

به نام خدا



آزمایش ششم

آزمایشگاه معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نویسندگان:

رادین چراغی ۴۰۱۱۰۵۸۱۵

آرین نوری ۴۰۱۱۰۶۶۶۳

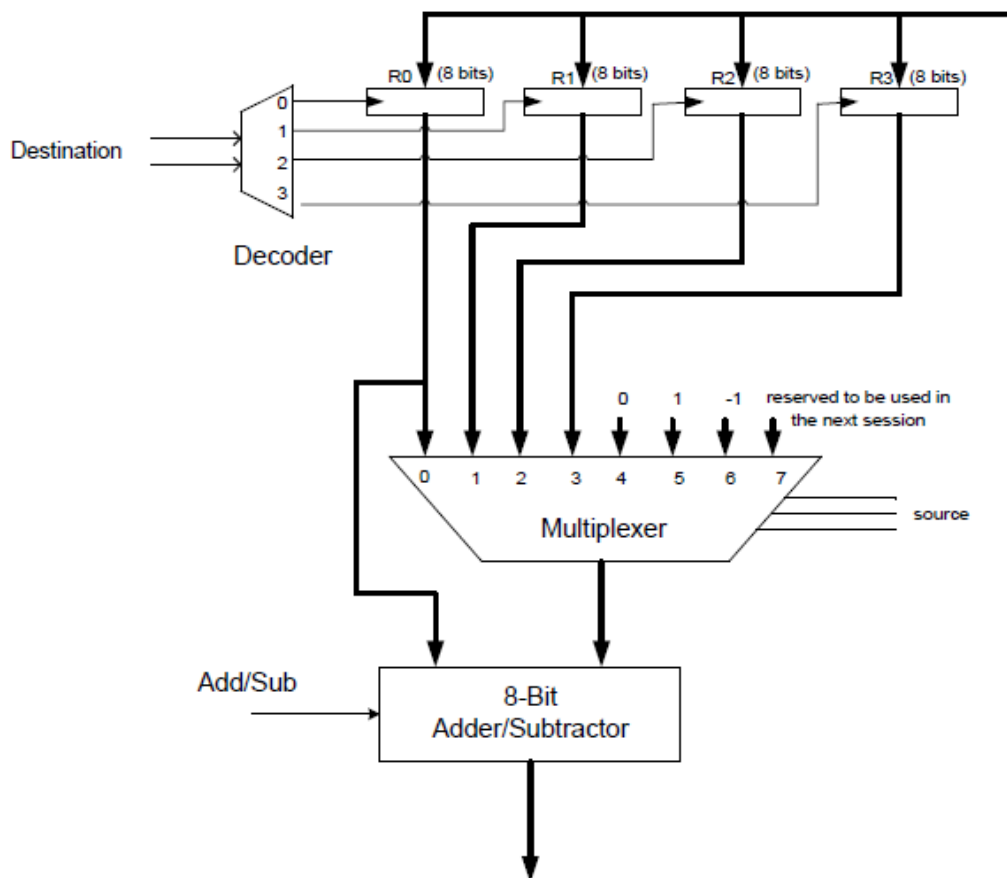
مبین پورعابدینی ۴۰۱۱۱۰۵۵۶

تاریخ ارائه تکلیف:

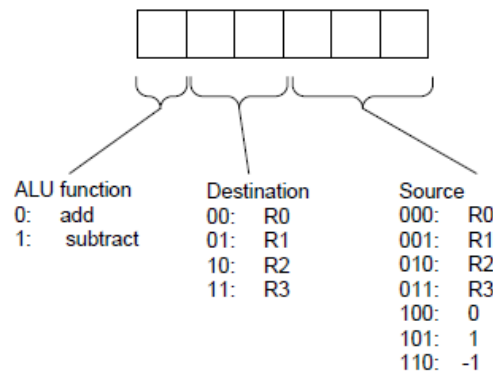
۱۴۰۳/۰۵/۱۹

مقدمه

هدف از انجام این آزمایش طراحی و پیاده‌سازی واحد محاسبات و مجموعه ثباتهای عمومی یک کامپیوتر ساده می‌باشد. این معماری در شکل ۱ نشان داده شده است. در این معماری، امکان انجام جمع و تفریق با انتخاب ثباتهای مبدا و ثبات نگهدارنده نتیجه (مقصد) فراهم می‌شود. همانطور که در شکل ۱ نیز مشخص شده است، در این کامپیوتر چهار ثبات عمومی $R0$ ، $R1$ ، $R2$ و $R3$ موجود می‌باشند که همگی هشت بیتی هستند. همچنین در این کامپیوتر یک واحد ALU وجود دارد. این واحد دو عملوند داشته که یکی از آنها همواره خروجی ۸ بیتی ثبات $R0$ بوده و دیگری محتوای یکی از ثباتهای $R0$ تا $R3$ یا مقادیر ثابت ۰، ۱ و -۱ می‌باشد که عملوند دوم از خروجی یک MUX ۸ به ۱ بدست خواهد آمد. دستورات این کامپیوتر نیز همانطور که در شکل ۲ مشخص است، ۶ بیتی می‌باشد. سه بیت اول آن ALU function بوده که جمع یا تفریق بودن عمل ALU را مشخص می‌کند. دو بیت بعدی $Destination$ یا مقصد خروجی ALU که یکی از ثباتها می‌باشد را مشخص کرده و در نهایت سه بیت آخر تعیین کننده عملوند دوم ALU می‌باشد.



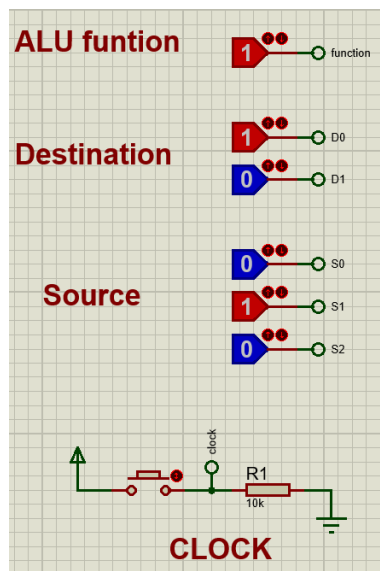
شکل ۱. معماری واحد محاسبات



شکل ۲. قالب دستورات شش بیتی

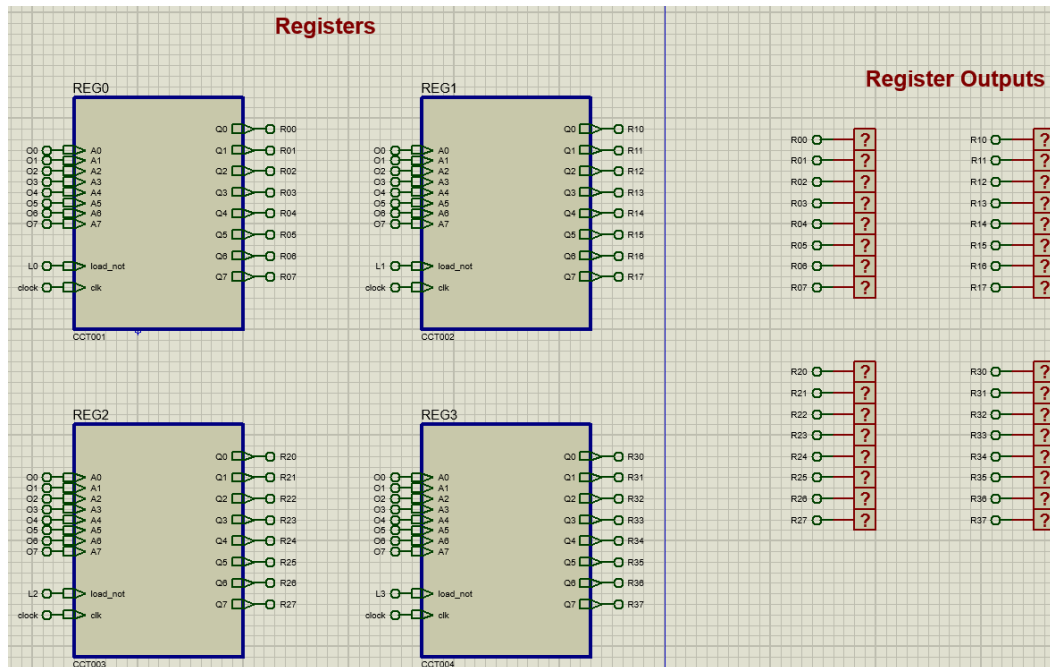
بخش اول: پیاده‌سازی در پروتئوس

در این بخش به سراغ پیاده‌سازی مدار در نرم‌افزار پروتئوس می‌رویم. ابتدا با استفاده از logic state و push button به ترتیب دستور ورودی ۶ بیتی و کلاک مدار را طراحی می‌کنیم. شکل ۳ بیانگر این عملیات می‌باشد.



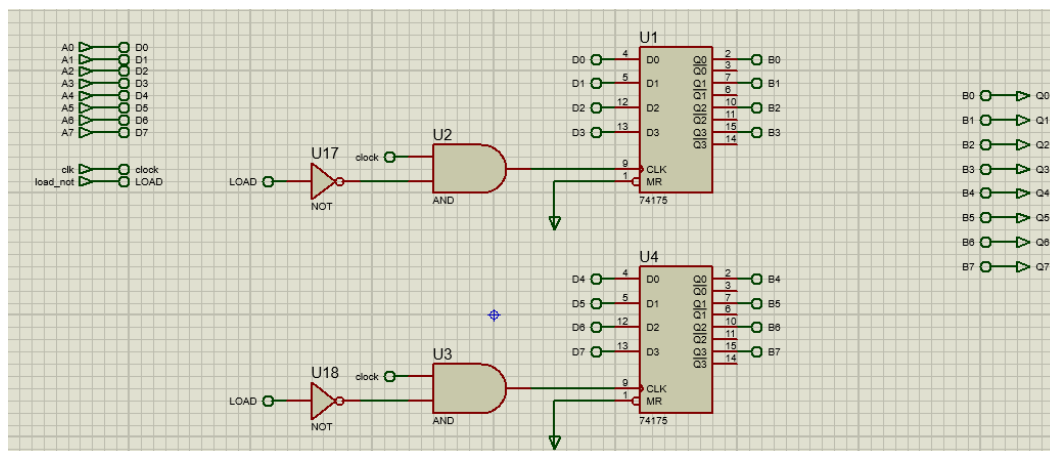
شکل ۳. ورودی‌ها و کلاک مدار

حال نوبت به طراحی رجیسترها می‌رسد. برای این کار ابتدا با استفاده از قابلیت default در قسمت subcircuit mode، یک ایسی رجیستر طراحی کرده و ورودی‌ها و خروجی‌های ۸ بیتی به علاوه یک ورودی فعال پایین (active low) load_not را مشخص می‌کنیم. سپس با استفاده از logic probe خروجی‌های این ثبات‌ها را نشان می‌دهیم. شکل ۴ نمای بیرونی ایسی‌های چهار رجیستر ۸ بیتی کامپیوتر و خروجی‌های آن‌ها را نمایش می‌دهد.

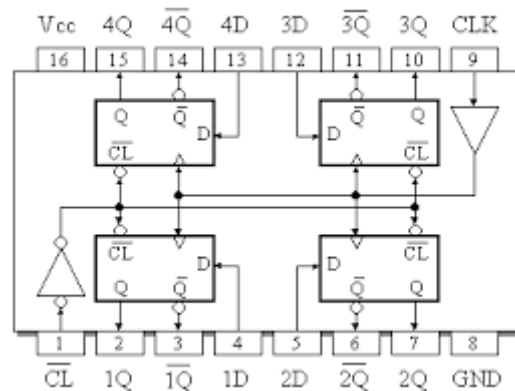


شکل ۴. نمای بیرونی ثبات‌ها و خروجی‌های آن‌ها

حال بخش داخلی هر ثبات را طراحی می‌کنیم. ابتدا ورودی‌ها و خروجی‌ها را در child sheet هر ثبات مشخص می‌کنیم. سپس دو ایسی 74175 که رجیسترهای ۴ بیتی می‌باشند را قرار داده و ورودی‌ها و خروجی‌ها را آن دو متصل می‌کنیم. از آن جایی که ثبات ۸ بیتی که طراحی کرده‌ایم یک ورودی load_not نیز دارد بایستی معکوس شده ورودی load_not را با ورودی کلاک AND کنیم و به کلاک ایسی‌های 74175 متصل کنیم. شکل ۵ مدار داخلی ثبات‌ها را نشان می‌دهد.

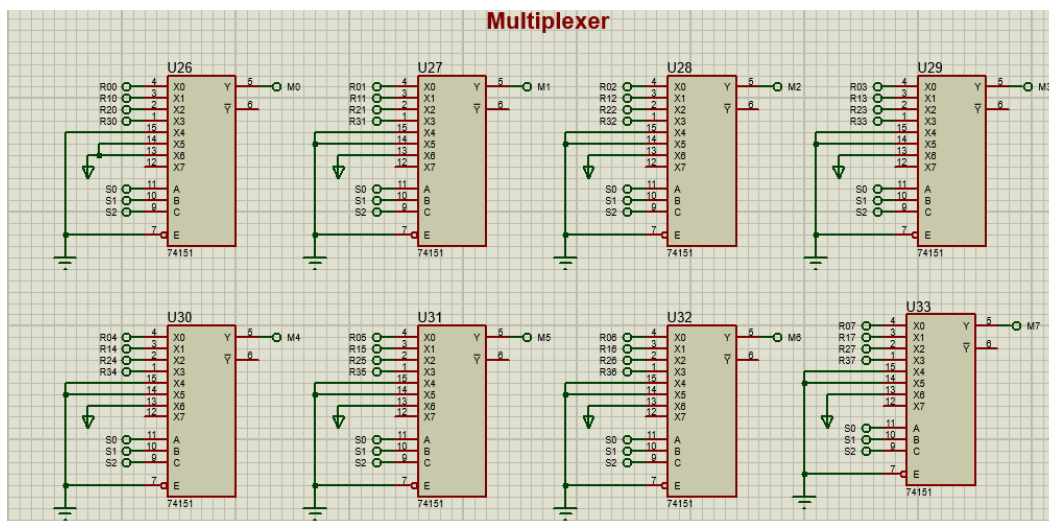


شکل ۵. مدار داخلی ثبات‌های ۸ بیتی

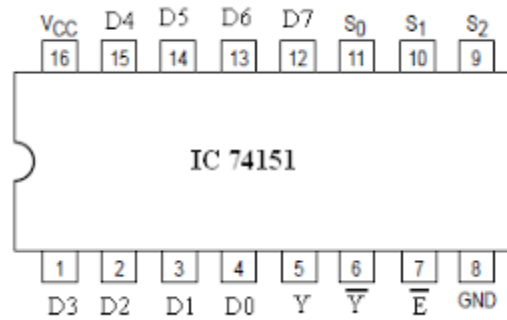


شکل ۶. ایسی ۷۴۱۷۵: ثابت ۴ بیتی

همانطور که گفته شد، در این کامپیوتر یک واحد ALU وجود دارد که این واحد دارای دو عملوند می‌باشد. یکی از آن‌ها همواره خروجی ۸ بیتی ثابت R0 بوده و دیگری محتوای یکی از ثابت‌های R0 تا R3 یا مقادیر ثابت 0، 1 و -1 می‌باشد که عملوند دوم از خروجی یک MUX ۸ به ۱ بدست خواهد آمد. برای طراحی MUX ۸ به ۱ از ایسی 74151 در پروتئوس استفاده می‌کنیم که یک MUX ۸ به ۱ می‌باشد. از آنجایی که محسبات کامپیوتر ۸ بیتی می‌باشد بایستی ۸ عدد از این ایسی‌ها را قرار دهیم. ورودی اول تا چهارم این ایسی‌ها بیت‌های ثابت‌های R0 تا R3 می‌باشد. ورودی پنجم تمامی این ایسی‌ها صفر بوده (عملوند دوم: صفر) و ورودی هفتم همگی آن‌ها یک (عملوند دوم: -1) است. ورودی ششم همگی این ایسی‌ها به جز ایسی اول صفر بوده و در ایسی اول یک می‌باشد (عملوند دوم: 1). ورودی‌های select این ایسی‌ها نیز همان بیت‌های Source در دستور ۶ بیتی ورودی می‌باشند. خروجی این MUX‌ها را M0 تا M7 در نظر می‌گیریم.

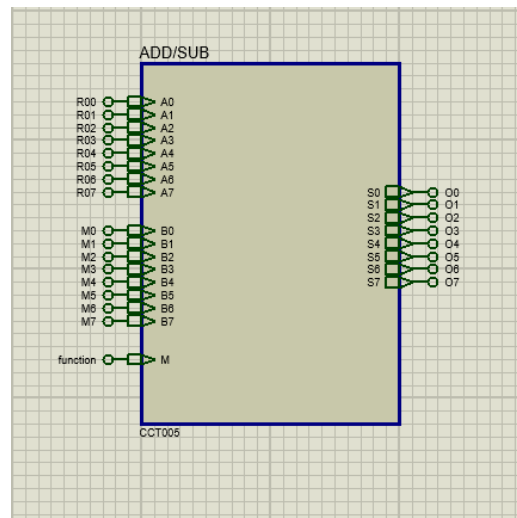


شکل ۷. MUX مدار، تشکیل شده از ۸ ایسی 74151



شکل ۸. ایسی ۷۴۱۵۱: MUX ۸ به ۱

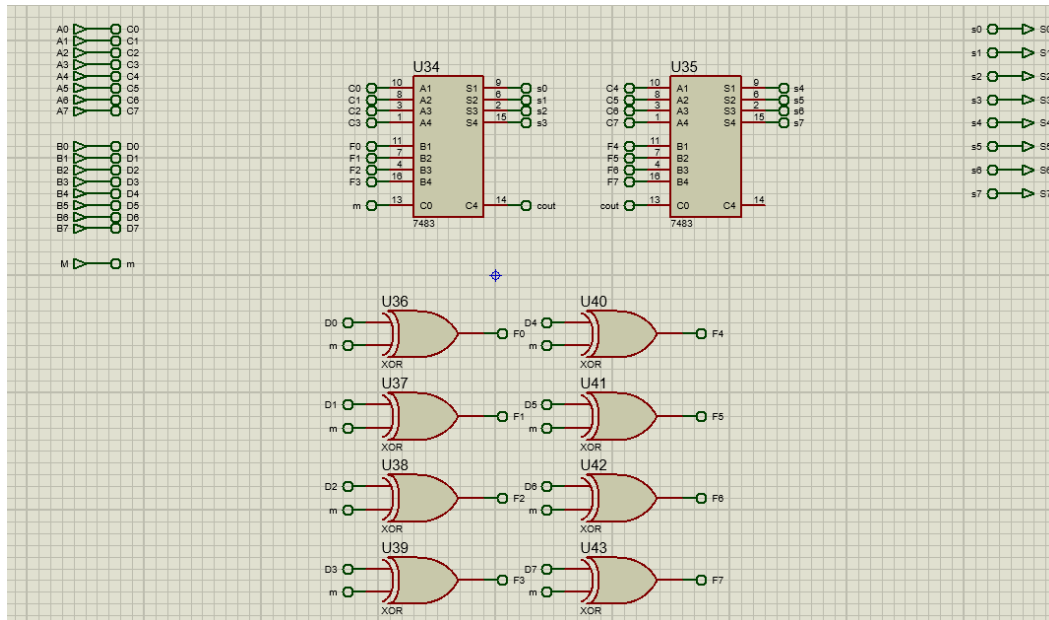
اکنون نیاز به یک واحد جمع/تفریق کننده داریم. برای این کار ابتدا با استفاده از قابلیت default در قسمت subcircuit mode، یک ایسی جمع/تفریق کننده طراحی کرده و ورودی‌ها و خروجی‌های ۸ بیتی به علاوه یک ورودی M که نوع عملیات آن را بیان می‌کند، را مشخص می‌کنیم. سپس خروجی‌های آن را 00 تا 07 در نظر می‌گیریم و به ورودی‌های ثبات‌ها متصل می‌کنیم. همچنین همانطور که گفته شد ورودی اول این واحد خروجی ثبات R0 و ورودی دوم آن خروجی MUX یا همان عدد ۸ بیتی M0 تا M7 می‌باشد. شکل ۴ نمای بیرونی این واحد را نشان می‌دهد.



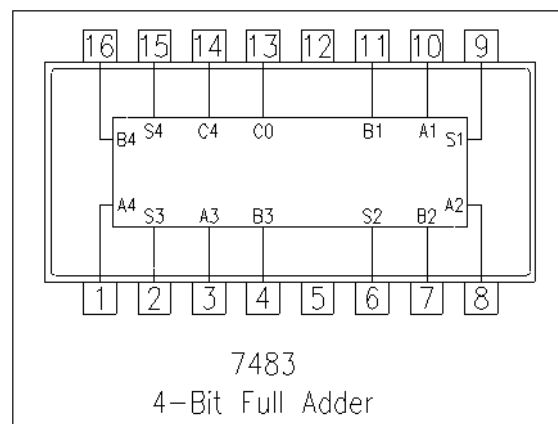
شکل ۹. نمای بیرونی واحد جمع/تفریق کننده

نوبت به طراحی مدار داخلی واحد جمع/تفریق کننده می‌رسد. پس از مشخص کردن ورودی‌ها و خروجی‌ها در child sheet این واحد، دو ایسی 7483 که جمع کننده ۴ بیتی می‌باشد را قرار می‌دهیم. ۴ بیت کم ارزش عدد اول ورودی را به ورودی‌های اول ایسی اول و ۴ بیت پر ارزش عدد اول ورودی را به ورودی‌های اول ایسی دوم متصل می‌کنیم. همچنین cout ایسی دوم را به cin ایسی اول وصل می‌کنیم. از آنجایی که مدار بایستی قابلیت تفریق کردن داشته باشد، ۸ گیت XOR قرار داده و بیت‌ها عدد دوم ورودی را با ورودی M که mode مدار است XOR می‌کنیم و به عنوان ورودی به ایسی‌های جمع کننده می‌دهیم. دلیل این کار

این می‌باشد که در هنگام تفریق کافی است عدد اول را با مکمل ۲ عدد دوم جمع کنیم. بنابراین با انجام عملیات XOR، در صورتی که در حالت تفریق باشیم، بیت‌های عدد دوم معکوس می‌شوند. حال کافی است ورودی M را به عنوان ورودی cin ایسی اول 7483 قرار دهیم تا عدد اول با مکمل ۲ عدد دوم جمع شود.

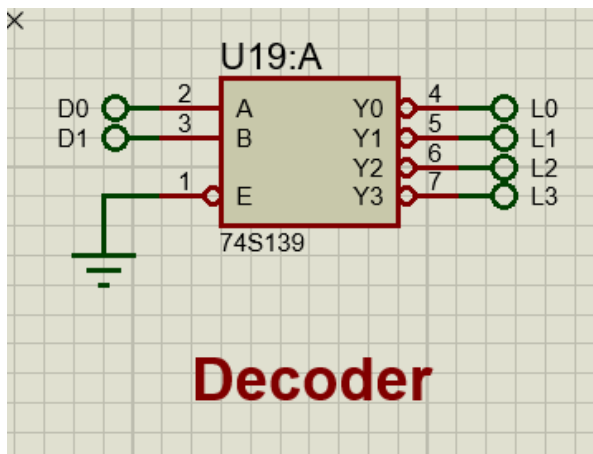


شکل ۱۰. مدار داخلی واحد جمع/تفریق کننده

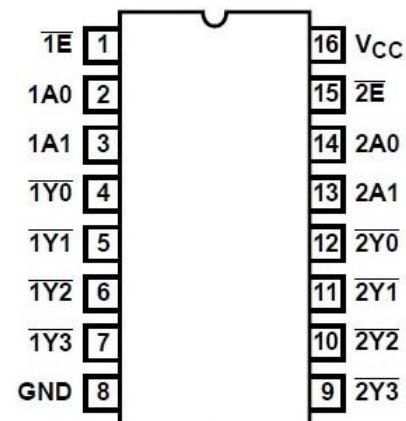


شکل ۱۱. ایسی ۷۴۸۳: جمع کننده ۴ بیتی

تنها قسمت باقی مانده از مدار طبق شکل ۱، ساز و کار طراحی ورودی‌های `load_not` ثبات‌ها می‌باشد. همانطور که در شکل نیز مشخص است نیاز به یک دیکودر دو به چهار داریم. بدین منظور از ایسی 74S139 در پروتئوس و در هنگام پیاده‌سازی روی برد از ایسی 74139 استفاده می‌کنیم. ورودی‌های این دیکودر همان Destination در دستور ۶ بیتی ورودی می‌باشد. همچنین خروجی‌های این دیکودر `active low` می‌باشند. بنابراین کافی است همین خروجی‌ها را به عنوان ورودی `load_not` به ثبات‌ها بدهیم.



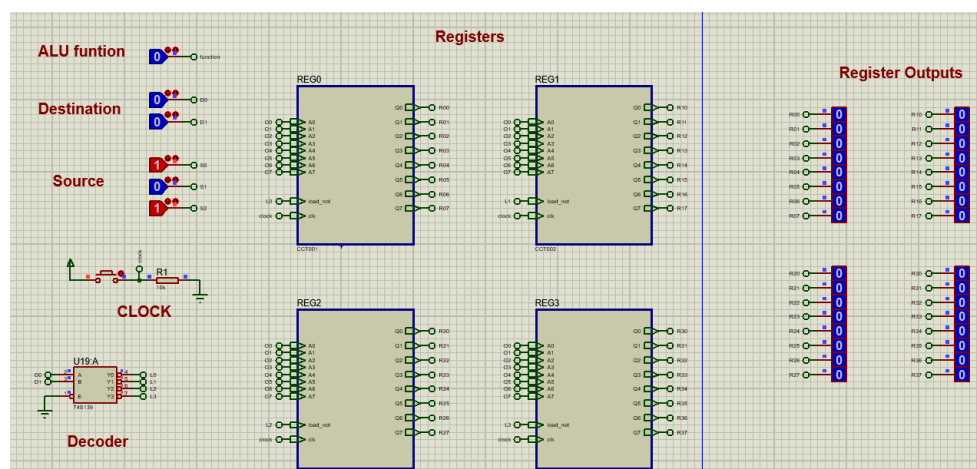
شکل ۱۲. مدار مشخص کننده ورودی `load_not` هر ثبات



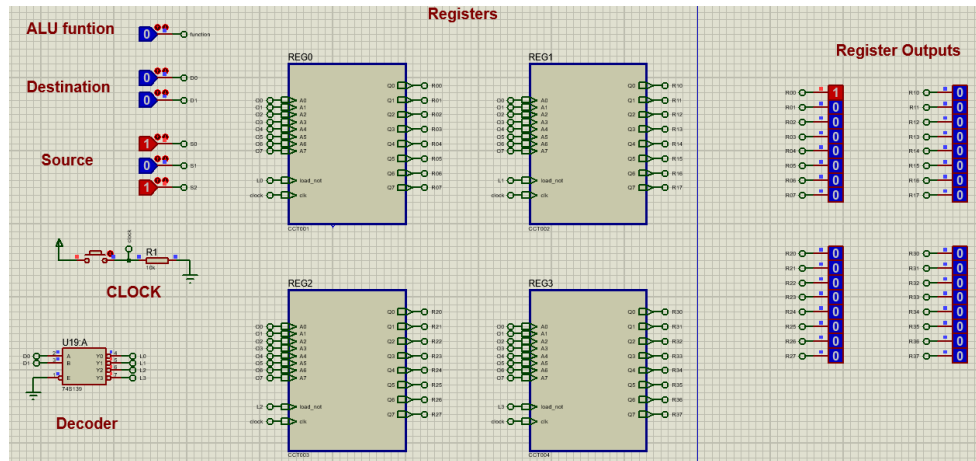
شکل ۱۳. ایسی ۷۴۱۳۹: دیکودر ۲ به ۴

آزمایش مدار در پروتئوس

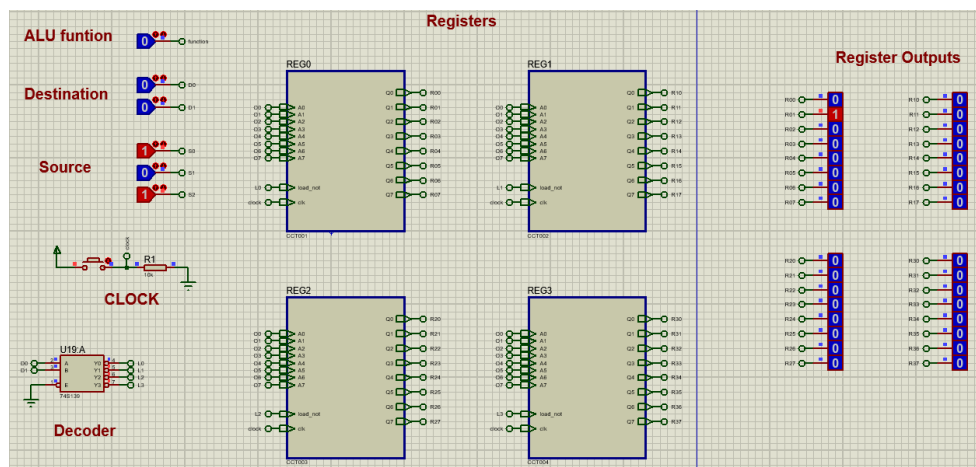
حال می‌خواهیم مدار را آزمایش کنیم. ابتدا عدد ۳ را در `RO` بارگذاری می‌کنیم. برای این کار بایستی سه بار `RO` با یک جمع شود و حاصل دوباره در `RO` قرار بگیرد. تصاویر زیر بیانگر این عملیات می‌باشند.



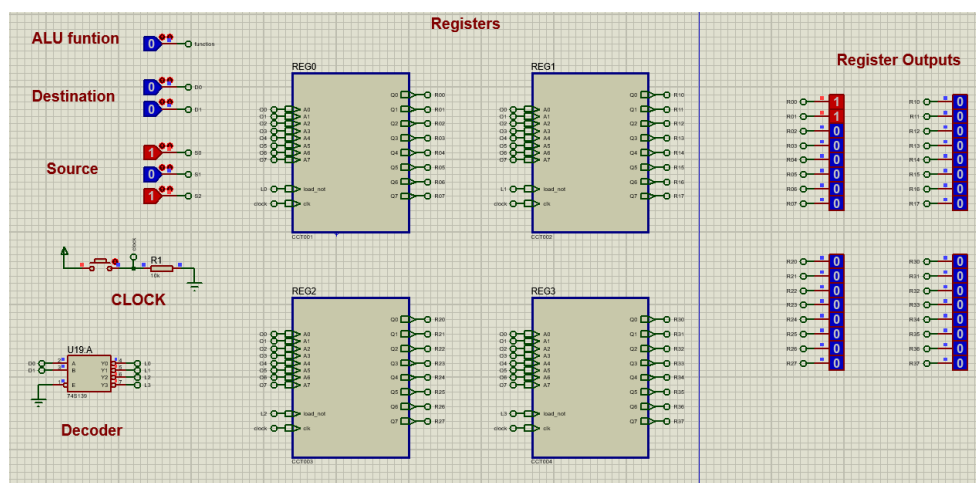
شکل ۱۴. آغاز انجام عملیات بارگذاری ۳ در `RO`



شکل ۱۵. جمع شدن عدد یک با R0 و قرارگیری حاصل در R0

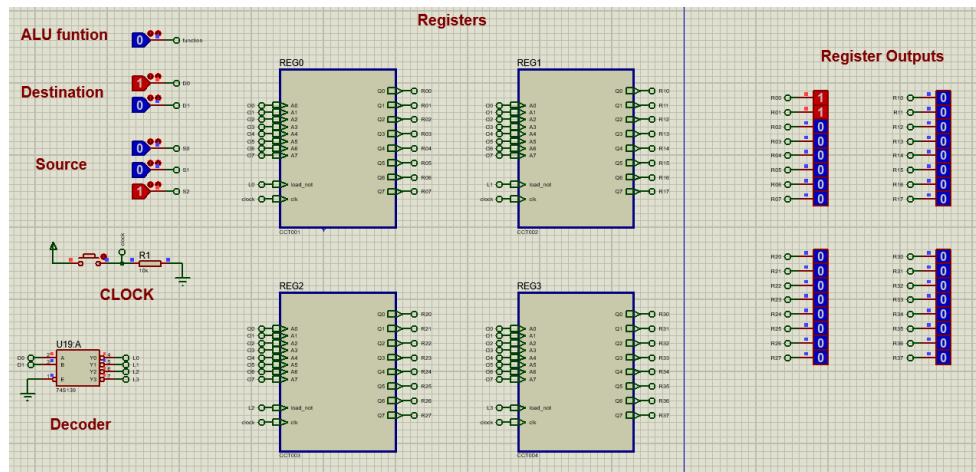


شکل ۱۶. جمع شدن عدد یک با R0 و قرارگیری حاصل در R0

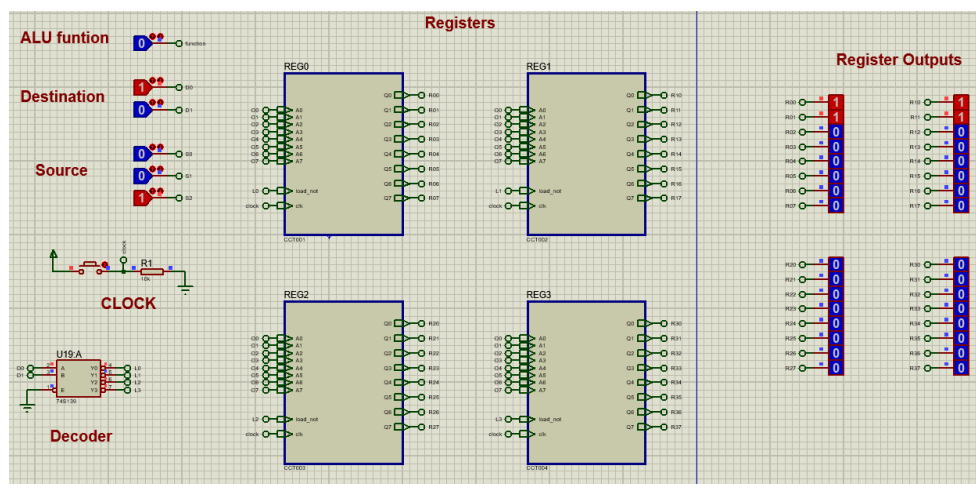


شکل ۱۷. جمع شدن عدد یک با R0 و قرارگیری عدد ۳ در R0

حال می‌خواهیم عدد ۳ را در R1 بارگذاری کنیم. برای این کار کافی است R0 را با 0 جمع کرده و حاصل را در R1 قرار دهیم.

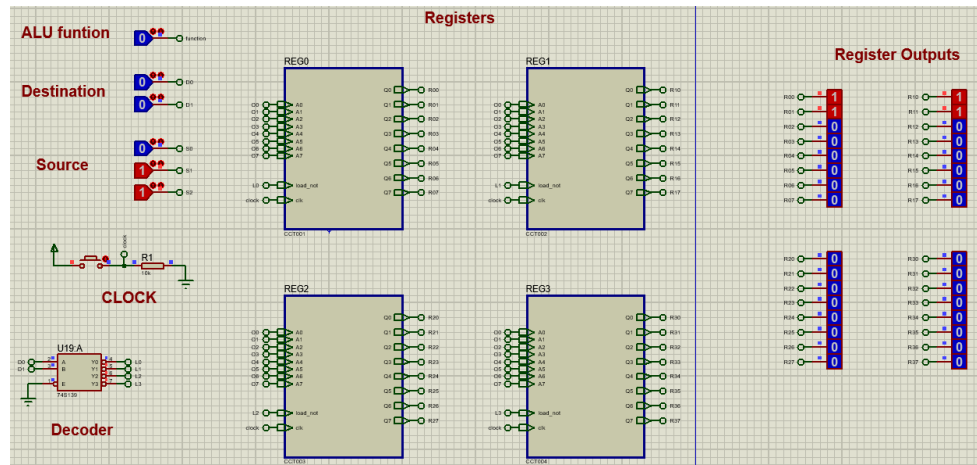


شکل ۱۸. آغاز انجام عملیات بارگذاری ۳ در R1

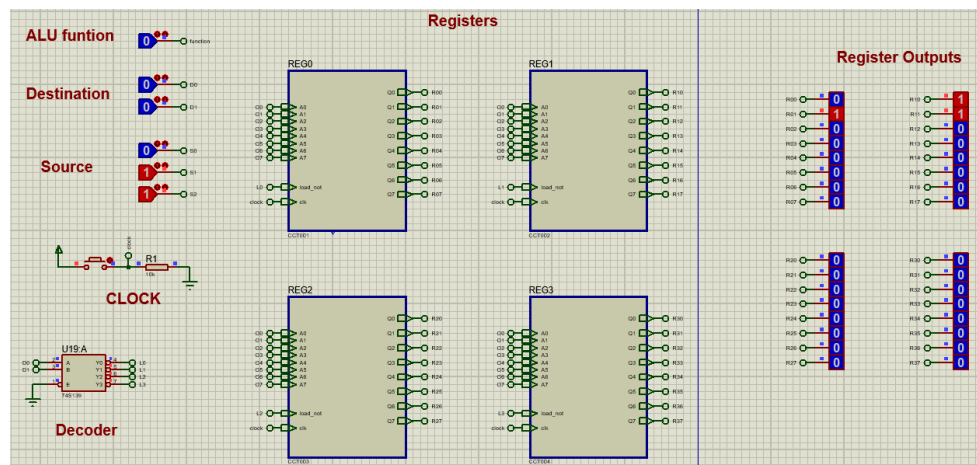


شکل ۱۹. بارگذاری شدن عدد ۳ در R1

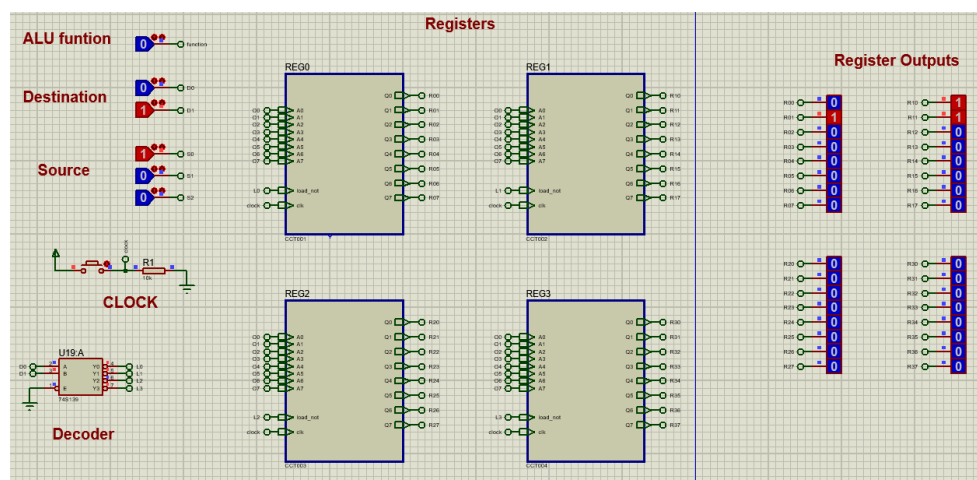
اکنون می‌خواهیم عدد ۵ را در R2 قرار دهیم. برای این کار ابتدا بایستی R0 را با 1- جمع کرده و سپس R0 را با R1 جمع می‌کنیم و حاصل را در R2 قرار می‌دهیم. تصاویر آورده شده در ادامه گزارش نشان‌دهنده این عملیات می‌باشند.



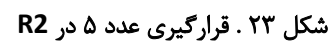
شکل ۲۰. تنظیم ورودی‌ها برای جمع عدد ۱- و R0 و قرارگیری حاصل در R0



شکل ۲۱. قرارگیری عدد ۲ در R0



شکل ۲۲. تنظیم ورودی‌ها برای جمع R1 و R0 و قرارگیری حاصل در R2



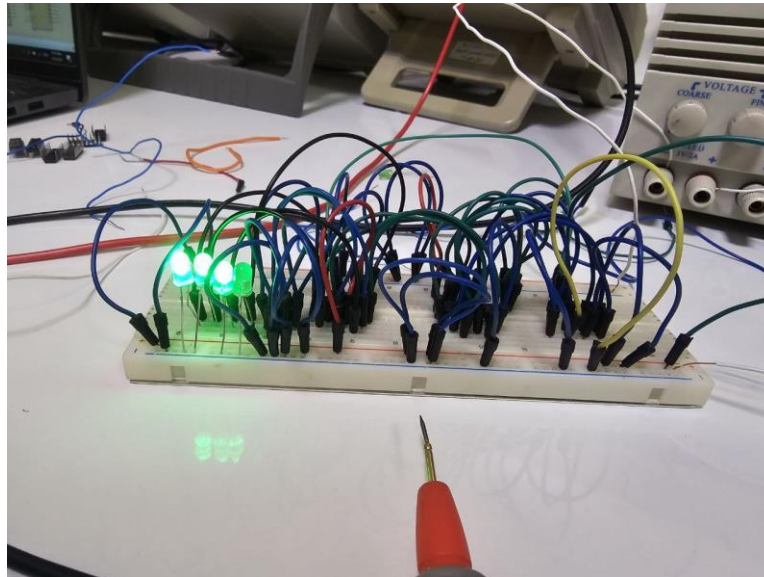
The diagram illustrates a 4-bit ALU and Register File circuit. The ALU section includes logic for addition, subtraction, and bit shifting, with inputs for ALU funtion, Destination, Source, and a shared 4-bit Clock. The Register File section contains four 4-bit registers (REG0, REG1, REG2, REG3) with outputs for Register Outputs. The circuit is implemented using a 74130 decoder and a 74135 ALU chip.

شکل ۲۴. تنظیم ورودی‌ها برای تفریق R2 و R0 و قرارگیری حاصل در R3

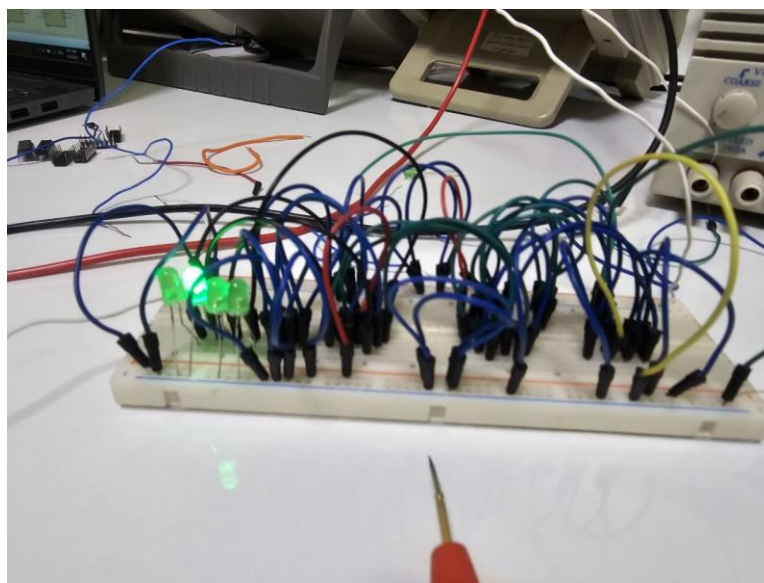


بخش دوم: پیاده‌سازی روی برد

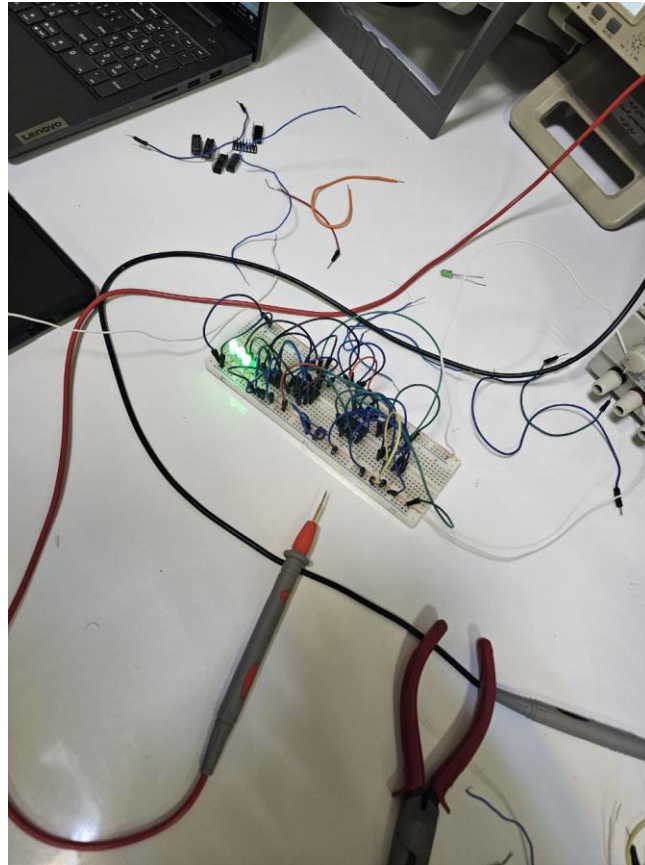
حال مدار را روی برد در آزمایشگاه پیاده‌سازی می‌کنیم. برای سادگی به جای چهار ثبات، فقط دو ثبات $R0$ و $R1$ را در نظر می‌گیریم. همچنین مالتی‌پلکسرها را ۴ به ۱ در نظر می‌گیریم که ورودی‌های آن‌ها به ترتیب بیت‌های $R0$ ، $R1$ ، ۱ و ۰-۱ می‌باشد. ابتدا با استفاده از یک برد و دو ایسی ۷۴۸۳ (جمع کننده ۴ بیتی) و ایسی ۷۴۸۶ (XOR) واحد جمع/تفریق کننده را مطابق توضیحات قبل می‌سازیم. تصاویر زیر مدار پیاده‌سازی شده این واحد را با ورودی و خروجی‌های مختلف نشان می‌دهند.



شکل ۲۶. پیاده سازی واحد جمع/تفریق کننده، ورودی‌ها: ۱۰ و ۴، خروجی: ۱۴

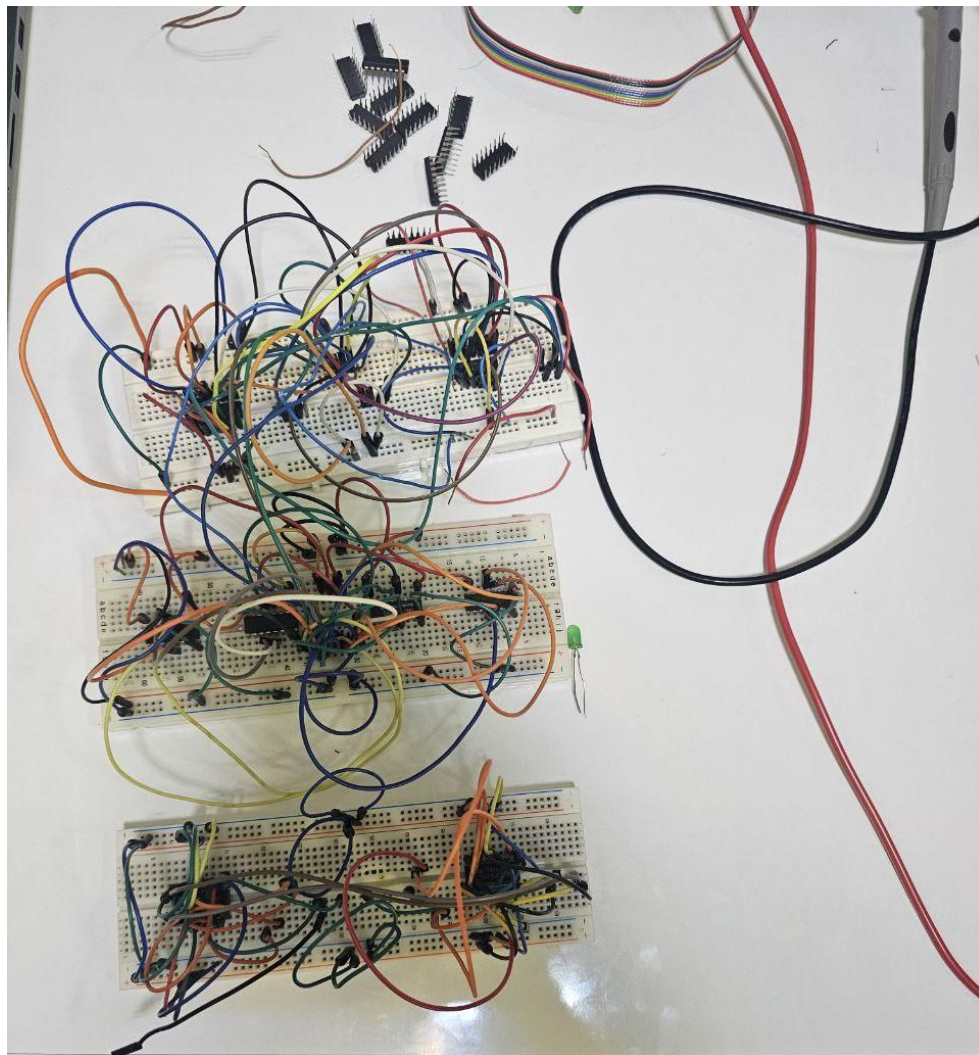


شکل ۲۷. پیاده سازی واحد جمع/تفریق کننده، ورودی‌ها: ۱ و ۳، خروجی: ۴



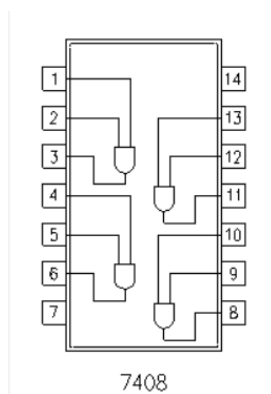
شکل ۲۷. پیاده سازی واحد جمع/تفریق کننده، ورودی‌ها: ۱۰ و ۴،
خروجی: ۱۴

با استفاده از یک برد و ۴ ایسی ۷۴۱۷۵ (ثبات ۴ بیتی) دو ایسی ۸ بیتی $R0$ و $R1$ را در برد پیاده‌سازی می‌کنیم. سپس با استفاده از دو برد و ۴ تا ایسی ۷۴۱۵۳ (مالتی‌پلکسر ۴ بیتی) بخش مربوط به MUX را طراحی می‌کنیم. از آن جایی که ایسی ۷۴۱۵۳ در واقع دو مالتی‌پلکسر ۴ به ۱ است فقط نیاز به چهار عدد از آن‌ها داریم. تصویر ۲۸ این سه برد را نشان می‌دهد.

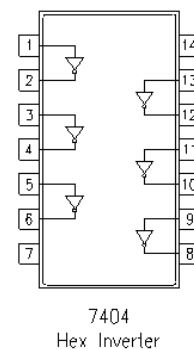


شکل ۲۸. پیاده سازی ثبات‌های ۸ بیتی و مالتی‌پلکسرها

حال همانطور که در بخش اول نیز توضیح داده شد نیاز به یک دیکودر برای طراحی `load_not` هر ثبات داریم. همچنین در مدار داخلی هر ثبات دیدیم که ورودی `load_not` با کلاک `AND` می‌شود و به عنوان `CLK` به ایسی‌های ۷۴۱۷۵ داده می‌شوند. بنابراین

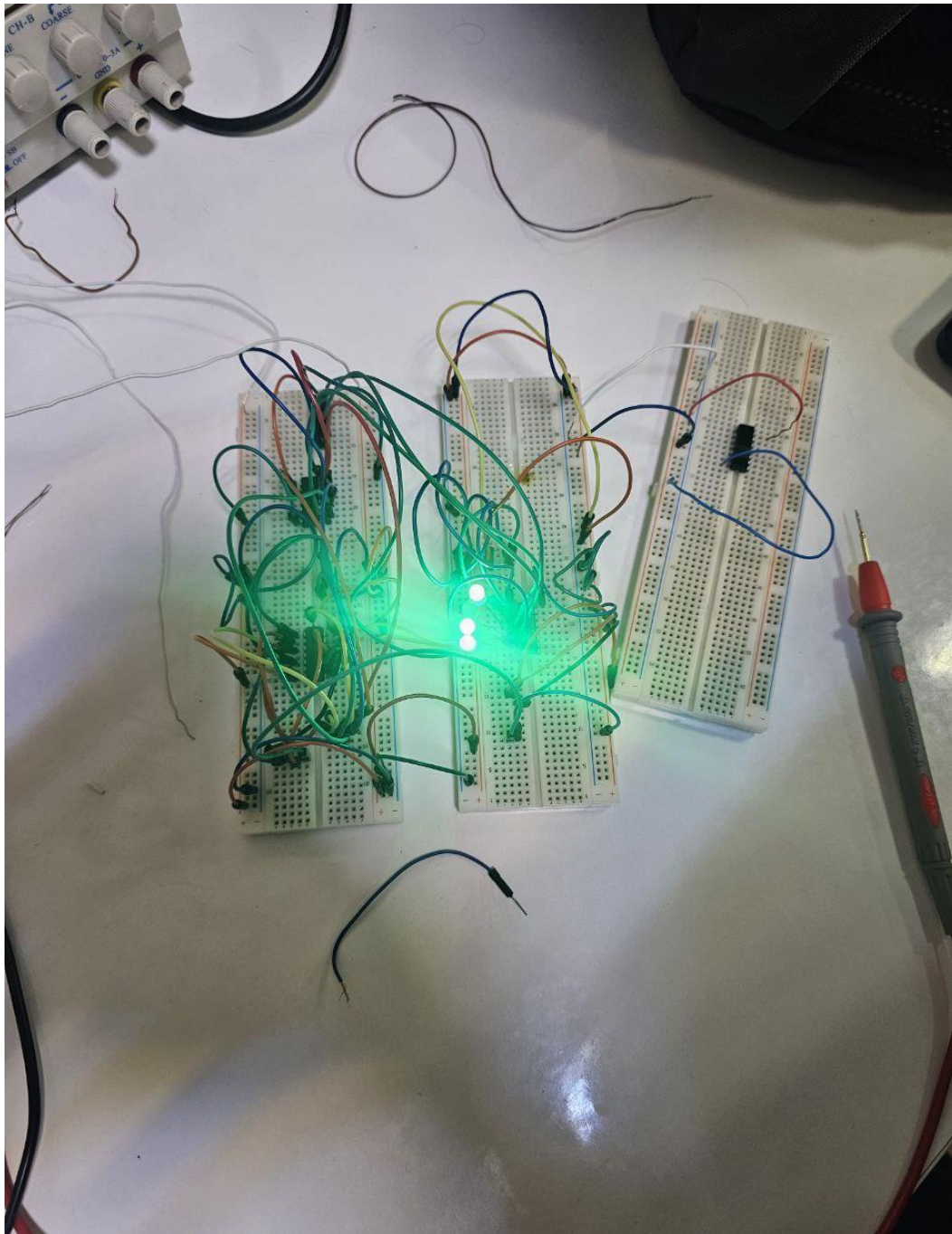


شکل ۲۹



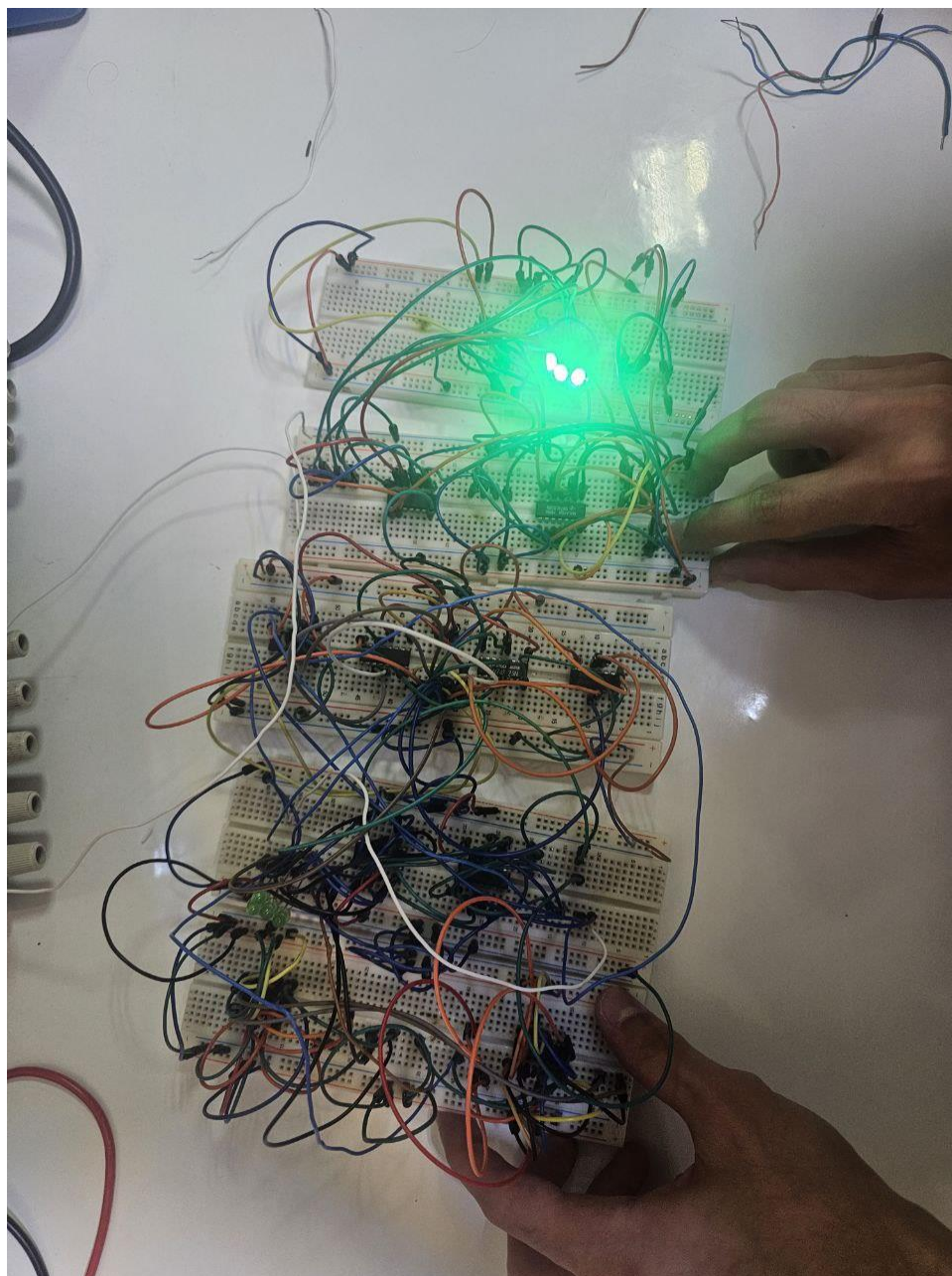
شکل ۳۰

با استفاده از یک ایسی ۷۴۱۳۹ و دو ایسی ۷۴۰۸ (AND) و ۷۴۰۴ (Hex Invertor) برد زیر را می‌سازیم. همچنین ورودی‌ها را نیز در این بوردها نشان می‌دهیم.

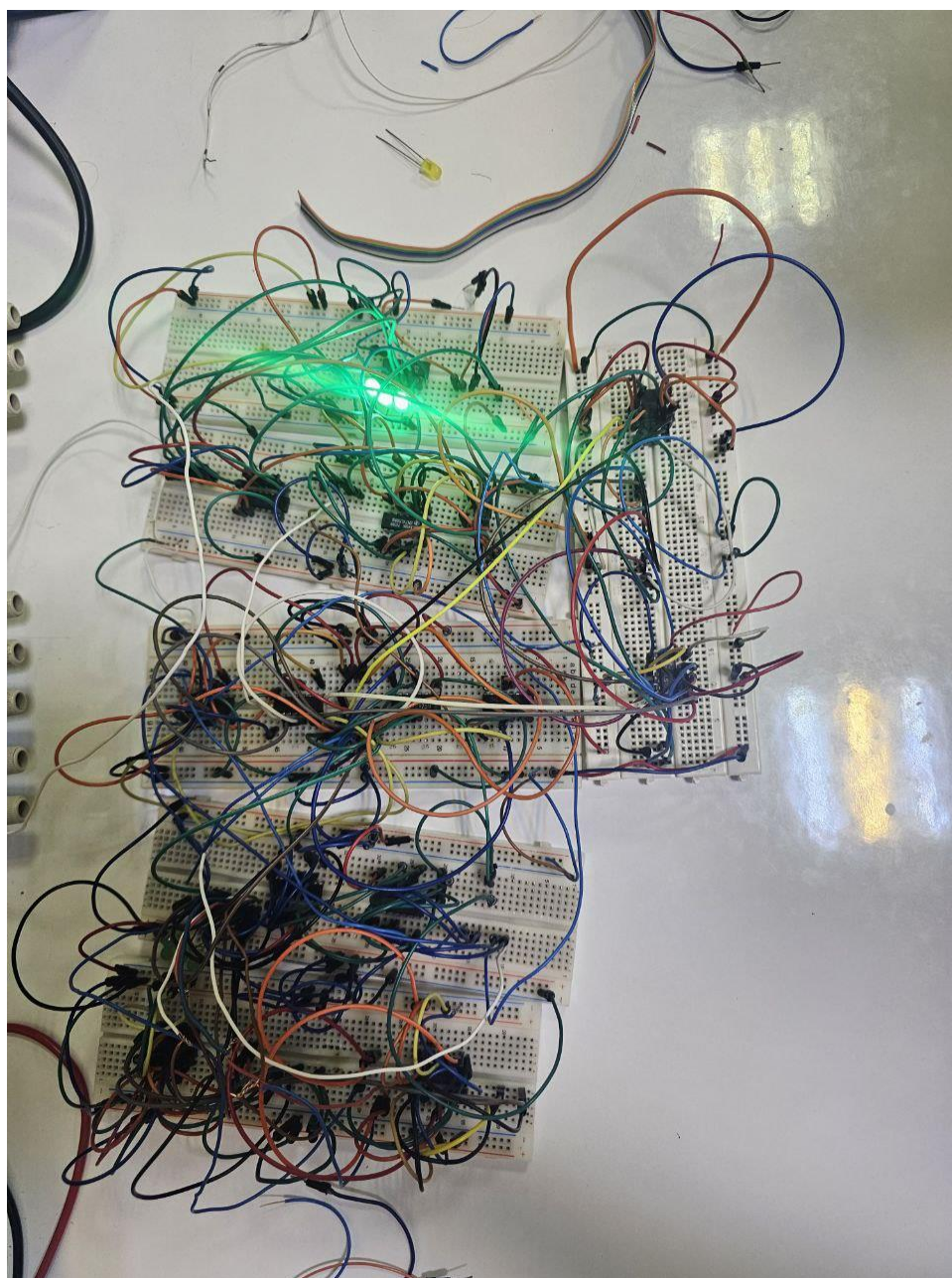


شکل ۳۱. پیاده سازی load_not و کلاک هر ثابت

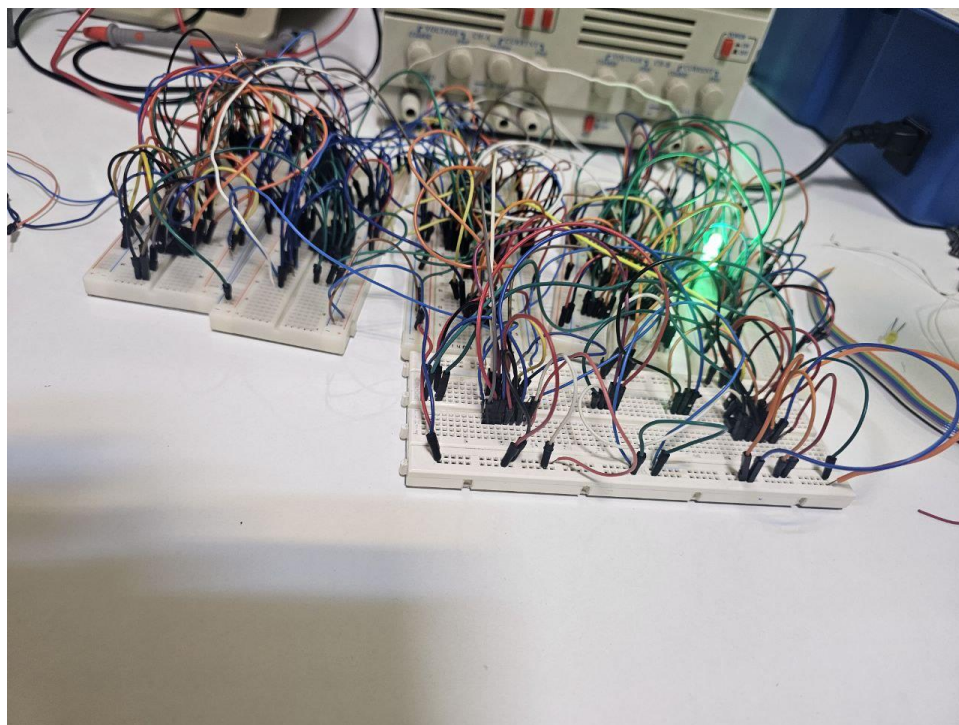
در نهایت مدار کلی را با اتصال بوردها به یک دیگر می‌سازیم. تصاویر زیر مدار کلی را نشان می‌دهند.



شکل ۳۲. تصویر مدار نهایی



شکل ۳۳. تصویر مدار نهایی



شکل ۳۴. تصویر مدار نهایی