به نام خداوند بخشنده مهربان

آزمایشگاه معماری کامپیوتر

دکتر سربازی

میترا قلی پورچناری - ۴۰۱۱۰۶۳۶۳
ملیکا علی زاده - ۴۰۱۱۰۶۲۵۵
نیکا قادری - ۴۰۱۱۰۶۳۲۸
بهار ۱۴۰۳

تاریخ گزارش: ۱۸ تیر ۱۴۰۳ | **طراحی** *carry select adder* | تمرین دوم

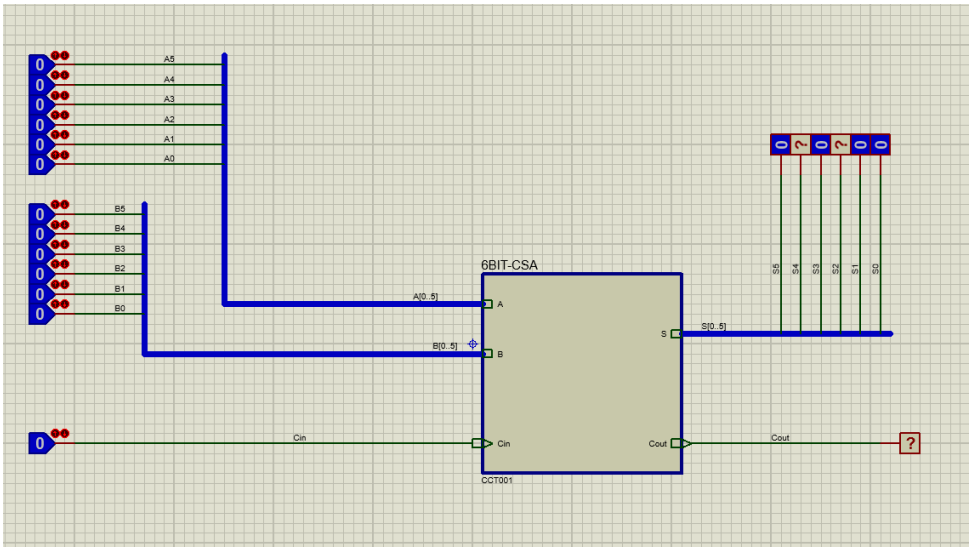
فهرست مطالب

۱ معرفی و بررسی هدف آزمایش

در این آزمایش دو عدد شش بیتی را با استفاده از جمع کننده های *csa* جمع می‌کنیم. مزیت این روش جمع نسبت به *carry ripple adder* این می‌باشد که لازم نیست منتظر بیت نقلی ورودی هر واحد شویم که از جمع کننده قبلی به دست می‌آید. کافی است یک بار فرض کنیم بیت نقلی یک است و محاسبات را انجام دهیم و یک بار هم فرض کنیم بیت نقلی صفر می‌باشد و جمع بزنیم. در آخر، با استفاده از یک مالتی پلکسر و با استفاده از بیت نقلی - که الان دیگر مقدارش را می‌دانیم - جواب درست را انتخاب می‌کنیم. تاخیر جمع کننده ها در این روش نسبت به روش انتشاری کمتر می‌باشد، چراکه فقط کافی است تاخیر یک *full adder* را به همراه تاخیر *multiplexer* ها در نظر بگیریم.

۲ پیاده سازی در پروتئوس

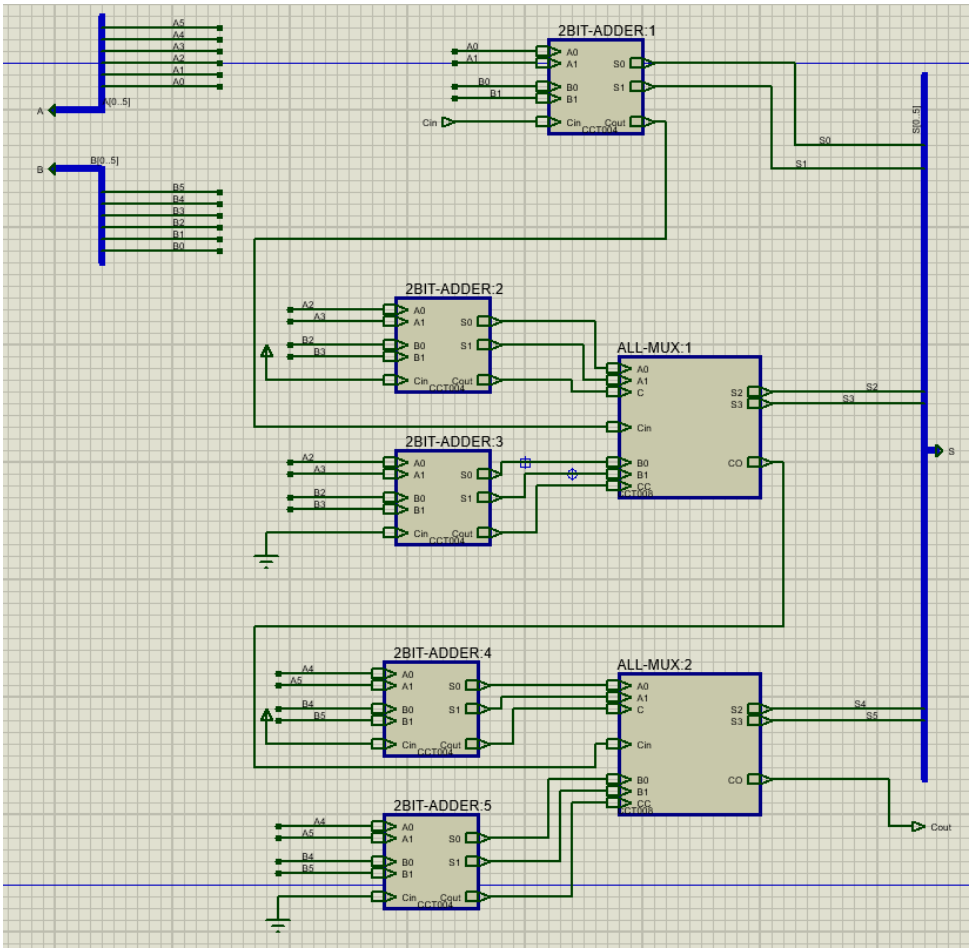
شمای کلی این برنامه به صورت زیر می‌باشد:



شکل ۱: *csa*

ورودی ها دو عدد شش بیتی A و B هستند و نتیجه جمع نیز در بیت های S و بیت کری $Cout$ قابل نمایش است. در ادامه ساختار درونی را به صورت سلسه مراتبی بررسی می‌کنیم.

۱.۲ csa

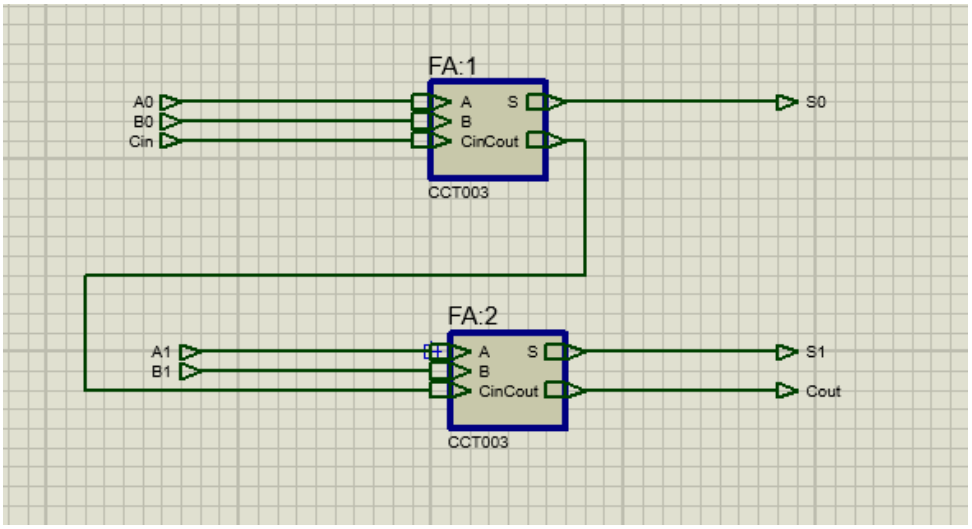


شکل ۲: ساختار داخلی csa

این بخش که در $child\,sheet$ قرار دارد، بدنه اصلی مدار را تشکیل می‌دهد. ابتدا وروی و خروجی ها با استفاده از bus گرفته می‌شوند یا خروجی داده می‌شوند. توجه شود که نام سیم های تکی که به گذرگاه متصل هستند باید با ورودی ها در ماژول $parent$ یکی باشد.
حال به شرح عملکرد مدار می‌پردازیم. از آنجایی که Cin به عنوان یکی از ورودی ها دارای مقدار مشخصی از همان ابتدا می‌باشد، لازم نیست از مالتی پلکسری استفاده کنیم و تنها باید از یک جمع کننده دو بیتی برای جمع $A[۰..۱]$، $B[۰..۱]$ و Cin استفاده کرد. بیت های S خروجی این جمع کننده $S۰$ و $S۱$ را در خروجی نهایی تشکیل می‌دهند و بنابراین به bus خروجی متصل هستند. خروجی نقلی این جمع کننده به عنوان Cin در مالتی پلکسر بعدی مورد استفاده قرار می‌گیرد.

حال برای جمع دو بیت بعدی، دوبار از جمع کننده استفاده می‌کنیم. یک بار با Cin برابر با صفر، و بار دیگر برابر با یک. سه خروجی هر یک از جمع کننده ها به یک مالتی پلکسر خاص به نام all_mux می‌روند. این مالتی پلکسر بیت سلکت خود را تحت عنوان Cin می‌گیرد که اگر صفر باشد، سه ورودی اول خود را خروجی می‌دهد و اگر یک باشد، سه ورودی دوم خود را انتخاب می‌کند. خروجی های S این مالتی پلکسر درواقع همان $S۲$ و $S۳$ هستند و بیت نقلی خروجی آن به همین صورت به مالتی پلکسر بعدی منتقل می‌شود.

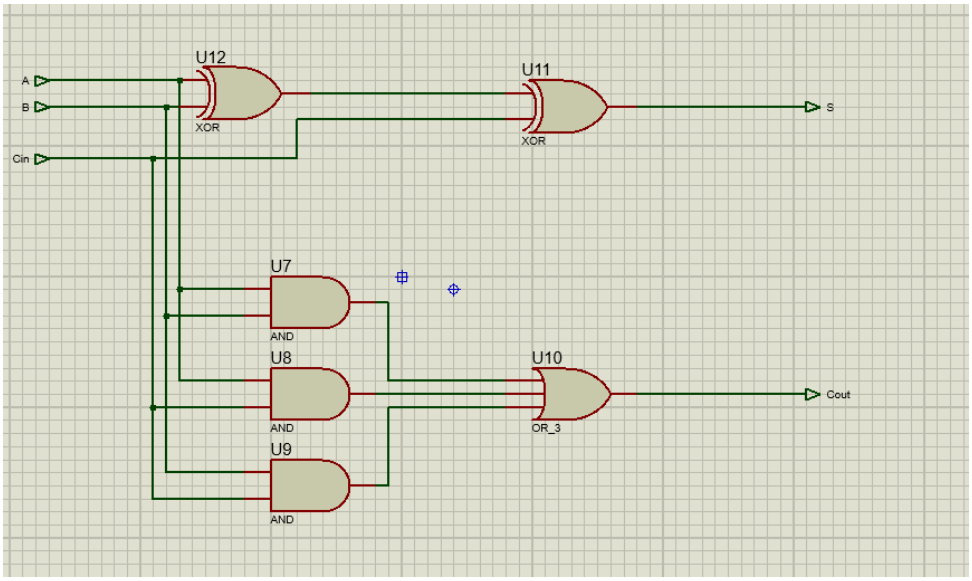
برای جمع دو بیت آخر نیز دقیقا مانند بند قبل عمل می‌کنیم با این تفاوت که خروجی $Cout$ مالتی پلکسر به عنوان بیت کری خروجی داده می‌شود و لازم نیست به ماژول دیگری وارد شود.
حال به توضیح ساختار دو ماژول پیاده سازی شده در این مدار می‌پردازیم: $۲bit\_adder$ و all_mux



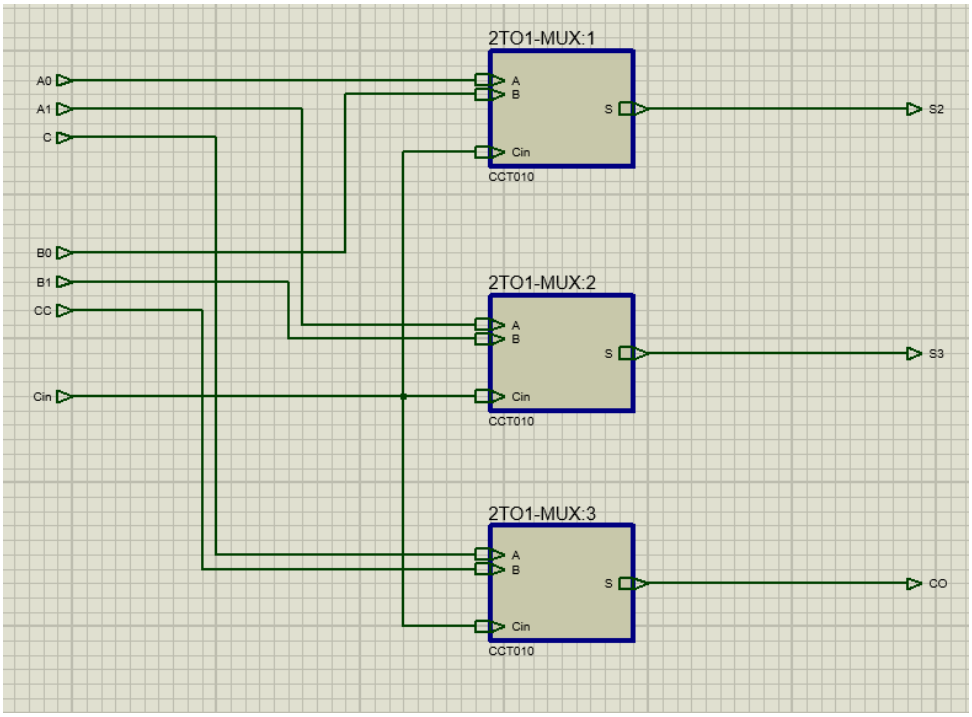
۱

شکل ۳: ۲ $bit\ adder$

این واحد یک جمع کننده انتشاری دو بیتی می‌باشد که با استفاده از دو واحد تمام جمع کننده ساخته شده است. ورودی نقلی جمع کننده دوم نیز از بیت خروجی نقلی جمع کننده اول تامین می‌شود.
ساختار یک تمام جمع کننده نیز به صورت زیر می‌باشد:

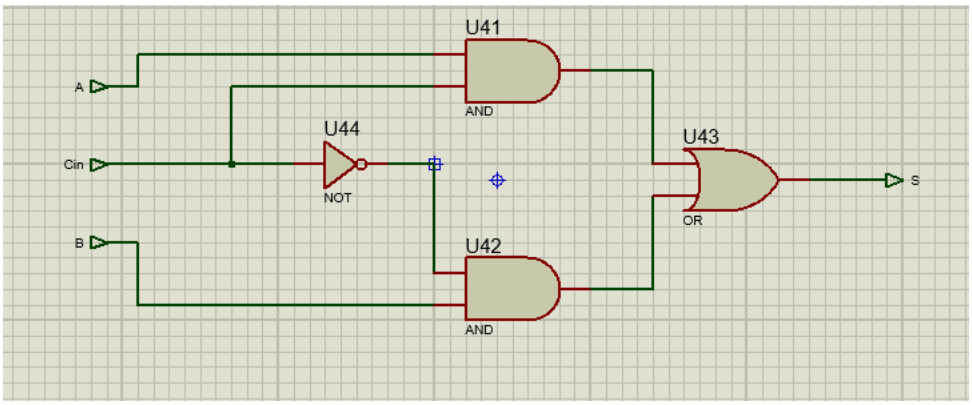


شکل ۴: fa



شکل ۵: $full\_mux$

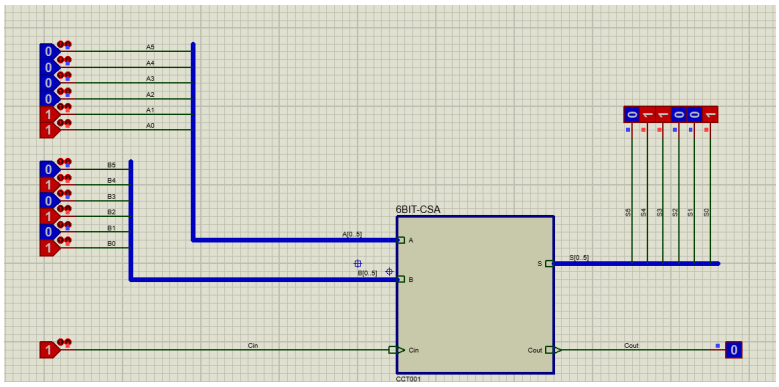
این قسمت از مدار شامل سه مالتی پلکسر دو به یک می‌باشد که از بین بیت های متناظر A و B خروجی خود را انتخاب می‌کنند. بیت انتخاب نیز همان ورودی Cin می‌باشد که بین هر سه مشترک است. ساختار یک واحد مالتی پلکسر دو به یک نیز همانند مدل های مدار منطقی به صورت زیر می‌باشد:



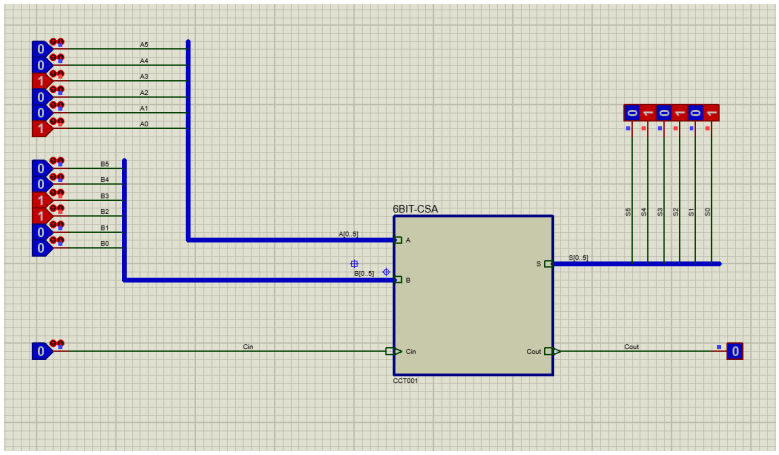
شکل ۶: $multiplexer$

۳ کامپایل و تست

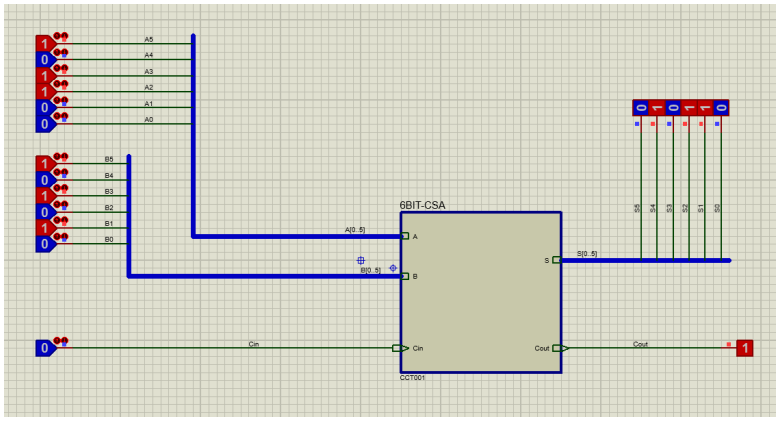
با استفاده از ورودی ها، می‌توان به ازای مقادیر مختلف، خروجی ها را آزمایش کرد. تعدادی از تست های انجام شده به صورت زیر می‌باشند:



شکل ۷: $۳(A) + ۲۱(B) + ۱(Cin) = ۲۴$



شکل ۸: $۹(A) + ۱۲(B) + ۰(Cin) = ۲۱$

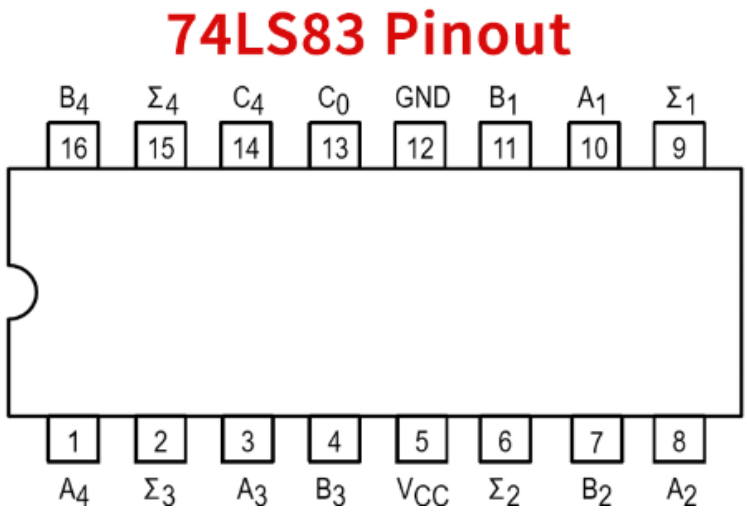


شکل ۹: $۴۴(A) + ۴۲(B) + ۰(Cin) = ۲۲ + ۶۴(Cout)$

۴ پیاده سازی فیزیکی

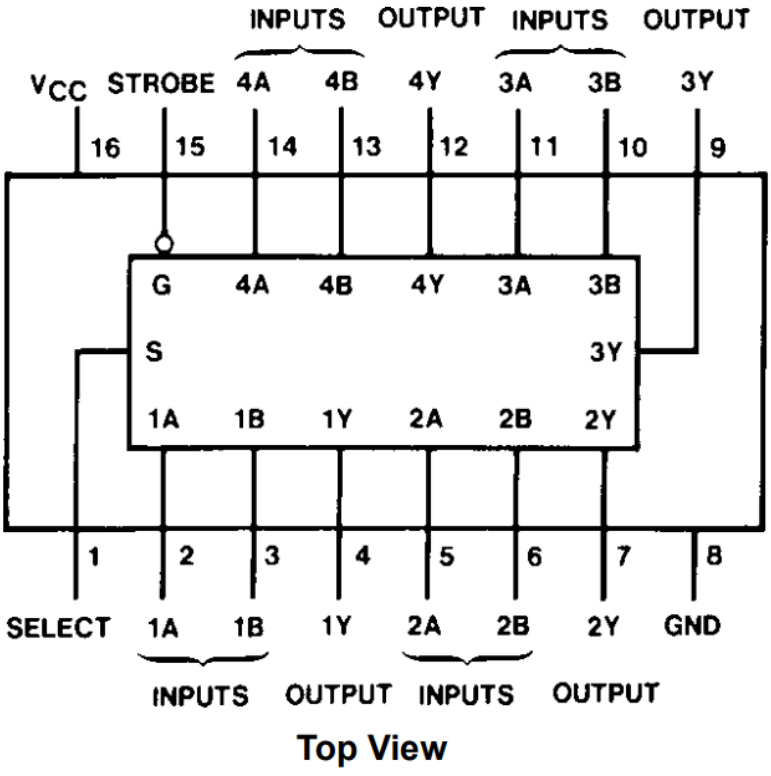
برای پیاده سازی این مدار، از دو آی سی $full\ adder(۷۴LS۸۳)$ و $multiplexer(۷۴LS۱۵۳)$ استفاده می‌کنیم و همانند مدل پروتئوس، مدار را می‌بندیم. البته چون جمع کننده دو بیتی پیدا نشد، از یک تمام جمع کننده چهاربیتی استفاده کردیم و دوبیت پرارزش هر دو ورودی آن را به سر منفی منبع تغذیه وصل کردیم تا در محاسبات مشکلی به وجود نیاید. البته توجه شود که در این حالت $Cout$ دیگر بیت نقلی را نمایش نمی‌دهد و برای گرفتن بیت نقلی از بیت سوم خروجی، یعنی $S۲$ باید استفاده کنیم.

