



گزارش کار چهارم

عنوان گزارش :

یک شمارنده بالشماری دهدهی آسنکرون با استفاده از هر فلیپ فالپ دلخواه
یک شمارنده پایین شمار دودویی 4 بیتی با استفاده از هر فلیپ فالپ

استاد:

مریم محبتی

گروه کلاس :

08

گروه انجام دهنده :

علیرضا اسلامی خواه ، فرزانه رحمانی ، ریحانه هاشم زاده

در ابتدا برای اینکه این شمارنده ها را پیاده سازی کنیم نیاز به فلیپ فلاپی داریم که از آن در شمارنده ها استفاده کنیم.

پس از یک T-flipflop ای استفاده میکنیم که پیاده سازی آن بدین شرح است:

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity T_FF is
5      port(
6          T, reset, clk: in std_logic;
7          Q: inout std_logic
8      );
9  end entity T_FF;
10
11  architecture Behavioral of T_FF is
12      begin
13          process (reset, clk) is begin
14              if reset = '1' then
15                  Q <= '0';
16              elsif(rising_edge(clk)) then
17                  Q <= Q xor T;
18              end if;
19          end process;
20
21  end Behavioral;
```

سپس به شرح یک شمارنده تصاعدی میپردازیم.

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity ascending_counter is
4      port(
5          Tin: in std_logic;
6          clock: in std_logic;
7          rst: in std_logic;
8          output: out std_logic_vector(3 downto 0)
9      );
10 end ascending_counter;
11 architecture Behavioral of ascending_counter is
12     component T_FF is
13     port(
14         T, reset, clk: in std_logic;
15         Q: inout std_logic
16     );
17     end component T_FF;
18     signal Q0, Q1, Q2, Q3: std_logic;
19
20     signal not_clk, not_Q0, not_Q1, not_Q2, not_Q3 : std_logic;
21 begin
22     T0: T_FF port map(T => Tin, reset => rst, clk => clock, Q => Q0);
23     not_Q0 <= not Q0;
24     T1: T_FF port map(T => Tin, reset => rst, clk => not_Q0, Q => Q1);
25     not_Q1 <= not Q1;
26     T2: T_FF port map(T => Tin, reset => rst, clk => not_Q1, Q => Q2);
27     not_Q2 <= not Q2;
28     T3: T_FF port map(T => Tin, reset => rst, clk => not_Q2, Q => Q3);
29     not_Q3 <= not Q3;
30     output(0) <= Q0;
31     output(1) <= Q1;
32     output(2) <= Q2;
33     output(3) <= Q3;
34 end Behavioral;

```

در اینجا ascending_counter که ما داریم سه پارامتر Tin و clock و rst را گرفته و یک وکتور 4 بیتی برمیگرداند.

سپس با توجه به تصویر از خط 22 به بعد 4 فلیپ فلاپ را که برای این کار لازم داریم با استفاده از portmap مقدار دهی میکنیم.

سپس از خط 30 به بعد خروجی های فلیپ فلاپ ها رو به وکتوری که برای خروجی تعریف کرده بودیم مقدار دهی میکنیم.

شمارنده نزولی :

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  entity descending_counter is
4      port(
5          Tin: in std_logic;
6          clock: in std_logic;
7          rst: in std_logic;
8          output: out std_logic_vector(3 downto 0)
9      );
10 end descending_counter;
11 architecture Behavioral of descending_counter is
12     component T_FF is
13     port(
14         T, reset, clk: in std_logic;
15         Q: inout std_logic
16     );
17     end component T_FF;
18     signal Q0, Q1, Q2, Q3: std_logic;
19     signal not_clk, not_Q0, not_Q1, not_Q2 : std_logic;
20 begin
21     T0: T_FF port map(T => Tin, reset => rst, clk => clock, Q => Q0);
22     not_Q0 <= not Q0;
23     T1: T_FF port map(T => Tin, reset => rst, clk => Q0, Q => Q1);
24     not_Q1 <= not Q1;
25     T2: T_FF port map(T => Tin, reset => rst, clk => Q1, Q => Q2);
26     not_Q2 <= not Q2;
27     T3: T_FF port map(T => Tin, reset => rst, clk => Q2, Q => Q3);
28     output(0) <= Q0;
29     output(1) <= Q1;
30     output(2) <= Q2;
31     output(3) <= Q3;
32 end Behavioral;

```

در اینجا هم مشابه بالا عمل کردیم فقط با این تفاوت که در clk فلیپ فلاپ ها به جای not کلاک قبلی خود کلاک را می‌گذاشتیم.

سپس به بررسی تست بنچ ها می‌پردازیم:

```
18 begin
19     UUT : ascending_counter
20     port map (
21         Tin => Tin,
22         clock => clock,
23         rst => rst,
24         output => output
25     );
26 process is
27 begin
28     Tin <= '1';
29     rst <= '1';
30     clock <= '0';
31     wait for clk_prd/2;
32     rst <= '0';
33     clock <= '1';
34     wait for clk_prd/2;
35     clock <= '0';
36     wait for clk_prd/2;
37     clock <= '1';
38     wait for clk_prd/2;
39     clock <= '0';
40     wait for clk_prd/2;
41     clock <= '1';
42     wait for clk_prd/2;
43     clock <= '0';
44     wait for clk_prd/2;
45     clock <= '1';
46     wait for clk_prd/2;
47     clock <= '0';
48     wait for clk_prd/2;
49     clock <= '1';
50     wait for clk_prd/2;
51     clock <= '0';
52     wait for clk_prd/2;
53     clock <= '1';
54     wait for clk_prd/2;
```

```
29  begin
30      UUT : counter_async
31      port map (
32          Tin => Tin,
33          clock => clock,
34          rst => rst,
35          output => output
36      );
37      process is
38      begin
39          Tin <= '1';
40          rst <= '1';
41          clock <= '0';
42          wait for clk_prd/2;
43          rst <= '0';
44          clock <= '1';
45          wait for clk_prd/2;
46          clock <= '0';
47          wait for clk_prd/2;
48          clock <= '1';
49          wait for clk_prd/2;
50          clock <= '0';
51          wait for clk_prd/2;
52          clock <= '1';
53          wait for clk_prd/2;
54          clock <= '0';
55          wait for clk_prd/2;
56          clock <= '1';
57          wait for clk_prd/2;
58          clock <= '0';
59          wait for clk_prd/2;
60          clock <= '1';
61          wait for clk_prd/2;
62          clock <= '0';
63          wait for clk_prd/2;
64          clock <= '1';
65          wait for clk_prd/2;
```

```

26  begin
27      UUT : descending_counter
28          port map (
29              Tin => Tin,
30              clock => clock,
31              rst => rst,
32              output => output
33          );
34      process is
35      begin
36          Tin <= '1';
37          rst <= '1';
38          clock <= '0';
39          wait for clk_prd/2;
40          rst <= '0';
41          clock <= '1';
42          wait for clk_prd/2;
43          clock <= '0';
44          wait for clk_prd/2;
45          clock <= '1';
46          wait for clk_prd/2;
47          clock <= '0';
48          wait for clk_prd/2;
49          clock <= '1';
50          wait for clk_prd/2;
51          clock <= '0';
52          wait for clk_prd/2;
53          clock <= '1';
54          wait for clk_prd/2;
55          clock <= '0';
56          wait for clk_prd/2;
57          clock <= '1';
58          wait for clk_prd/2;
59          clock <= '0';
60          wait for clk_prd/2;
61          clock <= '1';
62          wait for clk_prd/2;

```

جزئیات بیشتر تست بنچ ها در فایل های الصاق شده درون فایل زیپ قابل مشاهده است.

و در آخر خروجی بدست آمده از شمارنده آسنکرون:

