پارسا محمدپور – ۹۸۲۴۳۰۵۰

پویا جهانگیری – ۹۸۲۴۳۰۷۶

#### گزارش جلسه چهارم

در این جلسه ابتدا کد های VHDL قبلی که در پیش گزارش زده بودیم را نشان دادیم و آنها را در مدلسیم دوباره اجرا کردیم.

سپس به سراغ پیادهسازی تابع داده شده در قسمت ب رفتیم و آن را ابتدا در پروتئوس پیادهسازی کردیم و سپس کد VHDL آن را زدیم و سپس آن را در مدل سیم اجرا کردیم تا ببینیم نتایج خروجی در فایل TestBench به چه صورت است. ابتدا گزارش پروتئوس را می آوریم.

در این قسمت به ما فته شده بود تا با توجه به دیکودر پیادهسازی شده در قسمت قبل و گیت های پایه، تابع زیر را پیادهسازی کنیم. این تابع به صورت زیر میباشد:

 $F(A, B, C, D) = \sum m(1, 2, 4, 7, 8, 11, 13, 14)$ 

حال با توجه به این تابع، اگر کارنومپ آن را بنویسیم، باید در خانههایی که اعداد آن در بالا آمده است، یک بگذاریم و در بقیه جاها صفر بگذاریم. پس ابتدا باید بدانیم در یک کارنومپ چهار تایی، شماره خانهها به چه صورت است. پس ابتدا یک کارنومپ چهار تایی میکشیم که شماره خانه در آن مشخص باشد. این کارنومپ به صورت زیر میباشد:

AB	00	01	11	10
CD				
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

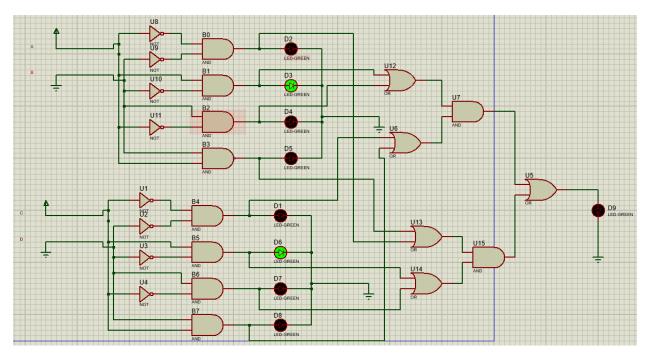
حال با توجه به شمارههای این جدول و تابع گفته شده، کارنومپ حاصل را رسم می کنیم. این کارنومپ به صورت زیر می باشد:

AB	00	01	11	10
CD				
00		1		1
01	1		1	
11		1		1
10	1		1	

همنطور که در این کارنومپ مشاهده می کنیم، این مدار مداری است که تنها در حالتی خروجی یک می شود که تعداد بیت های یک ورودی فرد باشد. یا همان مدار XOR چهار تایی می باشد. استفاده این مدار در parity bit ها و parity bit ها استفاده می شود.

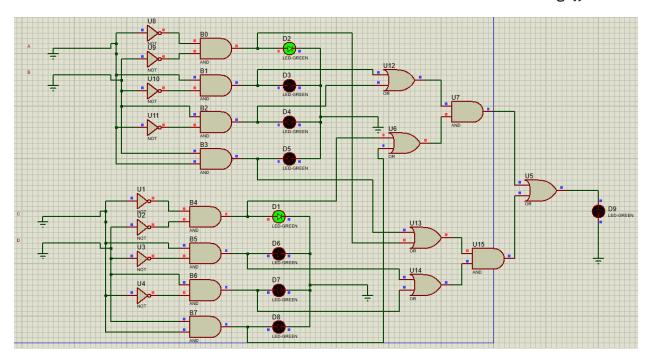
حال به سراغ پیادهسازی این مدار در پروتئوس میرویم و سپس حاصل امتحان کردن تمام ورودی های ممکن را در آن نشان میدهیم.

