دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

عنوان:

آزمایش دوم: واحد محاسبه با امکان انتخاب ثبات مبدا و مقصد

اعضای گروه

امیرحسین علمدار_۴۰۰۱۰۵۱۴۴

پیام تائبی_۴۰۰۱۰۴۸۶۷

ماهان بیهقی_۴۰۰۱۰۴۸۳۴

نام درس

آزمایشگاه معماری کامپیوتر

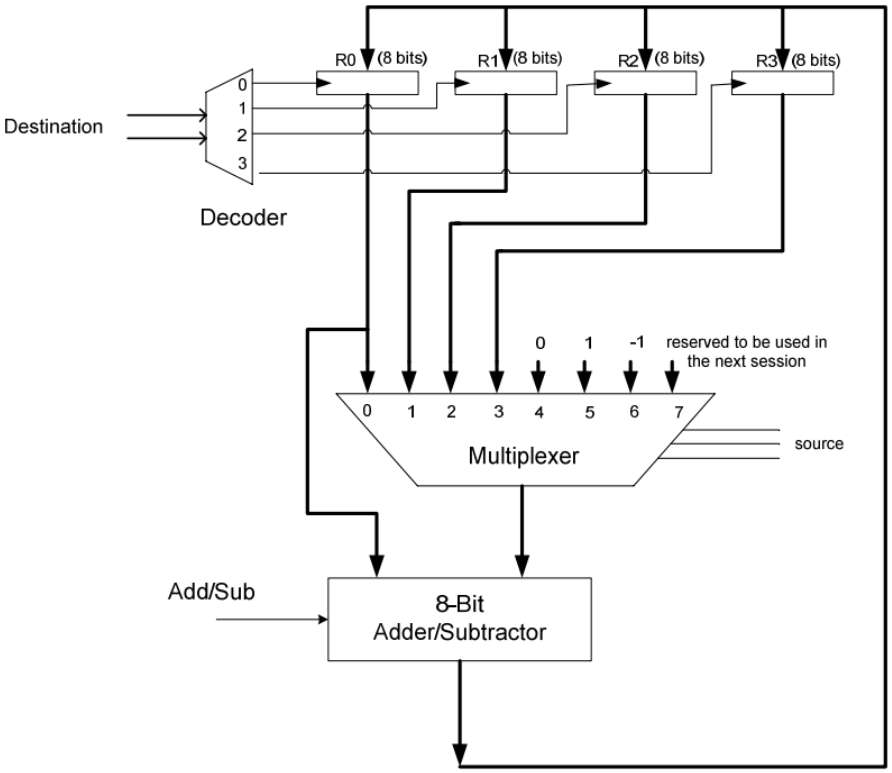
تابستان ۱۴۰۲

نام استاد درس

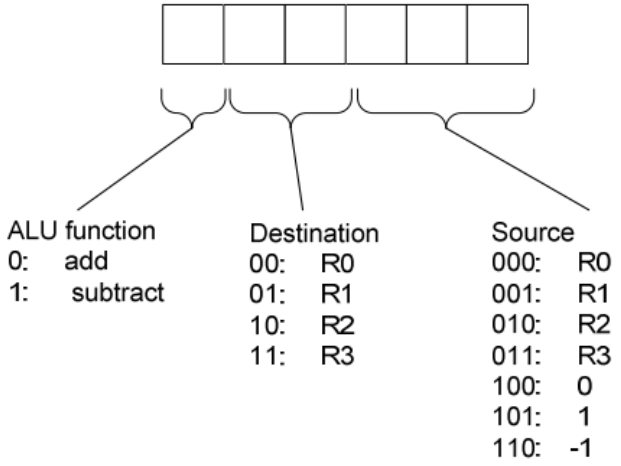
حمید سربازی آزاد

چکیده

در این آزمایش قصد طراحی و پیاده سازی یک واحد محاسبه به همراه ثبات های عمومی ماشین را داریم. این ماشین شامل ۴ ثبات ۸ بیتی است. در هر دستور می توانیم ثبات صفر را با ۱، ۰، ۱- و یا دیگر ثبات ها جمع بزنیم و نتیجه حاصل را در مقصد مشخص شده بریزیم. همانگونه که در شکل ۱ و ۲ مشاهده می کنید و طبق دستورکار، دستور داده شده با ماشین شامل ۶ بیت است و ۳ بیت پایین برای تعیین علموند دوم و پرارزش ترین بیت نیز برای تعیین عملگر استفاده می شود.پس با داده شدن این بیت ها، واحد محاسبات ثبات صفر را با عملوند مشخص شده جمع یا تفریق میکند. حال بیت های ۳ و ۴ باقی مانده اند، که این دو بیت نیز مشخص می کنند نتیجه واحد محاسبات در کدام یک از ثبات ها لود شود.



شکل ۱: شمای کلی طراحی مورد نظر، همانطور که مشاهده می کنید بیت های داده شده برای سلکت عملوند، عملگر و همچنین مقصد مناسب استفاده می شوند.



شکل ۲: همانگونه که توضیح داده شد، ۳ بیت پایین برای تشخیص علموند، ۲ بیت بعدی برای انتخاب ثبات مقصد و بیت بزرگ نیز برای تعیین جمع یا ضرب است.

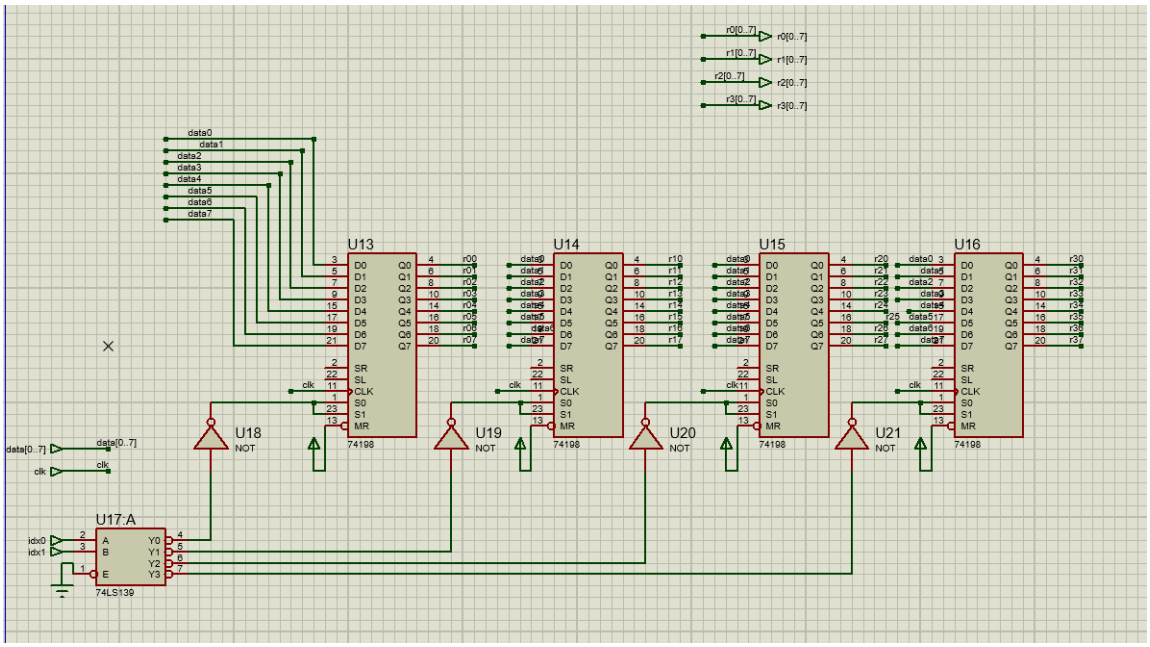
۱ نحوه طراحی

همانگونه که در شکل ۱ دیدید ، مدار به ۳ بخش اصلی تقسیم می شود. در بخش اول ۴ رجیستر ۸ بیتی داریم که سیگنال لود آن ها باتوجه به یک دیکودر مشخص میشود. دیکودر بیت های ۳ و ۴ دستور را می گیرد و باتوجه به آن ها ثبات هدف مشخص شده و با خوردن کلاک دیتا در آن لود می شود. در بخش بعد به یک مولتی پلکسر ۸ به ۱ نیاز داریم که پهنای هر ورودی ۸ بیت است. از آنجایی که این قطعه در پروتئوس وجود ندارد، آن را خودمان می سازیم. در بخش سوم یک واحد محاسبات داریم که با توجه به بزرگترین بیت، عملیات جمع یا تفریق را روی ورودی ها انجام می دهد.

۱-۱ رجیسترفایل

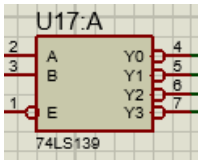
این ماژول ۸ بیت دیتا، ۲ بیت ایندکس ثبات مقصد لود، و سیگنال کلاک را ورودی میگیرد. خروجی ماژول همان مقادیر ثبات ها می باشد. ۸ بیت دیتا ورودی از خروجی واحد محاسبه جمه کننده/تفریق کننده می آید تا در ثبات هدف با توجه به ایندکس های داده شده لود شود. ۴ باس ۸ بیتی از ماژول خارج می شود تا به موتلی پلکسر ۸ به ۱ داده شود. دو ایندکس داده شده ابتدا به دیکودر ۷۴۱۳۹ وارد می شوند و با توجه به مقادیر آن ها، یکی از خروجی های دیکودر صفر و باقی یک می شوند. در واقع دیکودر نات خروجی هارا میدهد، حال باید یکبار این خروجی هارا نات کنیم تا برای سیگنال لود رجیستر ها از آن ها استفاده کنیم. حال برای رجیستر ها از قطعه ۷۴۱۹۸ استفاده می کنیم. این قطعه

علاوه بر لود قابلیت شیفت به راست و چپ را نیز دارد که از آن ها استفاده ای نمی کنیم. برای لود مقدار کافیست s_0 و s_1 هردو یک شوند بنابرین خروجی نات شده دیکودر را به هردو پایه وصل می کنیم. کلاک را نیز به پایه کلاک می دهیم. حال دیتای ورودی را به پایه های هر ۴ رجیستر وصل می کنیم تا با خوردن کلاک، دیتا باتوجه به ایندکس داده شده در ثبات مورد نظر لود شود. نکته ای که وجود دارد این است که مدار سینگل سایکل می باشد، لذا می تونیم همواره با هرکلاک در رجیستر ها مقدار را لود کنیم و مشکلی پیش نمی آید. در نهایت نیز خروجی رجیستر ها را خروجی میدهیم تا به ماژول بعدی ینی مولتی پلکسر بروند.(شکل ۳)

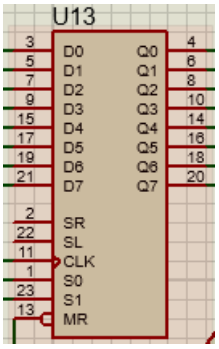


شکل ۳: شمای کلی رجیستر فایل، همانگونه که توضیح داده شد از قطعات ۷۴۱۳۹ و ۷۴۱۹۸ برای دیکودر و رجیستر استفاده شده است.

۱-۱-۱ قطعات رجیسترفایل



شکل ۴: دیکودر ۲ به ۴، خروجی ها را نات میکند و همچنین سیگنال enable اکتیو لو دارد که آن را به ۰ وصل می کنیم.

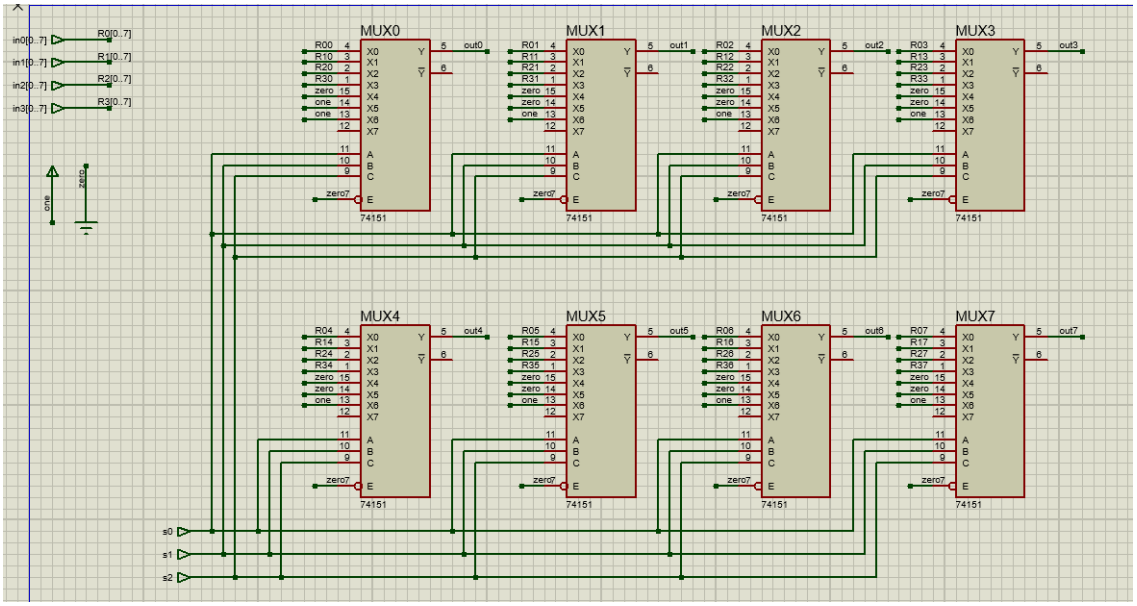


شکل ۵: شیفت رجیستر ۸ بیتی ۲ طرفه، در دو آزمایش گذشته تقریبا با این قطعه آشنا شدیم ولی اینجا فقط از قابلیت لود موازی آن استفاده می شود، برای اینکار کافیست s_0 و s_1 را یک کنیم و کلاک بخورد. همچنین MR همان کلییر است پس آن را یک می دهیم تا مقدار رجیستر ها پاک نشود. می توانستیم برای مدار دکمه ریست نیز بگذاریم که خواسته نشده بود.

شکل ۶: نات

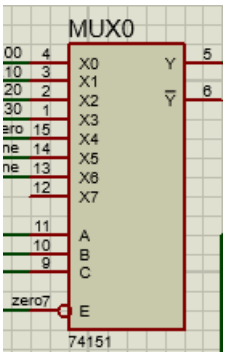
۱-۲ مولتی پلکسر ۸ به ۱

برای این بخش ۴ ورودی از رجیستر فایل، ۳ بیت برای انتخاب source وارد می شود و یک خروجی داریم که درواقع عملوند دوم را به جمع/تفریق کننده می دهد. دقت کنید پهنای ورودی ها و خروجی ها بجز بیت های انتخاب ۸ بیتی است. پس برای هر بیت از ۰ تا ۷، باید بین ۷ گزینه موجود، یکی را انتخاب کنیم. لذا به ۸ مولتی پلکسر نیاز داریم که هرکدام ۸ به ۱ باشند. ۴ ورودی اول تمام مولتی پلکسر ها، بیت های متناظر کرفته شده از رجیستر فایل و اعداد دستی داده شده است. اعداد دستی ۱ و ۰ و ۱- اند. برای ۰ و ۱- تمام بیت ها را ۰ و ۱ می دهیم و برای ۱ کافیست بیت مربوط به مولتی پلکسر اول را ۱ و باقی را ۰ دهیم.(شکل ۷)



شکل ۷: شمای کلی درون ماکس ۸ به ۱ با پهنای ۸ بیت، هر مولتی پلکسر برای سلکت یک بیت از ۸ بیت خروجی است و با کنار هم قرار گرفتن نتایج آن ها، ۸ بیت خروجی شکل می گیرد.

۱-۲-۱ قطعات ماکس ۸ به ۱



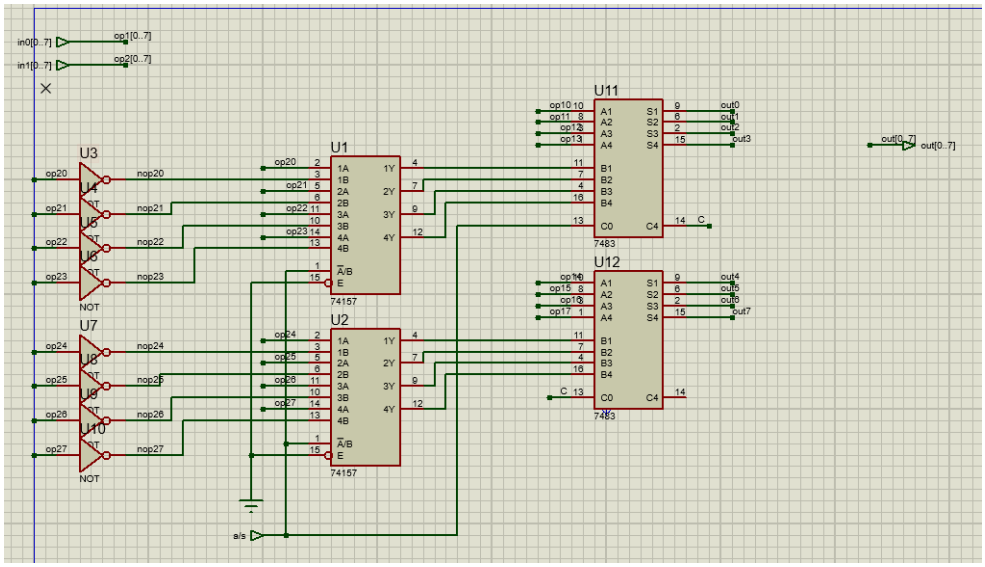
شکل ۸: قطعه ۷۴۱۵۱ تنها قطعه استفاده شده در این بخش است. پایه های A و B و C نیز برای سلکت عملوند دوم اند که با توجه به شکل ۱ و ۲، ترتیب دادن عملوند ها به پایه های مولتی پلکسر معلوم می شوند. پایه E نیز، enable اکتیو لو است و کافیست آن را ۰ بدهیم.

۱-۳ واحد محاسبات

ورودی ها شامل دو ورودی ۸ بیتی که یکی از مولتی پلکسر بخش قبل و دیگری مستقیما از ثبات شماره صفر می آیند. یک بیت ورودی دیگر نیز مشخص کننده جمع یا تفریق است. خروجی نیز نتیجه محاسبات است.

ابتدا تمام بیت های عملوند دوم را نات میکنیم. حال با استفاده از دو مولتی پلکسر ۴ بیتی ۲ به ۱،

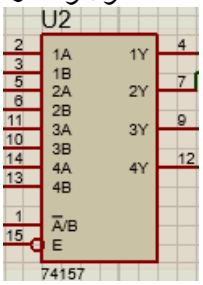
یکی از حالات را انتخاب می کنیم. این دومولتی پلکسر در کنار هم نقش یک مولتی پلکسر ۲ به ۱ ۸ بیتی را دارند، یعنی با توجه به بیت انتخاب، ۸ بیت اول یا بیا دوم را خروجی می دهند. حال ما ورودی اول را ۸ بیت اصلی و ورودی دوم را نات آنها می دهیم، به اینصورت اگر بیت انتخاب ۱ باشد و درحالت تفریق باشیم، نات بیت ها خروجی داده می شود و در حالت دیگر خود بیت ها خارج می شوند. حال ۲ فول ادر ۴ بیتی داریم که عملوند اول که همان خروجی ثبات شماره صفر بود، بصورت مستقیم ورودی اول آنهاست و عملوند دوم از مولتی پلکسر درون ماژول می آید. کری خروجی فول ادر اول را به کری ورودی فول ادر دوم می دهیم. حال نکته مکمل دوم این بود که بیت ها را نات و با یک جمع کنیم، پس برای اجرای اینکار، کری ورودی جمع کننده اول را بیت انتخاب جمع یا تفریق می دهیم تا اگر ۱ بود، بیت های نات شده که از مولتی پلکسر انتخاب شده اند، با ۱ نیز جمع شوند تا ورودی اول با مکمل دوم ورودی دوم جمع شود. اگر بیت انتخاب ۰ بود نیز اتفاقی نمیفتد و جمع ساده انجام می شود.(شکل ۹) خروجی ماژول نیز به رجیستر فایل می رود تا با خوردن کلاک در ثبات هدف لود شود.



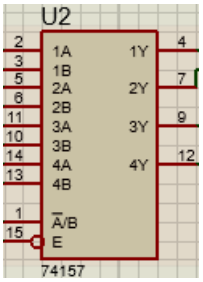
شکل ۹: واحد محاسبات شامل ۲ مولتی پلکسر ۲ به ۱ ۴ بیتی و ۲ فول ادر ۴ بیتی در کنار هم برای انجام مکلمل دو روی عملوند دوم درصورت یک بودن بیت مشخص کننده عملیات جمع یا تفریق.

