

---

## گزارشکار آزمایش چهارم – آزمایشگاه معماری کامپیوتر



### اعضای گروه:

امیراردلان دهقانپور ۴۰۱۱۰۵۹۰۱

رادین شاه دائی ۴۰۱۱۰۶۰۹۶

باربد شهرآبادی ۴۰۱۱۰۶۱۲۵

---

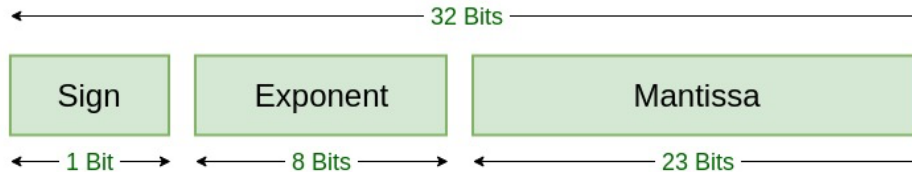
### شرح آزمایش:

در این آزمایش به طراحی یک واحد جمع و تفریق کننده ممیز شناور می‌پردازیم. برای پیاده‌سازی این آزمایش، از ابزار proteus استفاده شده است و همچنین مدار را مطابق استاندارد IEEE754 برای اعداد ممیز شناور ۳۲ بیتی طراحی کردیم. در ادامه به توضیح این استاندارد می‌پردازیم. پس از طراحی این مدار در ابزار پروتئوس، آن را با تست کیس‌های متفاوت تست می‌کنیم و صحت کارکرد مدار را نشان می‌دهیم.

## فرمت اعداد ممیز شناور ۳۲ بیتی:

عدد ممیز شناور ۳۲ بیتی در استاندارد IEEE 754 شامل سه بخش اصلی است:

- (۱) بیت علامت یا sign bit: ۱ بیت
- (۲) بیت‌های نما یا exponent bits: ۸ بیت
- (۳) بیت‌های کسری یا mantissa bits: ۲۳ بیت



شکل ۱

اگر بیت علامت 0 باشد، عدد مثبت و اگر بیت علامت 1 باشد، عدد منفی است. نما برای تعیین مکان اعشار استفاده می‌شود، اما به صورت "بایاس‌شده" ذخیره می‌گردد. در حالت تک‌دقتی، بایاس (bias) برابر با 127 است. مانتیسا نمایانگر قسمت کسری عدد است. در حالت نرمال، مانتیسا با یک عدد 1 فرضی در ابتدا نمایش داده می‌شود که ذخیره نمی‌شود.

با توجه به اینکه در صورت شرح آزمایش گفته شده بود که اعداد نرمال باید پیاده‌سازی شوند، تنها نیاز به پیاده‌سازی arithmetic این نوع اعداد ممیز شناور خواهیم بود. با توجه به توضیحات ارائه شده، یک عدد ۳۲ بیتی ممیز شناور نرمال در استاندارد IEEE754 مقدار زیر را به خود می‌گیرد:

$$(-1)^s \times 1.F \times 2^{E-127}$$

که s همان بیت sign می‌باشد. F نشانگر بیت‌های mantissa است و همچنین E نشانگر بیت‌های exponent می‌باشد.

## جمع و تفریق اعداد ممیز شناور ۳۲ بیتی:

در ادامه به arithmetic جمع و تفریق این اعداد می‌پردازیم. در این بخش صرفاً به الگوریتم کلی جمع و تفریق اعداد ممیز شناور می‌پردازیم و پیاده‌سازی دقیق این مراحل در ابزار proteus با استفاده از تراشه‌های متنوع را در بخش‌های بعدی گزارشکار توضیح می‌دهیم. این الگوریتم شامل ۳ بخش می‌باشد.

**بخش اول) هم‌تراز کردن اعداد:** در مرحله اول باید اعداد را به اصطلاح هم‌تراز یا align کنیم. برای جمع یا تفریق، باید نماهای دو عدد را برابر کنیم. عددی که نما کوچکتری دارد باید با تغییر مانتیسا به بالا هم‌تراز شود. به منظور این کار ابتدا بزرگترین نمای بین دو عدد را پیدا می‌کنیم. سپس تفاوت نماها را محاسبه می‌کنیم و در نهایت مانتیسی عددی که نما کوچکتری دارد را به اندازه تفاوت نماها به راست شیفت دهید و نما را برابر با نمای بزرگتر تنظیم کنید.

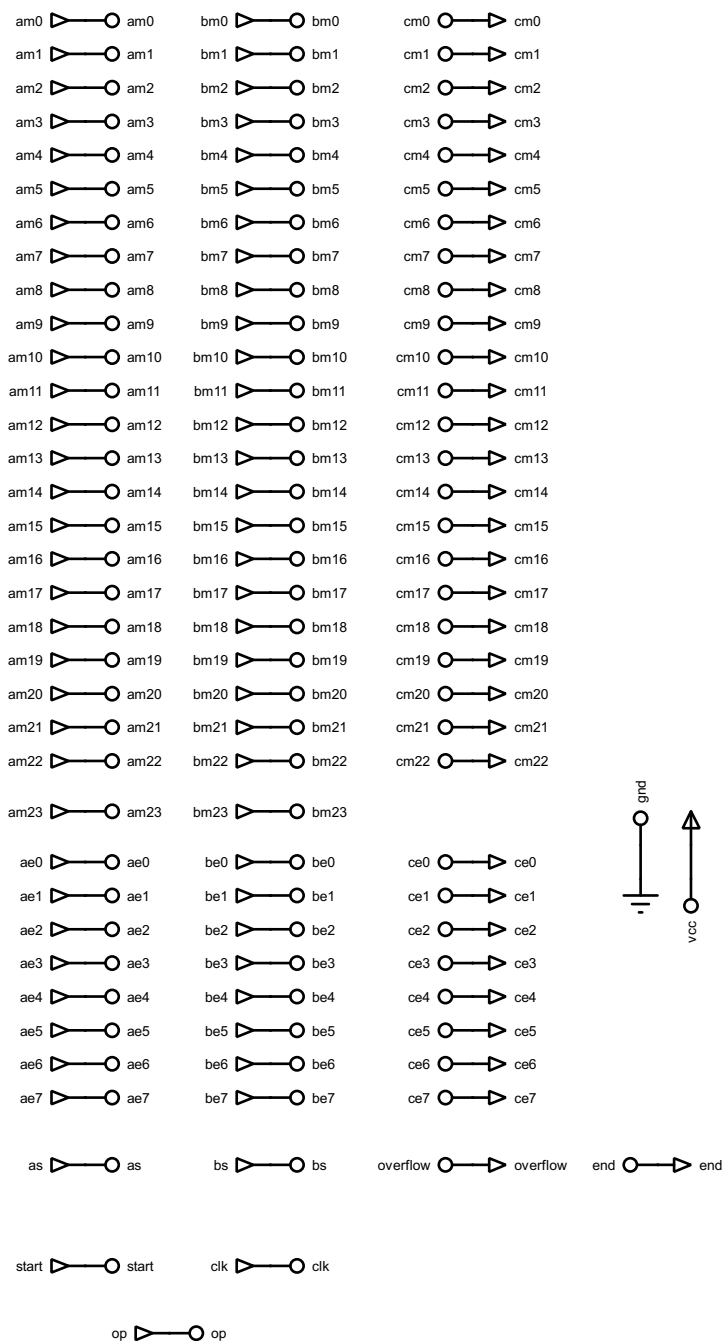
**بخش دوم) جمع یا تفریق مانتیساها:** پس از هم‌تراز کردن اعداد، مانتیسی‌ها (که مقدار 1.F هستند) را با یکدیگر جمع یا تفریق می‌کنیم. دقت کنید که در جمع یا تفریق کردن باید مقدار بیت sign دو عدد را نیز لحاظ کنیم.

