

به نام خدا

گزارش کار اول آزمایشگاه مدارهای منطقی



دانشکده مهندسی کامپیوتر

آزمایش اول: آشنایی با محیط‌های شبیه‌سازی

علی محمدزاده شبستری

۴۰۱۱۰۶۴۸۲

تابستان ۱۴۰۲

فهرست

مقدمه.....	۳
آشنایی با FRITZING.....	۳
آشنایی با LOGISIM.....	۵
آشنایی با PROTEUS.....	۹
جمع بندی.....	۱۲
منابع.....	۱۲

مقدمه

هدف این آزمایش آشنایی با سه نرم‌افزار شبیه‌سازی مدارهای منطقی به نام‌های Fritzing، Logisim و Proteus است.

این آزمایش در سه بخش (هر بخش برای هر نرم‌افزار) انجام می‌شود.

- در بخش اول، به کمک Fritzing با طرز کار و نحوه اتصالات یک بردبرد (breadboard) آشنا می‌شویم.
- در بخش دوم، با استفاده از Logisim یک مدار ترکیبی ساده (full adder) می‌سازیم.
- در بخش سوم، در فضای نرم‌افزار Proteus یک مدار ترکیبی پیچیده‌تر (carry look-ahead adder) را پیاده‌سازی می‌کنیم.

آشنایی با Fritzing

بخش تئوری ::

برای پیاده‌سازی مدارها باید قطعات آن‌ها را روی یک بردبرد قرار دهیم و از طریق آن سیم‌های اتصال را برقرار کنیم. لذا ابتدا لازم است با بردبرد آشنا شویم.

بردبرد (breadboard): یک تخته‌ی معمولاً پلاستیکی است که دارای سوراخ‌های ریز است و به راحتی می‌توانیم قطعات الکترونیکی را روی آن قرار دهیم. با استفاده از بردبرد می‌توانیم بدون نیاز به لحیم‌کاری قطعات را به هم وصل کنیم و نمونه اولیه‌ی مدار را پیاده‌سازی و آزمایش کنیم. می‌توانیم بردبرد را به چهار ناحیه تقسیم‌بندی کنیم:

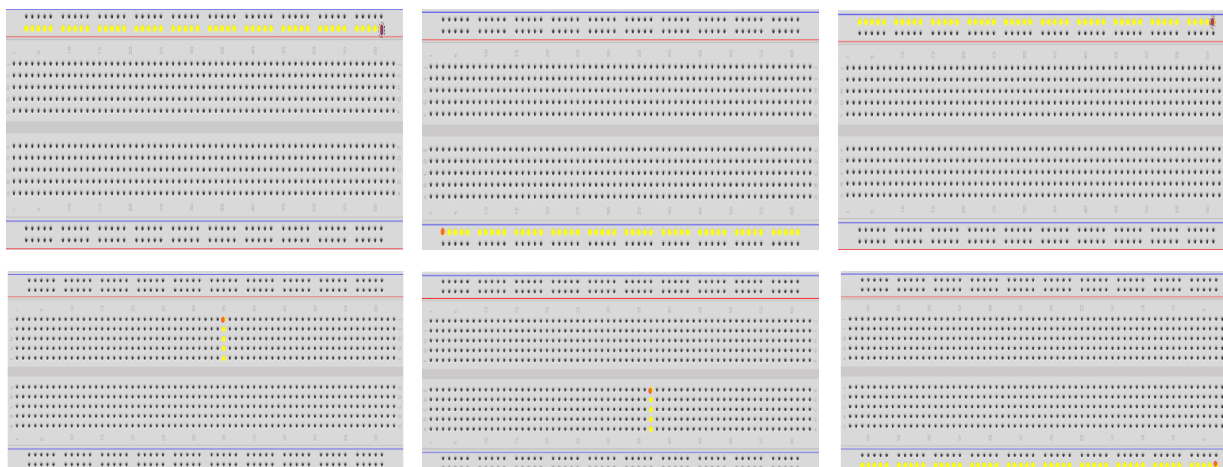
* خط تغذیه در بالای بردبرد

* خط تغذیه در پایین بردبرد

* بخش میانی بالا

* بخش میانی پایین

سوراخ‌های خط تغذیه بالا و همچنین سوراخ‌های خط تغذیه پایین همگی به صورت افقی به هم متصل هستند. ضمناً در هر یک از بخش‌های میانی، سوراخ‌هایی که در یک ستون قرار دارند به هم متصل هستند. لازم به ذکر است که هر کدام از بخش‌های میانی از یکدیگر جدا هستند و با هم هیچ ارتباط و اتصالی ندارند. خطی بین دو قسمت وجود دارد که آن‌دو را از یکدیگر جدا می‌کند و معمولاً محل قرارگیری ICها است.



