

# آزمایشگاه معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دکتر سربازی  
تابستان ۱۴۰۳

رادین چراغی، ۴۰۱۱۰۵۸۱۵  
مبین پورعابدینی، ۴۰۱۱۱۰۵۵۶  
آرین نوری، ۴۰۱۱۰۶۶۶۳



## گزارش آزمایش چهارم

### نتایج مورد انتظار

در این آزمایش با فعال شدن سیگنال Start، ضرب دو عدد دودویی محاسبه می‌شود. انتظار می‌رود نتیجه صحیح پس از چند سیکل ساعت، همزمان با فعال شدن سیگنال End در خروجی مشاهده شود.

### ۴-۲ آزمایش چهارم: جمع/تفریق کننده ممیز شناور

#### هدف

در این آزمایش (طی دو جلسه)، مدار یک جمع/تفریق کننده ممیز شناور را طراحی کرده و با استفاده از ابزار Proteus شبیه‌سازی می‌کنیم. پس از اطمینان از صحت عملکرد در شبیه‌ساز، آن را روی برد پیاده‌سازی می‌کنیم. مدار اولیه برای شبیه‌سازی را مطابق استاندارد IEEE-754 ۳۲ بیتی طراحی کنید. برای سهولت در پیاده‌سازی روی برد، تعداد بیت‌ها را از ۳۲ به ۱۲ کاهش دهید.

مشخصات مدار مورد نظر به قرار زیر است:

عملوند اول (ورودی): A

عملوند دوم (ورودی): B

حاصل جمع/تفریق (خروجی): C

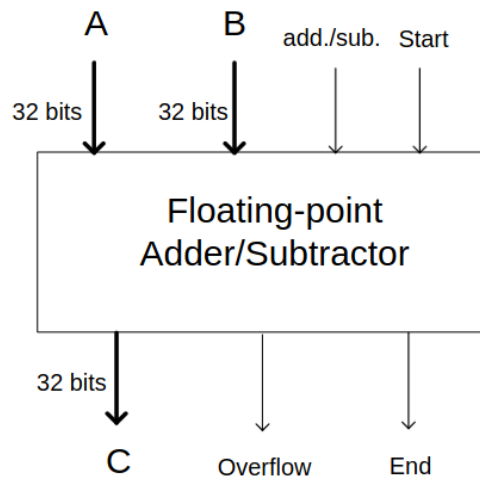
شروع عملیات (ورودی): Start

پایان عملیات (خروجی): End

سرریز (خروجی): Overflow

مشخص کننده جمع/تفریق (ورودی): add/sub

در این آزمایش قصد داریم یک جمع/تفریق کننده ممیز شناور ۳۲ بیتی به شکل زیر بسازیم:



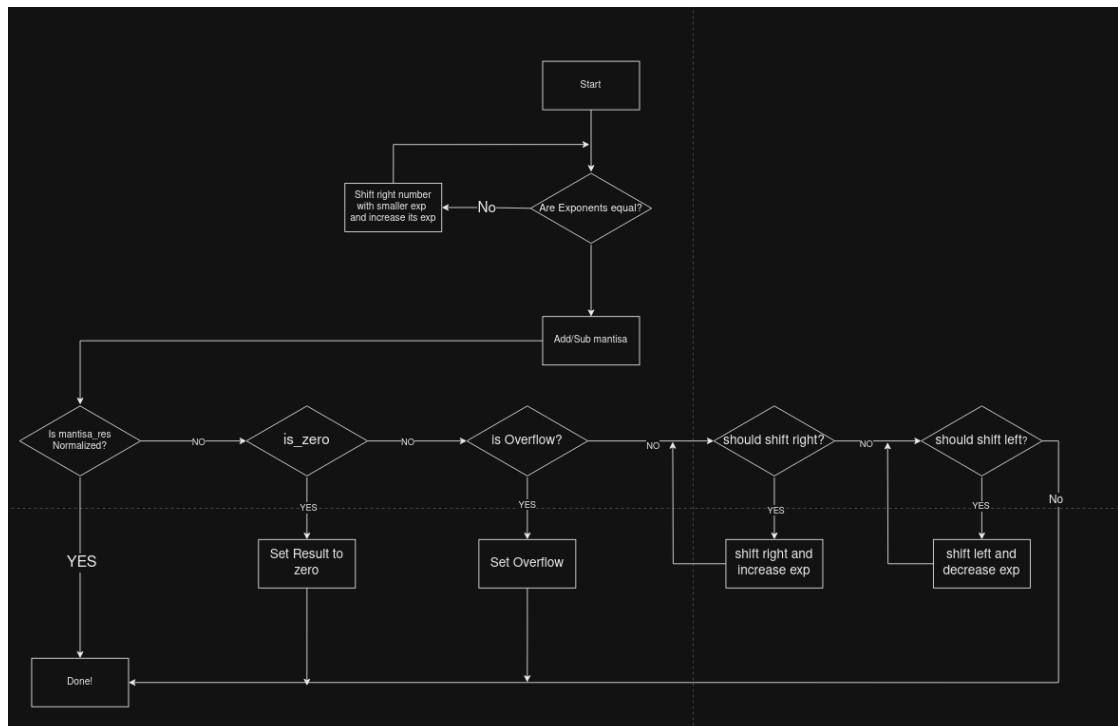
شکل ۴: جمع/تفریق کننده ممیز شناور



شکل (۱)

