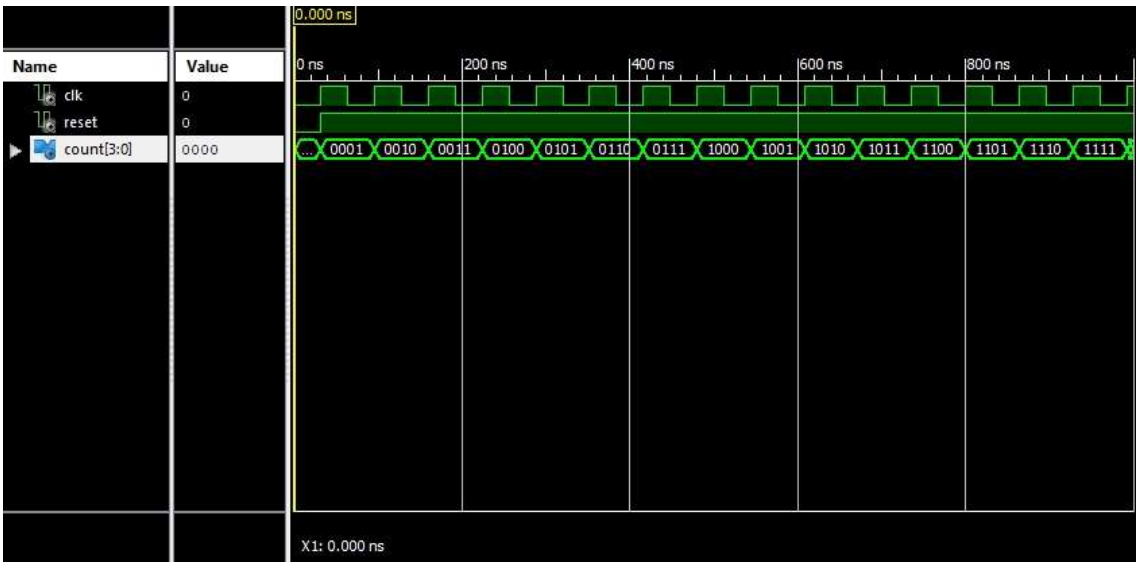
: RippleCounter4bits_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY RippleCounter4bits_tb IS
END RippleCounter4bits_tb;
```

```
ARCHITECTURE test OF RippleCounter4bits_tb IS

   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT RippleCounter4bits
   PORT(
       CLK : IN  std_logic;
       reset : IN  std_logic;
       count : OUT  std_logic_vector(3 downto 0)
      );
   END COMPONENT;


  --Inputs
  signal CLK : std_logic;
  signal reset : std_logic;

          --Outputs
  signal count : std_logic_vector(3 downto 0);

BEGIN

          -- Instantiate the Unit Under Test (UUT)
  uut: RippleCounter4bits PORT MAP (
        CLK => CLK,
        reset => reset,
        count => count
       );
```

```
-- Clock process definitions
Clk_process :process
begin
        Clk <= '0';
        wait for 32 ns;
        Clk <= '1';
        wait for 32 ns;
end process;
reset <= '0' ,'1' after 32 ns;

END;
```

خروجی شبیه سازی RippleCounter4bits_tb :



مدار Sequence detector برای رشته "۱۱۰۱" به صورت mealy :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mealy is
    Port ( clk : in STD_LOGIC;
           din : in STD_LOGIC;
           rst : in STD_LOGIC;
           dout : out STD_LOGIC
         );
end mealy;
architecture Behavioral of mealy is
    type state is (st0, st1, st2, st3);
    signal present_state, next_state : state;
    begin
    synchronous_process: process (clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                present_state <= st0;
            else
                present_state <= next_state;
            end if;
        end if;
    end process;

    output_decoder : process(present_state, din)
    begin
        next_state <= st0;
        case (present_state) is
```

```
when st0 =>
      if (din = '1') then
            next_state <= st1;
             dout <= '0';
     else
            next_state <= st0;
             dout <= '0';
      end if;
when st1 =>
      if (din = '1') then
            next_state <= st2;
           dout <= '0';
      else
            next_state <= st0;
             dout <= '0';
      end if;
when st2 =>
      if (din = '0') then
            next_state <= st3;
            dout <= '0';
      else
            next_state <= st2;
            dout <= '0';
      end if;
when st3 =>
      if (din = '1') then
            next_state <= st1;
            dout <= '1';
      else
```

```
                    next_state <= st0;
                    dout <= '0';
                end if;
            when others =>
                next_state <= st0;
        end case;
    end process;
end Behavioral;
```

مدار Sequence detector برای رشته "۱۱۰۱" به صورت moore :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity moore is
    Port ( clk : in STD_LOGIC;
           din : in STD_LOGIC;
           rst : in STD_LOGIC;
           dout : out STD_LOGIC
         );
end moore;
architecture Behavioral of moore is
    type state is (st0, st1, st2, st3, st4);
    signal present_state, next_state : state;
    begin
    synchronous_process: process (clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                present_state <= st0;
            else
```

