

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 آن ها را در مدلسیم دوباره اجرا کردیم.

سپس به سراغ پیاده سازی تابع داده شده در قسمت ب رفتیم و آن را ابتدا در پروتئوس پیاده سازی 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 پیاده سازی شده در قسمت قبل و گیت های پایه، تابع زیر را پیاده سازی کنیم. این تابع به صورت زیر می باشد:

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

حال با توجه به این تابع، اگر کارنومپ آن را بنویسیم، باید در خانه هایی 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 خانه ها به چه صورت است. پس ابتدا یک کارنومپ چهار تایی میکشیم که شماره خانه در آن مشخص باشد. این کارنومپ به صورت زیر می باشد:

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 شماره های این جدول و تابع گفته شده، کارنومپ حاصل را رسم می کنیم. این کارنومپ به صورت زیر می باشد:

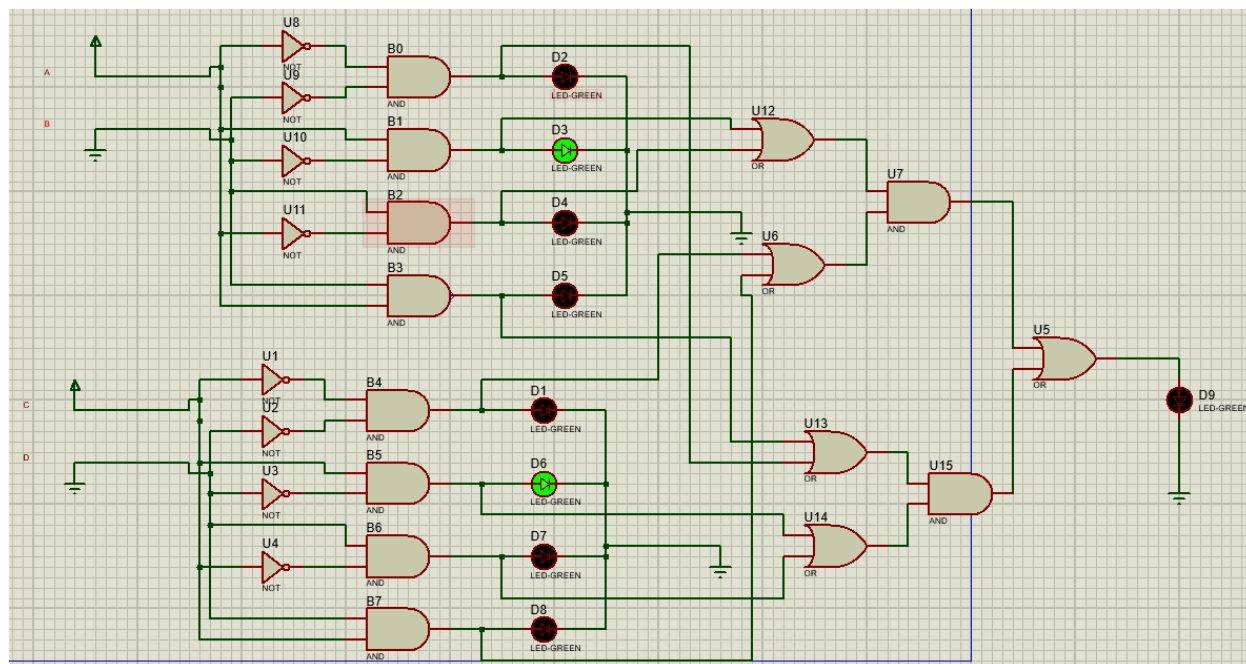
AB	00	01	11	10
CD				
00		1		1
01	1		1	
11		1		1
10	1		1	

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 full adder ها و parity bit ها استفاده می شود.

حال به سراغ پیاده سازی این مدار در پروتئوس می رویم و سپس حاصل امتحان کردن تمام ورودی های ممکن را در آن نشان می دهیم.

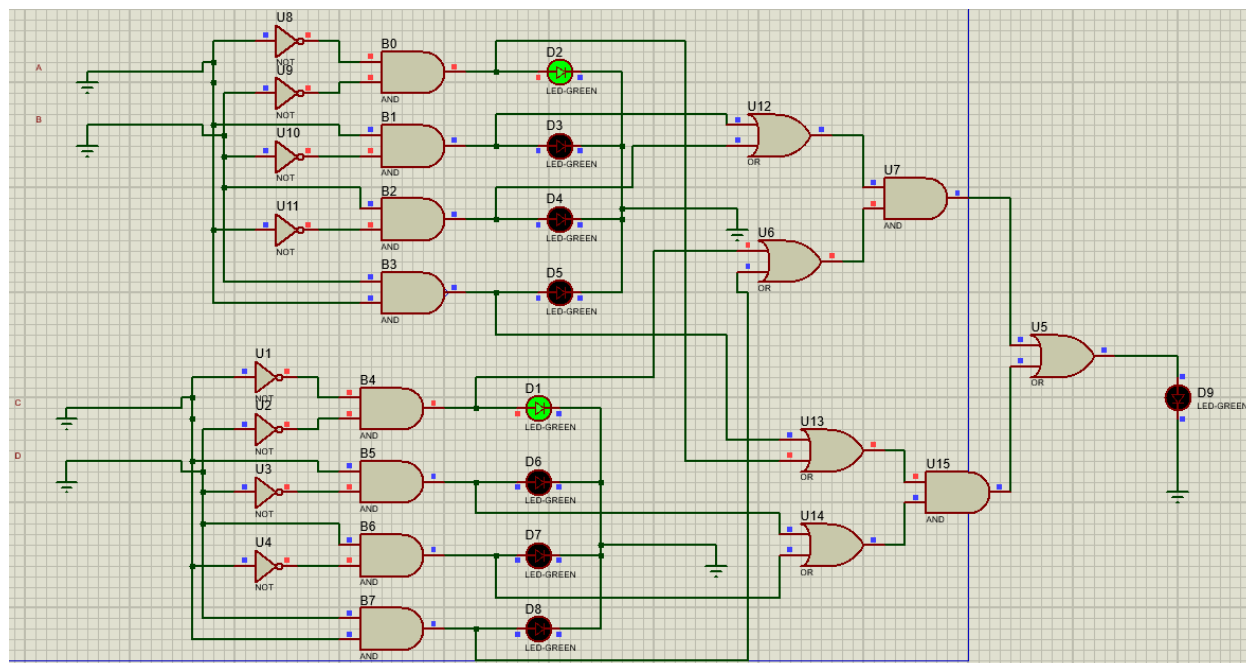
ابتدا به سراغ شماتیک مدار می رویم. با توجه به توضیحات داده شده، این مدار خروجی اش در حالت هایی یک است که تعداد یک های ورودی فرد باشد. همچنین ما در این پیاده سازی تنها مجاز به استفاده از گیت های پایخ و دیکودری که در قسمت قبیل پیاده سازی کردیم، هستیم.

بنابراین باید خط‌هایی از دیکودر را که ورودی‌های آن تعداد فرد (یک) بیت با ورودی یک دارد از دو تا ورودی اول (A و B) را با حالت‌هایی از دیکودر دوم (دیکودری برای C و D) که تعداد یک‌های ورودی زوج است (صفر یا دو) بدهد همزمان اتفاق بیفتند یا زوجیت هر دو گروه برعکس باشد هم می‌شود. پس شماتیک این مدار با توجه به توضیحات ارائه شده به صورت زیر می‌شود:

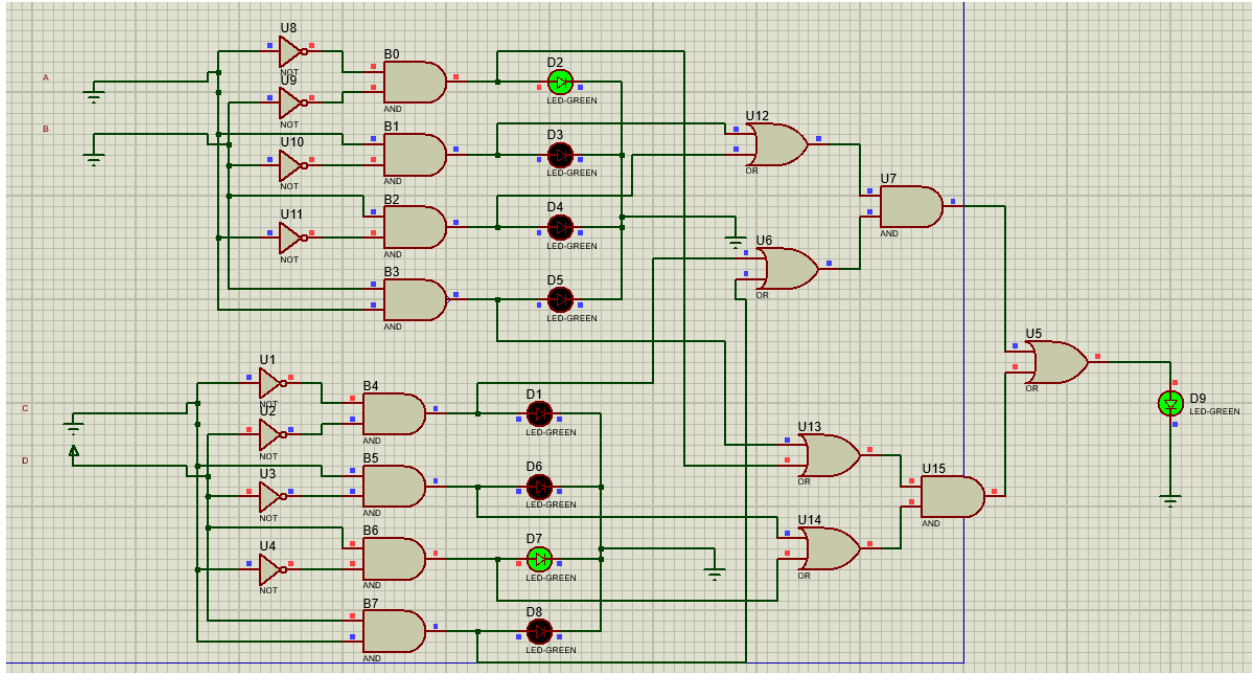


حال به این مدار تمام ورودی‌های ممکن را داده و سپس نتایج را می‌آوریم. این نتایج به صورت زیر می‌باشد:

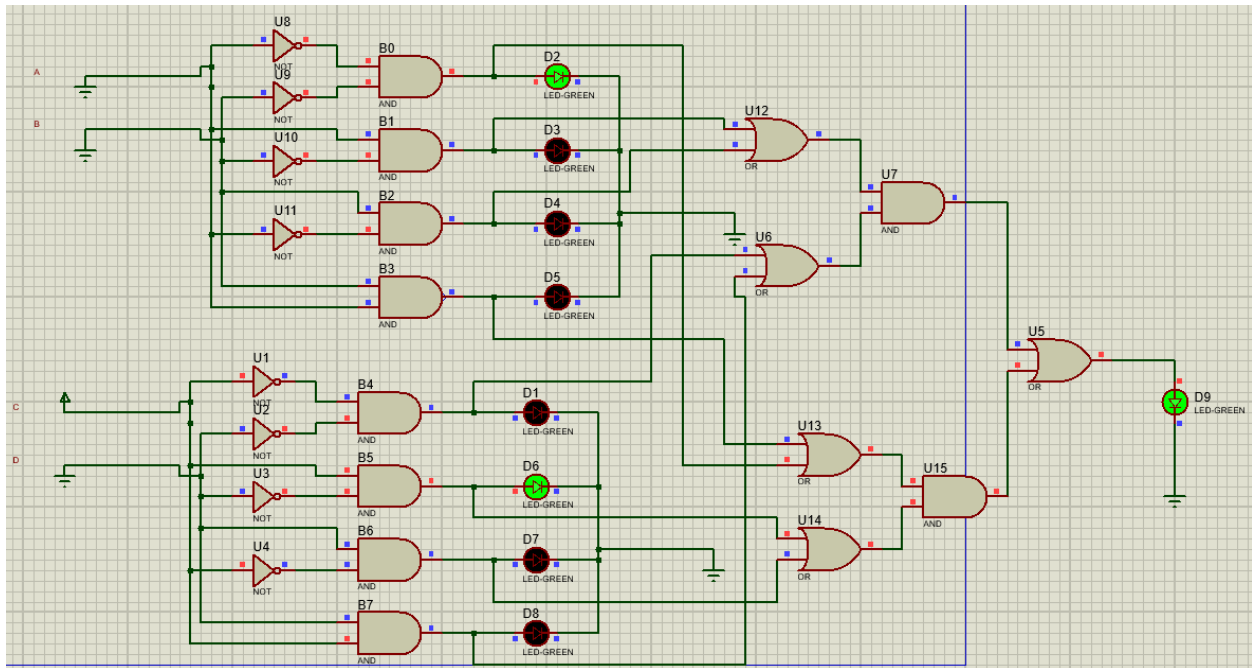
• خروجی حالت $A=0, B=0, C=0, D=0$:



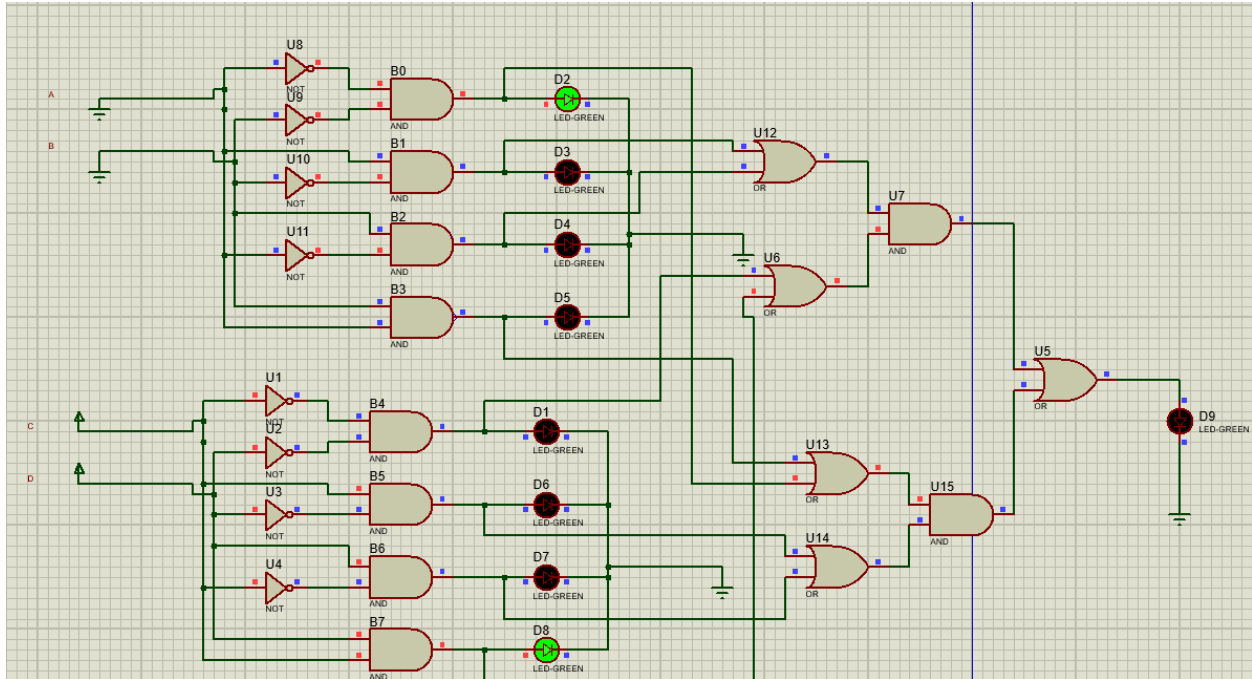
• خروجی حالت $A=0, B=0, C=0, D=1$:



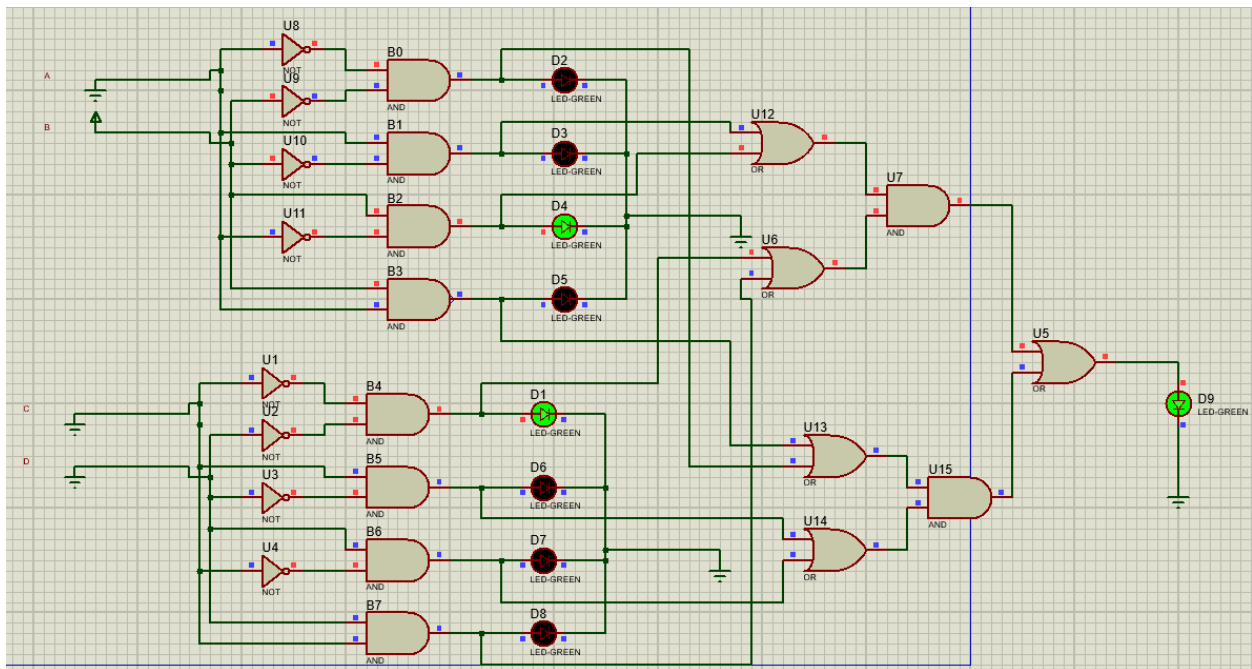
• خروجی حالت $A=0, B=0, C=1, D=0$:



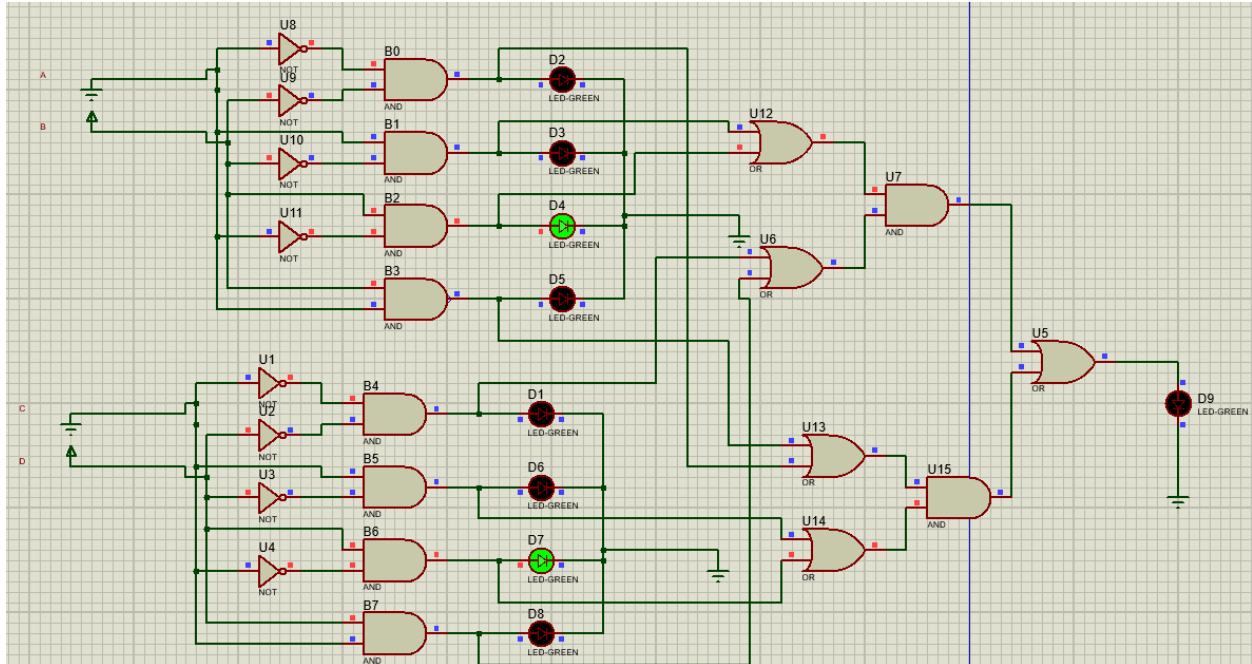
• خروجی حالت $A=0, B=0, C=1, D=1$:



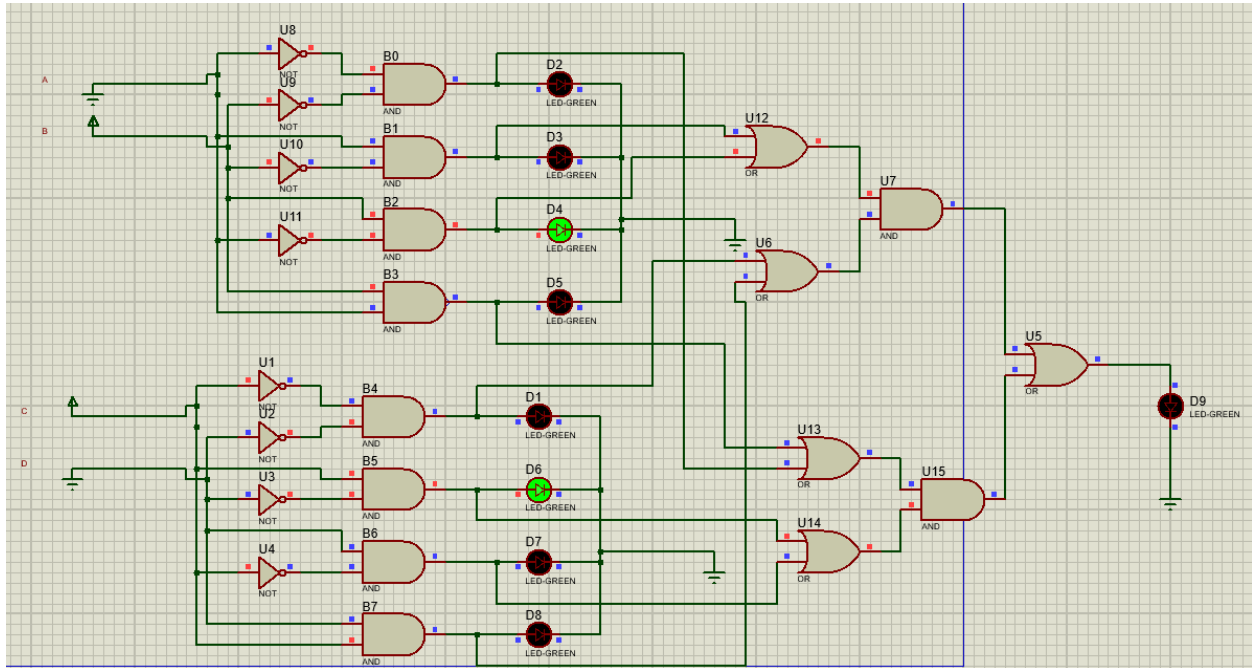
• خروجی حالت $A=0, B=1, C=0, D=0$:



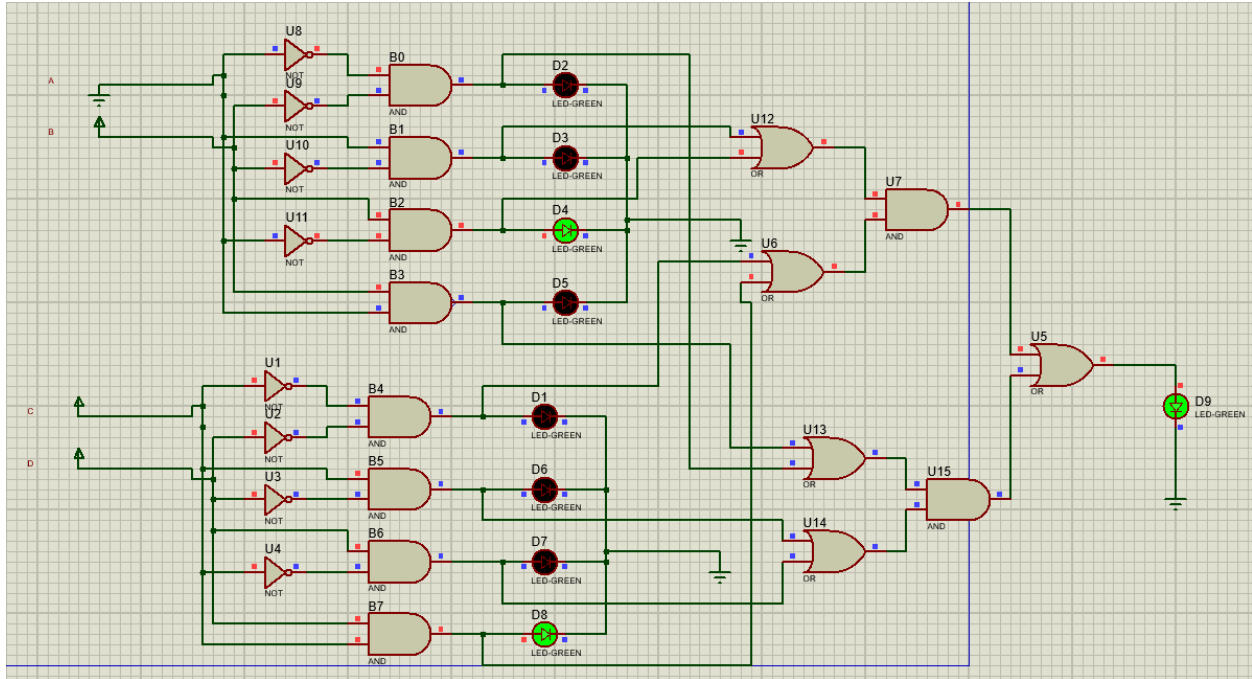
• خروجی حالت $A=0, B=1, C=0, D=1$:



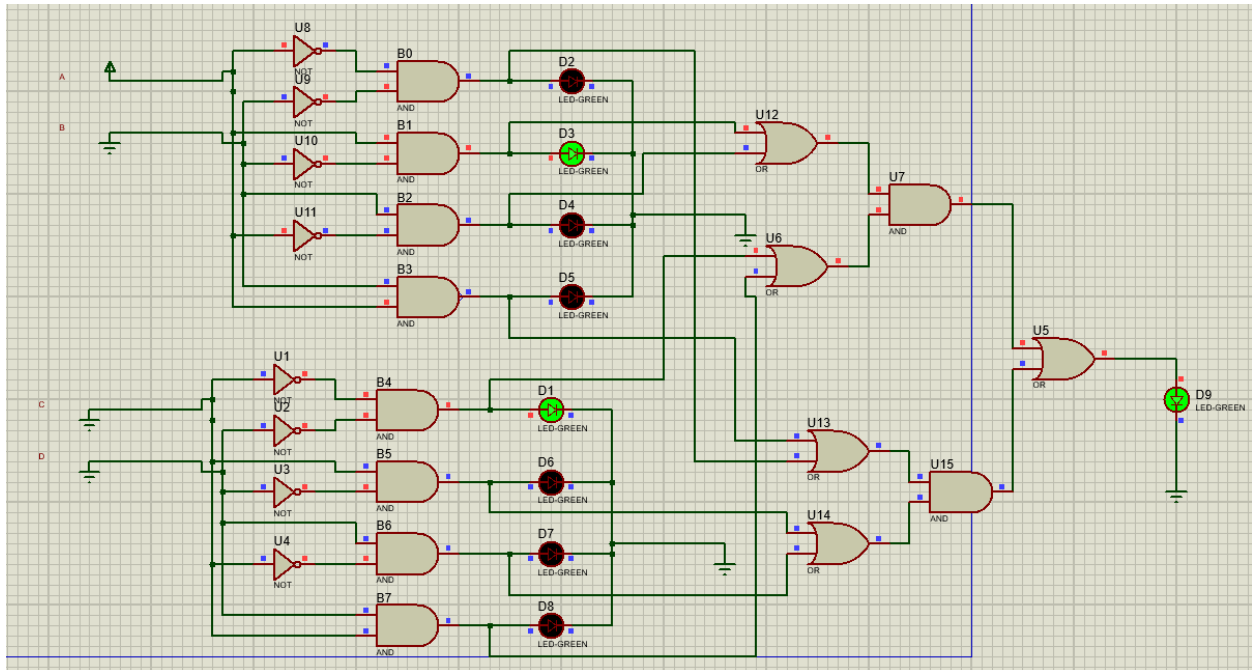
• خروجی حالت $A=0, B=1, C=1, D=0$:



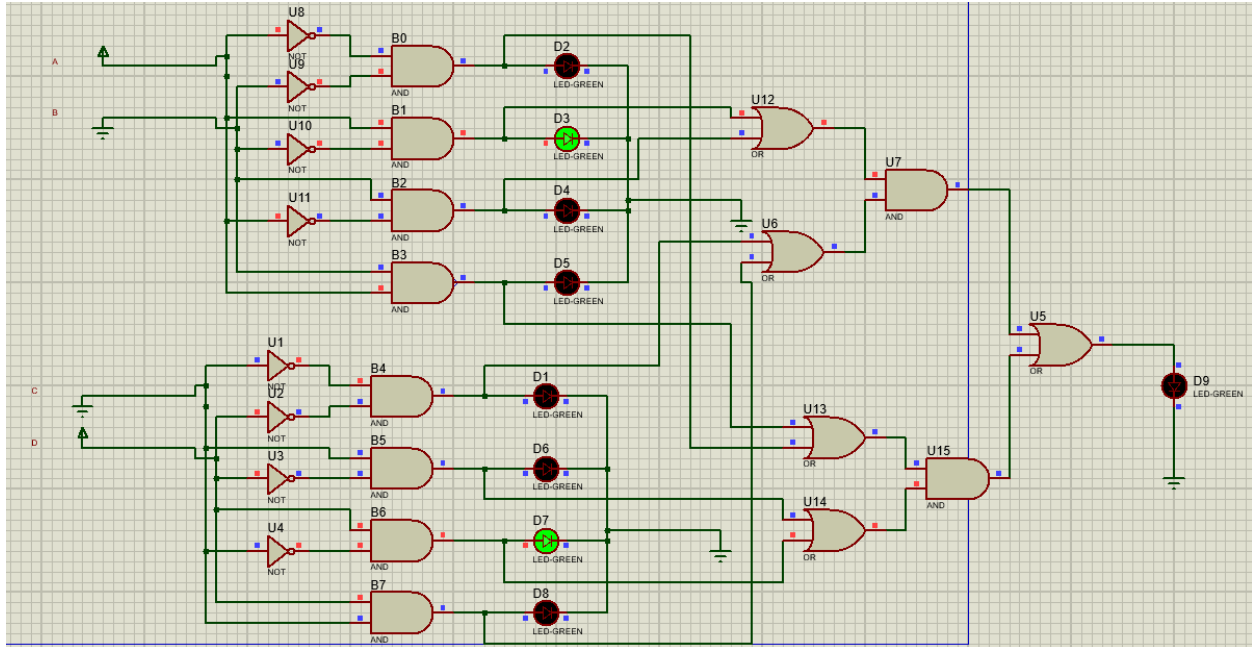
• خروجی حالت $A=0, B=1, C=1, D=1$:



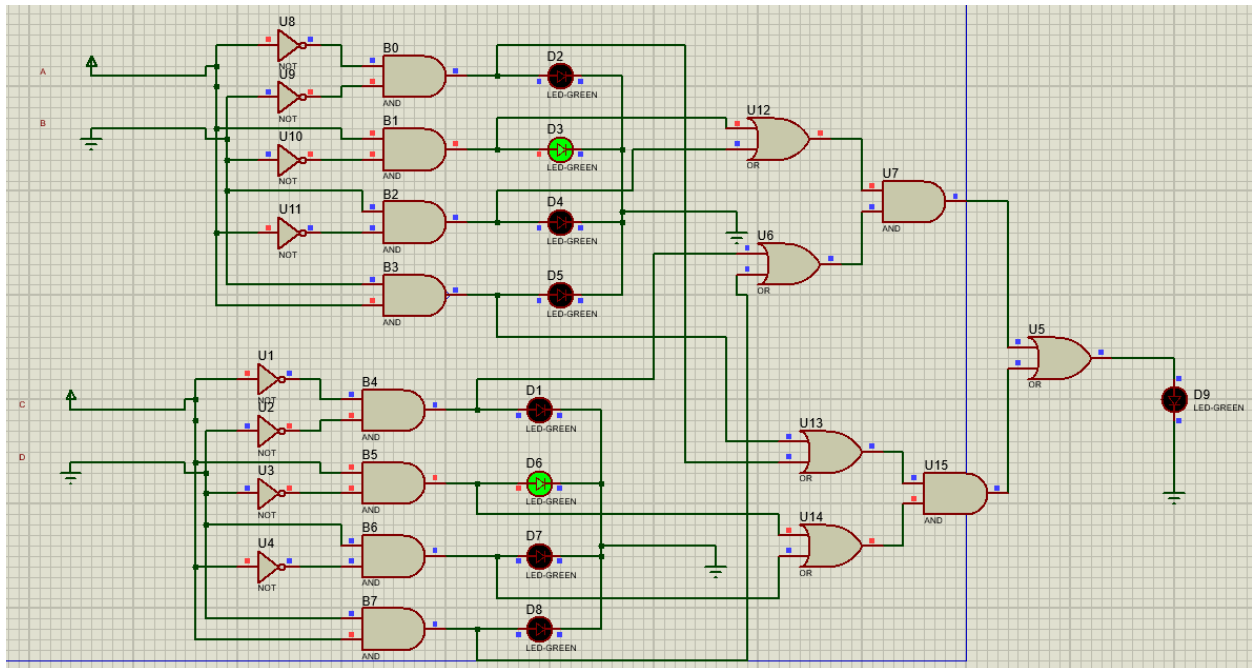
• خروجی حالت $A=1, B=0, C=0, D=0$:



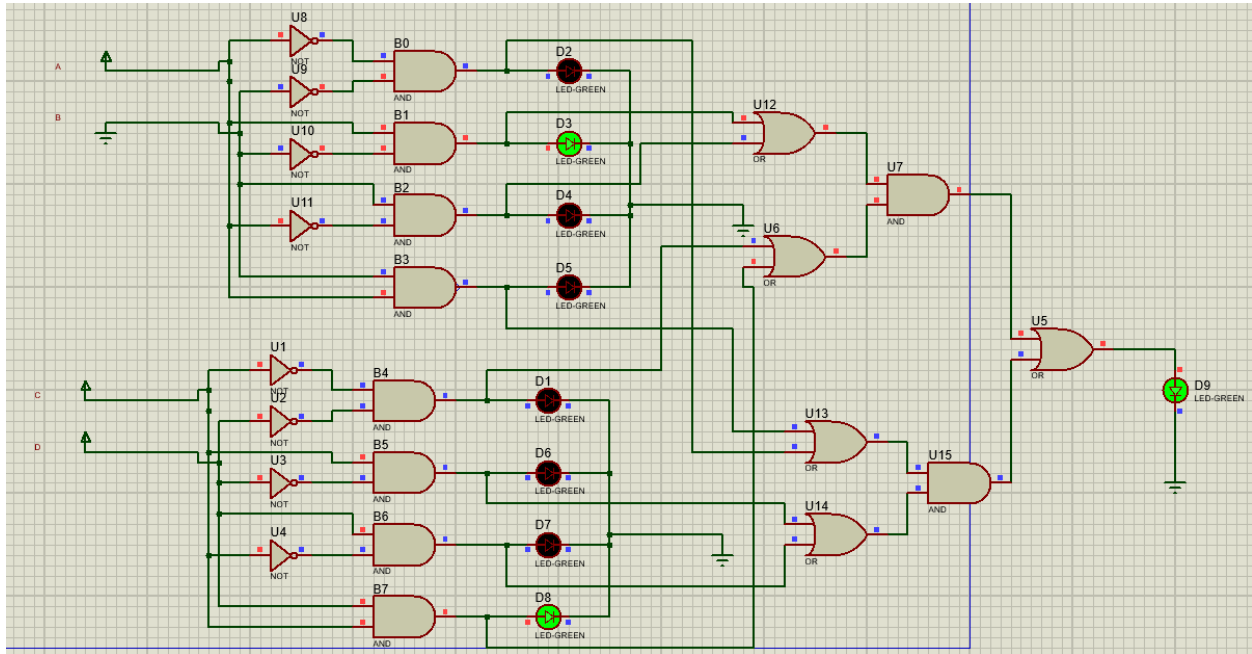
• خروجی حالت $A=1, B=0, C=0, D=1$:



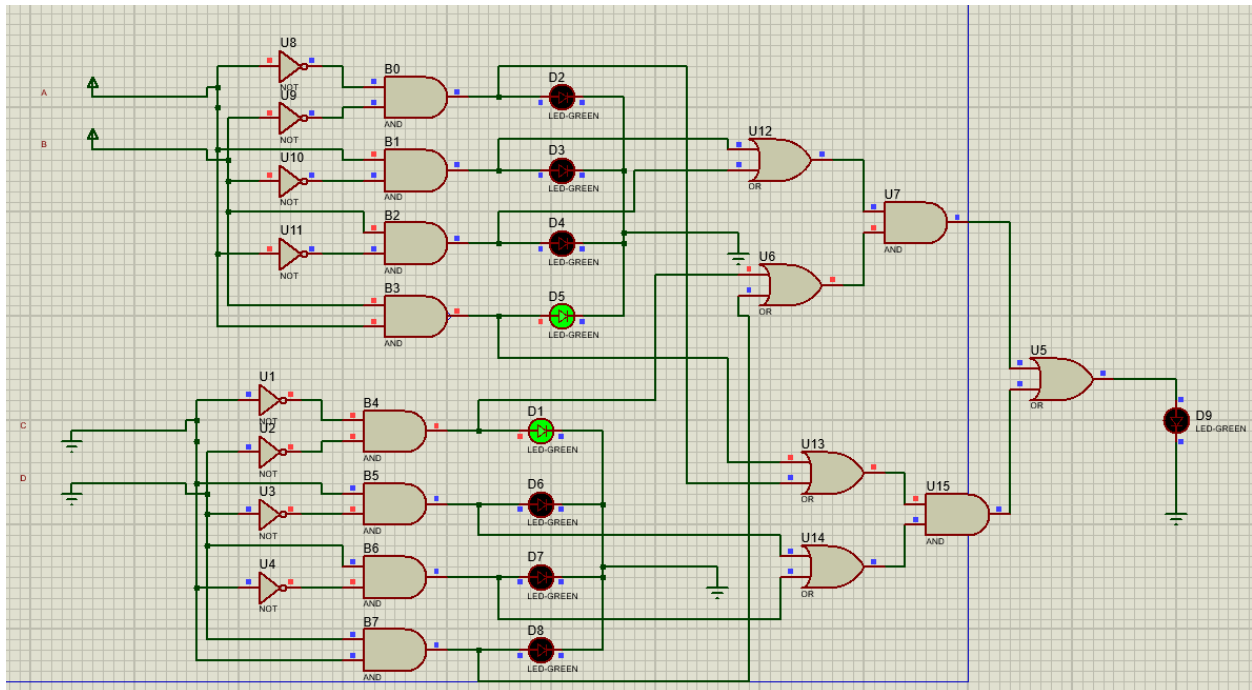
خروجی حالت $A=1, B=0, C=1, D=0$:



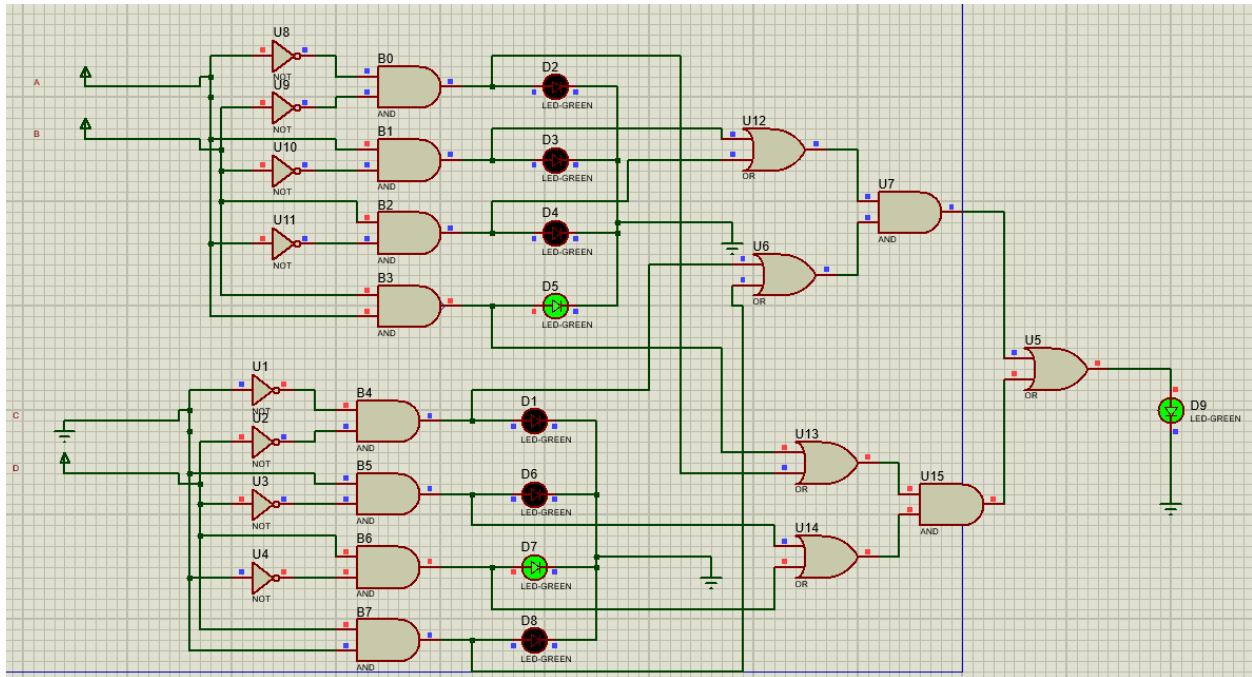
خروجی حالت $A=1, B=0, C=1, D=1$:



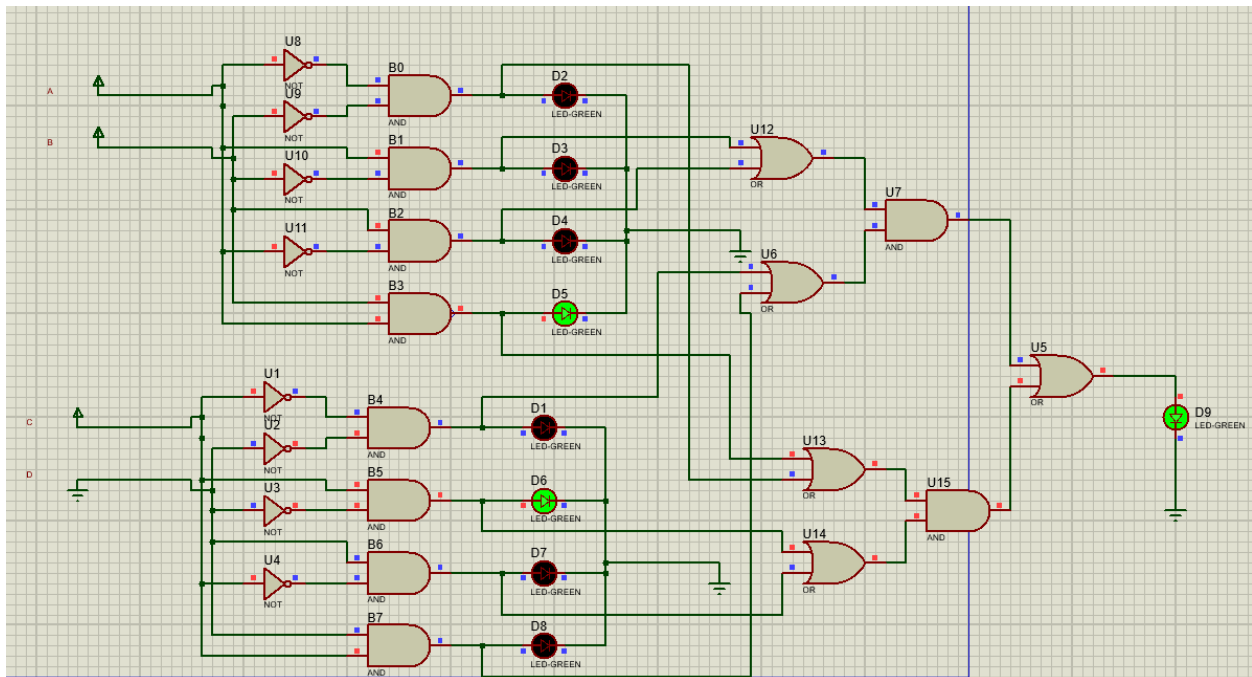
• خروجی حالت $A=1, B=1, C=0, D=0$:



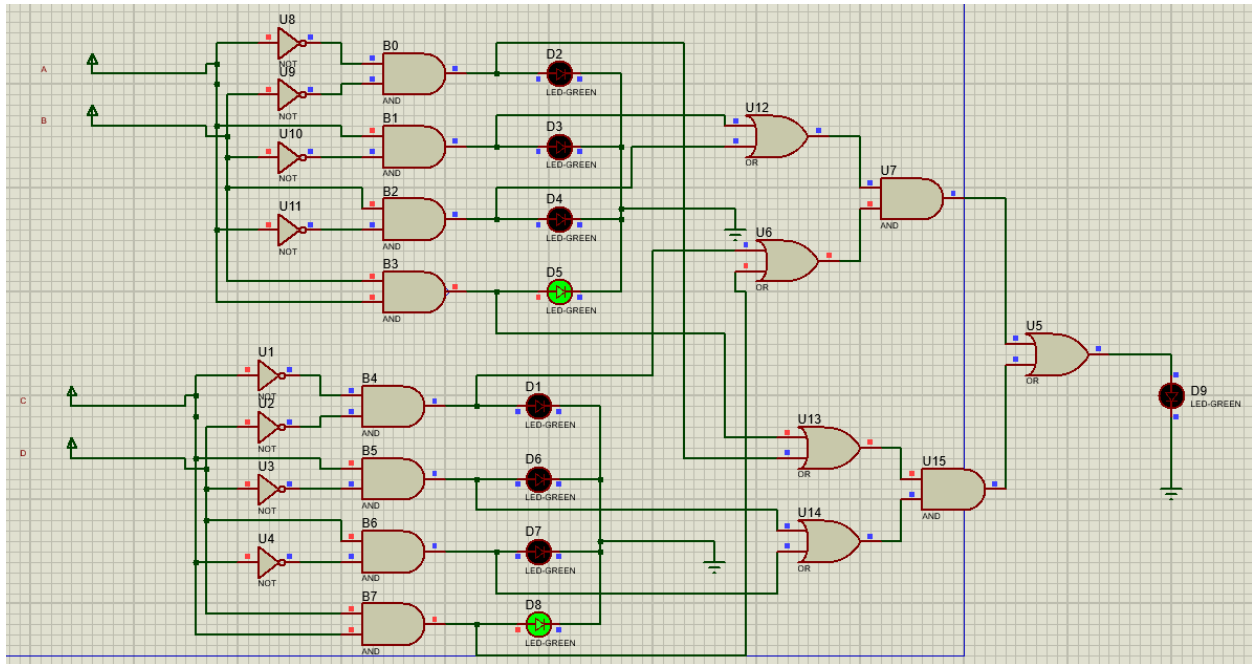
• خروجی حالت $A=1, B=1, C=0, D=1$:



• خروجی حالت $A=1, B=1, C=1, D=0$:



• خروجی 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 می‌توان آن را به راحتی با XOR کردن A و B باهم و XOR کردن C و D و XOR کردن نتایج‌شان بدست آورد. پس کد VHDL آن به صورت زیر خواهد بود:

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3
4  entity functioncode is
5  port(
6    a : in STD_LOGIC_VECTOR (3 downto 0);
7    b : out STD_LOGIC
8  );
9  end functioncode;
10
11 architecture bhv of functioncode is
12 begin
13
14   b <= ((a(0) xor a(1)) xor (a(2) xor a(3)));
15
16 end bhv;

```

سپس برای تست کردن این کد و اینکه آیا درست کار می‌کند یا خیر، فایل تست‌بنچ را نوشتیم که در آن همه ورودی‌های ممکن را به این ماژول می‌دهیم و سپس خروجی متناظر با آن را چک می‌کنیم. فایل تست‌بنچ به صورت زیر می‌باشد:

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 حاصل از نرم افزار مدلی سیم به صورت زیر می باشد:

