

۱- واحد تایمر با تنظیم دیوتی سائیکل

کد برنامه :

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.NUMERIC_STD.ALL;
4
5 entity Timer_8bits_PWM is
6     Port ( clk : in  STD_LOGIC;
7           Freq_in : integer;
8           Pulse_width : integer;
9           output : buffer STD_LOGIC);
10 end Timer_8bits_PWM;
11
12 architecture Behavioral of Timer_8bits_PWM is
13
14     COMPONENT DCM_IP
15     PORT(
16         CLKIN_IN : IN std_logic;
17         CLKFX_OUT : OUT std_logic
18     );
19     END COMPONENT;
20
21     signal clk_temp : STD_LOGIC ;
22 begin
23
24
25     Inst_DCM_IP: DCM_IP PORT MAP(
26         CLKIN_IN =>clk ,
27         CLKFX_OUT =>clk_temp
28     );
29 process ( clk_temp )
30
31     variable High : integer range 0 to 255 ;
32     variable i : integer range 0 to 255 := 0 ;
33
```

```

34 begin
35
36 High := (((Pulse_width)*(20*1000000/Freq_in))/256) ;
37
38 if rising_edge( clk_temp) then
39     i:=i+1 ;
40     OUTPUT <= '1' ;
41     if i=20*1000000/Freq_in then
42         i:=0 ;
43     elsif i > high then
44         OUTPUT <= '0' ;
45     end if ;
46
47 end if ;
48 end process ;
49
50
51 end Behavioral;
52

```

ابتدا با استفاده از IP Core مقدار فرکانس را به عدد ۲۰ مگاهرتز می‌رسانیم سپس مقدار دیوتی سیکل را با استفاده از سیگنال ۸ بیتی به پارامتر High تبدیل می‌کنیم. در واقع High میزان شمارش ۱ منطقی بودن را از مجموع میزان شمارش تعیین می‌کند. مجموع میزان شمارش نیز به صورت $20 \times \frac{1000000}{Freq_{in}}$ تعیین می‌شود. در واقع سیگنال ۲۰ مگاهرتز را ابتدا به ۱ مگاهرتز تبدیل می‌کنیم و سپس در معکوس نسبت مقدار فرکانس مطلوب به ۱ مگاهرتز ضرب می‌کنیم.

کد شبیه سازی :

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4
5  ENTITY test IS
6  END test;
7
8  ARCHITECTURE behavior OF test IS
9
10     -- Component Declaration for the Unit Under Test (UUT)
11
12     COMPONENT Timer_8bits_PWM
13     PORT(
14         clk : IN std_logic;
15         Freq_in : Integer;
16         Pulse_width : Integer;
17         output : buffer std_logic
18     );
19     END COMPONENT;
20
21
22     --Inputs
23     signal clk : std_logic := '0';
24     signal Freq_in : integer := 1000;
25     signal Pulse_width : integer := 75;
26
27     --Outputs
28     signal output : std_logic;
29
30     -- Clock period definitions
31     constant clk_period : time := 20 ns;
32
33 BEGIN

```

```

34
35     -- Instantiate the Unit Under Test (UUT)
36     uut: Timer_8bits_PWM PORT MAP (
37         clk => clk,
38         Freq_in => Freq_in,
39         Pulse_width => Pulse_width,
40         output => output
41     );
42
43     -- Clock process definitions
44     clk_process :process
45     begin
46         clk <= '0';
47         wait for clk_period/2;
48         clk <= '1';
49         wait for clk_period/2;
50     end process;
51
52
53     -- Stimulus process
54     stim_proc: process
55     begin
56         -- hold reset state for 100 ns.
57         wait for 100 ns;
58
59         wait for clk_period*10;
60         wait for 20 ms;
61         Freq_in <= 2000 ;
62         Pulse_width <= 190 ;
63
64         -- insert stimulus here
65
66         wait;

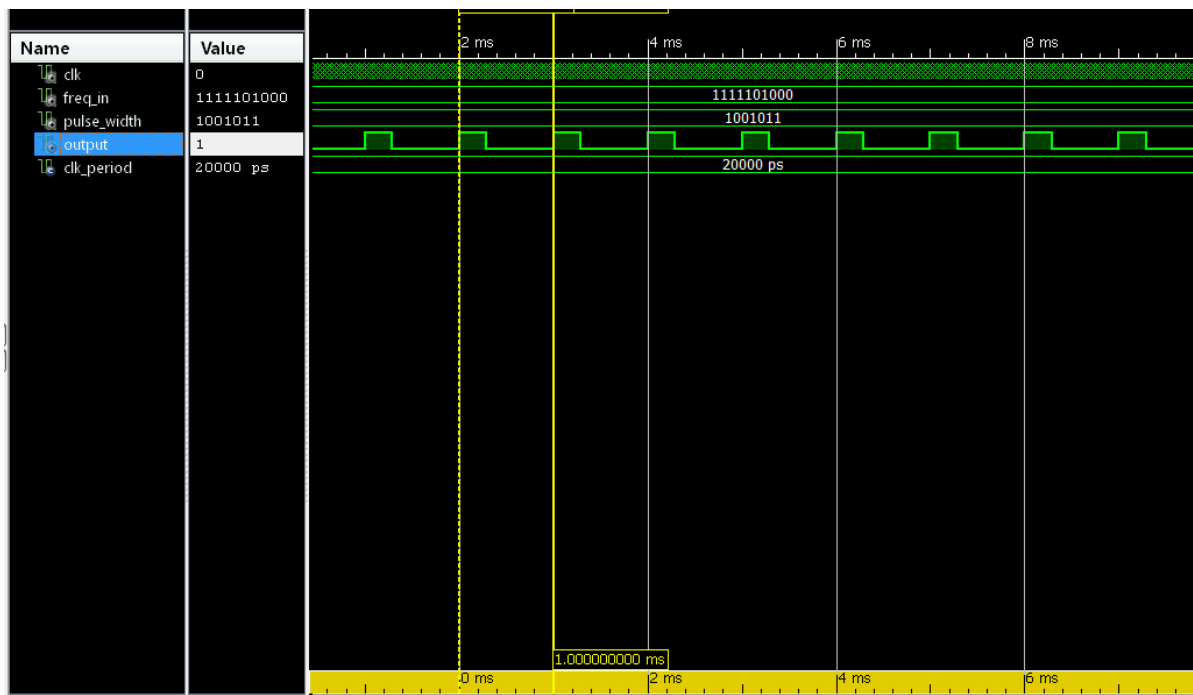
```

```

67     end process;
68
69 END;
70

```

خروجی :



فرکانس ۱ کیلو هرتز و دیوتی سایکل $\frac{75}{256}$ را مشاهده می کنیم.



فرکانس ۲ کیلو هرتز و دیوتی سایکل $\frac{190}{256}$ را مشاهده می کنیم.