

بسمه تعالی



دانشگاه صنعتی شریف

## جمع‌کننده / تفریق‌کننده اعداد ممیز شناور

درس:

آزمایشگاه معماری کامپیوتر

کیان بهادری (۹۹۱۰۵۳۱۲)  
امیرحسین براتی (۹۹۱۰۱۳۰۸)

## **هدف آزمایش:**

در این آزمایش سعی داریم یک جمع کننده/تفریق کننده ممیز شناور با استاندارد IEEE-754 بسازیم به طوری که بتواند دو عدد ۳۲ بیتی را از ورودی دریافت کند و بتواند حاصل جمع آنها را در خروجی نمایش دهد. طبق این استاندارد، ۲۳ بیت برای مانتیس، ۸ بیت برای نما و ۱ بیت برای در نظر گرفته می‌شود. ورودی‌های این مدار به صورت نرمالایز وارد می‌شود و خروجی هم باید به صورت نرمالایز باشد.

## **شرح آزمایش:**

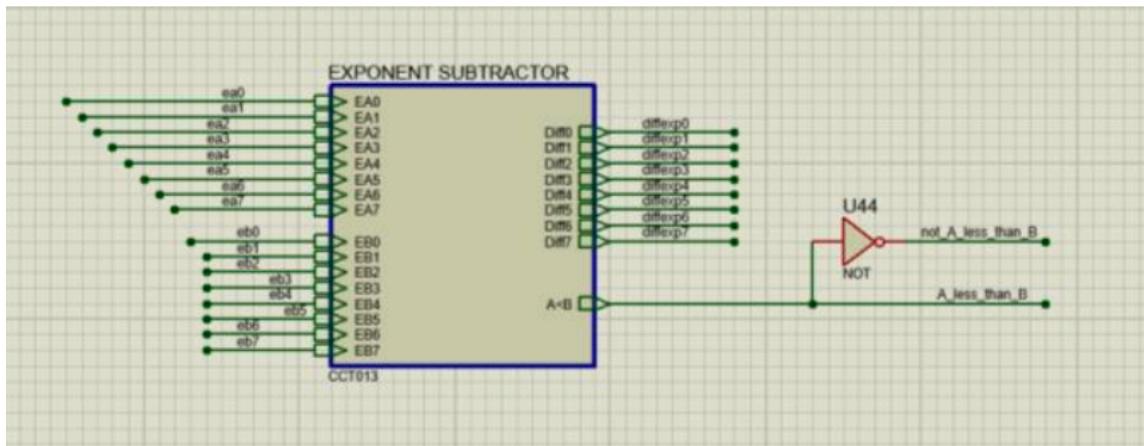
در ابتداء، اختلاف نمای اعداد A و B را محاسبه می‌کنیم. این مقدار برابر تعداد بیت‌هایی خواهد بود که لازم است مانتیس عدد با نمای کوچکتر به سمت راست شیفت بخورد. به همین ترتیب، کار بعدی شیفت دادن مانتیس عدد مورد نظر خواهد بود. سپس، نوبت به انجام عملیات بین مانتیس‌های دو عدد خواهد رسید. برای اینکه عملیات نهایی بین دو عدد را به دست بیاوریم، لازم است هم سیگنال عملیات و هم علامت دو عدد را داشته باشیم. پس از آن که عملیات روی مانتیس‌ها انجام شد، لازم است حاصل به دست آمده تا این مرحله را نرمالایز کنیم، بدین صورت که اگر در حین انجام عملیات بر روی مانتیس‌ها، حاصل به دست آمده شامل بیت کری بود، لازم است مانتیس را یک واحد به سمت راست شیفت بدهیم و همین‌طور بیت ۱ در جایگاه ۱۲۴ مانتیس قرار بگیرد و یک واحد هم به نما اضافه شود. در غیر این صورت، محاسبات مانتیس ما در ۲۴ بیت جا شده و فقط لازم است تا زمانی که بیت ۱۲۴ م برابر با ۱ نشده است، آن را یک واحد به سمت چپ شیفت می‌دهیم و یک واحد از نما کم می‌کنیم.

## **پیاده‌سازی آزمایش:**

اکنون نوبت به پیاده‌سازی مدار مربوطه به این آزمایش می‌رسد. همان طور که گفته شد مدار را به سه بخش اساسی تقسیم می‌کنیم. بخش محاسبه نما، بخش محاسبه مانتیس، بخش نرمالایز کردن. در ابتداء بخش اول یعنی محاسبه نمای حاصل را پیاده‌سازی می‌کنیم.

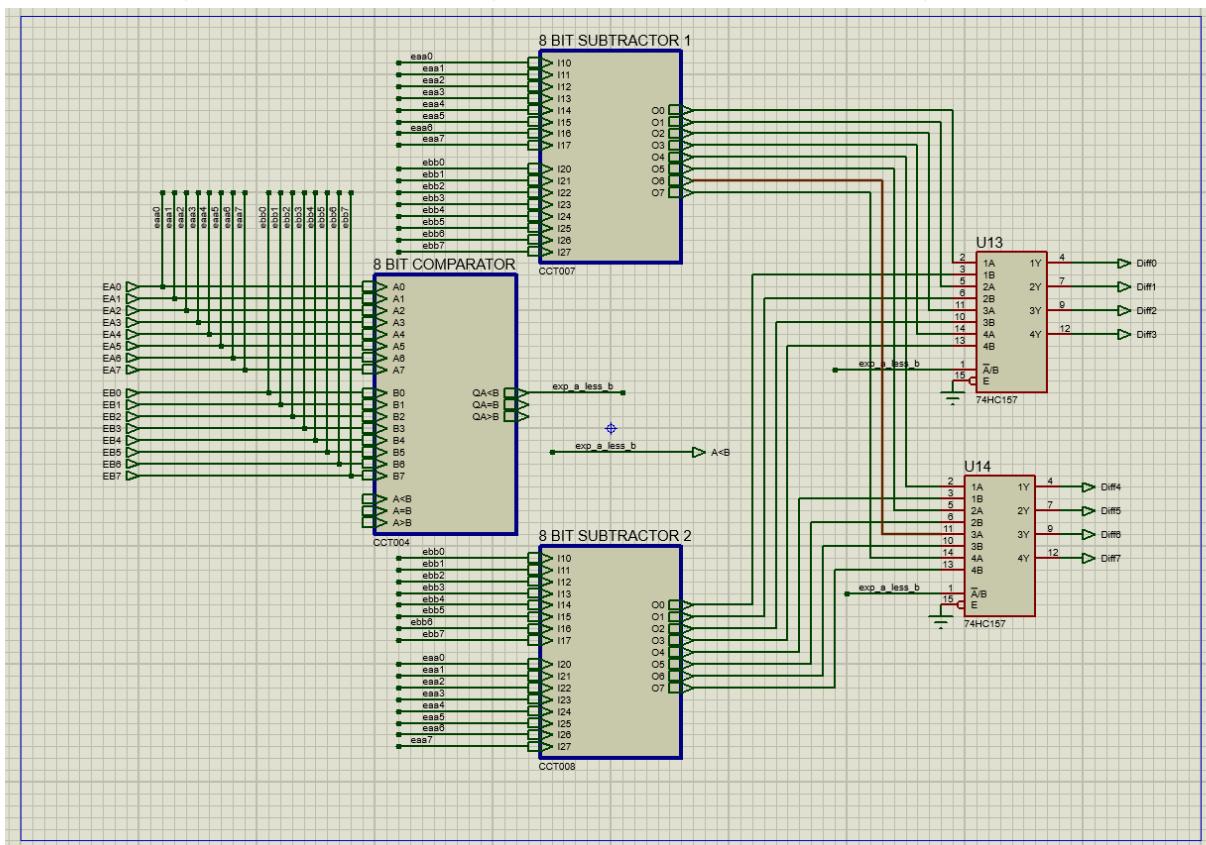
## بخش محاسبه نما:

برای اینکار یک مازول با نام exponent subtractor طراحی می‌کنیم. خروجی‌های این مدار یک بیت که نشان میدهد که نمای A از نمای B کوچک است بوده یا خیر و همین طور حاصل تفریق نمای بزرگ‌تر از نمای کوچک‌تر است. شکل کلی این مازول به صورت زیر خواهد بود:

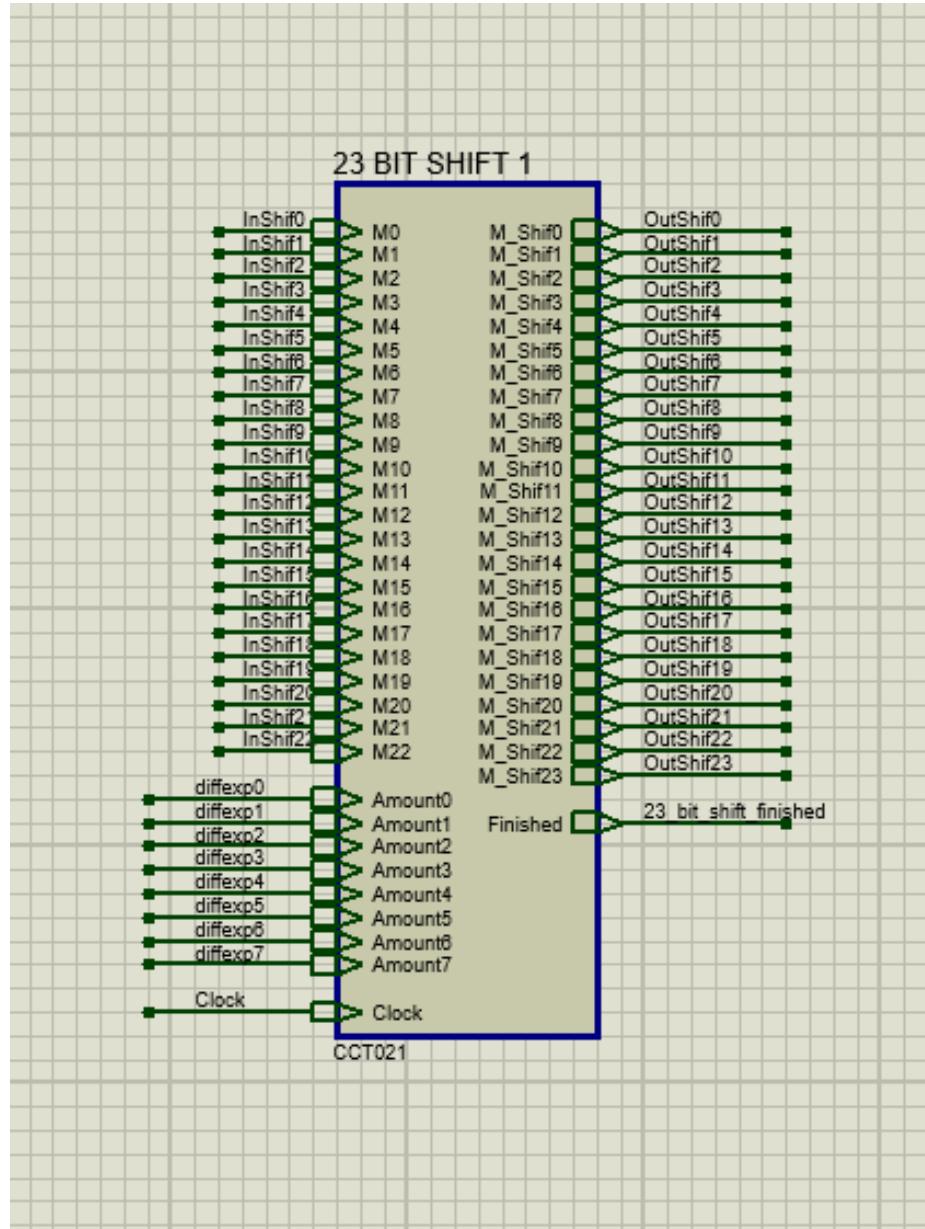


حال مدار داخلی این مازول را طراحی می‌کنیم. مدار داخلی را برای سادگی به این صورت طراحی می‌کنیم که ابتدا هر دو حالت تفریق نما را به دست می‌آوریم و با سپس با توجه به اینکه کدام یکی بزرگ‌تر بوده است مقدار مطلوب را با استفاده از دیکودر خروجی می‌دهیم.

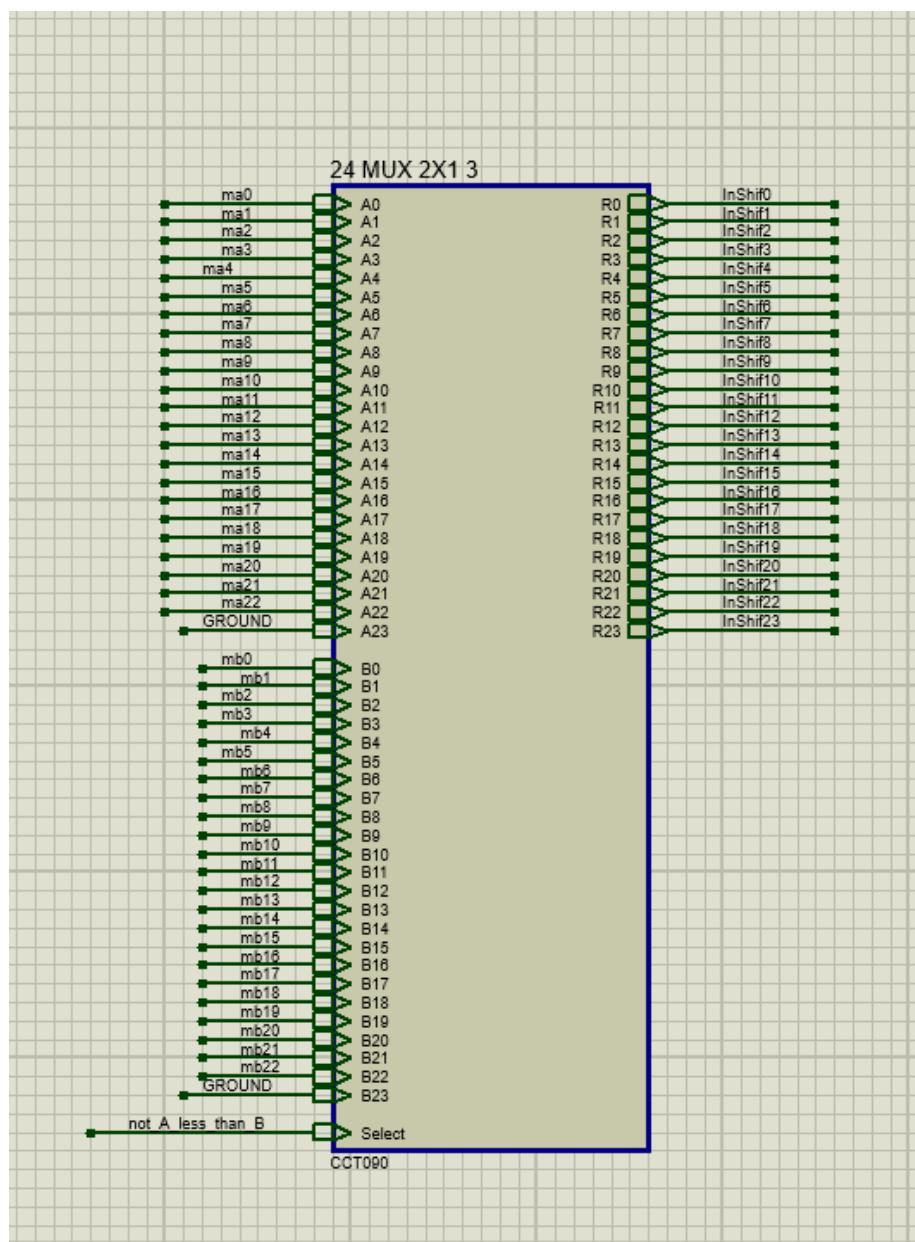
برای طراحی مدار مربوط به مقایسه کننده ۸بیتی، میتوانیم از یک مقایسه کننده آما استفاده کنیم یا با سری کردن دو مقایسه کننده ۴بیتی این مدار را پیاده سازی کنیم. برای ساختن تفریق کننده ۸بیتی هم مجدداً میتوانیم از یک مدار آماده استفاده کنیم یا با سری کردن دو تفریق کننده با هم، مدار مربوطه را پیاده کنیم.