برای بررسی پورت های این دو آیسی، از دیتاشیت آنها استفاده می‌کنیم. شماتیک این دو به صورت زیر می‌باشد:



شکل ۱۰: تمام جمع کننده

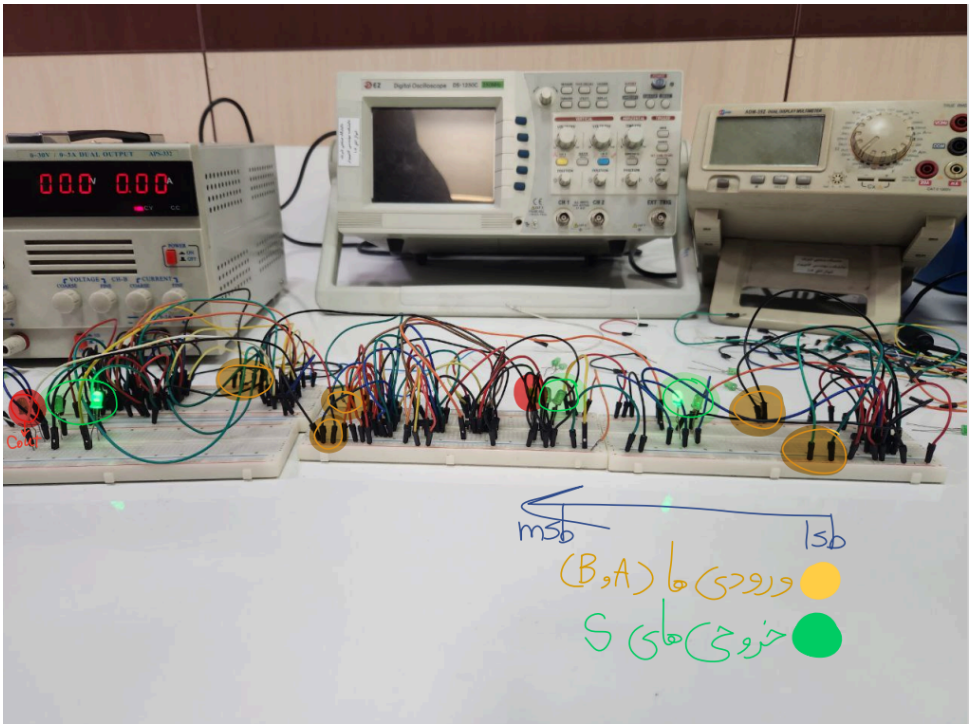
$A۰$ و $A۱$، را دو بیت ورودی عدد اول می‌گیریم و $B۰$ ، $B۱$ را ورودی دوم در نظر می‌گیریم. همچنین $A۲$و $A۳$، $B۲$ و $B۱$ باید صفر شوند. vcc و gnd را به ترتیب به سر مثبت و منفی به منبع تغذیه متصل می‌کنیم. خروجی ها $S۰$ و $S۱$ می‌باشند و بیت نقلی هم از پورت $S۲$ تامین می‌شود. Cin هم بسته به موقعیت می‌تواند صفر، یا یک باشد.



شکل ۱۱: مالتی پلکسر

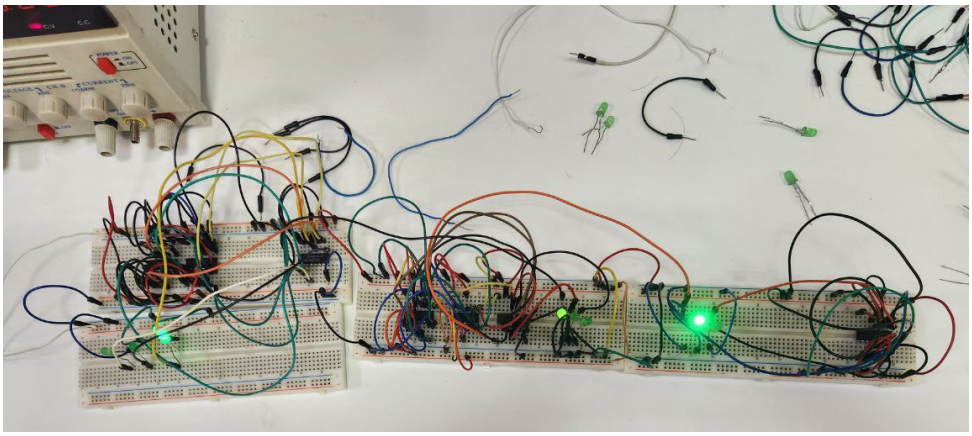
vcc و gnd به ترتیب به سر مثبت و منفی منبع تغذیه وصل هستند. ورودی سلکت همان $S۲$ در واحد قبلی می‌باشد. ورودی های مالتی پلکسر، از خروجی های تمام جمع کننده های متناظر تامین می‌شوند. چون هر جمع کننده دو بیتی می‌باشد، مالتی پلکسر فقط کافی است دو بیت خروجی دهد و نه چهار بیت. بنابراین به پورت های اضافی کاری نداریم.

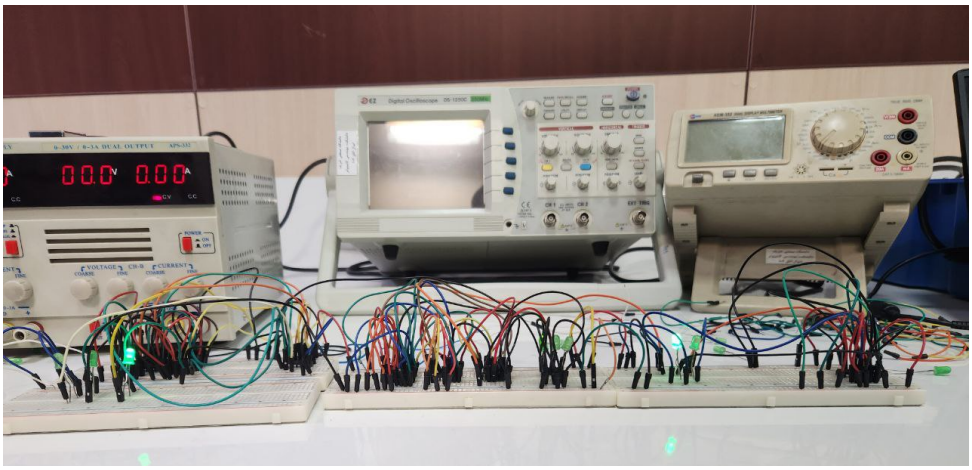
در نهایت، مدار به صورت زیر ساخته می‌شود:

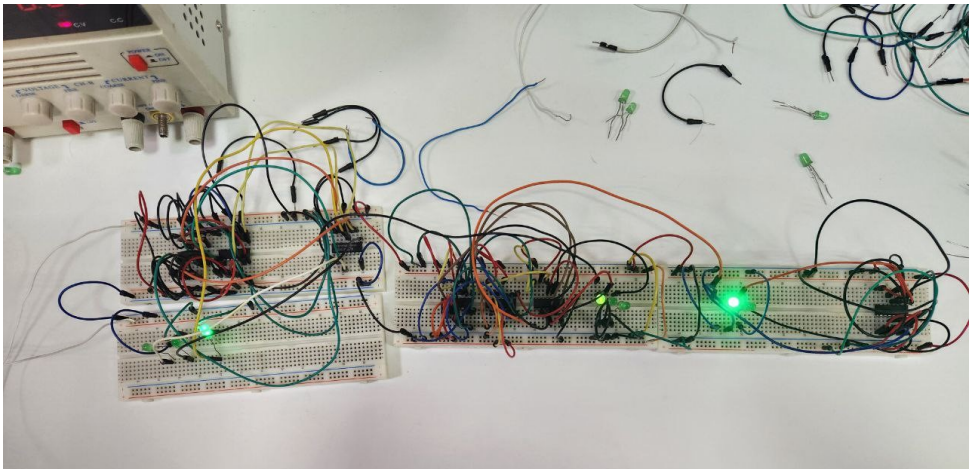


شکل ۱۲: مدار نهایی

در آخر نیز ورودی های مختلف را آزمایش می‌کنیم.







۵ نتیجه گیری

در این آزمایش با ساختار csa و هدف آن در سرعت بخشیدن به عملیات جمع آشنا شدیم. سپس یک واحد در ابعاد شش بیت را ابتدا در نرم افزار $proteus$ طراحی کرده و سپس، آن را به صورت فیزیکی در مدار پیاده سازی کردیم. لازم به ذکر است هنگام ساخت فیزیکی، در مواقعی، به دلایل مختلف از جمله نبود آیسی مورد نظر، ساختار مدار را اندکی متفاوت ساختیم به طوری که نتیجه نهایی تغییر نکند.

پایان