پارسا محمدپور – ۹۸۲۴۳۰۵۰

پویا جهانگیری – ۹۸۲۴۳۰۷۶

گزارش آزمایش دوم:

در این آزمایش از ما خواسته شده بود تا با استاده از جدول کارنو، عبارت های داده شده را تا حد ممکن ساده و سپس پیاده‌سازی کنیم.

- **الف)**

در این قسمت از ما خواسته شده تا عبارت زیر را با استفاده از جدو کارنو ساده‌سازی کرده و سپس مدار حاصل را پیاده‌سازی کنیم.

$F(A, B, C) = [(A' + B + C)(A + B)(A + C)]$

$= (A'B + AB + B + AC + BC)(A + C)$

$= A'BC + AB + ABC + AB + BC + AC + AC + ABC + BC$

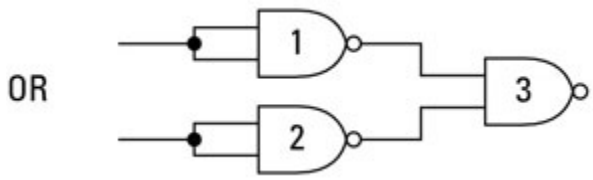
حال که مقدار معادل عبارت فوق را بدست آوردیم، با استفاده از جدول کرنو، مقدار ساده‌سازی شده آن را پیدا می‌کنیم:

A BC	0	1
00	0	0
01	0	1
11	1	111
10	0	1

همانطور که در جدول مشخص است، ما سه تا عبارت دو تایی داریم. این سه تا در جدول فوق با رنگ های یکسان مشخص شده اند. (اینکه در یک خانه سه تا عدد یک نوشته شده است برای این است که نشان دهیم عدد آن خانه چند بار مورد استفاده قرار گرفته است و رنگ های یکسان نشانه با هم در نظر گرفته شدن عبارت ها هستند). پس با استفاده از جدول کارنو به عبارت زیر می رسیم:

$F(A, B, C) = BC + AB + AC$

حال با توجه به عبارت بدست آمده به پیاده سازی مدار این عبارت می‌پردازیم. اما در هنگام پیاده‌سازی به دلیل اینکه تراشه گیت OR نداشتیم، از تراشه های گیت NAND برای پیاده‌سازی آن استفاده کردیم. برای این پیاده‌سازی (پیاده‌سازی گیت OR با گیت NAND) به صورت زیر عمل کردیم:



سپس مدار را مشابه آزمایش قبلی پیاده‌سازی کردیم و تمامی ورودی های مورد نظر را به آن اعمال کردیم و خروجی های متناظر را در جدول زیر وارد کردیم:

A	B	C	F
1	1	1	5.014
1	1	0	5.015
1	0	0	0.007
1	0	1	4.991
0	0	1	0.008
0	0	0	0.011
0	1	0	0.012
0	1	1	5.005

حال همانطور که در پیش توضیح داده شد به خاطر اتلاف و ... ممکن است اعداد بدست آمده خود صفر ولت و پنج ولت نباشند، و ما بازه بین سه یا سه و نیم ولت تا پنج ولت را برای یک منطقی، و بازه بین صفر ولت تا دو ولت را برای صفر منطقی در نظر می‌گیریم. بازه های بین را هم مقدار خالی می‌گذاریم که با هر تغییر خیلی کوچک در مدار، خروجی تغییر نکند یا با بروز نوسانات مختلف، مقدار خروجی، مدام تغییر نکند.

حال همانطور که از نتیجه جدول کارنو مشخص است، این گیت، گیت اکثریت (majority gate) می باشد. این گیت بدین گونه عمل می‌کند که اگر بیشتر از ۵۰٪ از ورودی ها یک باشند، خروجی آن نیز یک می‌شود. از کاربرد های این گیت می‌توان به جاهایی اشاره کرد که در آن چندین عامل تاثییرگذار در یک خروجی داریم و می‌خواهیم برآیند آن ها را بدانیم. علاوه بر این، در جمع کننده‌ها (adder)، تفریق کننده‌ها (subtractors)، توابع هش کننده (hash functions) و ... اشاره کرد.

- **ب)**

در این قسمت از ما خواسته شده است تا عبارت زیر را با استفاده از جدول کارنو ساده کرده و سپس مدار ساده‌سازی شده را پیاده‌سازی کنیم.

$$F(A, B, C, D) = \sum(1, 3, 4, 5, 6, 9, 11, 12, 14) + d(0, 7)$$

حال با توجه به اینکه در این سوال، اعداد داخل جدول را باید بدانیم، ابتدا در جدول زیر اعداد هر خانه را مشخص می‌کنیم.

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

حال با توجه به جدول بالا و اعداد داده شده در سوال، جدول کارنو این عبارت را بدست می‌آوریم:

AB CD	00	01	11	10
00	X	1	1	0
01	11	1	0	1
11	11	X	0	1
10	0	1	1	0

همانطور که می‌دانیم، می‌توان از مقادیر بی‌اهمیت (don’t care) در جدول، بنا به حالت مورد نظر، به عنوان صفر یا یک استفاده کرد. در این مثال مقادیر بی‌اهمیت با X نشان داده شده‌اند. مقدار X ای که در جدول با رنگ سبز در نظر گرفته شده است را برابر با یک در نظر می‌گیریم. سپس مانند مثال قبلی، رنگ‌های یکسان نشان دهنده این است که آن ها را با یکدیگر درنظر می‌گیریم. پس با توجه به جدول کارنو، به عبارت زیر می‌رسیم:

F(A, B, C, D) = DB’ + BD’ + A’D

حال مانند سوال قبل، مدار متناظر با این عبارت را پیاده‌سازی کردیم و سپس اعداد مورد نظر برای آن یادداشت کردیم. البته با توجه به اینکه مقدار ساده‌سازی شده این عبارت نشان می‌دهد که این عبارت مستقل از متغییر C می باشد و برای همین در مدار پیاده‌سازی شده، متغیر C را در نظر نگرفتیم تا هم تعداد حالت ها کمتر شود (از ۱۶ حالت به ۸ حالت رسیدیم) هم چون تاثییری در مدار و خروجی نداشت. اعداد اندازه گیری شده به شرح زیر می باشند:

A	B	D	F
1	1	1	0.350
1	1	0	5.018
1	0	0	0.005
1	0	1	5.012
0	0	1	5.010
0	1	1	5.011
0	1	0	5.005
0	0	0	0.004

همانطور که پیشتر توضیح داده شد برای ساده‌سازی در این مدار، یکی از مقادیر بی‌اهمیت را برابر با یک و دیگیری را بدون تغییر گذاشتیم. این مقادیر بی‌اهمیت حالت‌هایی هستند که بنا به استفاده می‌توان آن‌ها را صفر یا یک در نظر گرفت و رفتار مدار در این دو حالت برای ما اهمیتی ندارد. برای مثال اگر ورودی‌های این تابع، خروجی‌های تابعی دیگر باشند و آن تابع، خروجی مشابه مانند 0000 (مقادیر a, b, c, d در همین مثال) را در هیچ حالتی نتواند ایجاد کند، پس اهمیتی ندارد که ما در این مدار خروجی متناظر برای این حالت را چه چیزی در نظر بگیریم.