پس از اینکه مقدار تفاضل نما به دست آمد، نوبت به شیفت دادن مانتبس کمتر تا رسیدن به مقدار مطلوب خواهد رسید. برای اینکار یک مازول جدید با نام 23 Bit Shift طراحی می‌کنیم که همین وظیفه را بر عهده خواهد داشت. این مازول در ابتدا مانتبس عدد بنمای کمتر را ورودی گرفته و به اندازه اختلاف آن نماهای دو عدد شیفت خواهد داد. این مازول به صورت زیر است:

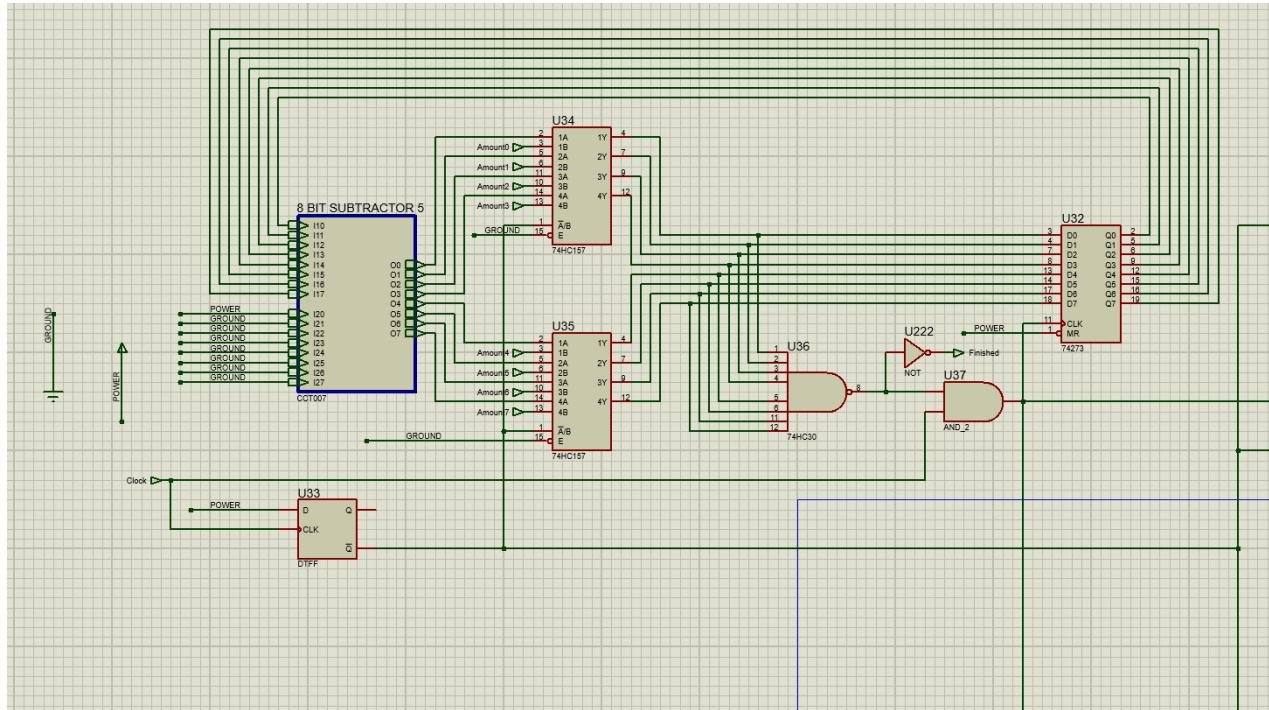


برای اینکه در ورودی این مدار مانتیس صحیح را به عنوان ورودی وارد کنیم، نیاز به یک مدار کمکی دیگر هم داریم که با توجه به اینکه کدام نما بزرگتر بوده است تصمیم بگیرد ورودی را به این مدار بدهد. برای این کار میتوانیم از یک MUX با ۲۳ ورودی استفاده کنیم. در صورتی که  $A < B$ ، باید مانتیس A را به ورودی مدار بدهیم و در غیر این صورت، مقدار B را به مدار خواهیم داد. بنابراین مدار این بخش تنها شامل یک MUX خواهد بود: که شکل آن به صورت زیر خواهد بود:

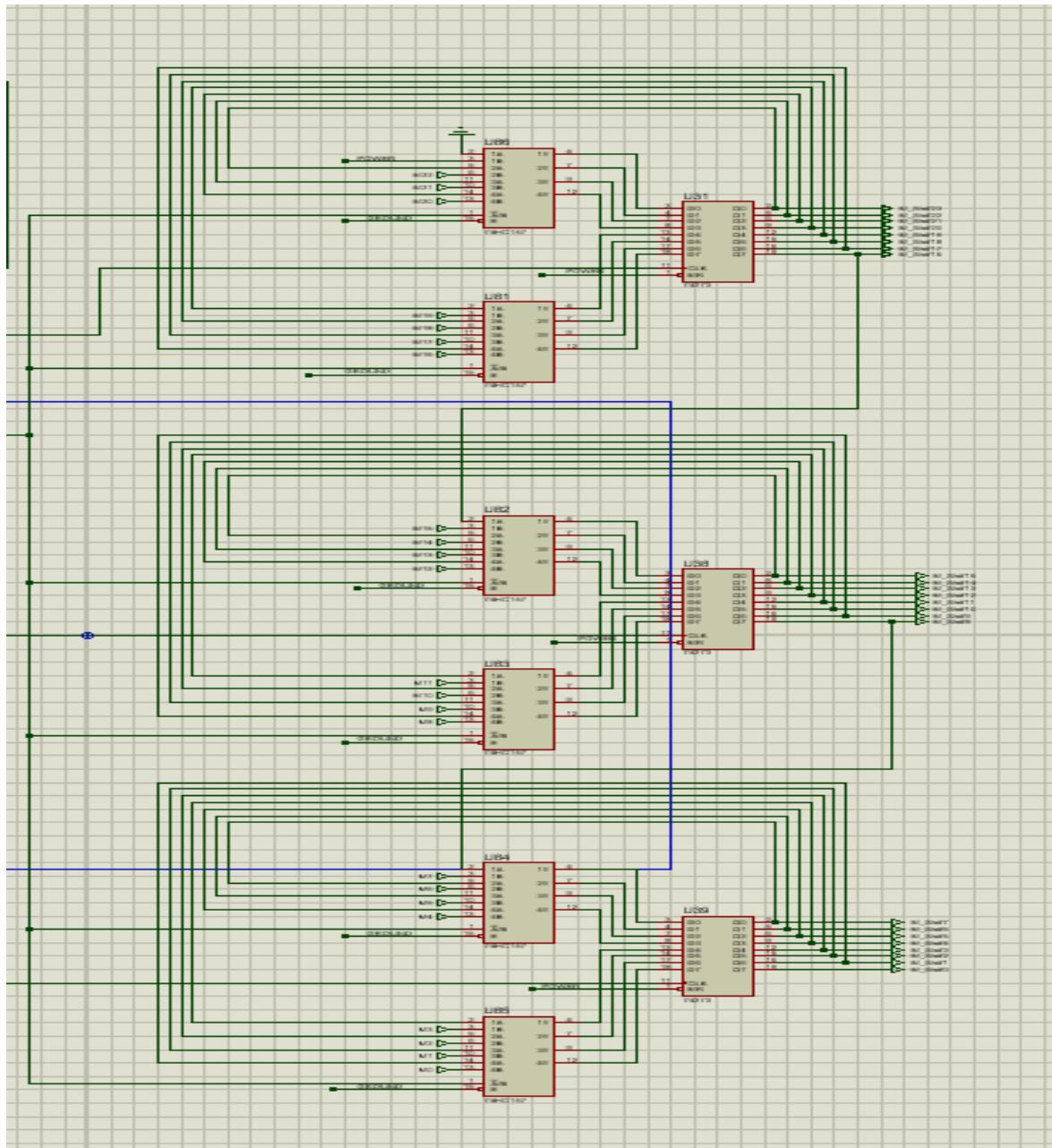


در نظر داشته باشید که برای این که بتوانیم در آینده سرزیز را تشخیص بدهیم، یک بیت صفر برای هر دو مانتیس در این مرحله نظر می‌گیریم. حال به پیاده سازی مدار داخلی شیفتدهنده خود باز می‌گردیم.

این شیفتدهنده در ابتدا برای یک بار باید مقدار ورودی را لود کند و در هر مرحله با یک کلک تا جایی که خروجی مدنظر آماده شود، مانتیس وارد شده را شیفت به راست بدهد. با توجه به گستردگی مدار، برای گزارش مدار را در دو بخش توضیح میدهیم. بخش اول شامل مربوط به عملیات بر روی نما خواهد بود. در این بخش باید در ابتدا، نما رو در یک رजیستر لود کرده و در هر مرحله تا زمانی که به صفر نرسیده بود، یک واحد از آن کم کنیم. مدار این بخش به صورت زیر خواهد بود :



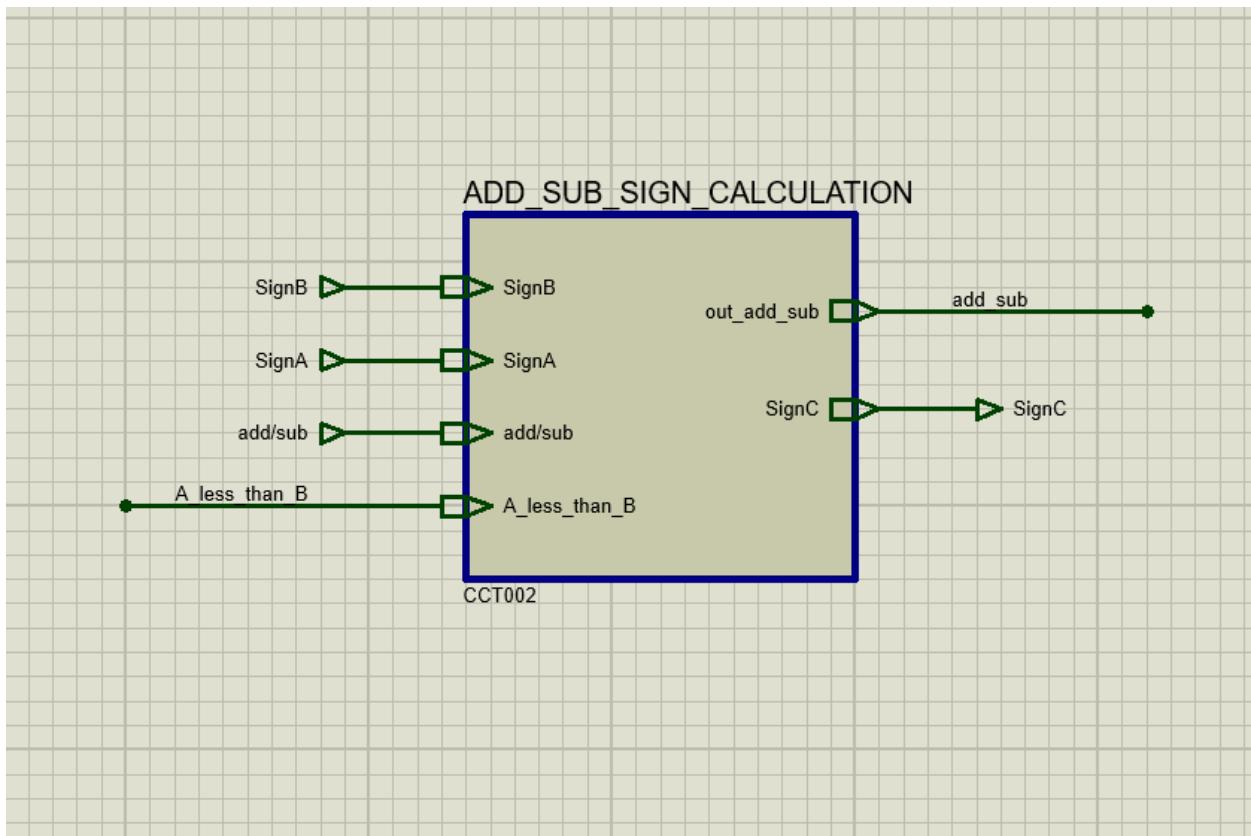
پس از طراحی بخش مربوط به نما، به طراحی قسمت شیفت مدار خواهیم رسید. در این بخش کافی است تا زمانی که کار مدار به پایان نرسیده بود، با هر کلاک مقدار مانتیس لود شده را یک واحد به راست شیفت بدهیم. برای عملیات شیفت دادن، میتوانیم از رجیستر آماده استفاده کنیم یا با سری کردن تعدادی رجیستر و دادن خروجی رجیستر قبلی به ورودی رجیستر مرحله بعد، مدار مربوط به شیفت را کامل کنیم. در پایین پیاده سازی این قسمت از مدار هم آمده است:



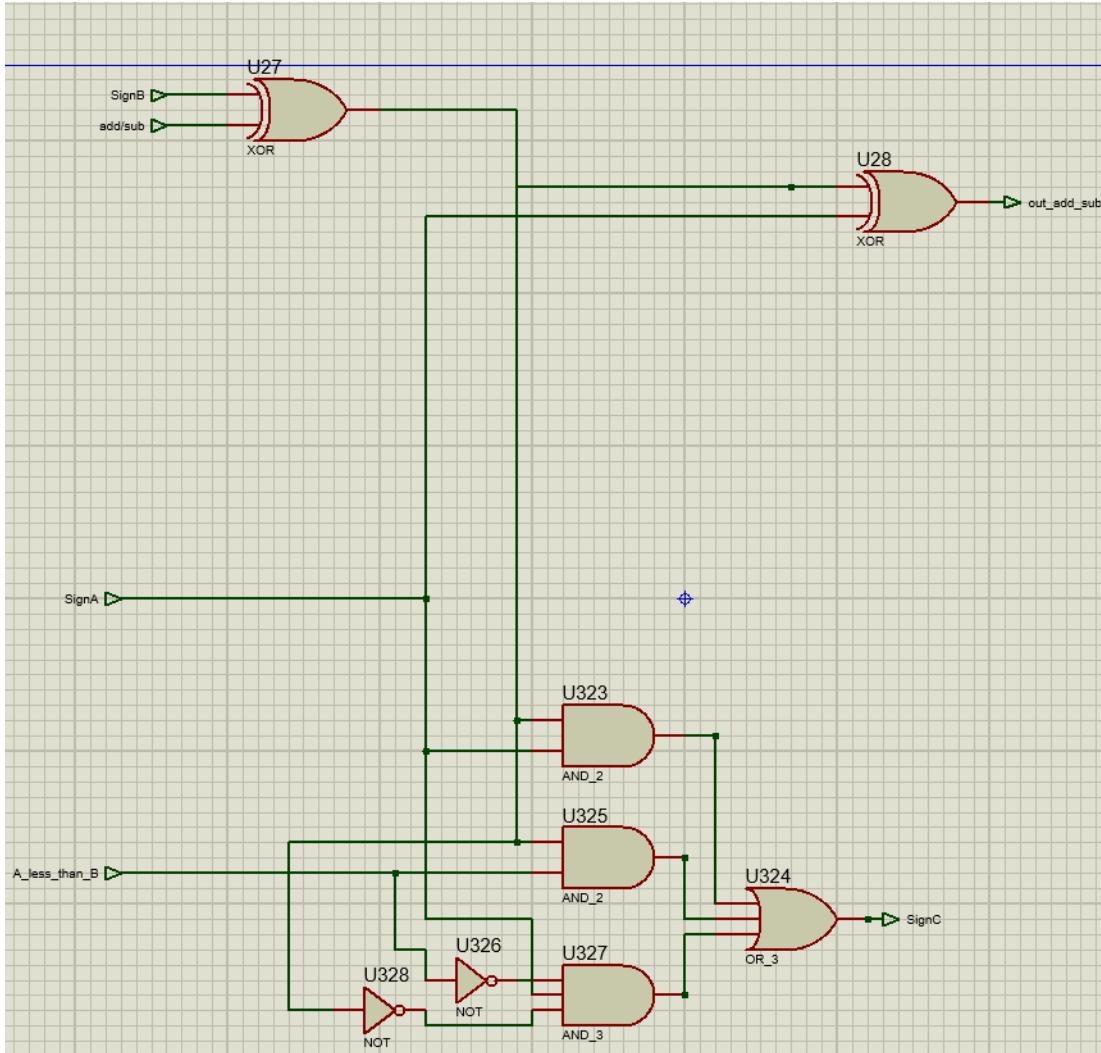
با این تفاسیر قسمت مربوط به عملیات‌های نما انجام شده است و وارد بخش دوم، یعنی عملیات مانتیس خواهیم شد.

## بخش محاسبه مانتیس:

پیش از هر چیز، تا این مرحله مقادیر مربوط به مانتیس‌ها به صورت کامل به دست آمده است. اکنون کافی است با توجه به علامت هر عملیات مدار و همین طور علامت هر دو عدد، تصمیم بگیریم که چه عملیاتی را انجام دهیم. در ابتدا باید تصمیم بگیریم که اکنون به کدام عملیات نیاز خواهیم داشت. به همین دلیل یک مژول طراحی می‌کنیم که وظیفه این کار را بر عهده دارد. ورودی‌های این مژول علامت دو عدد، عملیات و همین طور بیتی است که نشان می‌دهد که نمای  $A > B$  است. خروجی این مژول هم عملیات مدار و علامت جواب حاصل خواهد بود. شمای کلی مدار به صورت زیر خواهد بود:



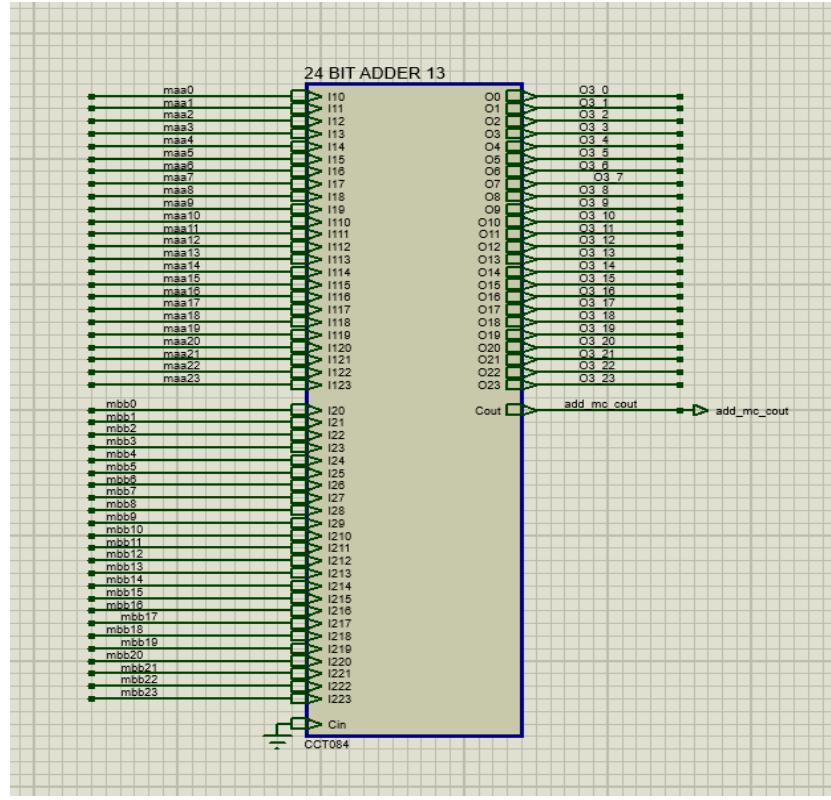
طراحی داخل این مدار به صورت زیر خواهد بود:



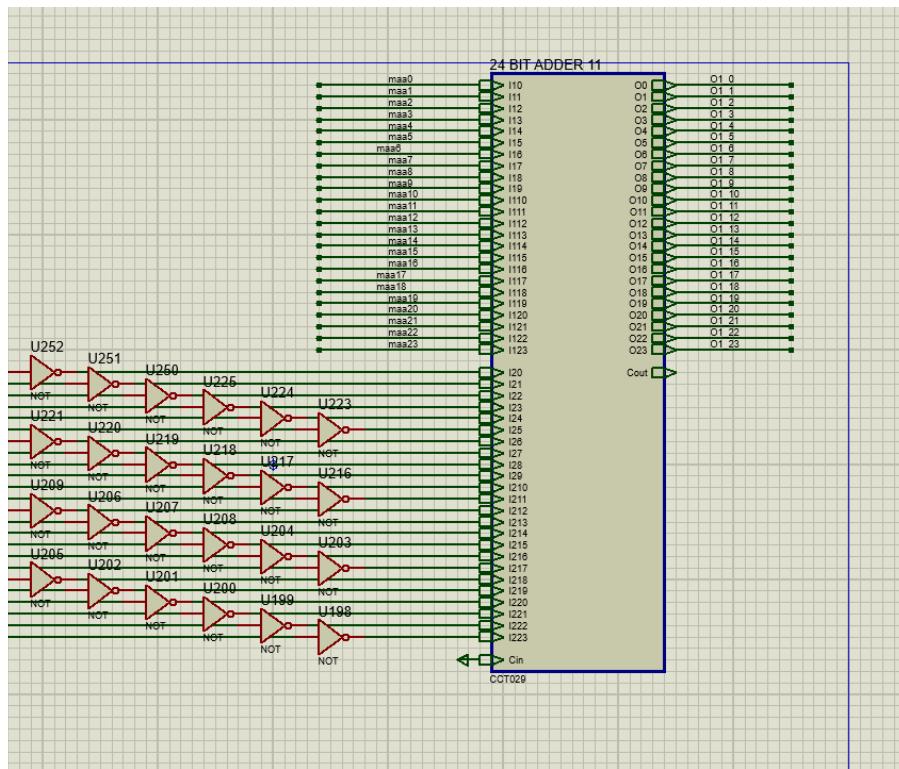
ابتدا با توجه به علامت عدد B و بیت عملیات، علامت عملیات نهایی را به دست می‌آوریم. میدانیم در صورتی تفriق رخ خواهد داد که یکی از این سه بیت برابر یک باشد. در غیر این صورت و با زوج بودن یکی از این دو بیت، عملیات جمع رخ خواهد داد.

حال به شرح بیت علامت جواب می‌پردازیم. در صورتی که هر دو عدد منفی بود، علامت جواب منفی خواهد شد. در صورتی که عدد شامل نمای بیشتر، علامت منفی داشت، حاصل به اجبار منفی خواهد بود. برای تشخیص این قسمت هم یک منطق ساده با استفاده از گیت‌های AND و OR کافی خواهد بود. اکنون با مشخص شدن سیگنال‌های عملیات و همین طور علامت خروجی، بخش مربوط به منطق مانتیس‌ها هم به راحتی قابل پیاده کردن خواهد بود. به این شرح که همه حالت‌های ممکن را به دست می‌آوریم و سپس با استفاده از عملیات واقعی مدار، خروجی را با MUX انتخاب می‌کنیم. در واقع یعنی هر سه حالت ممکن برای مانتیس، که شامل جمع مانیس A و B، تفriق A از B و برعکس خواهد بود.

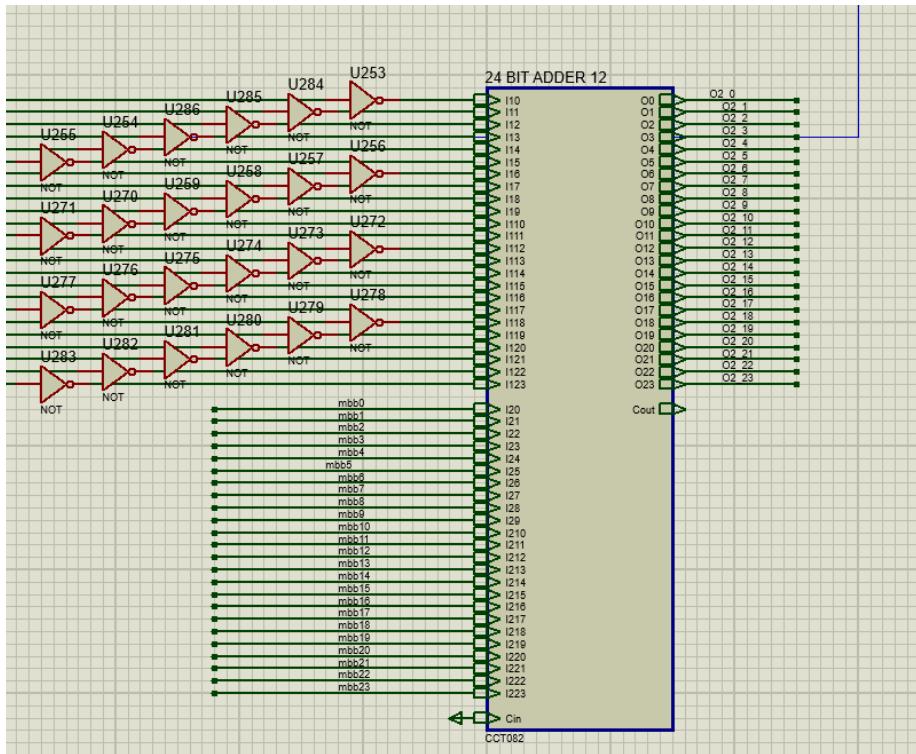
با توجه به تفاسیر گفته شده مدار مربوط به محاسبه ماتریس هم به صورت زیر خواهد بود:



جمع ماتریس‌ها  
1 Figure

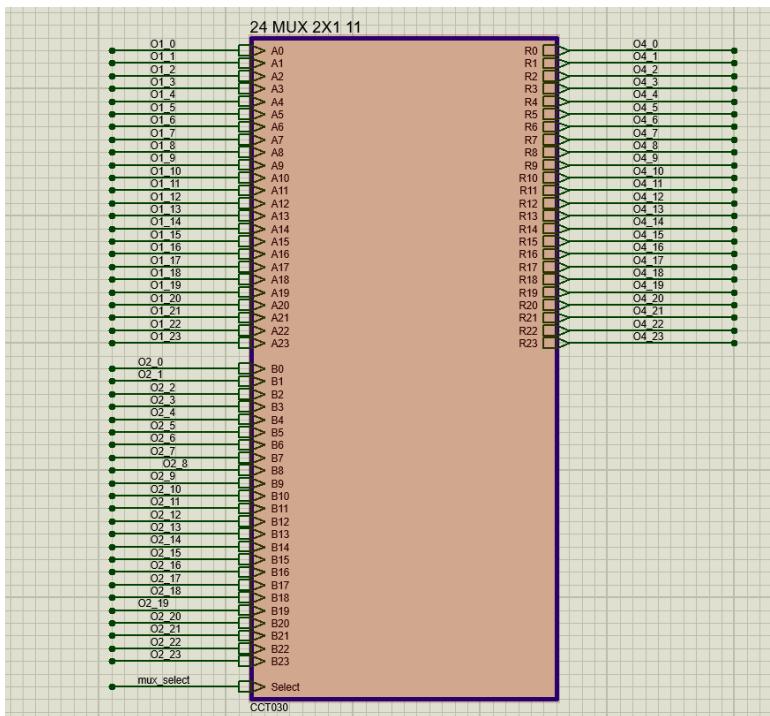


تفاضل A از B Figure

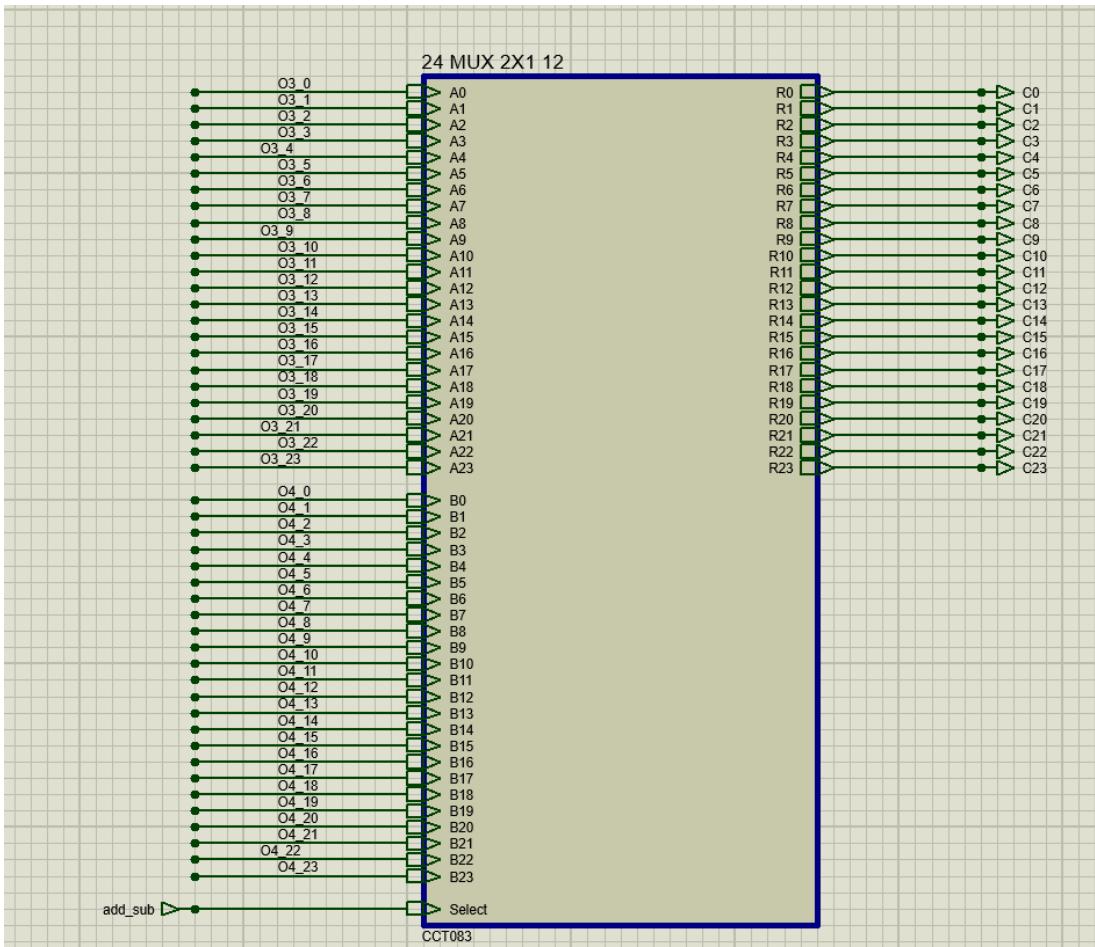


A 3 تفاضل باز Figure

حال با توجه به این که تفاضل هر دو مرحله را داریم، باید مقدار صحیح را با استفاده از MUX خروجی بدھیم. در مرحله اول، تفاضلی را انتخاب می‌کنیم که مقدار آن عددی مثبت باشد، یعنی عدد به دست آمده از این تفاضل منفی نباشد. پس از این مرحله نوبت به انتخاب بر اساس علامت عملیات است.



4 انتخاب تفاضل درست با توجه به بزرگتر بودن

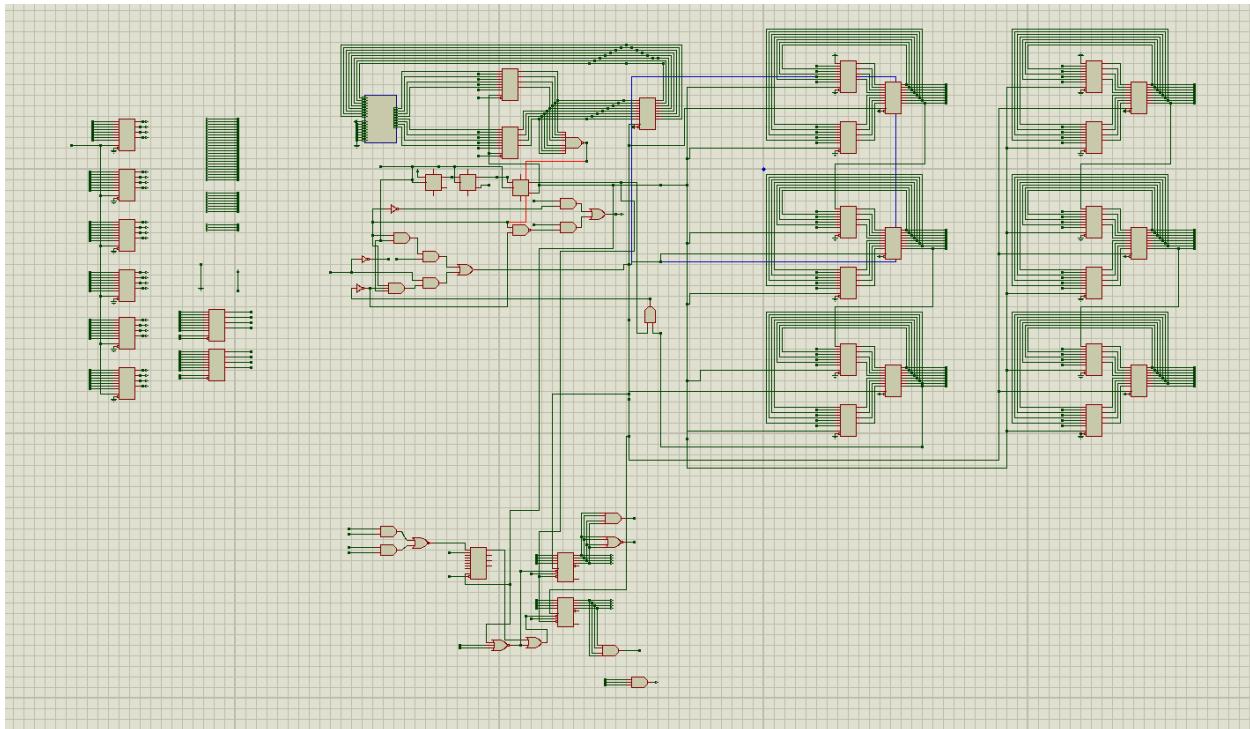


5 انتخاب بر اساس عملیات

حال با آماده شدن قسمت مانتیس، نوبت به نرمالایز کردن مدار خواهد رسید.

## نرم‌الایز کردن:

در این قسمت می‌بایست تا جایی مدار را به سمت چپ شیفت بدهیم که آخرین بیت سمت چپ برابر با ۱ باشد. با هر بار عملیات شیفت، باید مقدار نمای به دست آمده را با یک جمع کنیم. برای اینکار میتوانیم همانند قسمت شیفت مانتیس، از ترکیب کردن یک مدار شیفت دهنده با یک مدار برای افزایش نما استفاده می‌کنیم. از آنجایی که این مدار بسیار همانند قسمت شیفت مانتیس می‌باشد، از شرح کامل این قسمت در گزارش کار صرف نظر می‌کنیم و تنها عکس‌های مربوط به این قسمت را در این قسمت قرار می‌دهیم:



پس از این که این قسمت هم کامل شد، میتوانیم بگوییم که مدار ما به صورت کامل تشکیل شده است. اکنون نوبت به تست کردن این مدار می‌رسد.

## تست کردن مدار آزمایش:

برای تست این مدار، باید ورودی‌های مختلف با علامت‌های مختلف را به مدار بدهیم. با توجه به کامل بودن مدار در تست‌های انجام شده توسط تیم، از صحت مدار اطمینان کسب شده است. به همین دلیل در این قسمت به آوردن تنها یک مثال در این قسمت اکتفا می‌کنیم.

فرض کنید ورودی‌های مدار به این صورت باشند:

$$A = 0.45$$

$$B = 0.22$$

نمایش ۳۲ بیتی ممیز شناور این دو ورودی برابر است با:

$$\text{IEEE-754}(A) = 0\ 01111101\ 11001100110011001100110$$

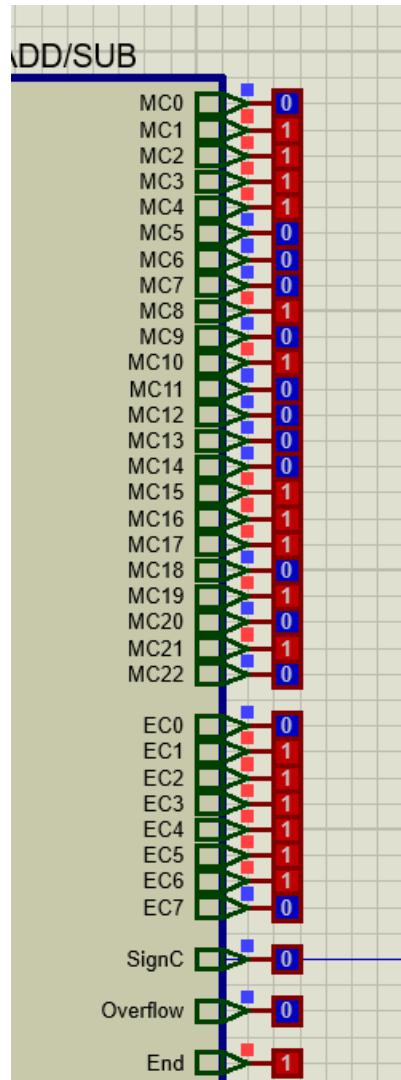
$$\text{IEEE-754}(B) = 0\ 01111100\ 11000010100011110101110$$

در اینجا میخواهیم دو عدد وارد شده را با هم جمع کنیم. این دو عدد را به عنوان ورودی به مدار می‌دهیم و سپس استارت را برابر با یک می‌کنیم و مقدار add/sub را برابر با صفر قرار می‌دهیم. توجه کنید که مدار به این صورت طراحی شده است که بیت استارت تا آخر عملیات باید برابر با یک باشد.  
می‌دانیم خروجی مربوط به این قسمت برابر است با:

$$C = A + B = 0.67$$

$$\text{IEEE-754}(C) = 0\ 01111110\ 01010111000010100011111$$

در صفحه بعد تصاویر مربوط به این عملیات آورده شده است.