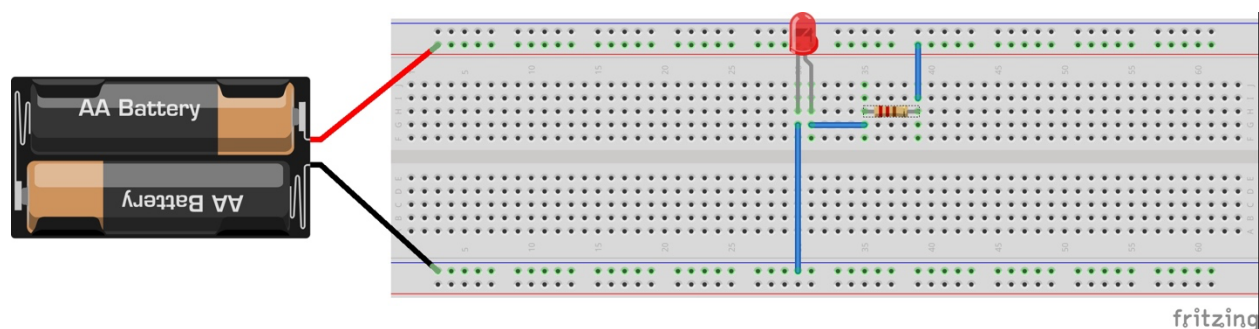
بخش عملی ::

آزمایش مدار LED:

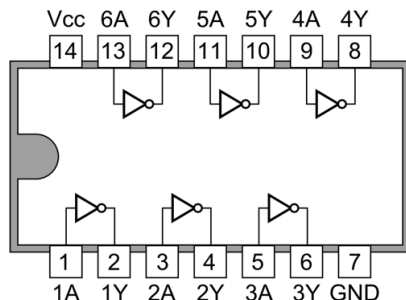
وسایل موردنیاز:

- * ال ای دی قرمز (LED Red (633nm
- * مقاومت ۲۲۰ اهم
- * منبع تغذیه ۳ ولت

می‌خواهیم روی یک بردبورد مداری بنویسیم تا یک LED روشن شود. هر دیود نورگسیل (LED) دارای دو پایه است که یکی از آن‌ها بلندتر است و باید به قطب مثبت منبع تغذیه (ولتاژ مثبت) وصل شود. چون ولتاژ باتری در حالت عادی زیاد است، می‌توانیم از یک مقاومت در مدار استفاده کنیم تا از سوختن LED جلوگیری شود. همان‌طور که در شکل زیر مشاهده می‌شود، ابتدا خطوط تغذیه بردبورد را به ترتیب به قطب مثبت و منفی منبع تغذیه (همان باتری) متصل می‌کنیم. در ادامه مقاومت و LED را به طور مناسب و صحیح در مدار قرار می‌دهیم و اتصالات لازم را ایجاد می‌کنیم. در آخر، از آنجایی که نرم‌افزار قابلیت شبیه‌سازی ندارد، با چک کردن سیم‌ها و اتصالات، از صحت عملکرد مدار مطمئن می‌شویم.