ابتدا به سراغ شماتیک مدار میرویم. با توجه به توضیحات داده شده، این مدار خروجیاش در حالتهایی یک است که تعداد یک های ورودی فرد باشد. همچنین ما در این پیادهسازی تنها مجاز به استفاده از گیت های پایخ و دیکودری که در قسمت قببل پیادهسازی کردیم، هستیم. بنابراین باید خطهایی از دیکودر را که ورودی های آن تعداد فرد (یک) بیت با ورودی یک دارد از دو تا ورودی اول (A و B) را با حالت هایی از دیکودر دوم (دیکودری برای C و D) که تعداد یک های ورودی زوج است (صفر یا دو) بدهد همزمان اتفاق بیفتد یا زوجیت هر دو گروه برعکس باشد هم میشود. پس شماتیک این مدار با توجه به توضیحات ارائه شده به صورت زیر می شود:

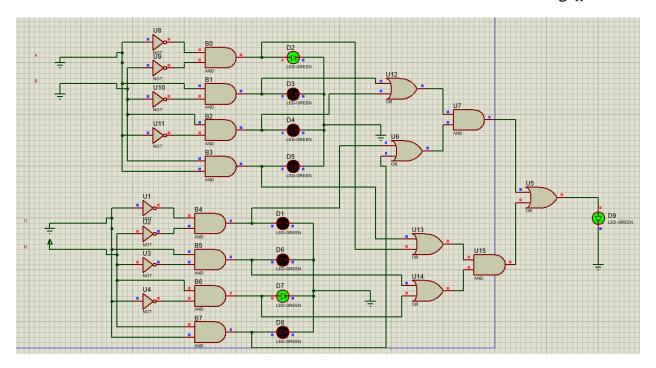


حال به این مدار تمام ورودی های ممکن را دااده و سپس نتایج را میآوریم. این نتایج به صورت زیر میباشد:

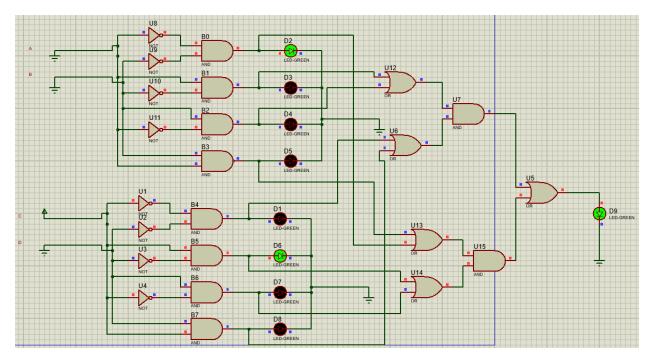
• خروجی حالت A=0, B=0, C=0, D=0:



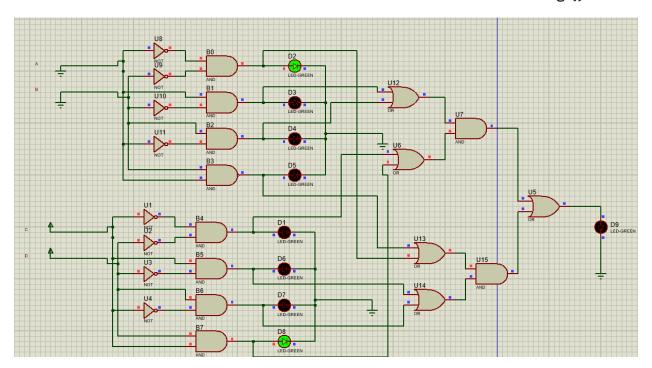
# • خروجي حالت A=0, B=0, C=0, D=1:



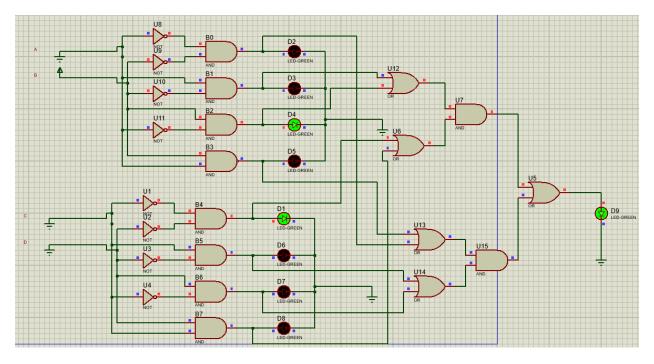
#### • خروجي حالت A=0, B=0, C=1, D=0:



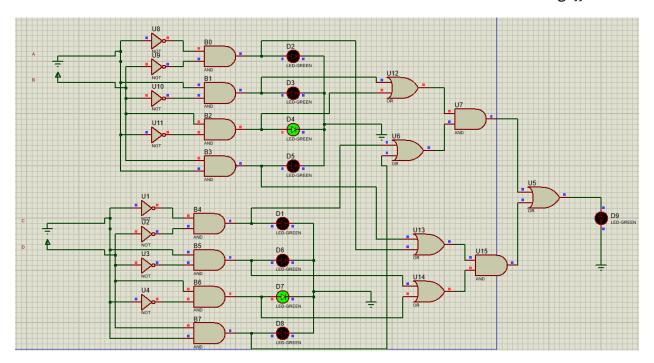
# • خروجي حالت A=0, B=0, C=1, D=1:



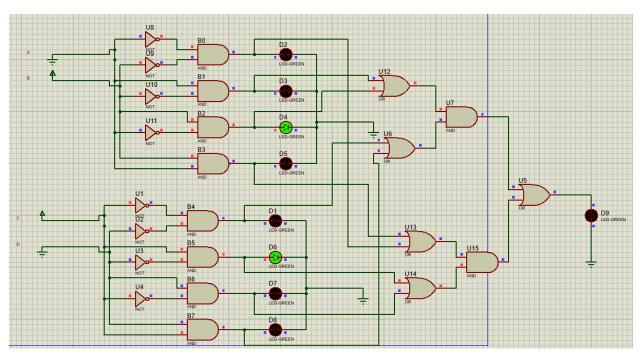
#### • خروجی حالت A=0, B=1, C=0, D=0



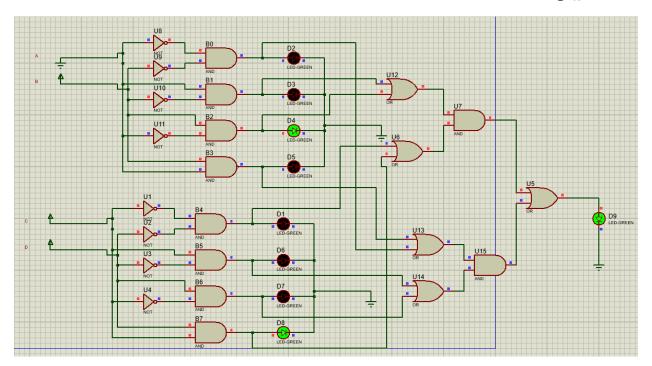
# • خروجي حالت A=0, B=1, C=0, D=1:



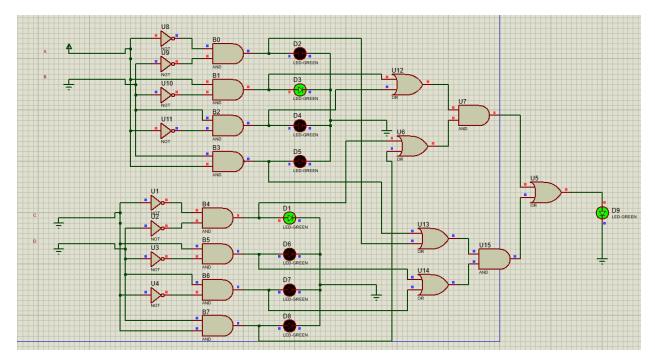
#### • خروجي حالت A=0, B=1, C=1, D=0



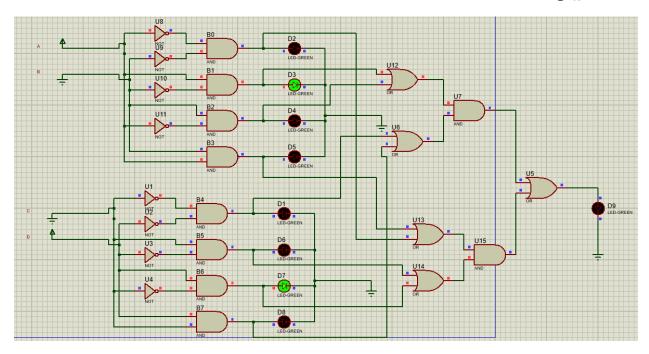
# • خروجي حالت A=0, B=1, C=1, D=1:



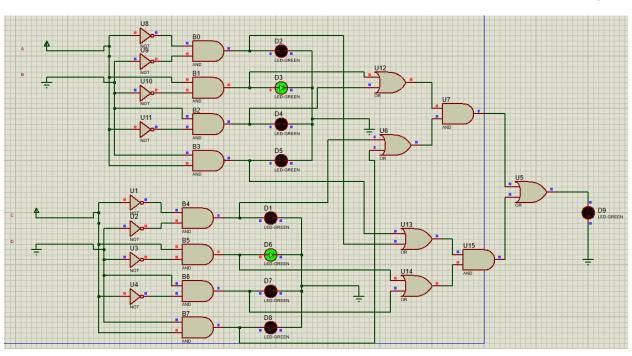
#### • خروجي حالت A=1, B=0, C=0, D=0:



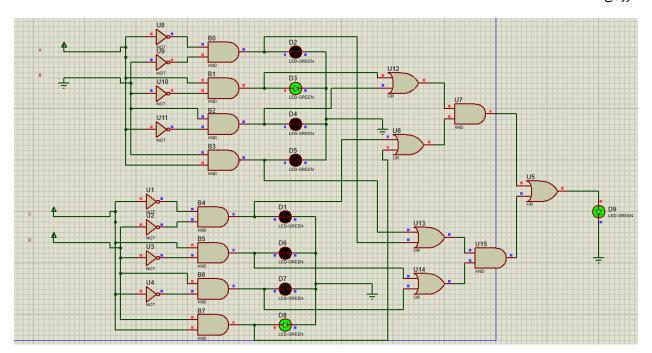
# • خروجي حالت A=1, B=0, C=0, D=1:



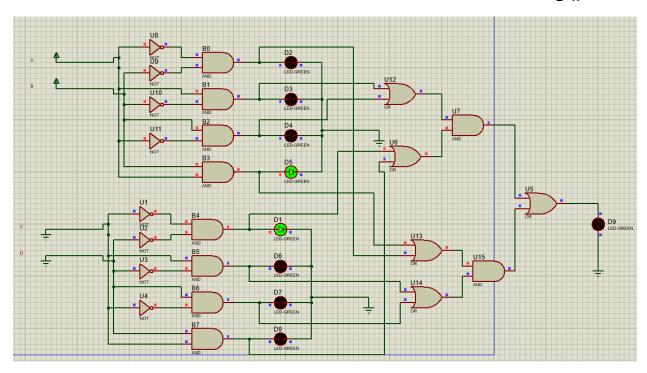
#### خروجی حالت A=1, B=0, C=1, D=0:



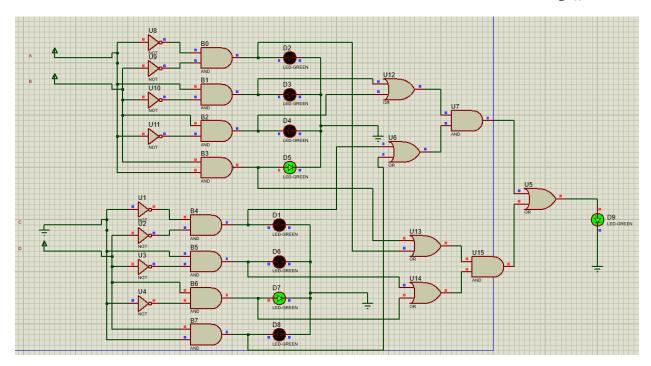
#### خروجی حالت A=1, B=0, C=1, D=1:



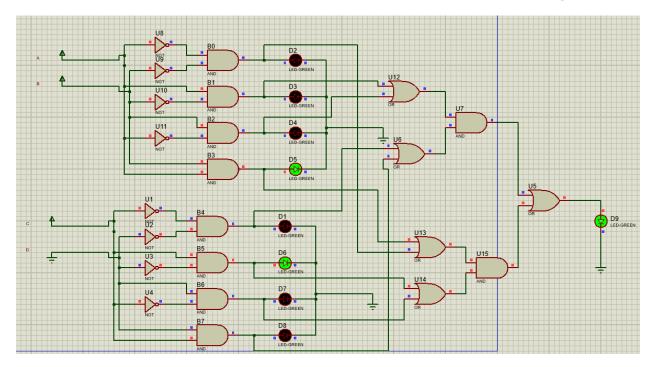
#### • خروجی حالت A=1, B=1, C=0, D=0:



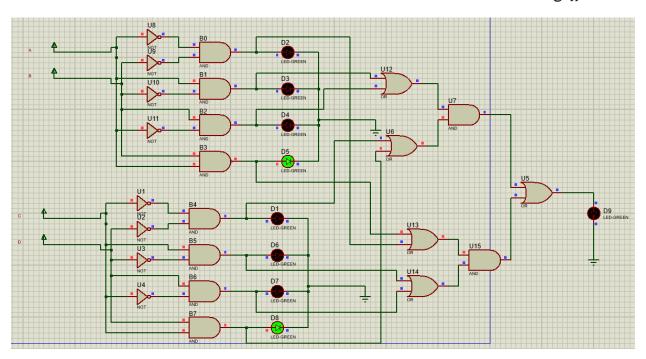
# • خروجي حالت A=1, B=1, C=0, D=1:



#### • خروجی حالت A=1, B=1, C=1, D=0:



#### • خروجي حالت A=1, B=1, C=1, D=1



حال به سراغ قسمت بعدی میرویم که پیادهسازی کد VHDL آن است. همانطور که توضیح دادیم این مدار مدار XOR چهارتایی است. پس می توان آن را به راحتی با XOR کردن A و A باهم و A باهم و A باهم و A کردن A و A باهم و A

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity functioncode is
port(
    a : in STD_LOGIC_VECTOR (3 downto 0);
    b : out STD_LOGIC
    end functioncode;

architecture bhv of functioncode is

begin

architecture bvv of functioncode is

begin

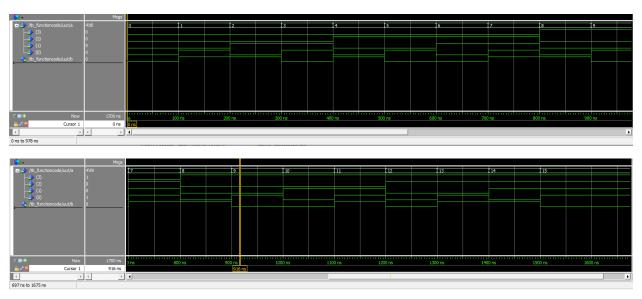
b <= ((a(0) xor a(1)) xor (a(2) xor a(3)));

end bvv;</pre>
```

سپس برای تست کردن این کد و اینکه آیا درست کار می کند یا خیر، فایل تستبنچ را نوشتیم که در آن همه ورودی های ممکن را به این ماژول می دهیم و سپس خروجی متناظر با آن را چک می کنیم. فایل تستبنچ به صورت زیر می باشد:

```
LTBRARY ieee;
USE ieee.std_logic_1164.ALL;
                                                                                    51
                                                                                                wait for 100 ns;
                                                                                                a <= "0101";
                                                                                    53
      ENTITY tb_functioncode IS
END tb_functioncode;
                                                                                               a <= "0110";
      ARCHITECTURE behavior OF tb_functioncode IS
          -- Component Declaration for the Unit Under Test (UUT)
                                                                                               a <= "0111";
      COMPONENT functioncode
PORT(
a: IN STD_LOGIC_VECTOR(3 downto 0);
b: OUT STD_LOGIC
                                                                                    59
wait for 100 ns;
                                                                                                a <= "1000";
                                                                                    62
                                                                                                wait for 100 ns;
                                                                                    65
          signal a : std_logic_vector(3 downto 0) := "0000";
                                                                                    66
                                                                                               a <= "1001";
         --Outputs
signal b : std_logic;
-- appropriate port name
                                                                                    68
                                                                                               wait for 100 ns;
                                                                                    69
70
71
72
73
74
75
76
77
78
79
80
                                                                                               a <= "1010";
         BEGIN
          -- Instantiate the Unit Under Test (UUT) uut: functioncode PORT MAP ( a \Rightarrow a, \\ b \Rightarrow b
                                                                                               wait for 100 ns;
a <= "1011";</pre>
                                                                                               a <= "1100";
       stim_proc: process
         begin
-- hold reset state for 100 ns.
wait for 100 ns;
                                                                                               a <= "1101";
                                                                                               wait for 100 ns;
a <= "1110";</pre>
         a <= "0001";
                                                                                    82
83
         wait for 100 ns;
                                                                                    84
85
86
                                                                                               wait for 100 ns;
a <= "11111";</pre>
         a <= "0010";
                                                                                    87
88
                                                                                               wait for 100 ns;
                                                                                    89
          wait for 100 ns;
a <= "0100";
                                                                                                wait;
                                                                                    90
91
          wait for 100 ns;
a <= "0101";</pre>
```

سپس به سراغ اجرای این کد در مدلسیم میرویم. نتیجه اجرای این کد در مدلسیم به صورت زیر میباشد:



همانطور که در تصویر مشاهده میکنیم، تنها در حالت هایی خروجی یک است، که ورودی یکی از اعداد گفته شده برای function باشد.

#### پیشگزارش آزمایش بعدی

در این قسمت باید ابتدا یک ماژولی طراحی کنیم که هم برای جمع و هم تفریق یک بیتی کار کند و سپس چند تا از آن ها را بهم وصل کنیم تا خروجی، بتواند حاصل جمع یا تفریق چهاربیتی را انجام دهد. کد های مورد نظر برای این کار زده شده است و در پیوست قرار دارد.(به دلیل زیاد بودنشان، آن ها را در اینجا نیاوردیم) همچنین خروجی خاصل از نرم افزار مدلی سیم به صورت زیر می باشد:

