

"به نام خالق رنگین کمان"



پیاده سازی:

فلیپ فلاپ های D,JK,T

شمارنده بالا شمار ددهی آسنکرون

شمارنده پایین شمار دودویی 4 بیتی

و تهیه TestBench برای هر کدام از آنها

اعضای گروه: نگین حقیقی، هلیا وفایی، ستاره باباجانی

استاد درس: دکتر مریم محبتی

نیم سال اول 1401-1402

## موضوع و اهداف:

این گزارش کار برای دو جلسه متوالی کلاس ساعت 10:30 الی 12:00 است. در جلسه اول، در ابتدا فلیپ فلاپ D را پیاده سازی کردیم و سپس فلیپ فلاپ های T,JK پیاده سازی شدند. در جلسه بعدی نیز باید دو شمارنده که یکی بصورت بالاشمار دهنده با هرنوع فلیپ فلاپ دلخواه و دیگری بصورت پایین شمار دهنده 4بیتی با هرنوع فلیپ فلاپ دلخواه می باشد را طراحی کنیم. هر یک از 5 مدار فوق، در ise پیاده سازی میشوند و عکس آنها، به همراه ورودی و خروجی تست بنچ آنها برای تایید صحت عملکرد آنها در این گزارشکار قرار میگیرد و کد تمامی آن ها نیز به پیوست ارسال میشود.

ابتدا عملکرد سه فلیپ فلاپ D,JK,T را در شکل زیر میتوانید مشاهده کنید.

فلیپ فلاپ D			فلیپ فلاپ JK				فلیپ فلاپ T		
D	Q(t+1)		J	K	Q(T+1)		T	Q(t+1)	
0	0	بازنشانی	0	0	Q(t)	بلا تغییر	0	Q(t)	بلا تغییر
1	1	نشاندن	0	1	0	بازنشانی	1	Q'(t)	متمم
			1	0	1	نشاندن			
			1	1	Q'(t)	متمم			

فلیپ فلاپ D:

این فلیپ فلاپ سه ورودی clock , reset , d دارد و خروجی آن در q ریخته میشود.

```
entity D_FF is
    Port ( d : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          q : out  STD_LOGIC);
end D_FF;

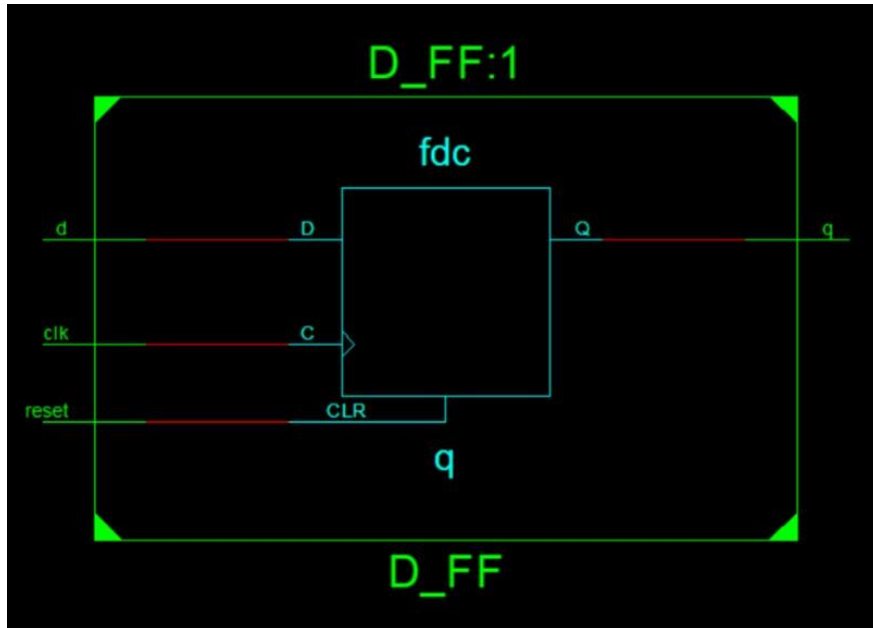
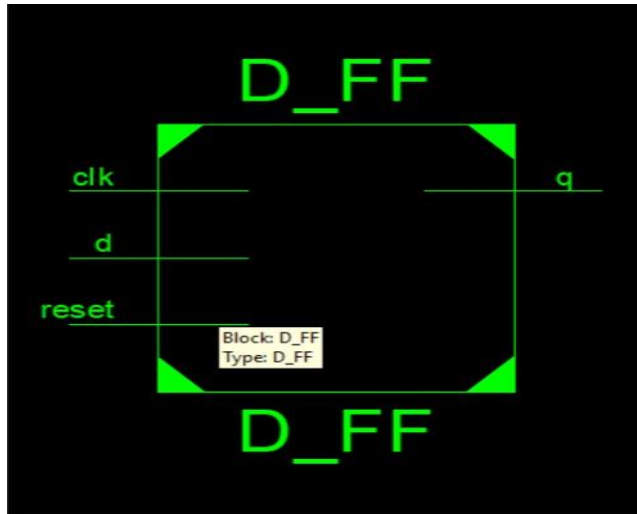
architecture Behavioral of D_FF is

begin

    q <= '0' when reset = '1' else
    d when clk'event and clk = '1';

end Behavioral;
```

همان طور که در شکل های زیر میبینید خروجی به ازای ورودی داده شده صحیح میباشد.(کد تست آن در فایل های ارسال شده موجود است).





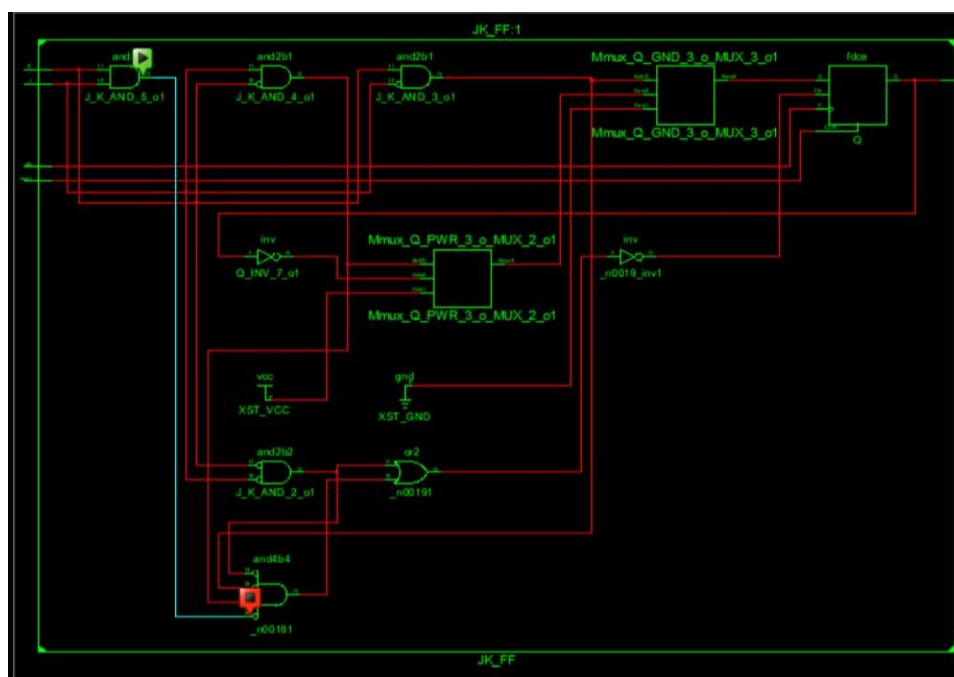
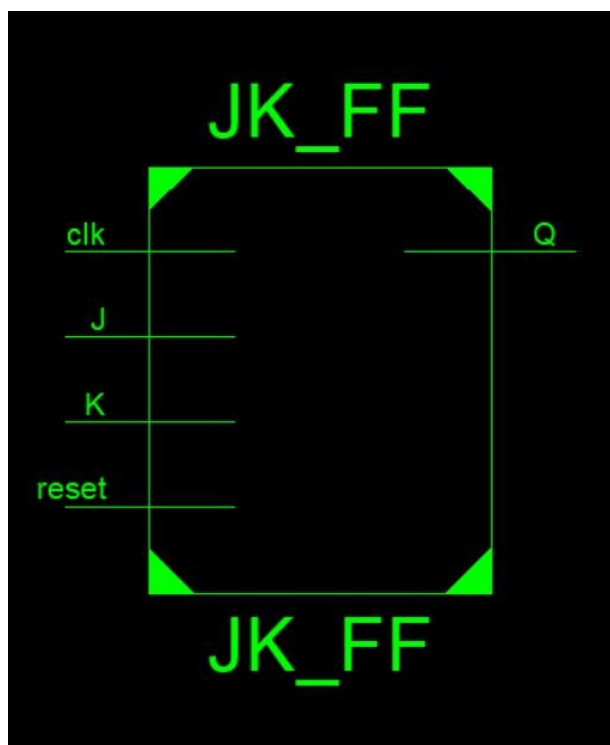
## فلیپ فلاپ JK:

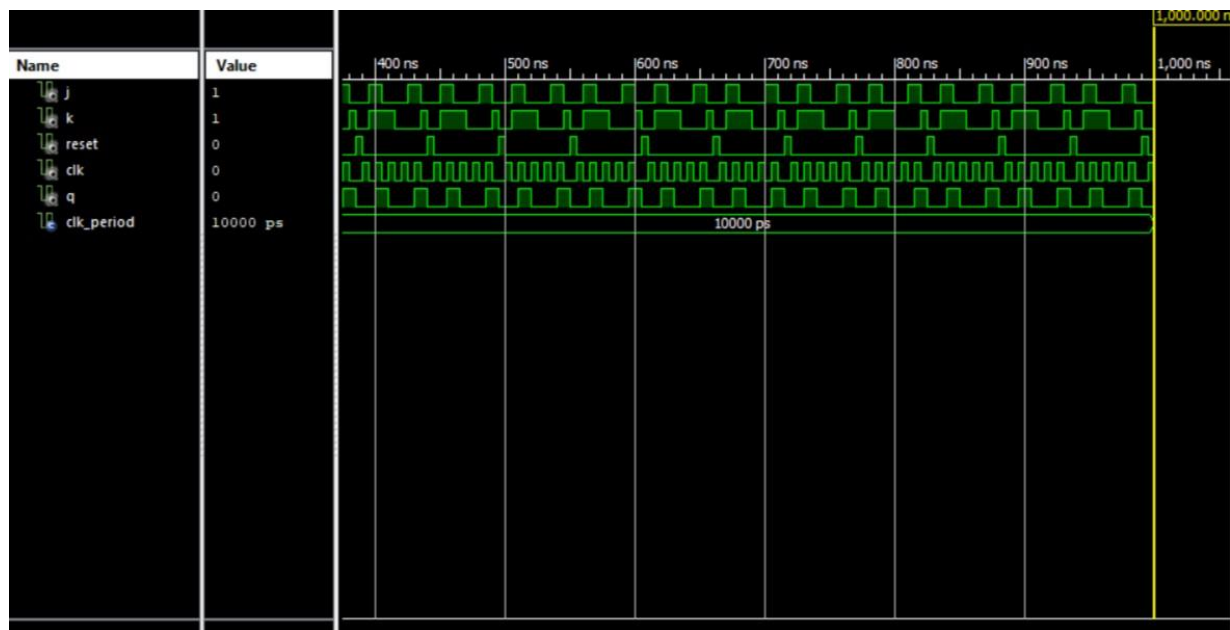
این فلیپ فلاپ 4 ورودی به نام های J,K,reset,clock و یک خروجی Q دارد. که بسته به مقدار هر کدام خروجی آن مشخص میشود.

```
entity JK_FF is
    Port ( J : in  STD_LOGIC;
          K : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          Q : inout STD_LOGIC);
end JK_FF;

architecture Behavioral of JK_FF is
begin
    process (reset, clk) is
    begin
        if reset = '1' then
            Q <= '0';
        elsif(rising_edge(clk)) then
            if (J='0' and K='0') then
                Q <= Q;
            elsif (J='0' and K='1') then
                Q <= '0';
            elsif (J='1' and K='0') then
                Q <= '1';
            elsif (J='1' and K='1') then
                Q <= not (Q);
            end if;
        end if;
    end process;
end architecture;
```

همان طور که در شکل های زیر میبینید خروجی به ازای ورودی داده شده صحیح میباشد.(کد تست آن در فایل های ارسال شده موجود است).





فلیپ فلاپ T:

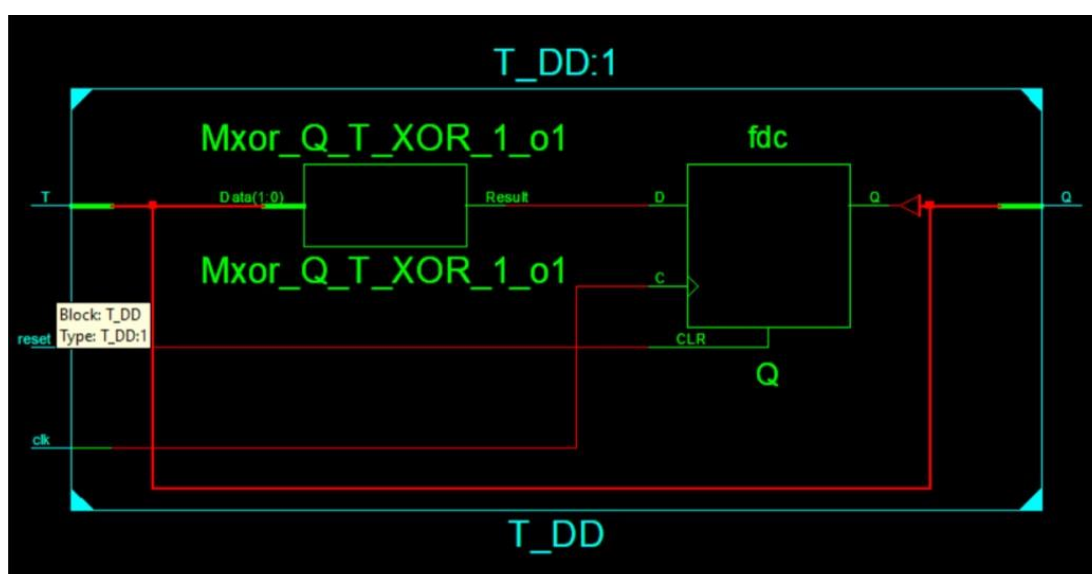
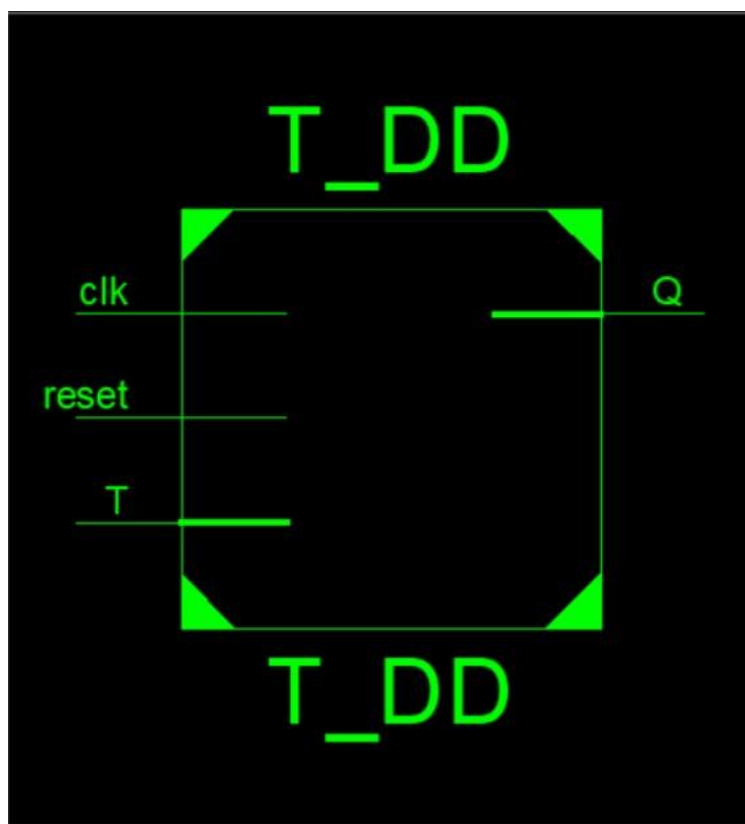
این فلیپ فلاپ 3 ورودی به نام های clock, reset, T و یک خروجی Q دارد.

که بسته به مقدار هر کدام خروجی آن مشخص میشود.

```
entity T_DD is
    Port ( T : in  STD_LOGIC;
          clk : in  STD_LOGIC;
          reset : in  STD_LOGIC;
          Q : inout STD_LOGIC);
end T_DD;

architecture Behavioral of T_DD is
begin
    process (reset, clk) is
    begin
        if reset = '1' then
            Q <= '0';
        elsif(rising_edge(clk)) then
            Q <= Q xor T;
        else
            Q <= Q;
        end if;
    end process;
end Behavioral;
```

همان طور که در شکل های زیر میبینید خروجی به ازای ورودی داده شده صحیح می باشد. (کد تست آن در فایل های ارسال شده موجود است).







حال به طراحی دو نوع شمارنده میپردازیم.

شمارنده بالا شمار دهندهی آسنکرون با فلیپ فلاپ دلخواه:

منطق این سیستم به این صورت است که شمارندهی طراحی شده، از 0 تا 9 را می‌شمارد و موقع رسیدن به 9 با 1 کردن سیگنال کنترلی reset مقدار خروجی را برابر 0 قرار می‌دهد. در این ساختار از 4 فلیپ فلاپ نوع T استفاده شده است. کد و شکل مربوط به این مدار را می‌توانید در تصاویر زیر مشاهده کنید.

```

component T_DD is
  port(
    T, reset, clk: in std_logic;
    Q: inout std_logic
  );
end component T_DD;

signal Q0, Q1, Q2, Q3: std_logic;

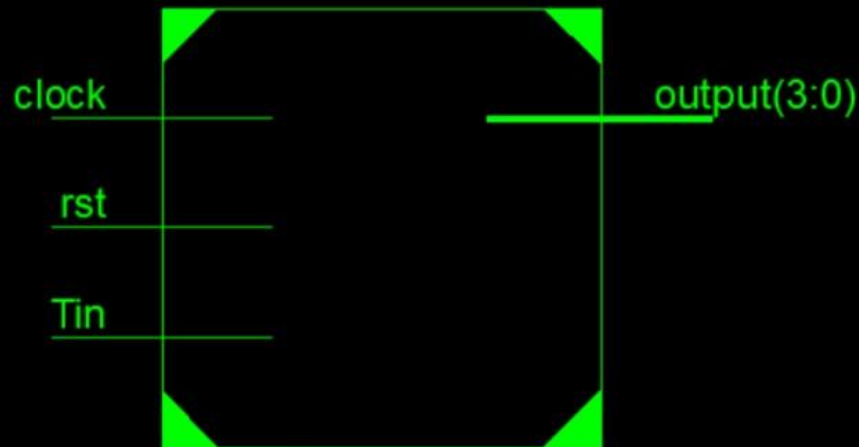
signal not_clk, not_Q0, not_Q1, not_Q2, not_Q3 : std_logic;

begin
  T0: T_DD port map(T => Tin, reset => rst, clk => clock, Q => Q0);
  not_Q0 <= not Q0;
  T1: T_DD port map(T => Tin, reset => rst, clk => not_Q0, Q => Q1);
  not_Q1 <= not Q1;
  T2: T_DD port map(T => Tin, reset => rst, clk => not_Q1, Q => Q2);
  not_Q2 <= not Q2;
  T3: T_DD port map(T => Tin, reset => rst, clk => not_Q2, Q => Q3);
  not_Q3 <= not Q3;
  output(0) <= Q0;
  output(1) <= Q1;
  output(2) <= Q2;
  output(3) <= Q3;

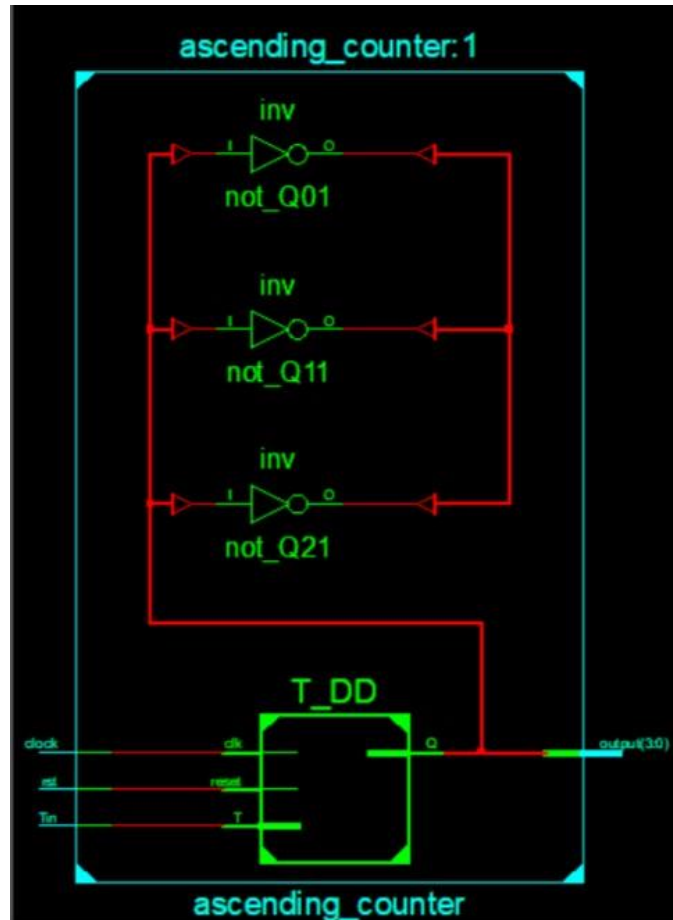
end Behavioral;

```

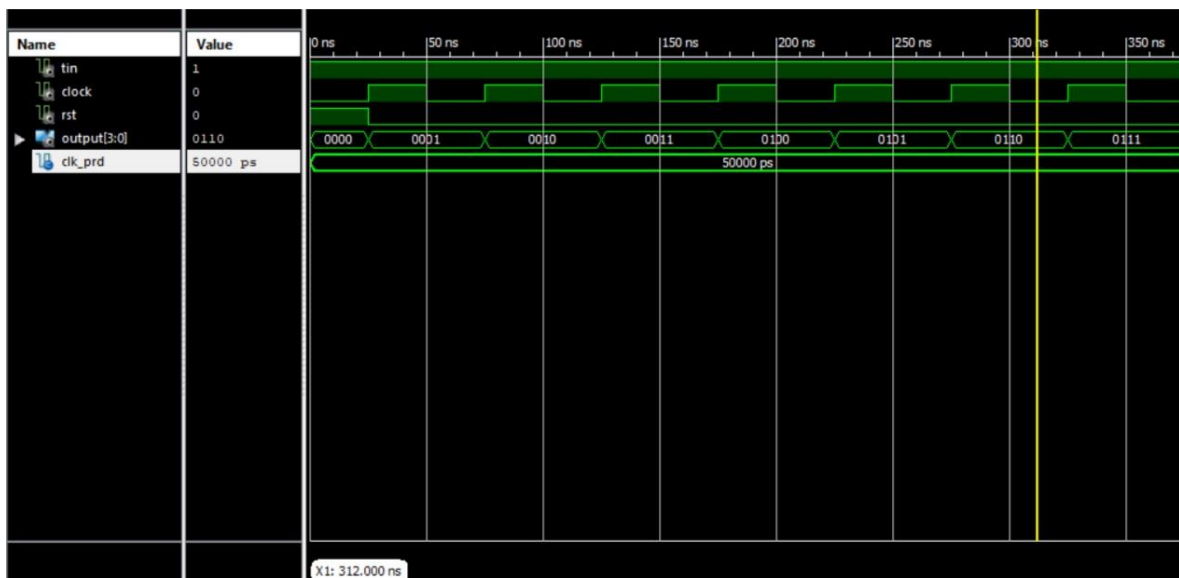
# ascending\_counter



# ascending\_counter



نتایج مربوط به این مدار را در تصویر زیر می‌توانید مشاهده کنید .



طراحی شمارنده پایین شمار دودویی 4 بیتی با استفاده از فلیپ فلاپ دلخواه:

منطق این سیستم به این صورت است که شمارنده‌ی طراحی شده، از 16 تا 0 را می‌شمارد و موقع رسیدن به 0 با 1 کردن سیگنال کنترلی reset مقدار خروجی را برابر 16 قرار می‌دهد. در این ساختار از 4 فلیپ فلاپ نوع T استفاده شده است. کد و شکل مربوط به این مدار را می‌توانید در تصاویر زیر مشاهده کنید.

```
entity descending_counter is
  port(
    Tin: in std_logic;
    clock: in std_logic;
    rst: in std_logic;
    output: out std_logic_vector(3 downto 0)
  );
end descending_counter;

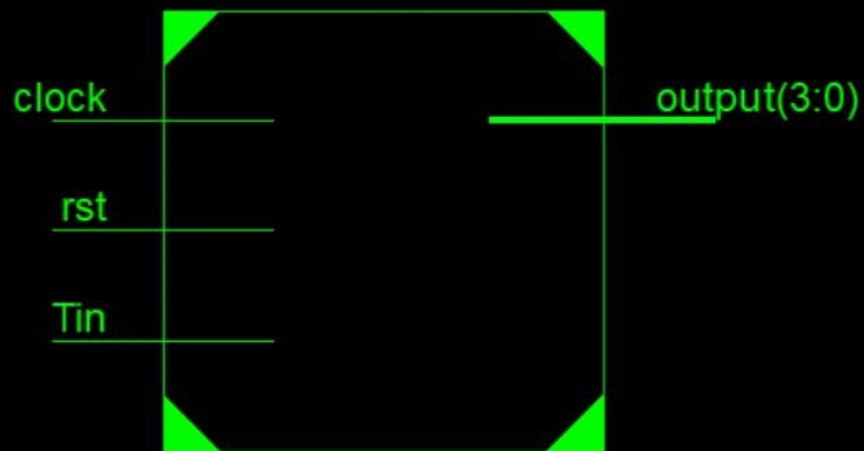
architecture Behavioral of descending_counter is

  component T_DD is
    port(
      T, reset, clk: in std_logic;
      Q: inout std_logic
    );
  end component T_DD;
  signal Q0, Q1, Q2, Q3: std_logic;

  signal not_clk, not_Q0, not_Q1, not_Q2 : std_logic;

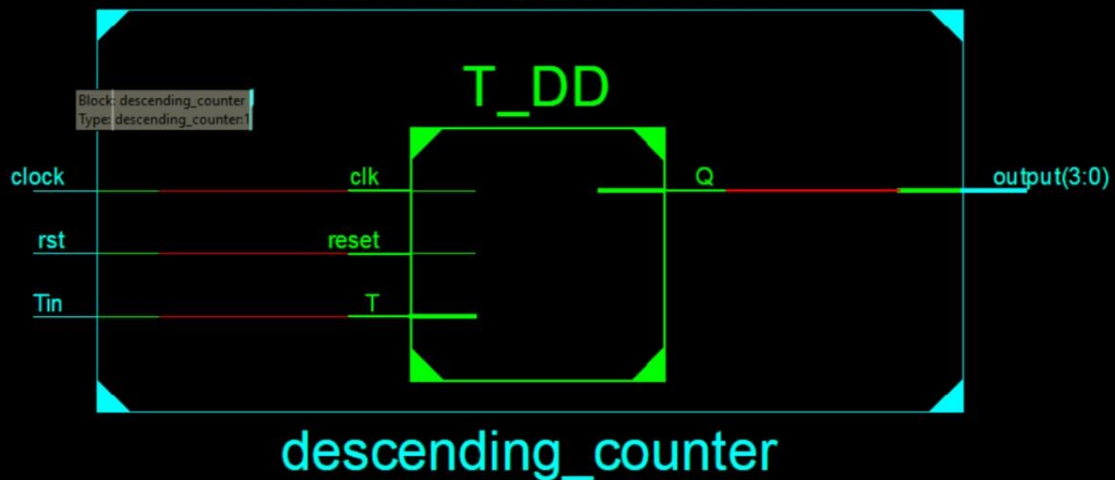
begin
  T0: T_DD port map(T => Tin, reset => rst, clk => clock, Q => Q0);
  not_Q0 <= not Q0;
  T1: T_DD port map(T => Tin, reset => rst, clk => Q0, Q => Q1);
  not_Q1 <= not Q1;
  T2: T_DD port map(T => Tin, reset => rst, clk => Q1, Q => Q2);
  not_Q2 <= not Q2;
  T3: T_DD port map(T => Tin, reset => rst, clk => Q2, Q => Q3);
  output(0) <= Q0;
  output(1) <= Q1;
  output(2) <= Q2;
  output(3) <= Q3;
```

# descending\_counter



# descending\_counter

## descending\_counter:1



نتایج مربوط به این مدار را در تصویر زیر می‌توانید مشاهده کنید .