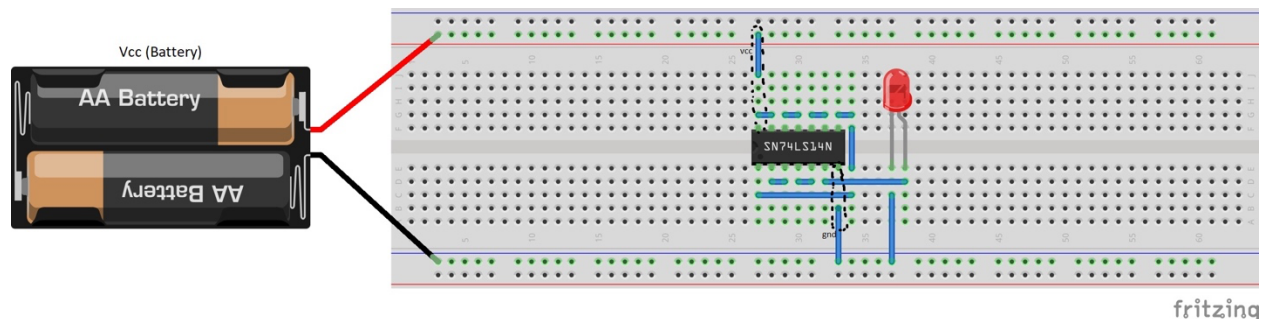
7404 Hex Inverters



۲- داخل تراشه ۷۴۰۴

تراشه 7404: این تراشه دارای ۶ گیت نات (inverter) و ۱۴ پایه است. پایه ۱۴ باید به Vcc و پایه ۷ باید به gnd وصل شود. همچنین Y ها صورت های نات شده ی A ها را به ما می دهند. پس برای ساختن شش گیت نات متوالی باید خروجی های Y را به A بعدی وصل کنیم تا به صورت زنجیروار نات بشوند. همچنین باید ۸ را به ۱ وصل کنیم.

در مدار زیر از تراشه SN74LS14N که در نرم افزار موجود بود استفاده کردیم. در نهایت اتصالات را بررسی می کنیم تا از صحت عملکرد مدار مطمئن شویم. Vcc به قطب مثبت باتری و gnd به قطب منفی باتری وصل شده است.

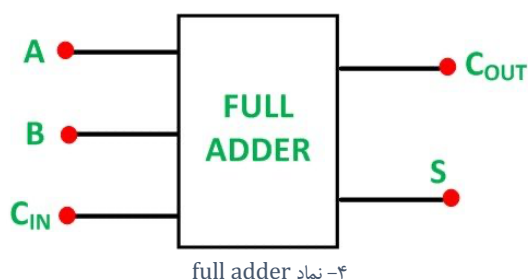


۳- مدار دارای تراشه ۷۴۰۴ و یک LED

آشنایی با Logisim

در این بخش می خواهیم جمع کننده تمام افزا (Full Adder) و جمع کننده/تفریق کننده ۴ بیتی بسازیم.

بخش تئوری ::

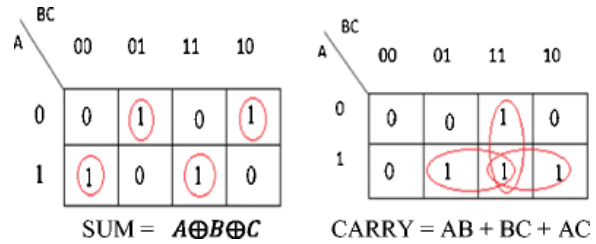


۴- نماد full adder

Full Adder: یک مدار ترکیبی است که برای جمع کردن دو بیت است. ورودی های آن بیت اول، بیت دوم و کری ورودی (احتمالاً از مرحله قبل) هستند. و در خروجی مجموع آن دو بیت و کری خروجی را می دهد. جدول درستی و جدول کارنو آن در زیر مشاهده می شود.

Inputs			Outputs	
A	B	C _{in}	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

۶- جدول درستی یک full adder



۵- جدول کارنوی یک full adder

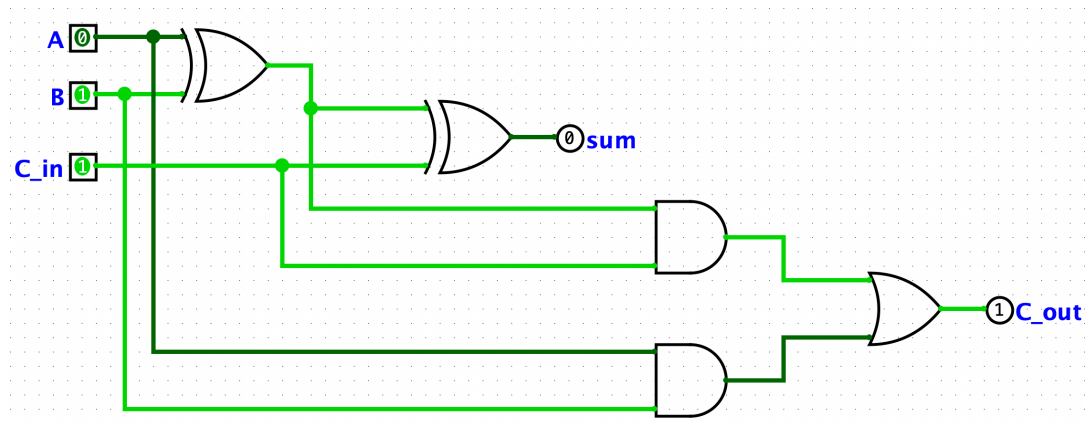
در جدول کارنوی بالا منظور از C همان کری ورودی (C_{in}) است. مشاهده می‌شود که تابع‌های خروجی به صورت زیر هستند:

$$sum = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

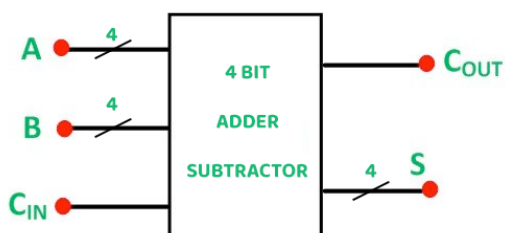
∴ بخش عملی ∴

مدار مربوطه را در فضای نرم‌افزار Logisim رسم می‌کنیم. برای تست می‌توانیم مقادیر $A=0$ و $B=1$ و $C_{in}=1$ را بدهیم. انتظار داریم جمع ۰ و ۱ و ۱ بشود ۰ با یک کری خروجی ۱. همان‌طور که پیداست مدار به درستی محاسبه کرده‌است.



۷- مدار full adder در برنامه Logisim

∴ بخش تئوری ∴



۷- جمع‌کننده/تفریق‌کننده چهار بیتی

4 Bit Adder/Subtractor: یک مدار ترکیبی است که

برای جمع کردن و تفریق کردن دو عدد چهاربیتی است. یک بیت دیگر (M) با مقدار C_{in} وجود دارد که مشخص می‌کند که باید جمع انجام شود یا تفریق.

ما می‌دانیم برای تفریق کردن دو عدد مثل a و b از مکمل دو استفاده می‌کنیم.

$$a + b = a + (-b) = a + b' + 1$$

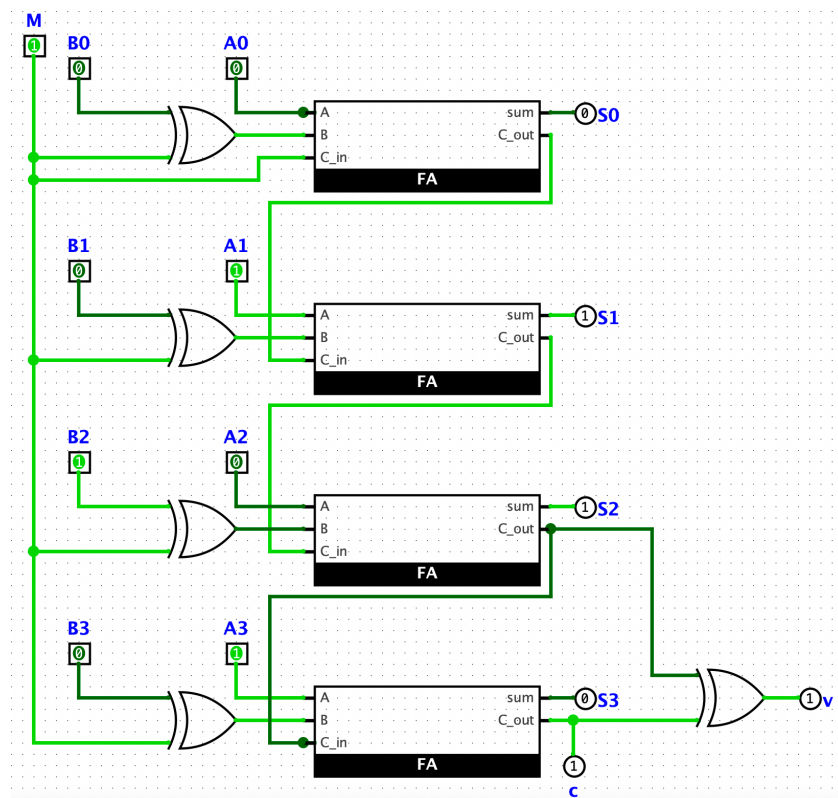
پس برای تفریق کردن کافی است a را با b' و ۱ جمع کنیم. برای جمع کردن هم همان a را با b جمع می‌کنیم. در قسمت تفریق ما باید بتوانیم عمل نات کردن b و همچنین اضافه کردن ۱ را با همان بیت C_{in} انجام دهیم. این کار با استفاده از گیت‌های XOR میسر می‌شود. به این صورت که هر بیت عدد b را با M XOR می‌کنیم.

* اگر $M=1$ باشد، آنگاه $b \oplus 1 = b'$ و عدد ۱ در قسمت C_{in} جمع می‌شود.

* اگر $M=0$ باشد، آنگاه $b \oplus 0 = b$ و هیچی به a و b اضافه نمی‌شود.

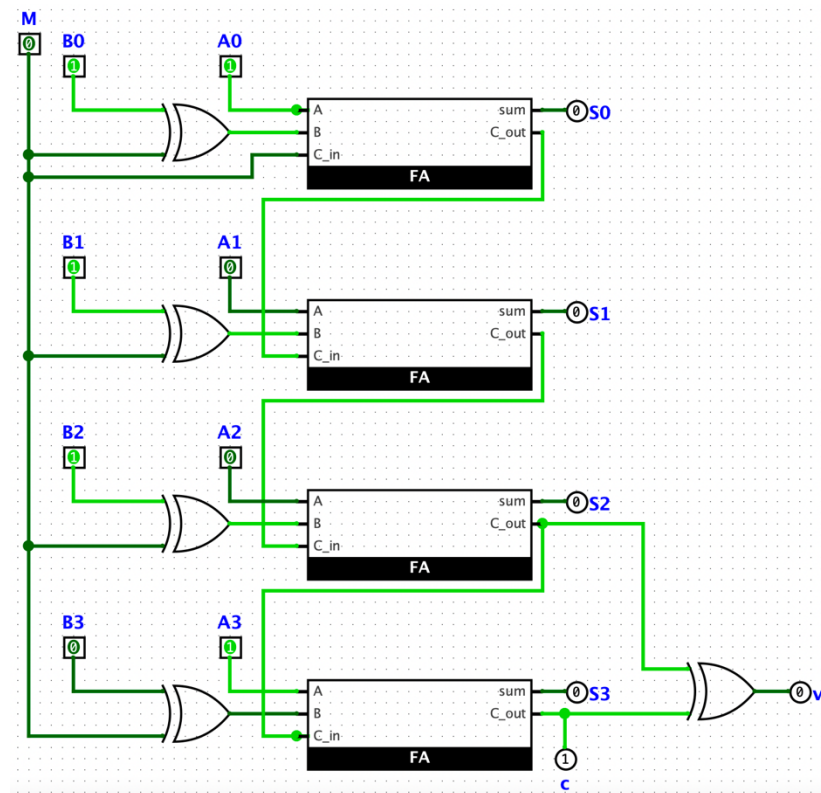
∴ بخش عملی ∴

4 Bit Subtractor



۸- مدار تفریق‌کننده چهار بیتی در برنامه Logisim

4 Bit Adder



۹- مدار جمع کننده چهار بیتی در برنامه Logisim

در دو تصویر بالا ما از Full Adder ای که در مرحله قبل ساختیم استفاده کردیم. به این صورت که در یک پروژه جدید، با استفاده از گزینه *Add Circuit* مدار ساخته شده توسط خودمان را به صورت باکس درآوردیم و آن را FA نامیدیم. این مدار یک خروجی **overflow** هم دارد که با علامت V نشان داده شده است. که حاصل $C_{in} \oplus C_{out}$ برای آخرین FA است. اگر ۱ باشد به معنی وقوع **overflow** است و اگر صفر باشد یعنی رخ نداده است.

برای امتحان کردن صحت کارکرد مدار ترکیبی ساخته شده، مقادیر امتحانی به ورودی‌هایمان داده‌ایم و آن را تست کرده‌ایم.

آشنایی با Proteus

بخش تئوری ::

آزمایشمان را مطرح کردن یک پرسش شروع می‌کنیم. در جمع‌کننده‌ای که در مرحله قبل با Logisim ساختیم، سرعت عملکرد مدار و تأخیر آن چگونه بود؟

ما برای هر مرحله از جمع‌کردن، باید کرای خروجی مرحله قبل را می‌گرفتیم تا بتوانیم جمع بعدی را انجام دهیم. و این یعنی برای جمع یا تفریق دو عدد ۸ بیتی، ۸ واحد زمانی تأخیر داریم. به این روش جمع کردن، اصطلاحاً Ripple Carry می‌گویند. آیا روش دیگری وجود دارد که از طریق آن بتوانیم جمع کردن بیت‌های با جایگاه یکسان را هم‌زمان و به صورت موازی انجام دهیم؟ در این صورت تأخیر ما بسیار کمتر خواهد شد. اما این کار مستلزم این است که کرای حاصل از جمع قبلی را خودمان جداگانه حساب کنیم. با این کار تعداد گیت‌های استفاده شده بیشتر می‌شود و مدار شلوغ‌تر می‌شود اما مزین آن زمان کمتر است. به این جمع‌کننده اصطلاحاً Carry Lookahead Adder (CLA) می‌گوییم.

برای ساخت این مدار ابتدا دو متغیر باینری به صورت زیر تعریف می‌کنیم:

$$P_i = a_i \oplus b_i \qquad G_i = a_i \cdot b_i$$

سپس خروجی sum و carry هر مرحله را برحسب آن دو می‌نویسیم.

$$S_i = P_i \oplus C_i \qquad C_{i+1} = G_i + P_i C_i$$

اکنون طبق فرمولی که به دست آورده بودیم، رابطه کرای هر مرحله را می‌نویسیم و به جای کرای مرحله قبل، از رابطه مربوط به آن استفاده می‌کنیم.

$$C_0 = \text{input carry}$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

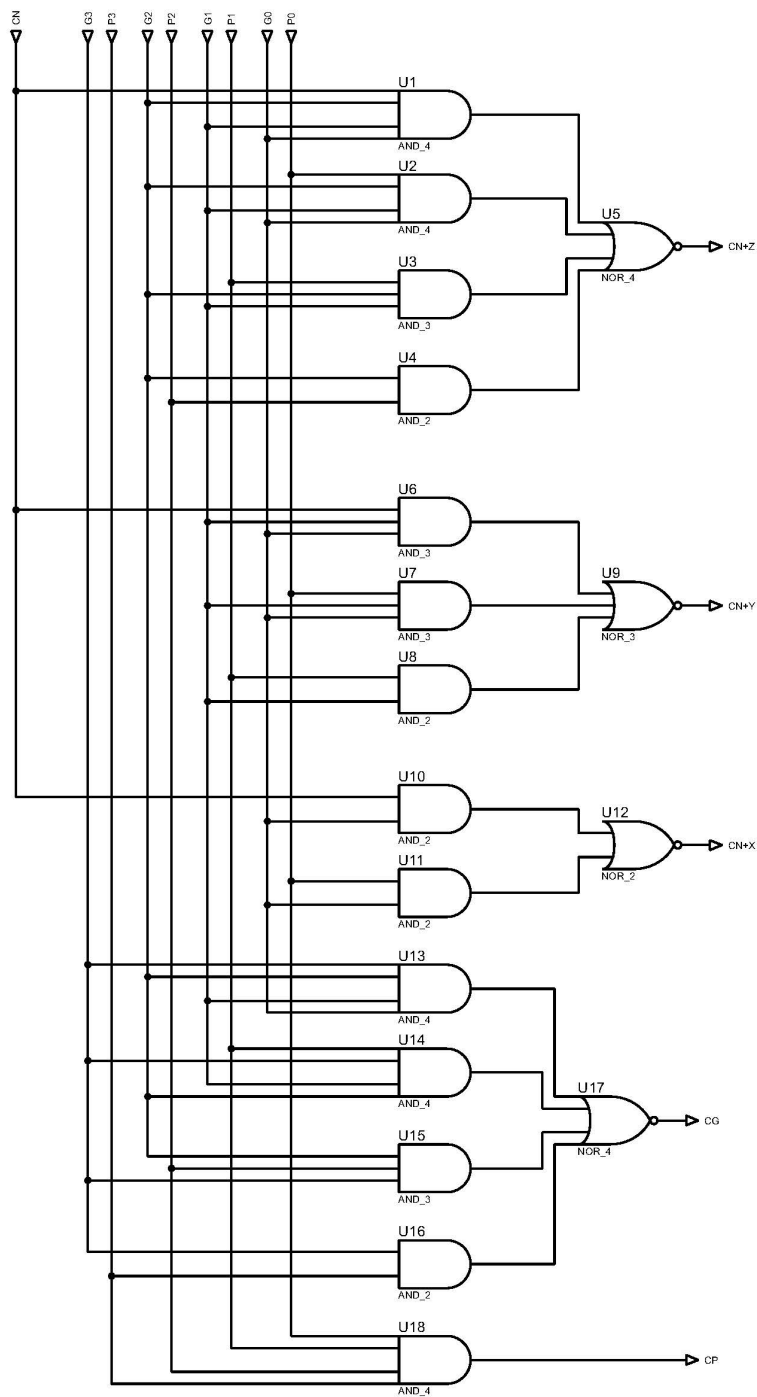
بخش عملی ::

قطعه 74182: ما می‌خواهیم از 74182 Carry Lookahead Generator استفاده کنیم. اما آی‌سی مورد

استفاده ما هیچ مدلی در پروتئوس ندارد. لذا خودمان شماتیک داخلی آی‌سی را طراحی می‌کنیم، به مدل 74182 اضافه می‌کنیم و شبیه‌سازی را انجام می‌دهیم. منطق مدار داخل آن (مطابق فرمول‌های قسمت قبل) در شکل زیر مشاهده می‌شود.

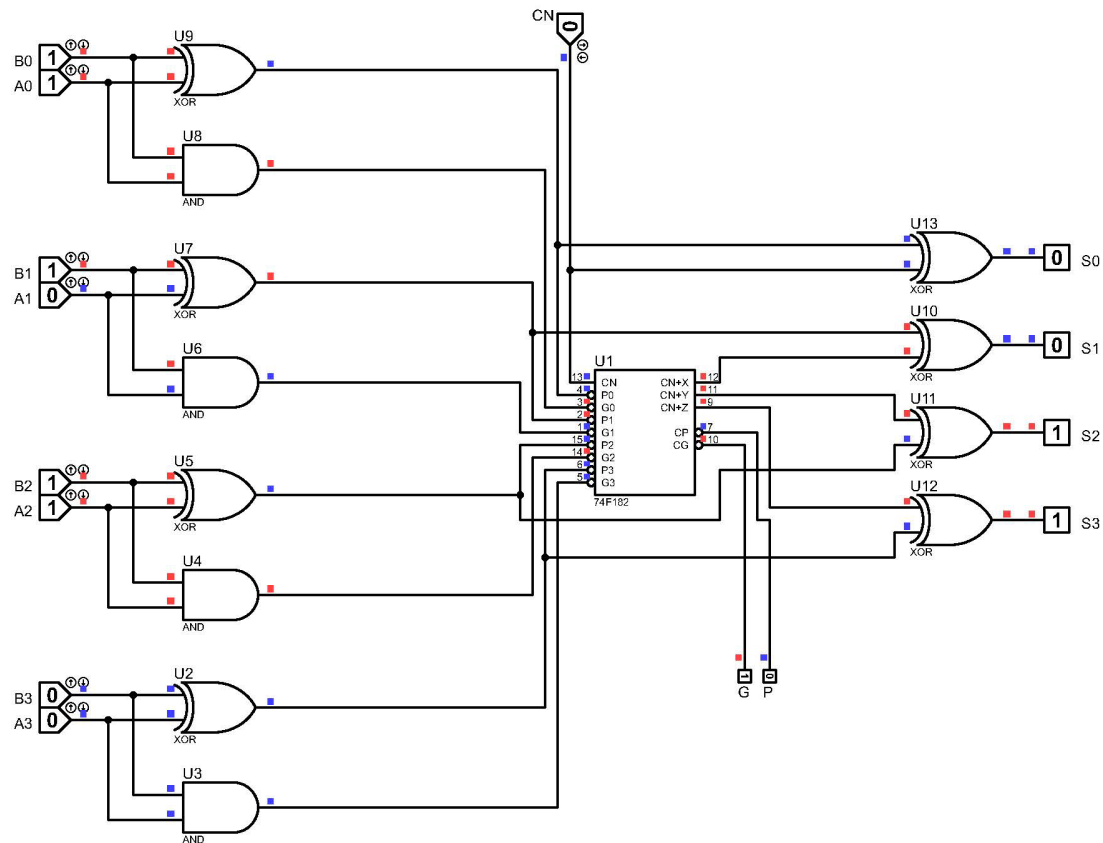
این مدار در پروتئوس به عنوان child sheet از صفر طراحی شده است و در نهایت با استفاده از گزینه Export