**بخش سوم) نرمال‌سازی نتیجه:** بعد از جمع یا تفریق، ممکن است نیاز به نرمال‌سازی نتیجه داشته باشید، اگر مانتیسا بزرگتر یا مساوی 2 شد، آن را به راست شیفت دهید و 1 به نما اضافه کنید. اگر مانتیسا کوچکتر از 1 شد، آن را به چپ شیفت دهید و 1 از نما کم کنید تا زمانی که مانتیسا به بازه نرمال برسد.

## پیاده‌سازی در شبیه‌ساز Proteus:

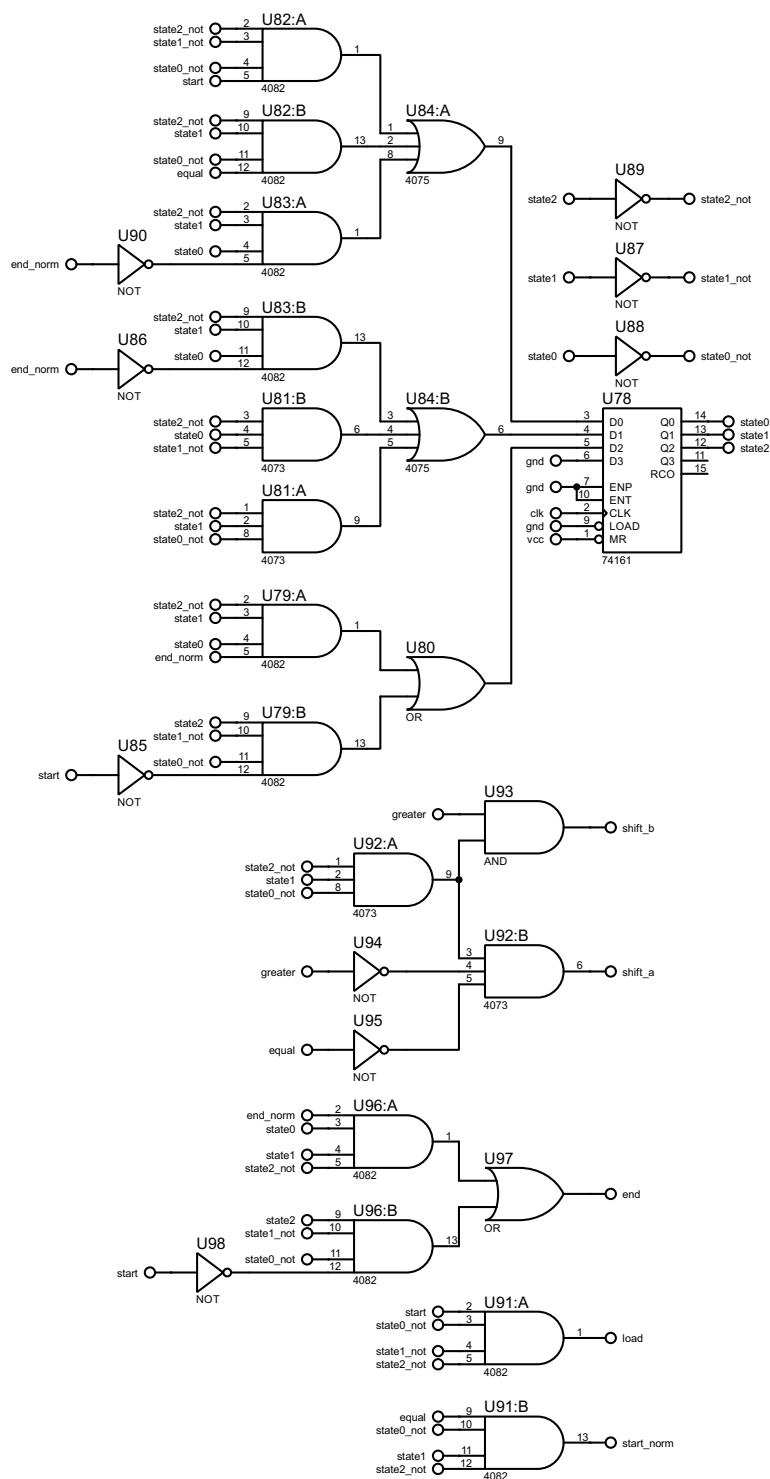
در این بخش به شرح پیاده‌سازی بخش‌های مختلف proteus می‌پردازیم.

(۱) ورودی و خروجی: ورودی‌های مدار شامل دو عدد ۳۲ بیتی ممیز شناور در استاندارد IEEE 754، سیگنال شروع عملیات (start)، سیگنال opcode برای تعیین نوع عملیات (جمع یا تفریق) و سیگنال ساعت (clock) می‌باشد. خروجی‌های مدار شامل عدد ۳۲ بیتی نتیجه عملیات، سیگنال overflow در صورت رخ دادن سرریز و سیگنال end برای نشان دادن پایان عملیات می‌باشد.



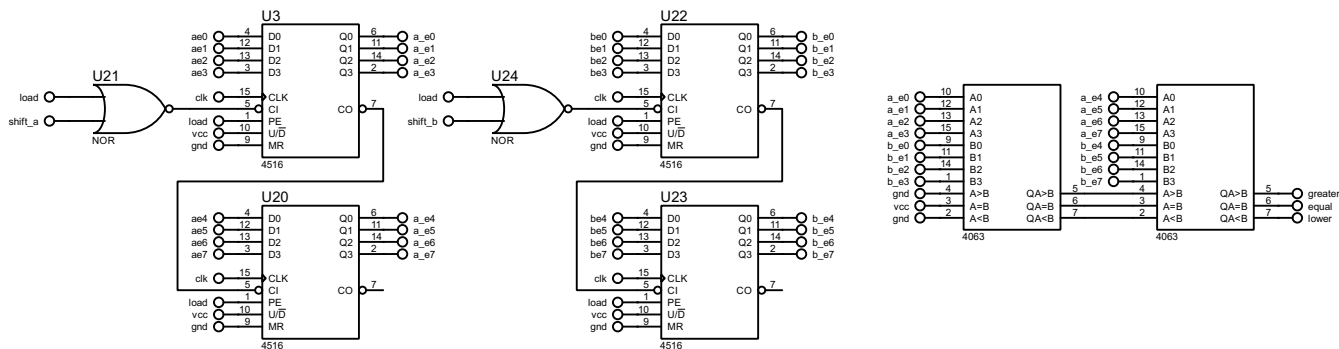
شکل ۲

۲) واحد کنترلی: واحد کنترل (Control Unit) در مدار ترتیبی جمع و تفریق اعداد ممیز شناور ۳۲ بیتی مسئول مدیریت و هماهنگی مراحل مختلف عملیات است. این واحد با دریافت سیگنال‌های ورودی مانند start و opcode، مراحل مختلفی از جمله هم‌تراز کردن اعداد، جمع یا تفریق مانتیساها، و نرمال‌سازی نتیجه را کنترل می‌کند. واحد کنترل ابتدا اعداد را هم‌تراز کرده، سپس عملیات جمع یا تفریق را بر اساس بیت‌های علامت انجام می‌دهد و در نهایت نتیجه را نرمال‌سازی می‌کند. در صورت بروز سرریز (overflow)، سیگنال مربوطه را فعال کرده و پس از اتمام عملیات، سیگنال end را روشن می‌کند.

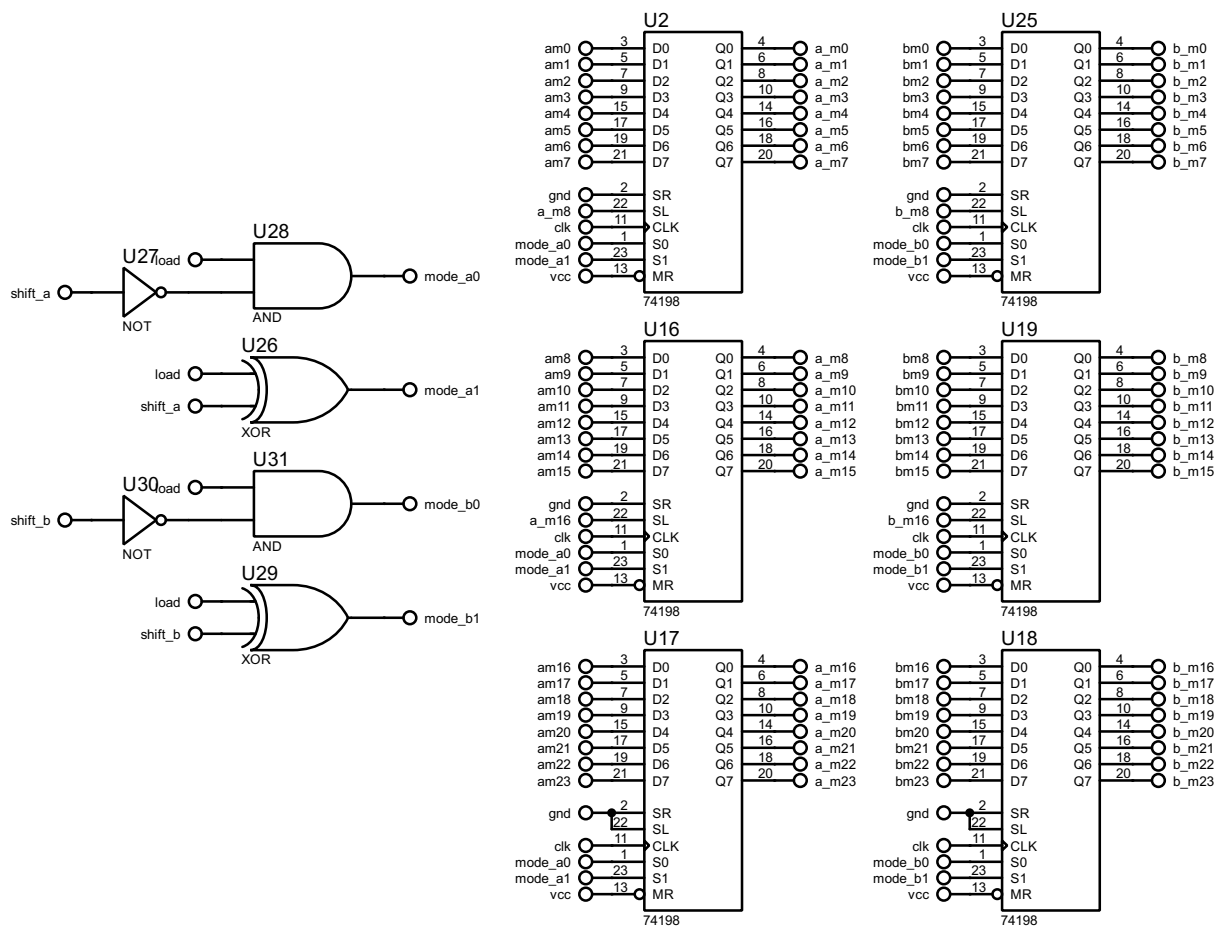


شکل 3

۳) واحد هم‌تراز سازی: یکی از بخش‌های مدار وظیفه مقایسه نماهای دو عدد ممیز شناور و محاسبه اختلاف آن‌ها را دارد. در این بخش، ابتدا نماهای دو عدد ورودی با هم مقایسه می‌شوند تا مشخص شود کدام نما بزرگتر است. سپس، اختلاف بین دو نما محاسبه می‌شود. این اختلاف برای هم‌تراز کردن ممیزها استفاده می‌شود؛ به این صورت که ممیزی عددی که نمای کوچکتری دارد، به اندازه این اختلاف به راست شیفت داده می‌شود تا نماهای دو عدد برابر شوند و عملیات جمع یا تفریق به درستی انجام گیرد.