پیش گزارش آزمایش بعدی:

این آزمایش ما پیاده سازی مالتی‌پلکسر، دیکودر و انکودر را مرور می کنیم و سپس به پیاده سازی آن با گیت های پایه در زبان VHDL می‌پردازیم. سپس یک عبارت جبری را مانند سری های قبلی داریم که آن را ساده سازی می کنیم و سپس با استفاده از دیکودر این مدار را پیاده سازی می کنیم. همچنین کد های VHDL برای مدار های خواسته شده این سوال، به صورت زیر می باشد:

- Decoder:

کد VHDL ماژول مورد نظر و همچنین عکس حاصل از شبیه‌سازی به صورت زیر می باشد:

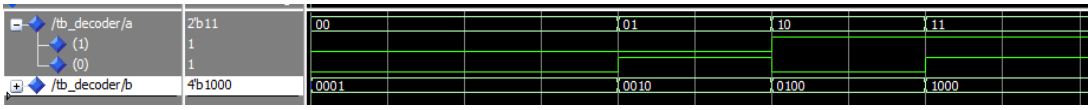
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity decoder is
 port(
  a : in STD_LOGIC_VECTOR(1 downto 0);
  b : out STD_LOGIC_VECTOR(3 downto 0)
 );
end decoder;

architecture bhv of decoder is
begin

b(0) <= not a(1) and not a(0);
b(1) <= not a(1) and a(0);
b(2) <= a(1) and not a(0);
b(3) <= a(1) and a(0);

end bhv;
```



- Encoder:

کد VHDL این ماژول مورد نظر به همراه عکس حاصل از شبیه‌سازی آن به صورت زیر است(در آخرین ورودی شبیه‌سازی نشان دهنده این است که اولویت با ورودی سمت چپی a است):

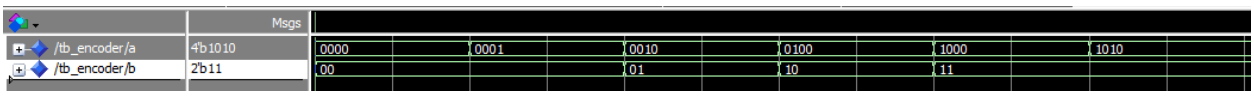
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity encoder is
 port(
  a : in STD_LOGIC_VECTOR(3 downto 0);
  b : out STD_LOGIC_VECTOR(1 downto 0)
 );
end encoder;

architecture bhv of encoder is
begin

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

end bhv;
```



- Multiplexer:

در تصویر زیر، کد VHDL ماژول به همراه عکس از شبیه‌سازی موجود می‌باشد:

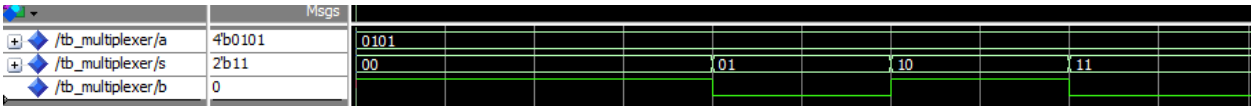
```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity multiplexer is
 port(
  a : in STD_LOGIC_VECTOR(3 downto 0);
  s : in STD_LOGIC_VECTOR(1 downto 0);
  b : out STD_LOGIC
 );
end multiplexer;

architecture bhv of multiplexer is
begin

b <= (a(0) and not s(0) and not s(1))
        or (a(1) and s(0) and not s(1))
        or (a(2) and not s(0) and s(1))
        or(a(3) and s(0) and s(1));

end bhv;
```



آموزه‌های این جلسه:

در این جلسه آزمایشگاه کار مجددا با ولت‌متر و منبع تغذیه و ... را یاد گرفتیم. علاوه بر آن به دلیل نبود تعداد کافی، مجبور شدیم گیت OR را با گیت‌های NAND پیاده سازی کنیم که در نوع خود جالب بود و همچنین کارکردن با مدارهایی که شلوغ می‌شوند و در آن‌ها سیم زیاد می‌شود، کارکردیم. علاوه بر این‌ها کار کردن با جدول کارنو یادآوری شد و همچنین برای پیش‌گزار آزمایش بعدی، کارکردن با modelsim و کدهای VHDL یادآوری شد و همنین پیاده‌سازی در سطح گیت ماژول‌های گفته شده نیز باعث یادآوری آن‌ها شد.