Graphics خروجی تصویری گرفته شده‌است.



۱۰- مدار داخلی تراشه ۷۴۱۸۲

اکنون به مدار اصلی می‌پردازیم. پس از تکمیل مدار آن را اجرا (ران) می‌کنیم تا ببینیم با توجه به LogicState هایی که دادیم، خروجی‌های LogicProbe ما به چه صورتی هستند. (صحت عملکرد آن را بررسی کنیم)



۱۱- مدار جمع کننده/تفریق کننده چهار بیتی با تراشه ۷۴۱۸۲

Cn is the carry input,

Cn+4 is the carry output.

/P=0 indicates, that the 74182 will propagate a carry through,

/G=0 indicates, that the 74182 will generate a carry.

جملات انگلیسی فوق از این [لینک](#) آورده شده‌اند.

جمع بندی

در آزمایش اول درس آزمایشگاه مدارهای منطقی به آشنایی با محیط‌های شبیه‌سازی و انجام چند آزمایش ساده در نرم‌افزارهای Fritzing و Logisim و Proteus پرداختیم.

در Fritzing نحوه کار بردبرد را بررسی کردیم و در ادامه مدار LED و مدار معکوس‌کننده را با استفاده از تراشه SN74LS14N پیاده‌سازی کردیم. به دلیل عدم توانایی شبیه‌سازی کامل، خودمان همه اتصالات را بررسی کردیم. این نرم‌افزار به ما اجازه کار با بوردها و تراشه‌های متنوع را می‌دهد و همچنین به دلیل داشتن تصویر نزدیک به واقعیت برای هر المان، محیط جذابی دارد.

در Logisim به پیاده‌سازی دو مدار ترکیبی جمع‌کننده تمام‌افزا و جمع‌کننده/تفریق‌کننده ۴ بیتی پرداختیم. این برنامه دارای قابلیت شبیه‌سازی بی‌درنگ را دارد که باعث جذابیت بیشتر آن می‌شود. در این نرم‌افزار ما می‌توانیم انواع گیت‌ها، حافظه‌ها و فلیپ‌فلاپ‌ها را داشته باشیم ولی نهایتاً در سطح گیت هستند و امکان کار با تراشه‌ها وجود ندارد.

در Proteus به ساخت یک جمع‌کننده با پیش‌بینی کری نقلی پرداختیم. ما از آی‌سی 74182 استفاده کردیم، اما شماتیک داخلی آن را خودمان طراحی کردیم. هرچند می‌توانستیم بدون آن هم مدار را طراحی کنیم، طراحی با این آی‌سی تجربه متفاوت و جدیدتری را برای ما ایجاد کرد. این نرم‌افزار قابلیت‌های بسیار زیادی (از جمله گیت‌ها و تراشه‌های متعدد) را دارد و می‌توانیم با اجرا کردن مدار خروجی آن را ببینیم.

منابع

Digital Design, M. Morris Mano

منابعی که از تصاویر موجود در آن‌ها استفاده شده‌است:

https://commons.wikimedia.org/wiki/File:7404_Hex_Inverters.PNG
<https://circuitglobe.com/half-adder-and-full-adder-circuit.html>