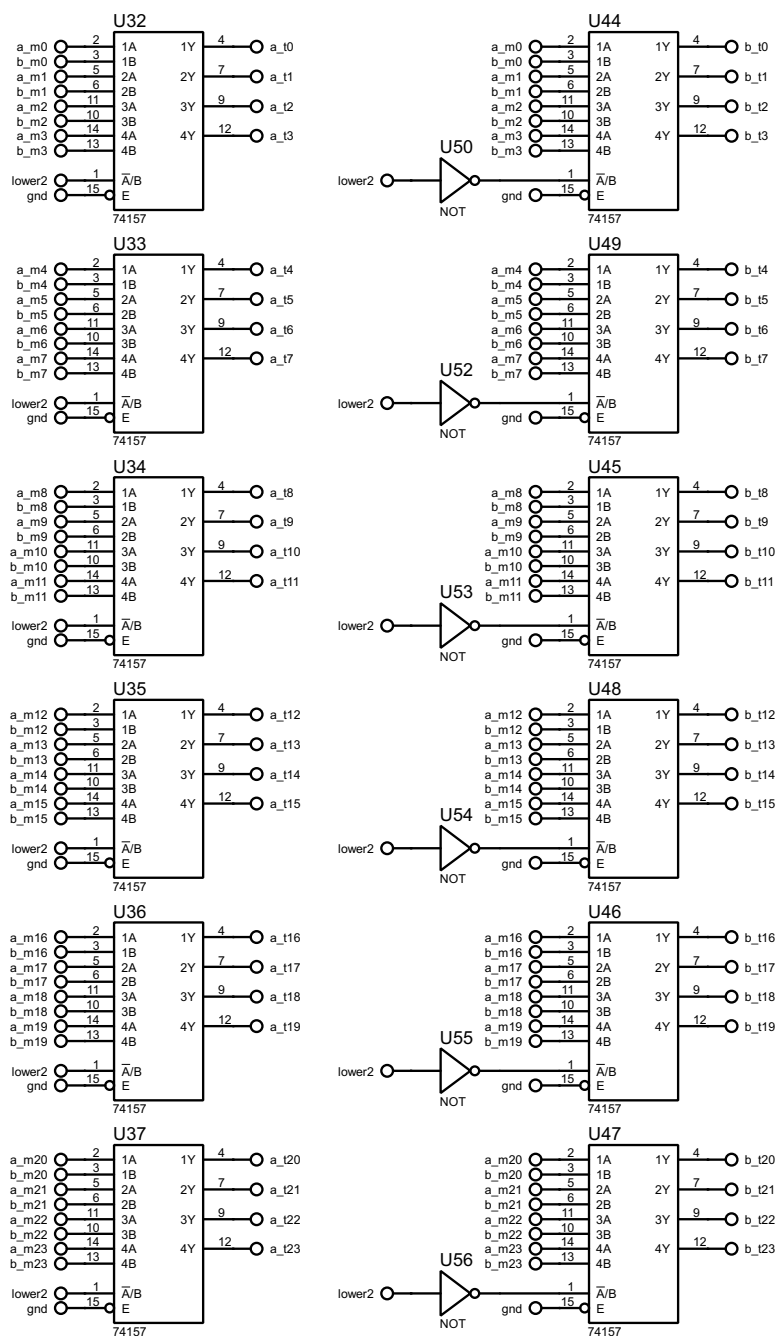


شکل ۴

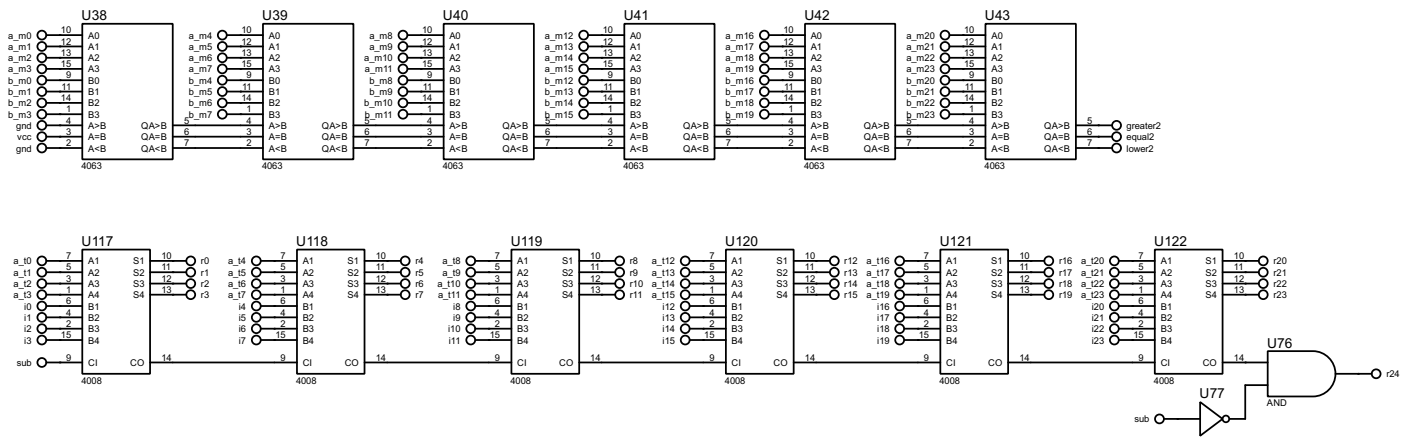


شکل ۵

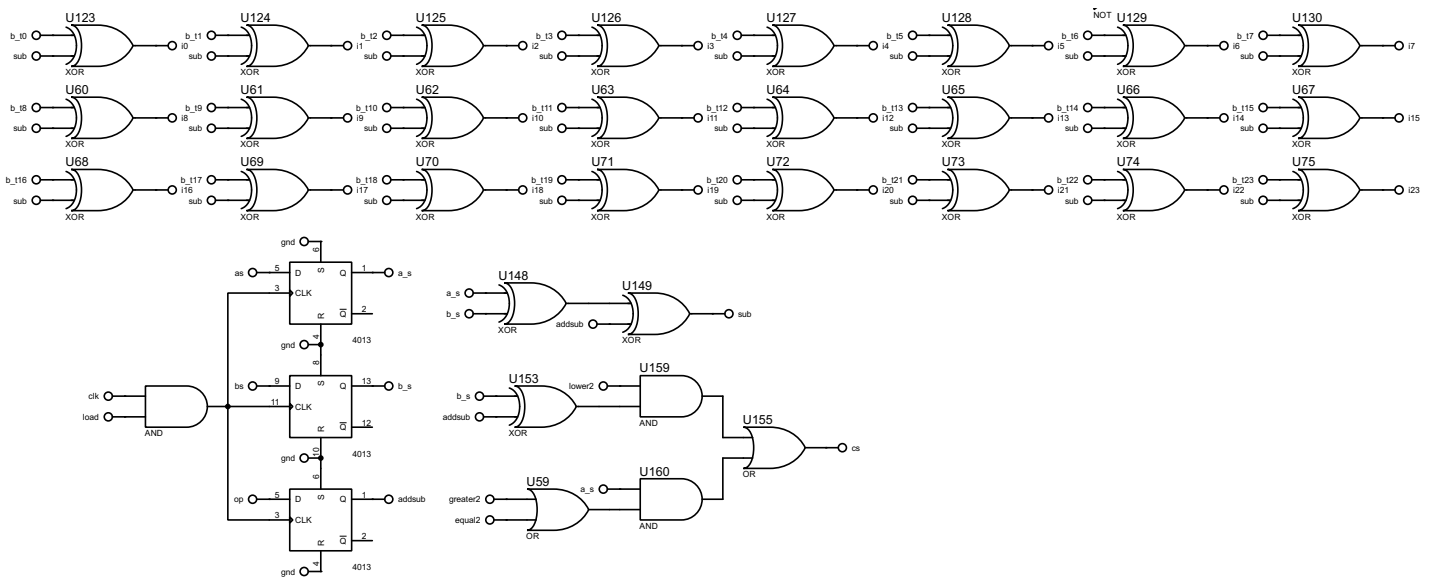
۴) **واحد جمع و تفریق:** بخشی از مدار وظیفه جمع و تفریق مانتیسه‌های دو عدد ممیز شناور را به عهده دارد. پس از هم‌تراز شدن نماها، این بخش عملیات جمع یا تفریق را بر روی مانتیسه‌های دو عدد انجام می‌دهد. برای این کار، ابتدا بر اساس سیگنال opcode که نوع عملیات را مشخص می‌کند (جمع یا تفریق)، عملیات مناسب را انتخاب می‌کند. اگر بیت‌های علامت دو عدد مشابه باشند (هر دو ۰ یا هر دو ۱)، مانتیسه‌ها جمع می‌شوند؛ در این حالت، نتیجه نهایی به صورت مجموع دو مانتیسه به دست می‌آید. اگر بیت‌های علامت متفاوت باشند (یکی ۰ و دیگری ۱)، مانتیسه‌ای عددی که دارای علامت منفی است، از مانتیسه‌ای عدد مثبت تفریق می‌شود. در این وضعیت، نتیجه نهایی به صورت تفاضل دو مانتیسه محاسبه می‌شود. این واحد با استفاده از مدارهای منطقی و جمع‌کننده‌های دو بیتی، عملیات جمع و تفریق را انجام می‌دهد و نتیجه را برای مرحله بعدی (نرمال‌سازی) آماده می‌کند.



شکل 6

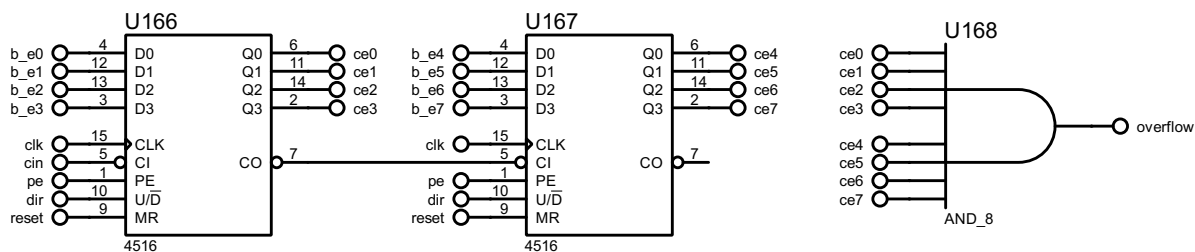


شکل 7

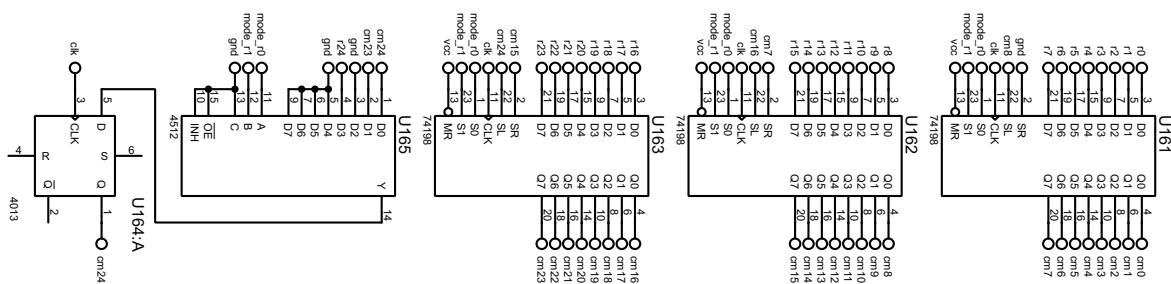


شکل 8

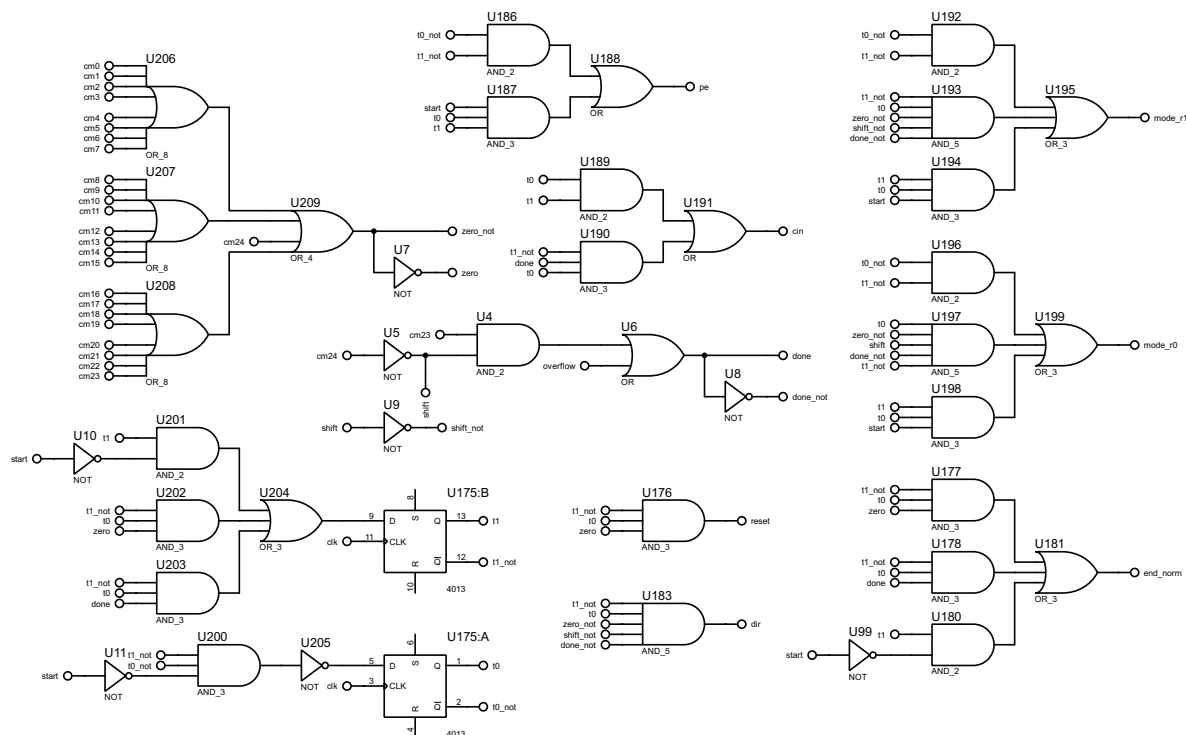
**(۵) واحد نرمال سازی:** بخشی از مدار مسئول نرمال سازی نتیجه عملیات جمع و تفریق مانتیساهاى دو عدد ممیز شناور است. پس از انجام جمع یا تفریق، ممکن است مانتیسا نیاز به نرمال سازی داشته باشد تا در دامنه معتبر قرار گیرد. در این بخش، ابتدا بررسی می شود که آیا مانتیسا بزرگتر یا مساوی 2 است. اگر اینطور باشد، مانتیسا به اندازه یک بیت به راست شیفت می شود و مقدار 1 به نما اضافه می شود تا عدد نرمال شود. اگر مانتیسا کوچکتر از 1 باشد، مانتیسا به اندازه یک بیت به چپ شیفت داده می شود و مقدار 1 از نما کاسته می شود تا مانتیسا به بازه نرمال برسد. این واحد با استفاده از شیفت کننده ها و مقایسه کننده ها، مانتیسا را به درستی نرمال کرده و نتیجه را برای مرحله پایانی آماده می کند.



شکل 9



شکل 10



شکل 11



## توضیحات کلی:

مدار ترتیبی جمع و تفریق اعداد ممیز شناور ۳۲ بیتی شامل چندین بخش اصلی است که هر کدام وظیفه خاصی را در فرآیند پردازش اعداد ممیز شناور بر عهده دارند. این بخش‌ها شامل واحد هم‌تراز کردن، واحد جمع و تفریق، و واحد نرمال‌سازی است. پیاده‌سازی این مدار در نرم‌افزار Proteus با استفاده از تراشه‌ها و بلوک‌های منطقی مختلف انجام می‌شود. در ادامه، به توضیح این بخش‌ها و تراشه‌های مورد استفاده در Proteus خواهیم پرداخت.

### بخش اول: هم‌تراز کردن اعداد

در این بخش، واحد هم‌تراز کردن وظیفه مقایسه نماهای دو عدد ممیز شناور را بر عهده دارد. این واحد با استفاده از تراشه‌های مقایسه‌کننده (مثل 7485) و مدارهای محاسباتی، بزرگترین نما را بین دو عدد تعیین کرده و اختلاف نماها را محاسبه می‌کند. سپس، مانتیسای عدد با نمای کوچکتر به اندازه این اختلاف به راست شیفت داده می‌شود تا نماهای دو عدد برابر شوند. در Proteus، این فرآیند به کمک بلوک‌های منطقی و رجیسترهای شیفت پیاده‌سازی می‌شود که به تنظیم صحیح مانتیساها کمک می‌کند.