```
                present_state <= next_state;
            end if;
        end if;
end process;
next_state_decoder : process(present_state, din)
begin
        next_state <= st0;
        case (present_state) is
            when st0 =>
                if (din = '1') then
                    next_state <= st1;
                else
                    next_state <= st0;
                end if;
            when st1 =>
                if (din = '1') then
                    next_state <= st2;
                else
                    next_state <= st0;
                end if;
            when st2 =>
                if (din = '0') then
                    next_state <= st3;
                else
                    next_state <= st2;
                end if;
            when st3 =>
                if (din = '1') then
                    next_state <= st4;
```

```
                else
                    next_state <= st0;
                end if;
            when st4 =>
                if (din = '1') then
                    next_state <= st2;
                else
                    next_state <= st0;
                end if;
            when others =>
                next_state <= st0;
        end case;
    end process;
    output_decoder : process(present_state)
    begin
        case (present_state) is
            when st0 =>
                dout <= '0';
            when st1 =>
                dout <= '0';
            when st2 =>
                dout <= '0';
            when st3 =>
                dout <= '0';
            when st4 =>
                dout <= '1';
            when others =>
                dout <= '0';
        end case;
```

```
    end process;
end Behavioral;
```

مدار moore و mealy بالا به صورت مجتمع :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity moore_mealy is
    port(
        din : in std_logic ;
        dout_moore ,dout_mealy : out std_logic;
        Clk : in std_logic;
        Rst : in std_logic
        );
end moore_mealy;
architecture Behavioral of moore_mealy is
    component mealy is
            port(
        din : in std_logic ;
        dout : out std_logic;
        clk : in std_logic;
        rst : in std_logic
        );
    end component mealy;
    component moore is
            port(
        din : in std_logic ;
        dout : out std_logic;
        clk : in std_logic;
```

```
            rst : in std_logic
            );
        end component moore;

begin
        mealy_state_machine: mealy port map(din => din , dout => dout_mealy,clk => Clk , rst => Rst);
        moore_state_machine: moore port map(din => din , dout => dout_moore , clk => Clk , rst => Rst);
end Behavioral;
```

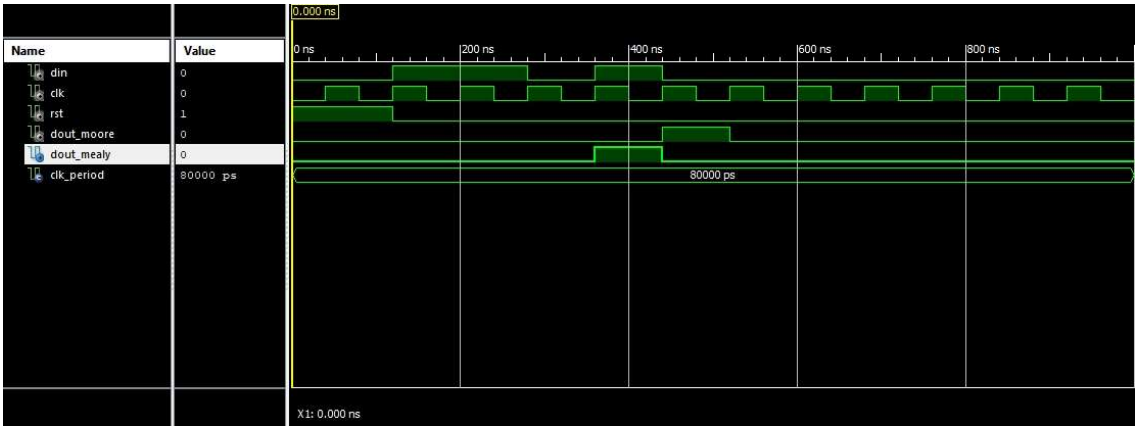
: Moore_mealy_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY moore_mealy_tb IS
END moore_mealy_tb;
ARCHITECTURE test OF moore_mealy_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT moore_mealy
   PORT(
       din : IN  std_logic;
       dout_moore : OUT  std_logic;
       dout_mealy : OUT  std_logic;
       Clk : IN  std_logic;
       Rst : IN  std_logic
      );
   END COMPONENT;
  --Inputs
  signal din : std_logic := '0';
  signal Clk : std_logic ;
```

```
  signal Rst : std_logic ;
        --Outputs
  signal dout_moore : std_logic;
  signal dout_mealy : std_logic;
  -- Clock period definitions
  constant Clk_period : time := 80 ns;
BEGIN
        -- Instantiate the Unit Under Test (UUT)
  uut: moore_mealy PORT MAP (
       din => din,
       dout_moore => dout_moore,
       dout_mealy => dout_mealy,
       Clk => Clk,
       Rst => Rst
      );
  -- Clock process definitions
  Clk_process :process
  begin
                Clk <= '0';
                wait for Clk_period/2;
                Clk <= '1';
                wait for Clk_period/2;
  end process;
  -- Stimulus process
  stim_proc: process
  begin
    Rst <= '1' ;
wait for 3*Clk_period/2;
    Rst <= '0' ;
```

