



گزارش کار اول آزمایشگاه مدارهای منطقی

آشنایی با محیط‌های شبیه‌سازی

دکتر شاهین حسابی

آبان ۱۴۰۰

نویسنده: علیرضا حبیب‌زاده
شماره دانشجویی: ۹۹۱۰۹۳۹۳

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

مقدمه

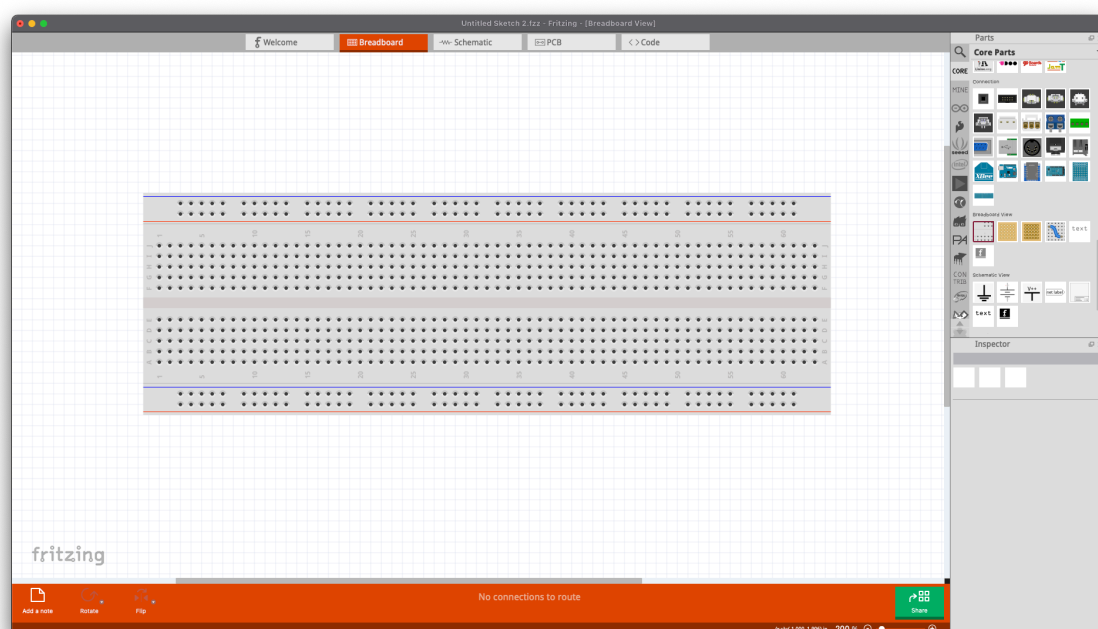
هدف از این آزمایش آشنایی با محیط‌های نرم‌افزاری شبیه‌سازی مدارهای منطقی است. این آزمایش در سه بخش انجام میشود. در بخش اول، به کمک نرم افزار Fritzing با طرز کار و نوع اتصالات یک بردبرد (Breadboard) آشنا خواهیم شد. در بخش دوم آزمایش، یک مدار ترکیبی ساده را با نرم‌افزار Logisim رسم و تست می‌کنیم و بالاخره در بخش سوم، مدار ترکیبی پیچیده‌تری را با نرم افزار Proteus خواهیم ساخت.

فهرست مطالب

i	مقدمه
۱	۱ رسم مدار با Fritzing
۱	۱.۱ اتصالات داخلی بردبرد
۲	۲.۱ مدار LED
۴	۳.۱ نات نات نات نات نات نات
۶	۲ ساخت مدار با Logisim
۶	۱.۲ جمع‌کننده کامل
۶	۱.۱.۲ شرح آزمایش
۷	۲.۲ جمع‌کننده/تفریق‌کننده ۴ بیتی
۹	۳ ساخت مدار با Proteus
۹	۱.۳ جمع‌کننده‌ی Carry-Look-Ahead
۱۲	۴ نتیجه و بحث

۱ | رسم مدار با Fritzing

هدف از این آزمایش بررسی اتصالات بردبورد و نحوه‌ی کار با آن است. این نرم‌افزار تنها برای طراحی مدار است و فقط می‌توان با آن وصل بودن اتصالات را بررسی کرد. بنابراین قابلیت شبیه‌سازی در آن وجود ندارد.

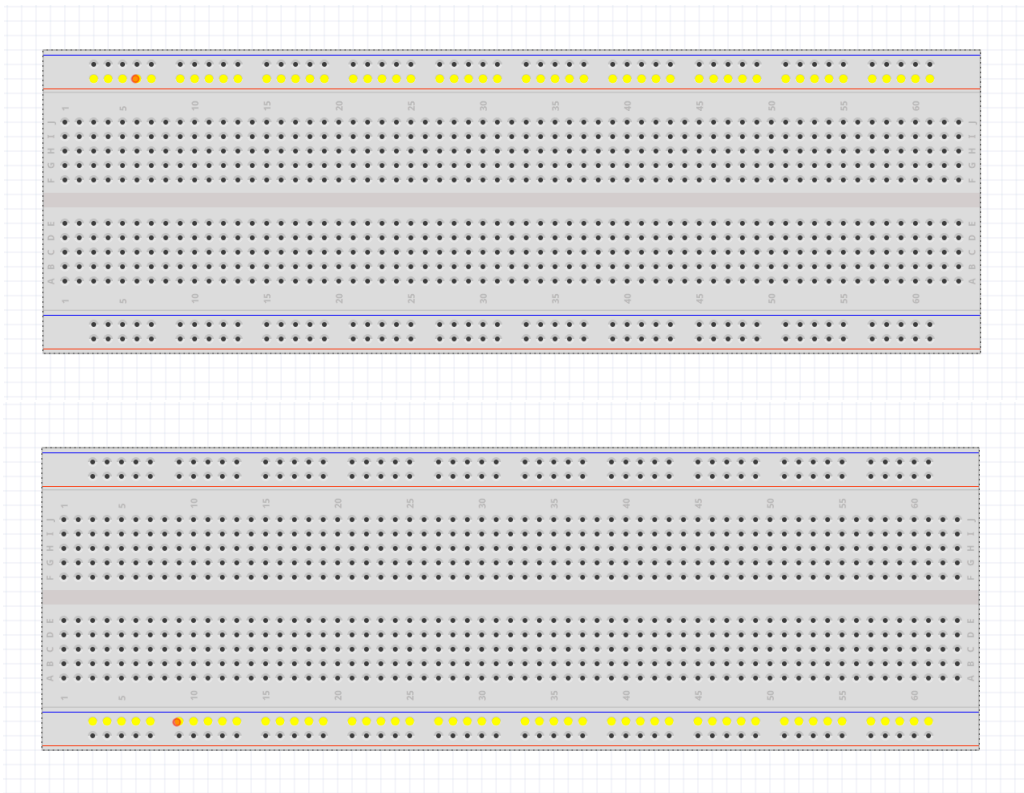


شکل ۱.۱: محیط کار نرم‌افزار Fritzing

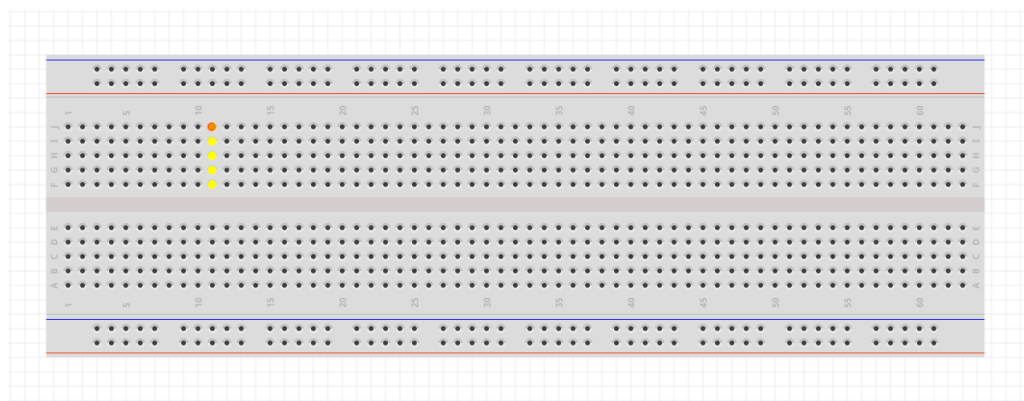
۱.۱ اتصالات داخلی بردبورد

در بالا و پایین برد بورد چهار نوار تغذیه وجود دارند که از هم مجزا هستند یعنی هیچ نواوری به نوار دیگر وصل نیست اما هر نوار در عرض کل برد بورد وصل است.

در میانه‌ی برد بورد ستون‌ها وجود دارند که هر ستون به اعضای خود وصل است. اما ستون‌های بالا و پایین شکاف میانی به هم وصل نیستند.



شکل ۲.۱: دو تا از از نوارهای تغذیه بردبرد



شکل ۳.۱: یکی از ستون‌های اصلی

۲.۱ مدار LED

وسایل مورد نیاز

۱. LED

۲. مقاومت ۲۲۰ اهم

۳. منبع تغذیه ۳ ولت

تئوری آزمایش

برای این آزمایش باید دقت کنیم پایه‌های LED دارا جهت هستند و می‌بایست به مثبت و منفی بودن پایه‌ها دقت کرد. برای تشخیص این امر به این نکته توجه می‌کنیم که داخل LED پایه‌ای که به تکه‌ی فلزی کوچک‌تر متصل است پایه‌ی مثبت است. در صورتی که داخل LED معلوم نبود به این نکته توجه داریم که معمولاً پایه‌ی مثبت LED کمی بلندتر از پایه‌ی منفی است.

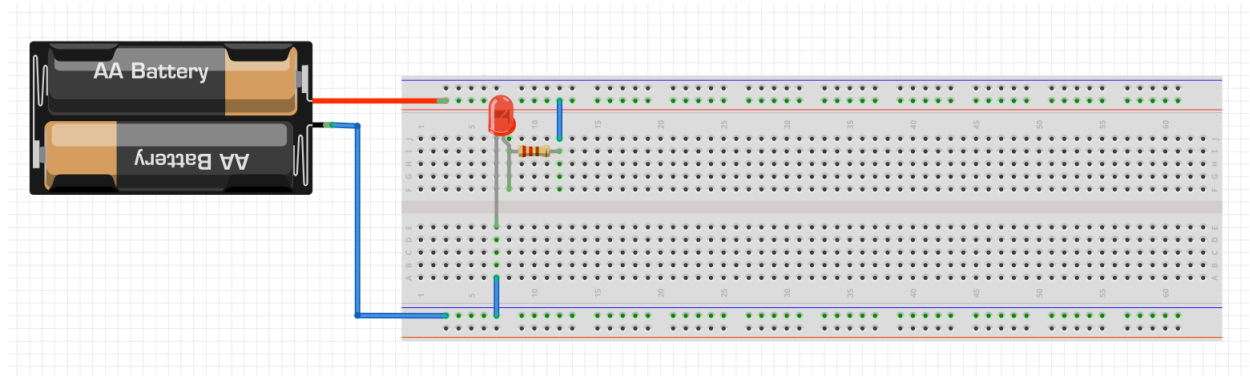


شکل ۴.۱: یک LED معمولی

شرح آزمایش

مطابق شکل مدار را به شکل صحیح بسته و به جهت پایه‌های LED توجه داریم. همچنین برای تمیزی مدار باتری را به خطوط تغذیه بردبورد وصل می‌کنیم. در ضمن اهمیتی ندارد مقاومت را در چه سمت LED قرار می‌دهیم زیرا در هر دو صورت معادله‌ی مدار یکسان است.

در انتها با قابلیت نرم‌افزار بررسی می‌کنیم تا اتصالات از ابتدای سر مثبت باتری تا جایی که جریان به سر منفی می‌رسد برقرار باشد.



شکل ۵.۱: مدار بخش ۲

۳.۱ نات نات نات نات نات

وسایل مورد نیاز

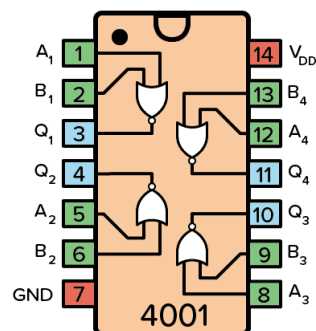
۱. LED

۲. ۶ گیت نات

۳. منبع تغذیه ۳ ولت

تئوری آزمایش

در نرم افزار به طور مستقیم قطعه ای با گیت نات وجود ندارد. اما می دانیم با گیت NOR می توان گیت نات را ساخت. برای این کار دو سر ورودی NOR را به هم وصل می کنیم. با استفاده از دو آی سی گیت NOR که هر کدام چهار گیت دارند، شش گیت نات مورد نیاز را خواهیم داشت.



شکل ۶.۱: مدار داخلی ۴۰۰۱

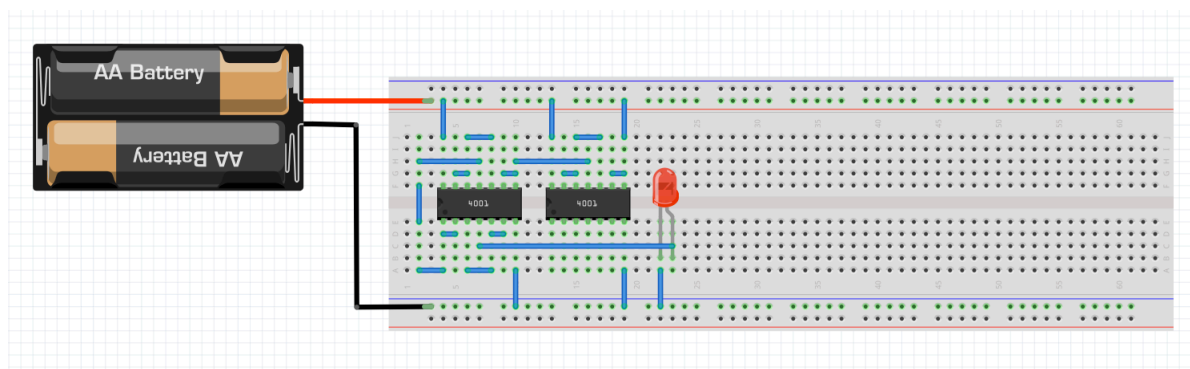
شرح آزمایش

مطابق شکل ابتدا آی سی را روی شکاف بردبرد قرار داده و سپس تغذیه ی آی سی ها را مطابق دیتاشیت شان به آن ها وصل می کنیم. معمولاً پایه ی بالا سمت چپ VCC و پایه ی پایین سمت راست منفی یا زمین است. دقت کنید که این بالا و پایین به شرطی است که نیم دایره ی فرورفته ی آی سی را در سمت چپ قرار دهیم و قطعه را نگاه کنیم.

حال که تغذیه ی آی سی ها وصل شد، آن ها آماده ی کار هستند و آن ها را مطابق توضیحات بخش تئوری وصل می کنیم.

نکته ی مهم دیگر این که معمولاً دیودهای نوری یا همان LED برای ولتاژ حداکثر حدود ۵.۱ ولت طراحی شده اند. که اینجا با اتصال باتری ۳ ولت بدون مقاومت ممکن است باعث سوختن LED شویم.

همچنین بعضی از آی سی ها برای کار کردن به تغذیه ی بیشتر از ۳ ولت و در حدود ۵ ولت احتیاج دارند که اینجا فرض می کنیم هم LED و هم آی سی با ولتاژ ۳ ولت به درستی کار می کنند.



شکل ۷.۱: مدار بخش ۳

۲ | ساخت مدار با Logisim

۱.۲ جمع‌کننده کامل

تئوری آزمایش

Adder Full یک مدار ترکیبی دیجیتال است که سه ورودی گرفته و دو خروجی می‌دهد. یکی از این سه ورودی ره به عنوان کری ورودی می‌شناسیم و دو تای دیگر متغیرهایی هستند که قرار است جمع شوند. خروجی این مدار نیز یکی بیت جمع و دیگر بیت کری خروجی است.

Inputs			Outputs	
A	B	C – IN	Sum	C – Out
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

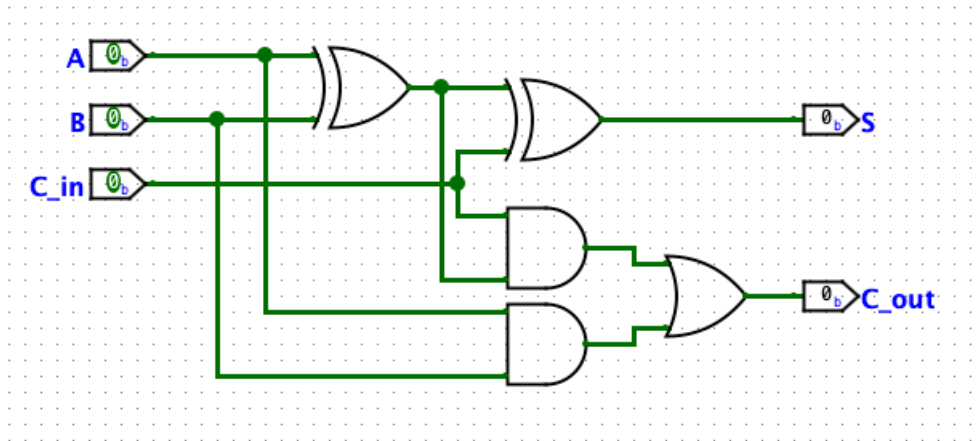
شکل ۱.۲: جدول صحت یک جمع‌کننده کامل

مطابق این جدول برای بیت‌های خروجی یک مدار بدست می‌آید که آن را در نرم‌افزار رسم کرده و شبیه‌سازی می‌کنیم.

۱.۱.۲ شرح آزمایش

ورودی‌ها و خروجی‌ها را به تعداد مناسب قرار داده و برای آن‌ها لیبل مناسب می‌زنیم.

حال گیت‌ها را قرار داده و اتصالات را برقرار می‌کنیم.



شکل ۲.۲: مدار جمع کننده کامل در نرم افزار Logisim

در نهایت با استفاده از ابزار «دست» که از منوی بالای صفحه قابل انتخاب است ورودی های مدار را تغییر می دهیم و تاثیر آن را در خروجی ها می بینیم.

۲.۲ جمع کننده/تفریق کننده ۴ بیتی

برای کم کردن B از A کافی است هر دو عدد را در مکمل دو ببریم و سپس با جمع کننده ی عادی آن ها را جمع بزنیم. حاصل نیز در مکمل دو خواهد بود.

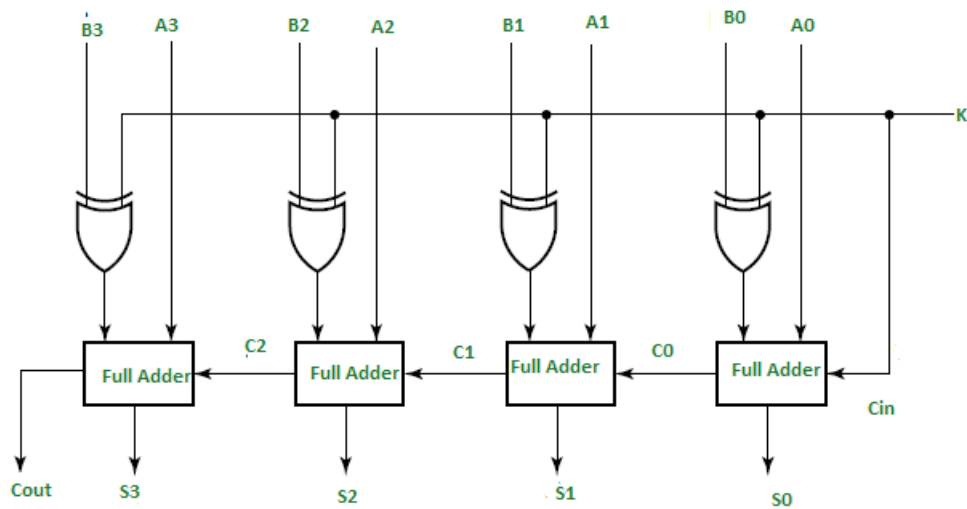
همچنین می دانیم برای پیدا کردن مکمل دوی یک عدد باید بیت های آن را معکوس کرده سپس عدد نهایی را با یک جمع کنیم. برای معکوس کردن بیت ها کافی است آن ها را با ورودی ای که اگر یک شود این اتفاق بیفتد XOR کنیم. که اگر این ورودی کنترلی یک باشد خروجی معکوس بیت ها و در غیر این صورت خود آن ها خواهد بود.

حالا این ورودی های معکوس شده ی مشروط را با هم جمع می کنیم. آن یکی هم که باید جمع می کردیم را در کری ورودی جمع کننده ی ابتدایی قرار می دهیم. نتیجه ی نهایی در شکل آمده است.

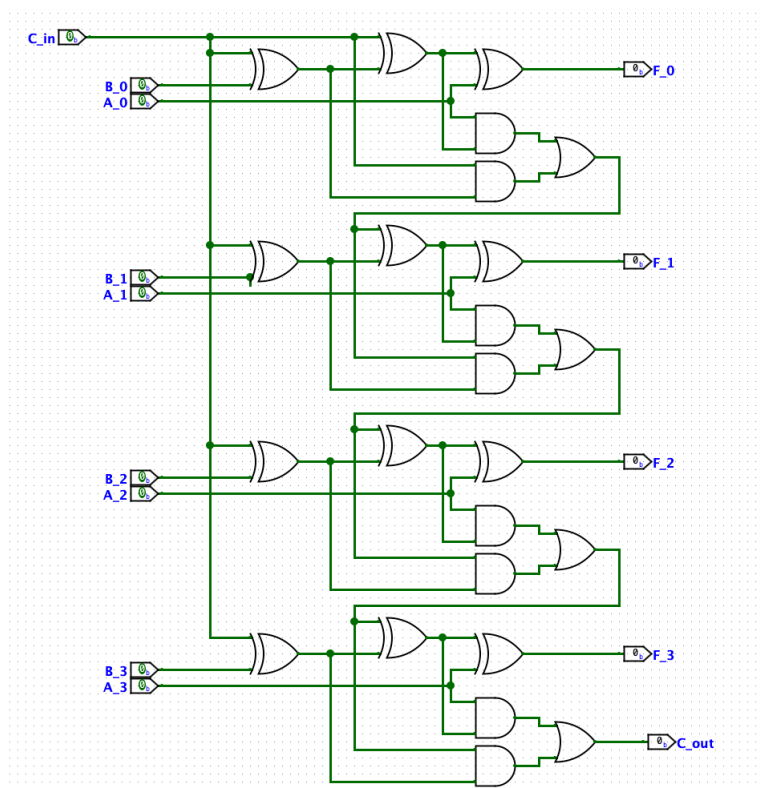
شرح آزمایش

مطابق شکل نشان داده شده و مطابق مدار جمع کننده ی کامل که در بخش قبل بستیم مدار نهایی را می بندیم.

در انتها می توانیم کمی با آن بازی کنیم و آن را برای حالات مختلف تست کنیم.



شکل ۳.۲: شماتیک مدار جمع‌کننده/تفریق‌کننده کنترلی



شکل ۴.۲: مدار نهایی

۳ | ساخت مدار با Proteus

۱.۳ جمع کننده ی Carry-Look-Ahead

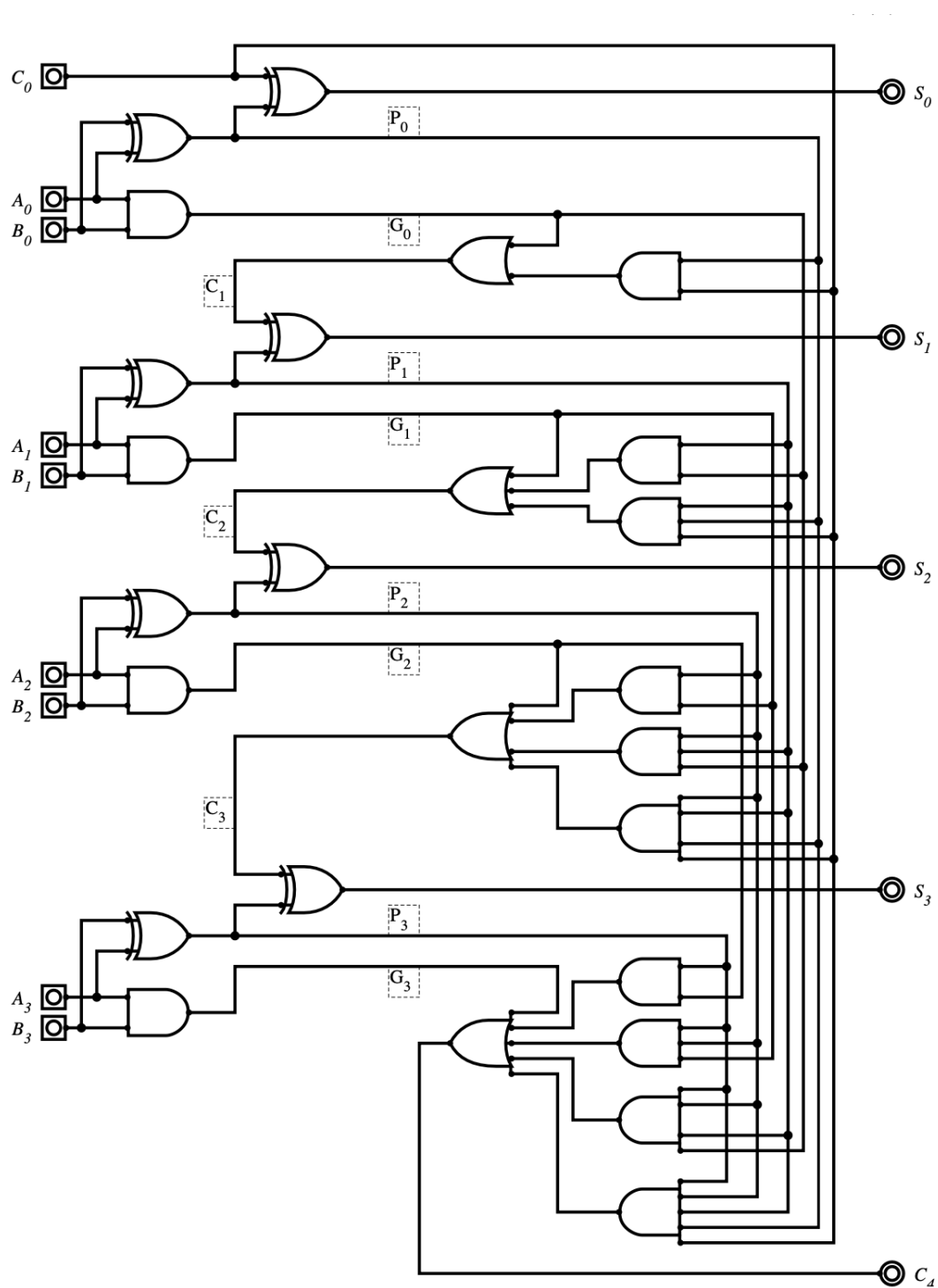
مقدمه و تئوری آزمایش

برای ساخت جمع کننده هایی با تعداد بیت بالا معمول است که جمع کننده های کوچک را پشت سر هم می بندند. این کار از جهاتی بسیار مفید است. چرا که به تمیزتر شدن مدار و قطعه قطعه یا ماژولار شدن آن کمک می کند. اما مشکلی که این روش دارد این است که هر جمع کننده برای رسیدن به جواب نهایی باید بیت کرای ورودی خود را بداند و این عدم آگاهی زنجیروار تا اولین جمع کنند می رسد به طوری که جمع کننده ی بیت آخر باید به تعداد تاخیر همه ی جمع کننده های قبلی خود صبر کند تا جواب قطعی درست بدهد.

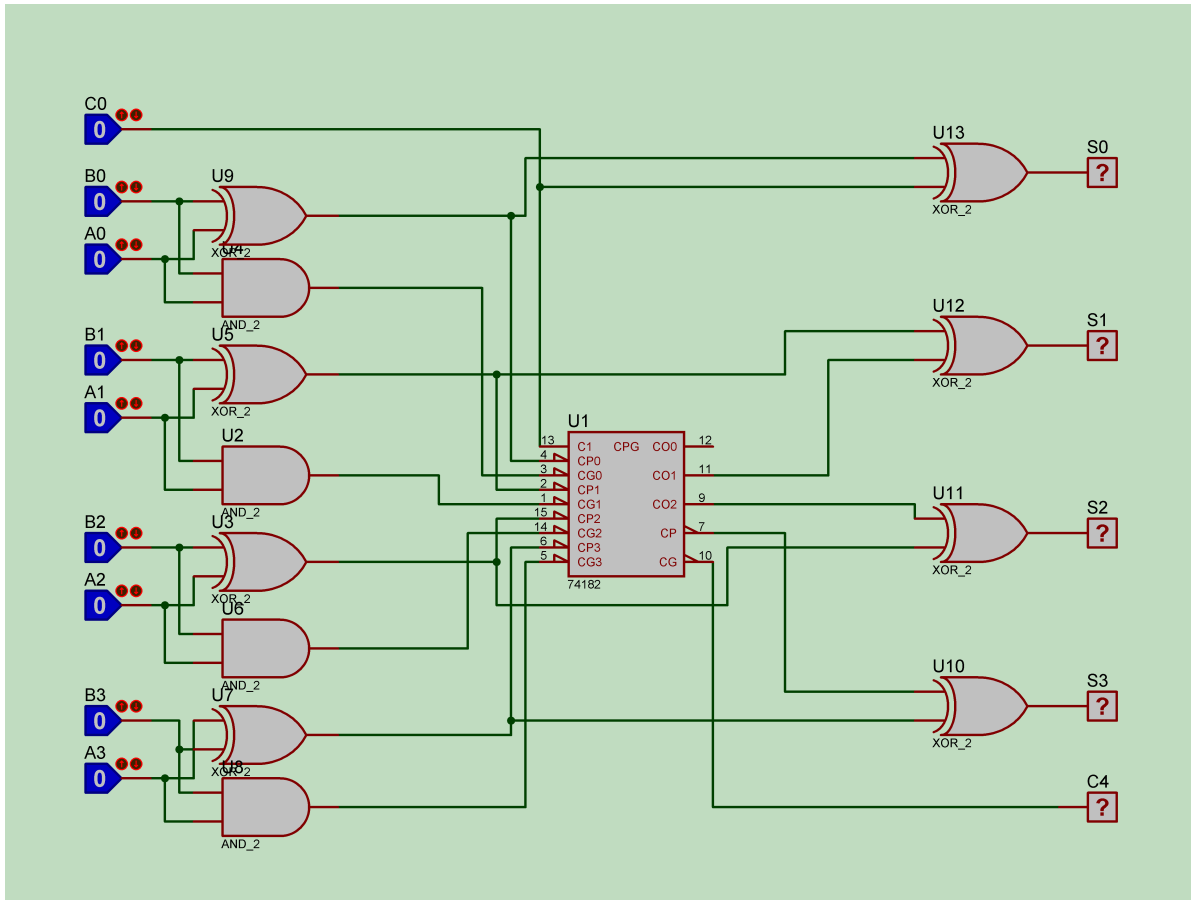
برای حل این مشکل از قطعه ای به نام Carry Look-ahead Generator استفاده می شود که به طور خلاصه این زمان تاخیر را کاهش می دهد و کرای ها را با سرعت بالاتری محاسبه می کند و در اختیار جمع کننده های دیگر می گذارد. و در واقع این وابستگی پشت هم قطعات و جمع شدن تاخیر آن ها را از بین می برد.

شرح آزمایش

مطابق شکل قطعه ی Carry Look-ahead Generator را از کتابخانه ی نرم افزار Proteus پیدا کرده و قرار می دهیم سپس مطابق تئوری گیت ها و ورودی ها و خروجی ها را وصل می کنیم و برای آن ها نام می گذاریم.



شکل ۱.۳: مدار داخلی یک جمع کننده ۴ بیت با Carry Look-ahead Generator باز



شکل ۲.۳: جمع کننده‌ی ۴ بیتی سریع

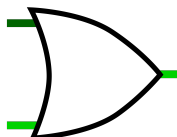
۴ | نتیجه و بحث

در اولین جلسه از آزمایشگاه مدارهای منطقی با کار با سه نرم افزار مهم آشنا شدیم.

The logo for Fritzing, featuring the word "fritzing" in a white, lowercase, sans-serif font, centered within a solid red rectangular background.

شکل ۱.۴: لوگوی نرم افزار Fritzing

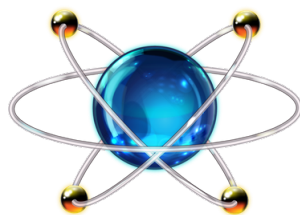
نرم افزار اول Fritzing بود که اوپن سورس است ولی قابلیت های چندانی از جمله شبیه سازی ندارد و تنها برای طراحی مدارات و قطعات است. البته از مزیت های این برنامه نزدیک به واقعیت بودن قطعات است و این امر برای طراحی بسیار مناسب است.



شکل ۲.۴: لوگوی نرم افزار Logisim

نرم افزار دومی که با آن کار کردیم Logisim بود که این هم اوپن سورس است و ظاهر بسیار ساده ای دارد اما این یکی برخلاف قبلی امکان شبیه سازی مدارها را به صورت در لحظه دارد به طوری که در حین اجرا نیز می توان ورودی ها را تغییر داد و نتیجه را درجا مشاهده کرد. اما محدودیت این نرم افزار در کم بودن تنوع قطعات و ساده بودنش است به طوری که تنها گیت ها و برخی قطعات محدود را می توان در آن استفاده کرد.

در انتها به نرم افزار معروف پولی فقط ویندوزی Proteus رسیدیم. این نرم افزار ظاهر شلوغ و اعصاب خردکنی دارد اما در عوض بسیار کامل بوده و امکانات بسیار زیادی دارد و همچنین تنوع قطعات آن بسیار فراوان است و گاه از زیاد بودن تنوع و مدل های یک قطعه اعصاب آزمایشگر را خرد می کند. اما در نهایت این تنوع و امکانات زیاد و امکان شبیه سازی بسیار خود آن باعث شده برای اکثر مدارها فراتر از نیاز آزمایشگر باشد.



شکل ۳.۴: لوگوی نرم افزار Proteus