### بخش دوم: جمع یا تفریق مانتیساها

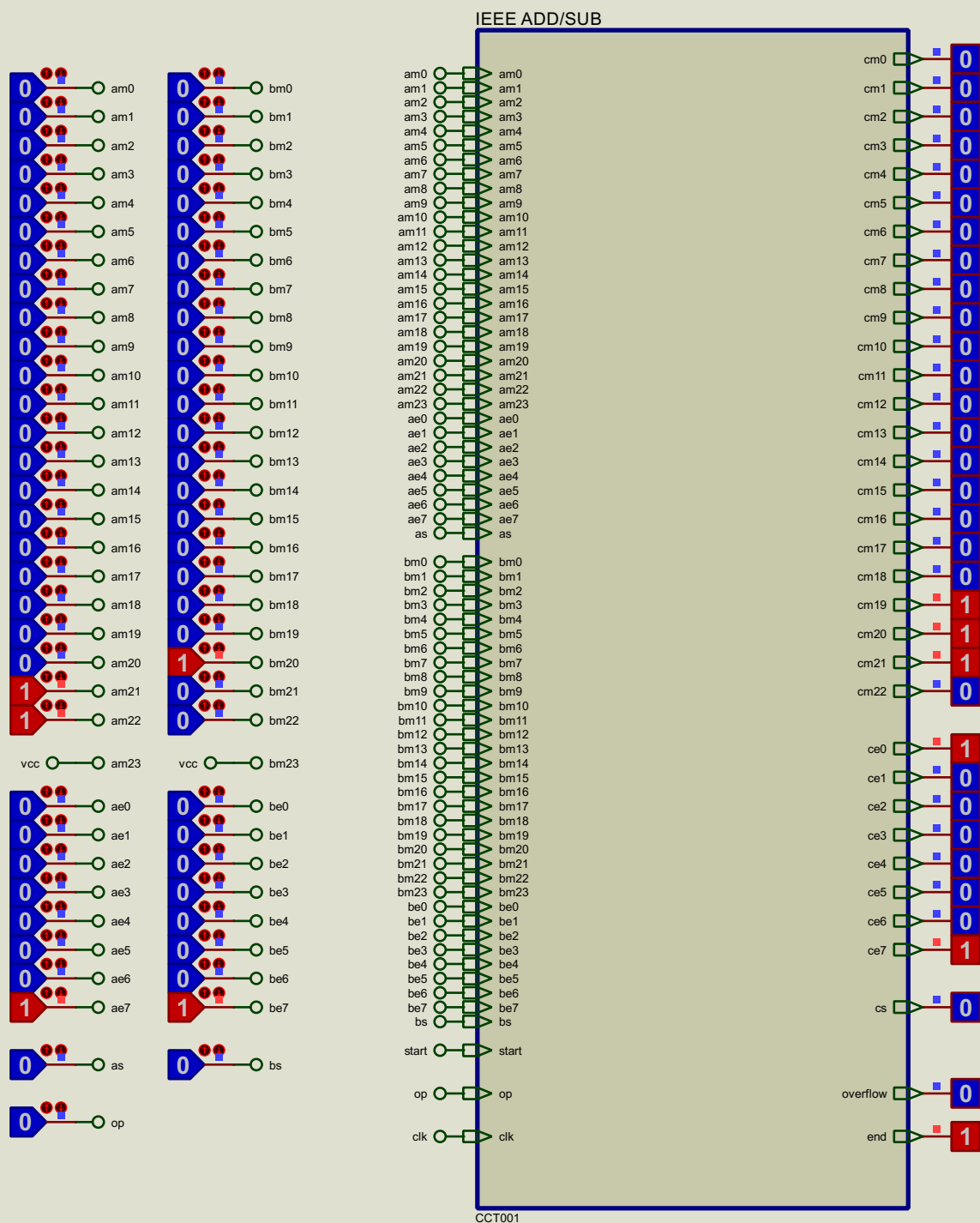
پس از هم‌تراز کردن، مانتیساها باید جمع یا تفریق شوند. این عملیات با استفاده از جمع‌کننده‌های بیتی و مدارهای منطقی (مثل 7483 برای جمع‌کننده‌های 4 بیتی) انجام می‌شود. اگر بیت‌های علامت دو عدد مشابه باشند، مانتیساها جمع می‌شوند و اگر متفاوت باشند، مانتیسای کوچک‌تر از مانتیسای بزرگ‌تر تفریق می‌شود. در Proteus، این بخش با استفاده از جمع‌کننده‌های کامل و دقت بالا برای پردازش صحیح مانتیساها پیاده‌سازی می‌شود.

### بخش سوم: نرمال‌سازی نتیجه

واحد نرمال‌سازی به تنظیم مانتیسا و نما پس از جمع یا تفریق می‌پردازد. اگر مانتیسا بزرگتر یا مساوی 2 باشد، به اندازه یک بیت به راست شیفت می‌شود و مقدار 1 به نما اضافه می‌شود. اگر مانتیسا کوچکتر از 1 باشد، به اندازه یک بیت به چپ شیفت داده می‌شود و مقدار 1 از نما کاسته می‌شود. در Proteus، این بخش به کمک بلوک‌های شیفت‌کننده و مقایسه‌کننده‌های بیتی پیاده‌سازی می‌شود تا مانتیسا به بازه نرمال برسد.