```
    din <= '1' ;
    wait for Clk_period;
    din <= '1' ;
    wait for Clk_period;
    din <= '0' ;
    wait for Clk_period;
   din <= '1' ;
   wait for Clk_period;
   din <= '0' ;
   wait;
end process;
END;
```

خروجی شبیه سازی moore_mealy_tb :



مدار تشخیص دورشته "۰۱۱۰ " و "۰۱۰۱" به صورت moore :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity moore_0101_or_0110 is
    Port ( clk : in STD_LOGIC;
           din : in STD_LOGIC;
           rst : in STD_LOGIC;
          p_state : out std_logic_vector(2 downto 0);
          n_state : out std_logic_vector(2 downto 0);
           dout : out STD_LOGIC);
end moore_0101_or_0110;
architecture Behavioral of moore_0101_or_0110 is
    type state is (st0, st1, st2, st3, st4, st5, st6);
    signal present_state, next_state : state;
    begin
    synchronous_process: process (clk)
    begin
        if rising_edge(clk) then
            if (rst = '1') then
                present_state <= st0;
            else
                present_state <= next_state;
            end if;
        end if;
    end process;
    next_state_decoder : process(present_state,din)
    begin
        case (present_state) is
            when st0 =>
```

```
        if (din = '0') then
            next_state <= st1;
            n_state <= "001";
        else
            next_state <= st0;
            n_state <= "000";
        end if;
    when st1 =>
        if (din = '1') then
            next_state <= st2;
            n_state <= "010";
        else
            next_state <= st1;
            n_state <= "001";
        end if;
    when st2 =>
        if (din = '1') then
            next_state <= st4;
            n_state <= "100";
        else
            next_state <= st3;
            n_state <= "011";
        end if;
    when st3 =>
        if (din = '1') then
            next_state <= st6;
            n_state <= "110";
        else
            next_state <= st1;
            n_state <= "001";
```

```
            end if;
        when st4 =>
            if (din = '1') then
                next_state <= st0;
                n_state <= "000";
            else
                next_state <= st5;
                n_state <= "101";
            end if;
        when st5 =>
            if (din = '1') then
                next_state <= st2;
                n_state <= "010";
            else
                next_state <= st1;
                n_state <= "001";
            end if;
        when st6 =>
            if (din = '1') then
                next_state <= st4;
                n_state <= "100";
            else
                next_state <= st3;
                n_state <= "011";
            end if;
       when others =>
            next_state <= st0;
            n_state <= "000";
end case;
```

```
end process;
output_decoder : process(present_state)
begin
      case (present_state) is
            when st0 =>
                 dout <= '0';
                 p_state <= "000";
            when st1 =>
                  dout <= '0';
                   p_state <="001";
            when st2 =>
                dout <= '0';
                p_state <= "010";
            when st3 =>
                 dout <= '0';
                p_state <= "011";
           when st4 =>
                  dout <= '0';
                 p_state <= "100";
           when st5 =>
                  dout <= '1';
                 p_state <= "101";
           when st6 =>
                  dout <= '1';
                 p_state <= "110";
            when others =>
                  dout <= '0';
                 p_state <= "111";
```

```
            end case;
        end process;
end Behavioral;
```

: moore_0101_or_0110_tb

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY moore_0101_or_0110_tb IS
END moore_0101_or_0110_tb;
ARCHITECTURE test OF moore_0101_or_0110_tb IS
   -- Component Declaration for the Unit Under Test (UUT)
   COMPONENT moore_0101_or_0110
   PORT(
        clk : IN  std_logic;
        din : IN  std_logic;
        rst : IN  std_logic;
       p_state : out std_logic_vector(2 downto 0);
       n_state : out std_logic_vector(2 downto 0);
        dout : OUT  std_logic
       );
   END COMPONENT;
  --Inputs
  signal clk : std_logic ;
  signal din : std_logic :='1';
  signal rst : std_logic ;
--Outputs
  signal dout : std_logic;
  signal p_state :  std_logic_vector(2 downto 0);
```

```
  signal n_state :  std_logic_vector(2 downto 0);
  -- Clock period definitions
  constant clk_period : time := 80 ns;
BEGIN
        -- Instantiate the Unit Under Test (UUT)
  uut: moore_0101_or_0110 PORT MAP (
       clk => clk,
       din => din,
       rst => rst,
      p_state => p_state,
      n_state => n_state,
       dout => dout
      );
  -- Clock process definitions
  clk_process :process
  begin
                clk <= '0';
                wait for clk_period/2;
                clk <= '1';
                wait for clk_period/2;
  end process;
  -- Stimulus process
  stim_proc: process
  begin
               Rst <= '1' ;
         wait for Clk_period;
                Rst <= '0' ;
                wait for  Clk_period *3/2;
                din <= '0' ;
```

```
        wait for  Clk_period ;
        din <= '1' ;
        wait for  Clk_period ;
        din <= '1' ;
        wait for  Clk_period ;
        din <= '0' ;
        wait for  Clk_period;
        din <= '1' ;
        wait for  Clk_period ;
        din <= '0' ;
        wait for  Clk_period ;
        din <= '1' ;
        wait for  Clk_period ;
        din <= '1' ;
        wait;
  end process;
END;
```

خروجی شبیه سازی moore_0101_or_0110_tb :