۱-۳-۱ قطعات واحد محسبات

علاوه برقطعات شکل های ۱۰ و۱۱، از ۸ نات موجود در شکل ۶ نیز استفاده شده است.



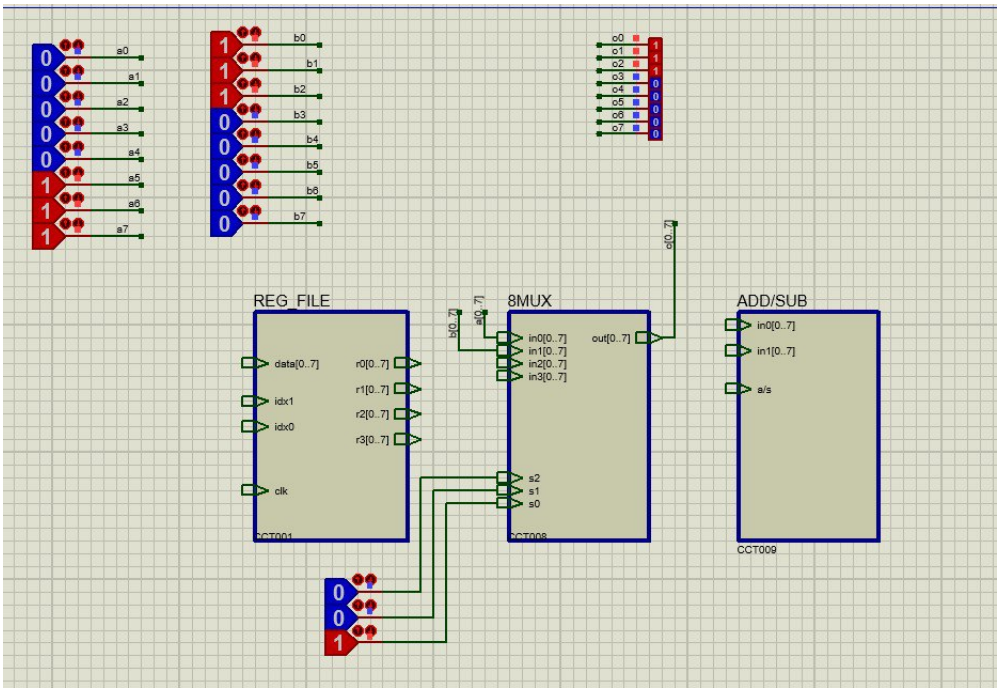
شکل ۱۰: جمع کننده، دوعدد ۴ بیتی ورودی و یک بیت کری ورودی می گیرد و ۴ بیت و یک کری خروجی میدهد.



شکل ۱۱: مولتی پلکسر ۴ بیتی ۲ به ۱، دو عدد ۴ بیتی ورودی می گیرد و باتوجه به ۰ یا ۱ بودن بیت انتخاب، اولی یا دومی را خروجی میدهد. همچنین enable اکتیو لو دارد که آن را ۰ میدهیم.

۱-۴ مدار نهایی

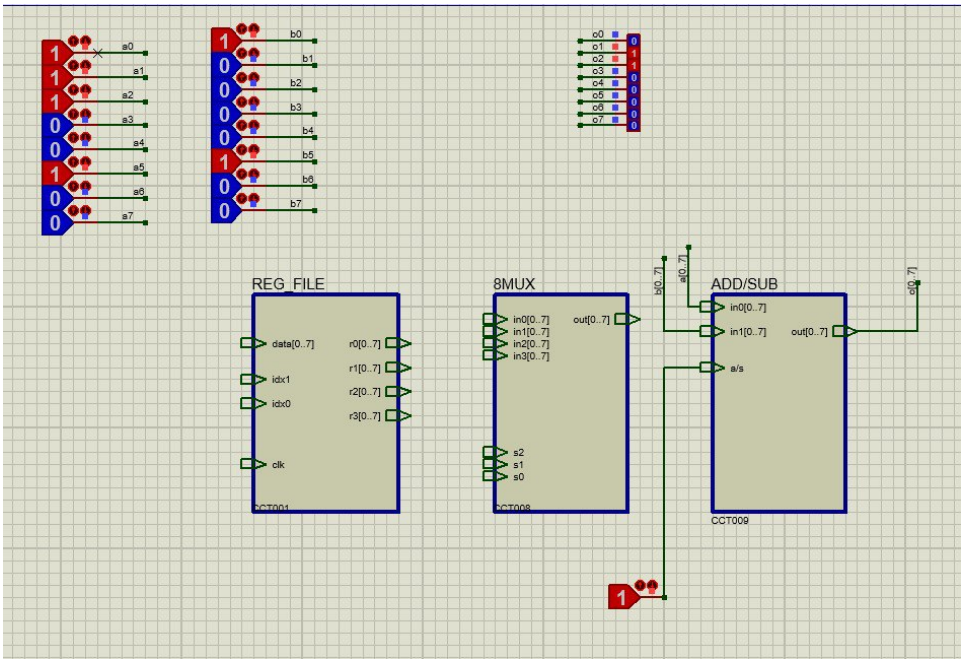
ابتدا ماژول های نهایی مولتی پلکسر و واحد محاسبات را به تنهایی تست میکنیم. این دو ماژول ترکیبی اند و تست آنها به راحتی قابل انحام است، پس از تست آن ها میتوانیم کل مدار را تست کنیم تا اگر مشکلی در رجیستر فایل بود نیز متوجه شویم.(شکل ۱۲ و ۱۳ نمونه هایی از تست های انجام شده هستند.)



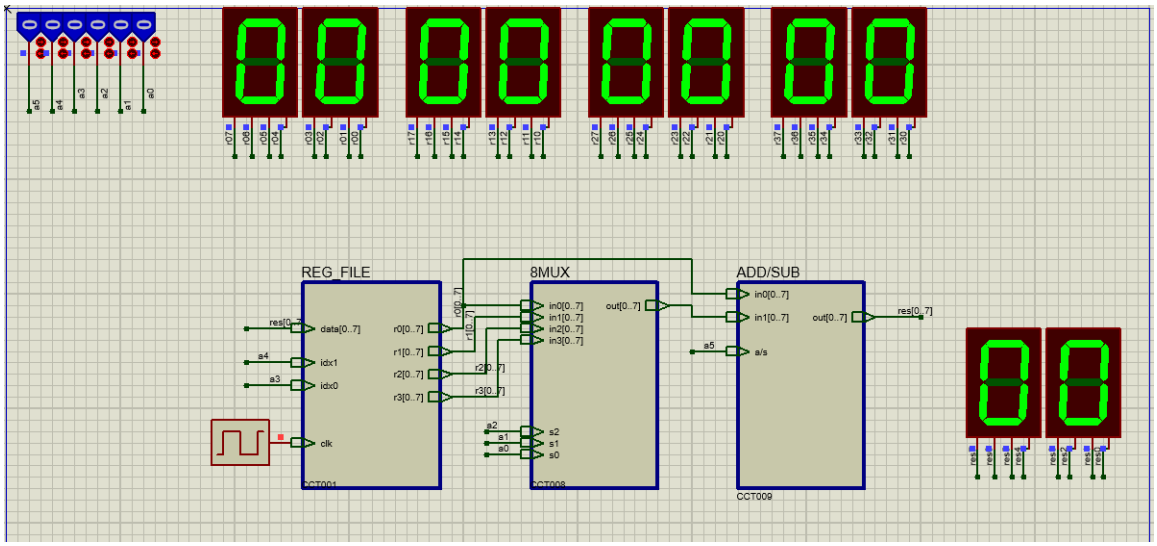
شکل ۱۲: مولتی پلکسر باید ورودی in۱ را انتخاب کند که به درستی آن را خروجی می دهد.

حال با اطمینان از درستی بخش های ترکیبی، مدار را کامل و آن را تست می کنیم. در شکل ۱۳ مشاهده می کنید که خروجی های رجیستر فایل به مولتی پلکسر رفته اند و در آنجا با توجه به اینستراکشن، یکی از ۷ گزینه موجود برای عملوند دوم انتخاب و به alu داده می شود. برای علموند اول نیز ثبات r۰ می آید و alu باتوجه به ببیت عملیات، آن هارا جمع یا تفریق می کنید و سپس نتیجه را به رجیسترفایل میدهد و با خوردن کلاک، دیتا در ایندکس گفته شده توسط دستور، لود می شود.

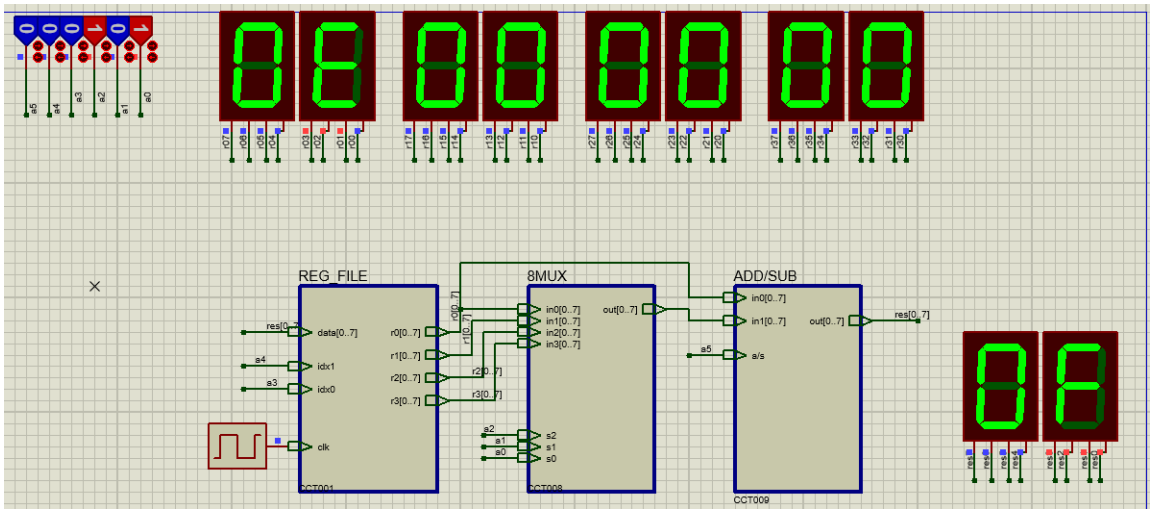
در ادامه دو حالت نهایی را تست و ضمیمه می کنیم.



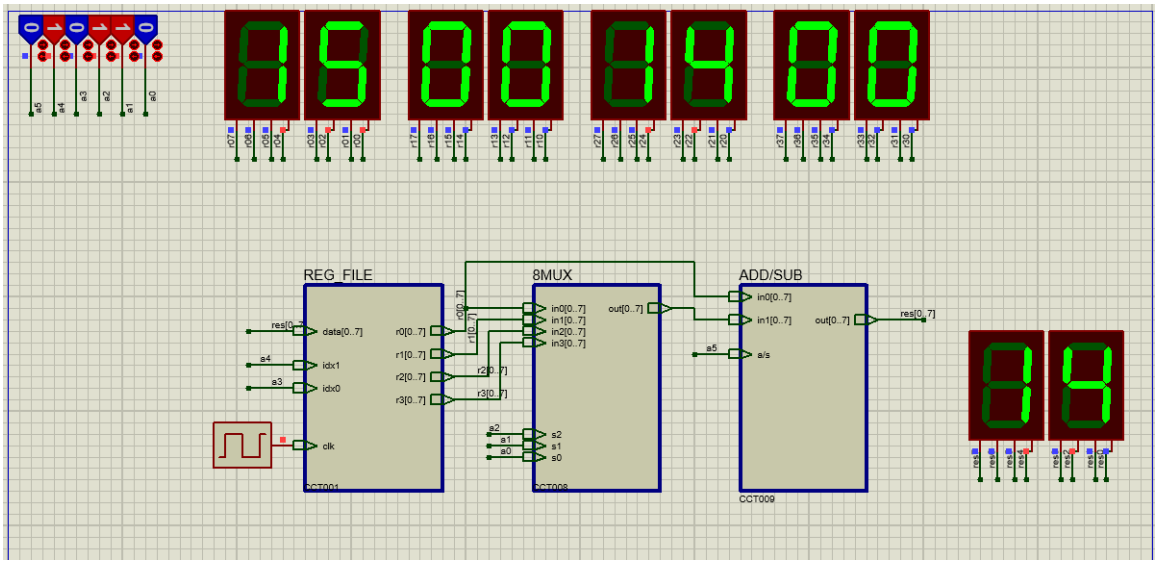
شکل ۱۳: alu به درستی ۰۰۱۰۰۱۱۱ را منهای ۰۰۱۰۰۰۰۱ می کند و نتیجه ی ۰۰۰۰۰۱۱۰ را خروجی میدهد.



شکل ۱۴: دستور در بالا چپ شکل داده می شود . segment ۷ های بالا نیز به ترتیب از چپ به راست مقادیر ثبات های ۰ تا ۳ را نشان می دهند. در سمت راست نیز مقادیر خروجی alu نشان داده می شود تا برای دیباگ و اطمینان از درستی مدار استفاده شوند. دقت کنید که مقادیر segment ۷ ها هگز میباشد.



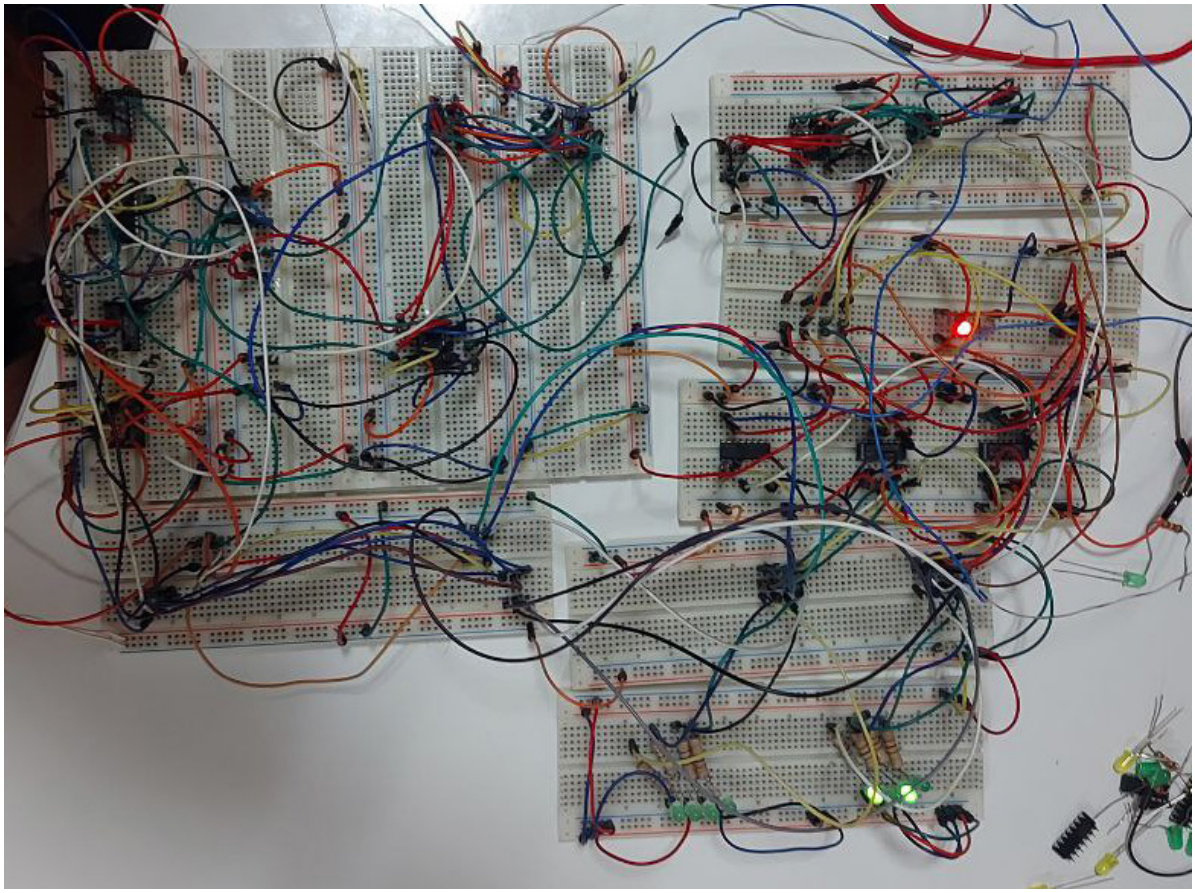
شکل ۱۵: در این قسمت دستور داده شده هدف را ثبات ۰ و مبدا رو عدد ۱ قرار داده است و دستور آن جمع است. پس با هر لبه بالارونده باید مقدار ثبات اول یکی زیاد شود. عکس در لحظه ای گرفته شده است که مقدار ۱۴ در ثبات صفر لود شده است و alu مقدار بعدی یعنی ۱۵ را محاسبه کرده و منتظر است با یک شدن کلاک این مقدار در ثبات لود شود. اگه به کلاک نیز دقت کنید آبی است و در لحظه بعد که قرمز شود مقدار ۱۵ در ثبات لود خواهد شد.



شکل ۱۶: در این تست، دستور داده شده مبدا را منفی یک و مقصد را ثبات r۲ قرار داده است،. همچنین عملیان نیز جمع است. پس باید مقدار ثبات شماره صفر یکی کم و در ثبات r۲ ریخته شود. از آنجایی که در کلاک های بعد این ثبات r۰ و منفی یک اعداد ثابتی اند، مقدار r۲ ثابت و برابر با ۲۰ می ماند.

۲ چالش های بستن مدار در آزمایشگاه

از آنجایی که بستن مداری با این تعداد بیت و مولتی پلکسر در زمان مورد نظر و با امکانات آزمایشگاه ممکن نبود، حالت ساده شده ی مدار را بستیم. به اینصورت که دو رجیستر در رجیستر فایل کافی بود و همچنین برای راحت شدن کار، ۱- را پیاده سازی نکردیم تا بتوانیم از مولتی پلکسر های ۴ به ۱ بجای مولتی پلکسر های ۸ به ۱ استفاده کنیم. همچنین به علت خراب بودن اکثر قطعات ۷۴۱۹۸ و ۸ بیتی بودن آن ها، آن هارا با ۷۴۱۷۳ جایگزین کردیم که دیتاشیت آن در فایل گزارش ضمیمه شده است. این قطعه write enable دارد که باعث شد بیت مربوط به انتخاب ثبات را به آسانی با یک نات به ثبات های مورد نظر بدهیم. دقت کنید که بعد از ساده سازی ها ذکر شده، دستور ورودی شامل ۴ بیت میشد، یک بیت برای مشخص کردن جمع یا تفریق، یک بیت برای مشخص کردن مقصد و دو بیت برای انتخاب مبدا بین دو رجیستر موجود و ۰ و ۱. به همین علت اگر بیت رجیستر مقصد ۰ می بود باید آن را نات می کردیم و حاصل را به write enable ۷۴۱۷۳ میدادیم تا دیتا در ثبات r_0 لود شود. پس نات این بیت به we رجیستر اول و خود بیت به we رجیستر دوم می رفت. در بخش دوم نیز، بجای ۸ مولتی پلکسر ۸ تایی، از ۴ مولتی پلکسر ۴ تایی استفاده کردیم. برای این بخش نیز از قطعه ۷۴۱۵۳ استفاده شد که دیتا شیت آن ضمیمه شده است. کاربرد این قطعه برای انتخاب یکی از ۴ بیت ورودی می باشد. در انتها نیز در بخش سوم برای ،alu بجای دو فول ادر و دو مولتی پلکسر، از نصف این تعداد استفاده شد. همچنین بجای قطعات ،not از قطعات nand استفاده شد و هرورودی را با خودش nand کردیم تا نات آن خروجی داده شود.



شکل ۱۷: همانگونه که در ارائه حضوری شرح داده شد، بخش بالا چپ مدار مربوط به مولتی پلکسر ها و نات کردن و انتخاب یکی از دو ۴ بیت اصلی و نات شده است. بخش بالا راست مدار فول ادر ۴ بیتی و بخش بالای LED ها مربوط به رجیستر فایل است. در نهایت یک LED قرمز مشاهده می شود که جهت اطمینان از درستی کلاک بوده است. LED های پایین نیز خروجی های رجیسترها هستند. به عنوان مثال در این شکل مقدار رچیستر دوم برابر با ۱۰۱۰ یا همان ۱۰ می باشد.