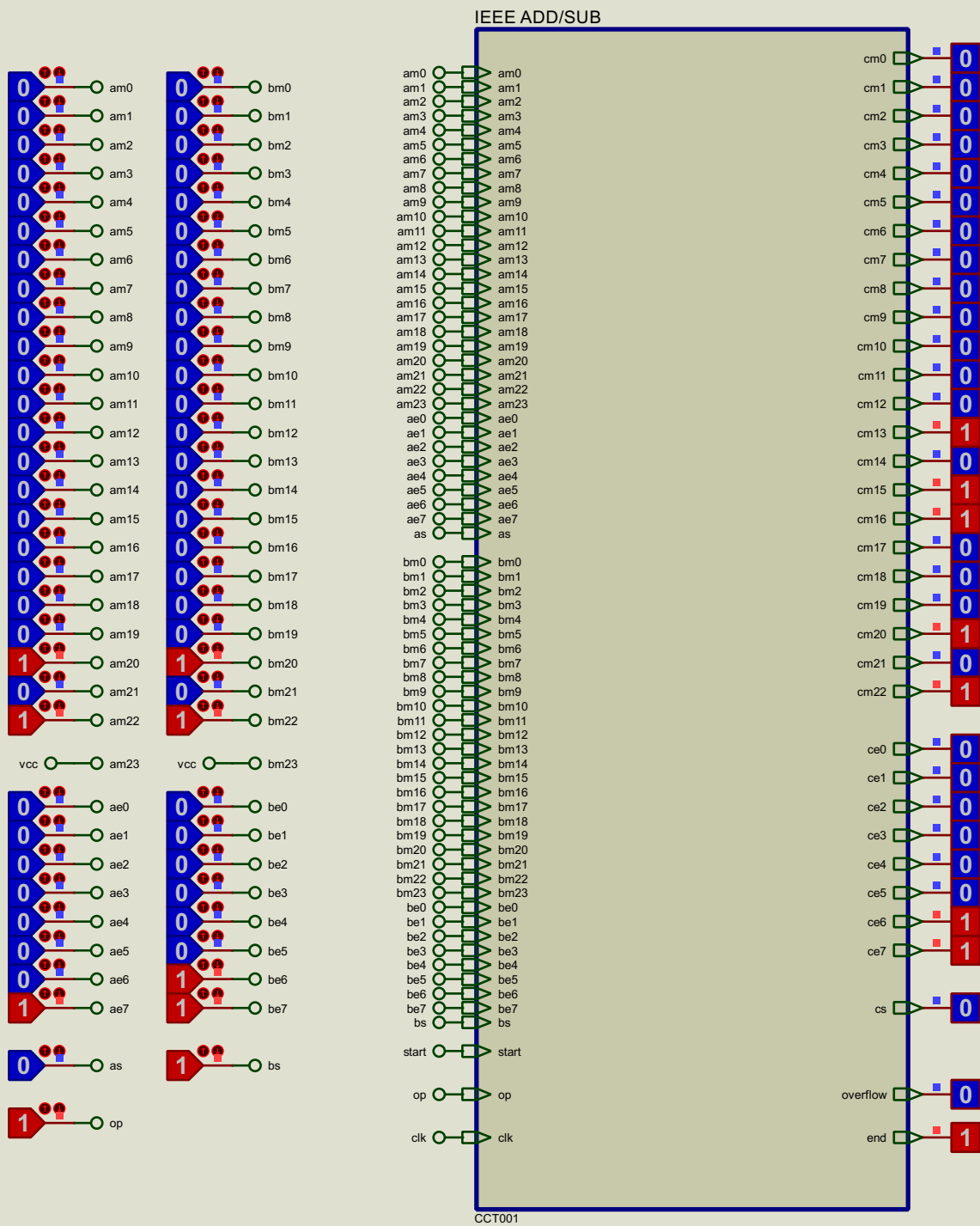
### تراشه‌های استفاده شده در Proteus

برای پیاده‌سازی این مدار در Proteus، از تراشه‌ها و بلوک‌های مختلفی استفاده می‌شود. تراشه‌های مقایسه‌کننده، جمع‌کننده‌های بیتی، و شیفت‌کننده‌ها از جمله اجزای کلیدی هستند. این تراشه‌ها با هم همکاری کرده و واحد کنترل را برای هماهنگی بین مراحل مختلف فراهم می‌کنند. در Proteus، این تراشه‌ها به صورت بلوک‌های منطقی درون مدارات طراحی شده و به یکدیگر متصل می‌شوند تا عملیات جمع و تفریق به درستی انجام شود.



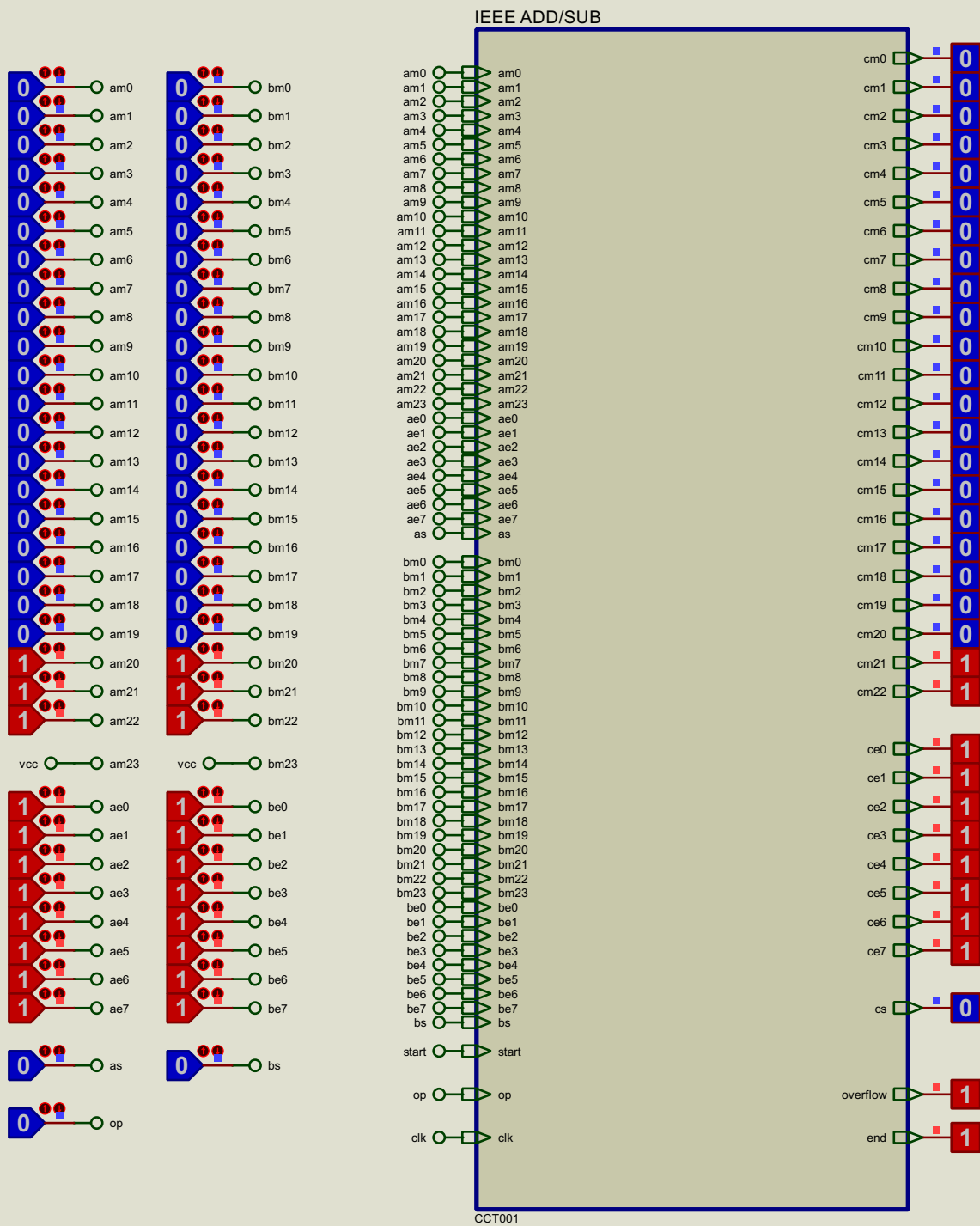
شکل 12

تست جمع دو عدد



شکل 13

تست تفریق دو عدد



شکل 14

تست overflow