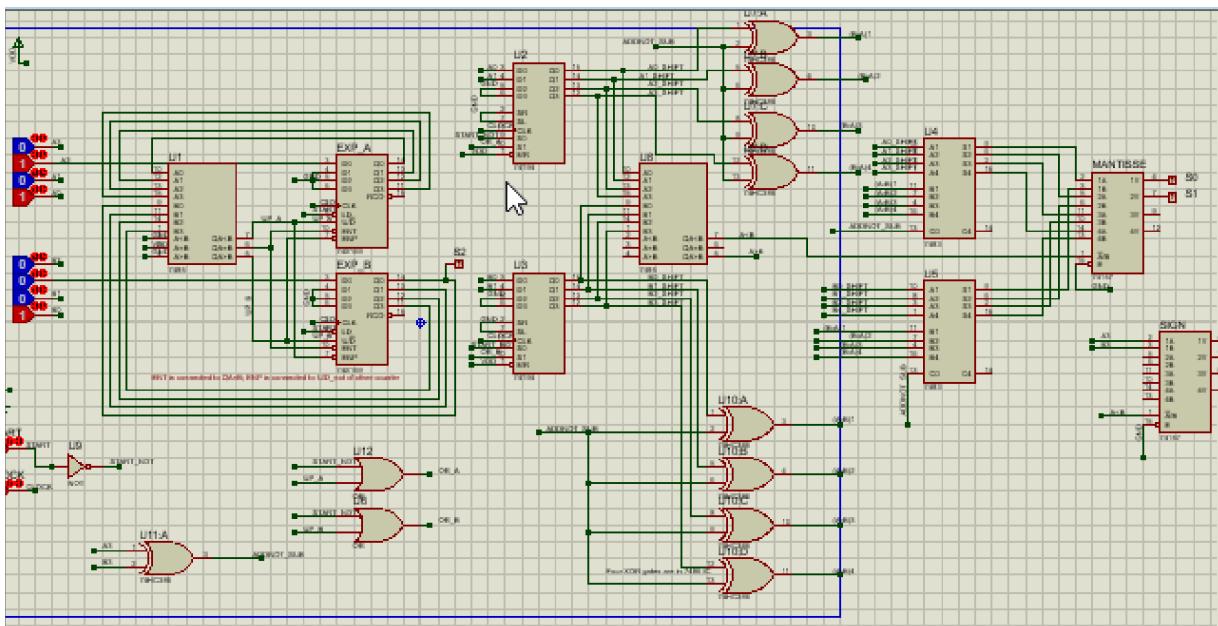


6 خروجی مدار به ازای ورود مقادیر گفته شده Figure

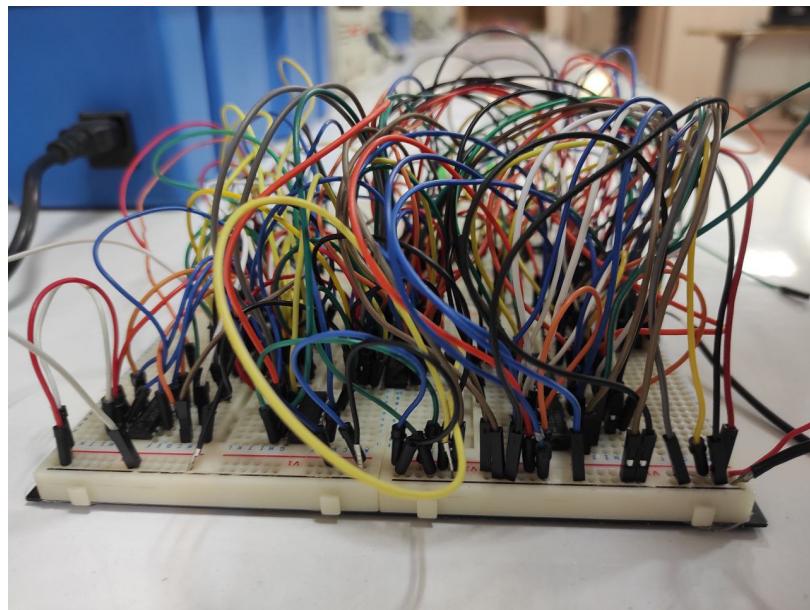
دقت شود تا زمانی که سیگنال End برابر با یک نشده است، خروجی صحیح برای مدار آماده نشده است. از طرفی چون مقدار این جمع در ۳۲ بیت جا می‌گرفت، مقدار Overflow برابر با صفر شد.

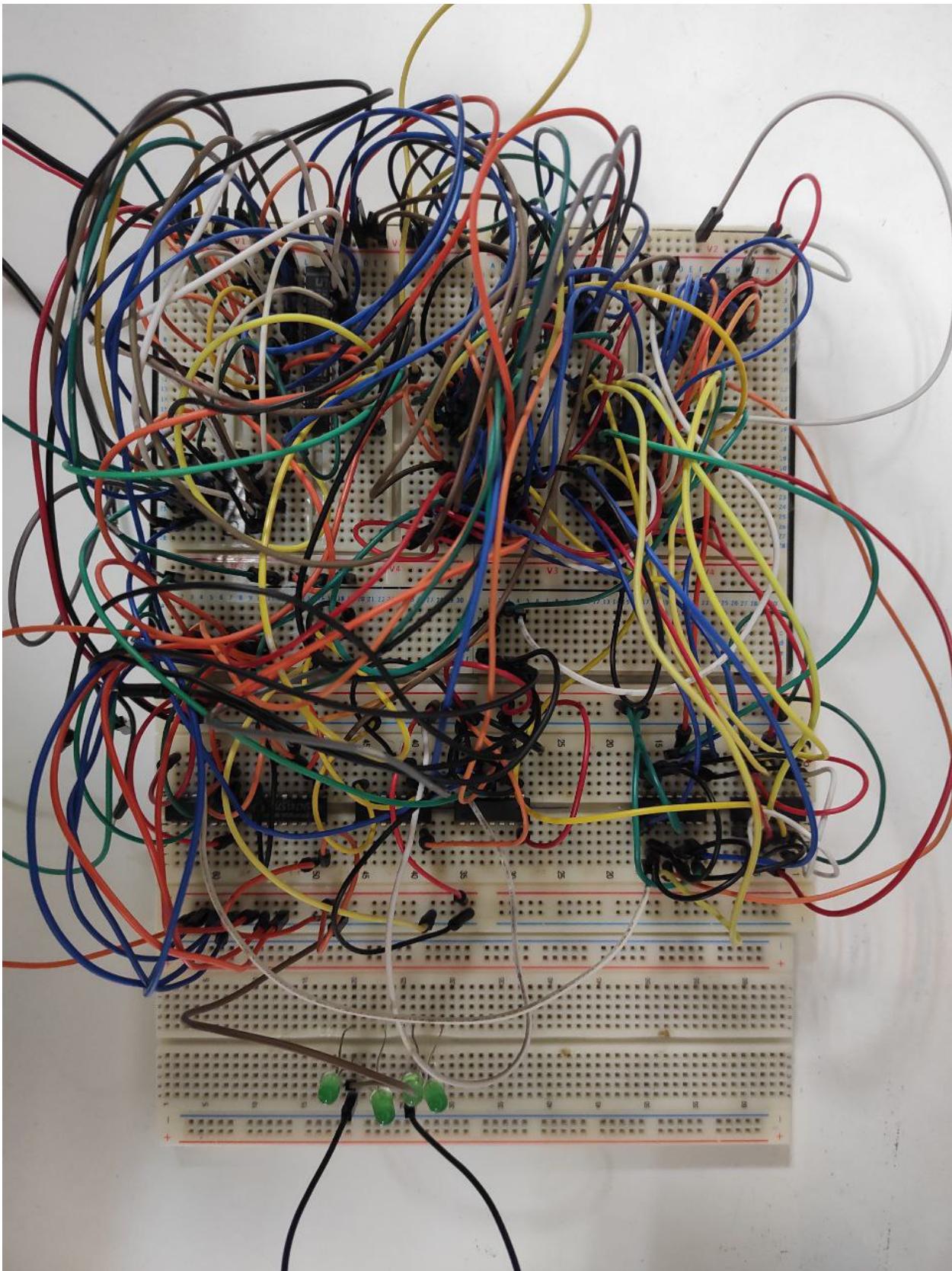
## بستن مدار روی بردبورد:

پس از اینکه مدار را برای ۳۲ بیت طراحی کردیم، نیاز داریم که آن را روی بردبورد پیاده‌سازی کنیم. از آنجایی که حجم مدار زیاد است، این مدار را ۴ بیت طراحی می‌کنیم. به این صورت که یک بیت را برای علامت در نظر می‌گیریم، یک بیت برای نما و دو بیت هم برای بخش مانتیس. به همین ترتیب نیاز پیدا می‌کنیم که یک مدار جدید برای ۴ بیت طراحی کنیم. مدار طراحی شده برای ۴ بیت به صورت زیر خواهد بود:



با توجه به مدار فوق که مجدد برای ۴ بیت طراحی کردیم، اکنون بخش پیاده‌سازی روی بردبورد را مطابق با مدار انجام می‌دهیم.





پایان