## ۱ الگوریتم جمع/تفریق کننده:

برای راحتی تر ارائه کردن الگوریتم یک Diagram معادل الگوریتم در draw.io کشیده و ارائه می‌دهیم:



شکل (۱.۱)

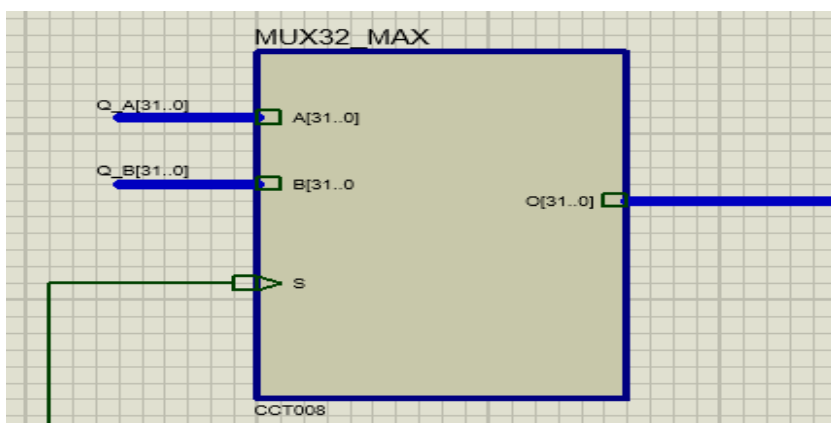
این الگوریتم سه بخش اساسی دارن:

- یکسان کردن نما های اعداد ورودی:  
در این مرحله اعداد لود شده در رجیستر را به گونه ای شیفت میدهم که بعد از تعدادی مرحله نمای شان برابر شود برای مثال اگر اعداد ورودی مان  $A = 2^5 * (1/0010)$ ,  $B = 2^6 * (1/0101)$  در این مرحله عدد با مبنای کوچک تر را (در اینجا  $A$ ) یک واحد به راست شیفت میدهم و یک واحد به نمای آن اضافه میکنیم که سبب میشود اعداد مان به  $A = 2^6 * (0/1001)$ ,  $B = 2^6 * (1/0101)$  تبدیل شود.
- جمع/تفریق کردن مانتیسا ها:  
در این بخش چون اعداد ورودی هم نما شده اند برا جمع/تفریق کردن شان میتوانیم از نما فاکتور گرفته و دو عدد ۲۴ بیتی باقی مانده را باهم جمع کنیم و در نهایت نمایی که داشتیم را در عدد به دست آمده ضرب کنیم معادل اش در مثال ما میشود:  $A+B = 2^6 * (0/1001 + 1/0101) = 2^6 * (1/1110)$ .
- نرمال کردن خروجی:  
برای نرمال کردن خروجی باید توجه کنیم که ممکن است جمع دو عدد ۲۴ بیتی ما عددی با ۲۵ بیت باشد در نتیجه در چنین حالتی باید جواب نهایی را قبل از خروجی دادن یک واحد به راست شیفت دهیم و به نمای آن یک واحد اضافه کنیم، حالتی دیگری که ممکن است رخ دهد این است که عدد خروجی ما بیت پر ارزش یک اش سمت راست بیت ۲۴ ام باشد در این صورت چون لازم است عدد نهایی مان نرمال شده باشد باید عدد را آنقدر به چپ شیفت دهیم (و نمای آن را هر بار یک واحد کم کنیم) تا بیت ۲۴ ام عدد مان ۱ شود و به شکل نرمال در بیاید.  
باید توجه کنیم که اگر خروجی نهایی مان ۰ میشد در حلقه بی نهایت گیر نکنیم.

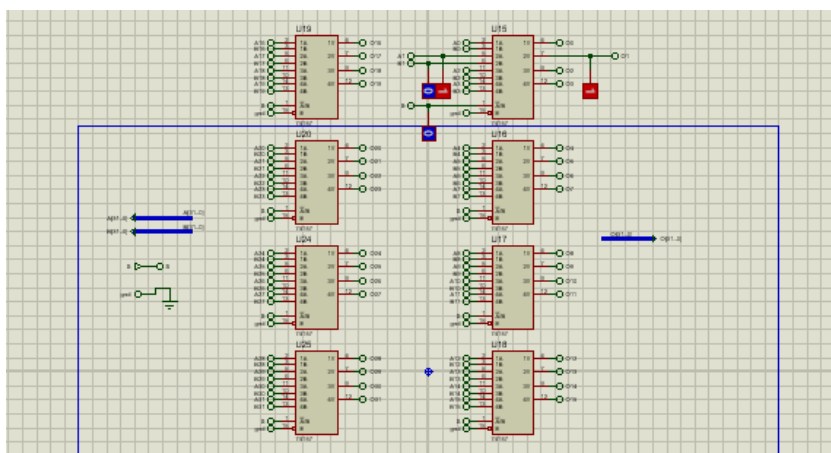
## ۲ پیاده سازی در پروتئوس:

### ۱.۲ ساخت ماژول های کمکی:

- ماژول MUX ۳۲ بیتی:

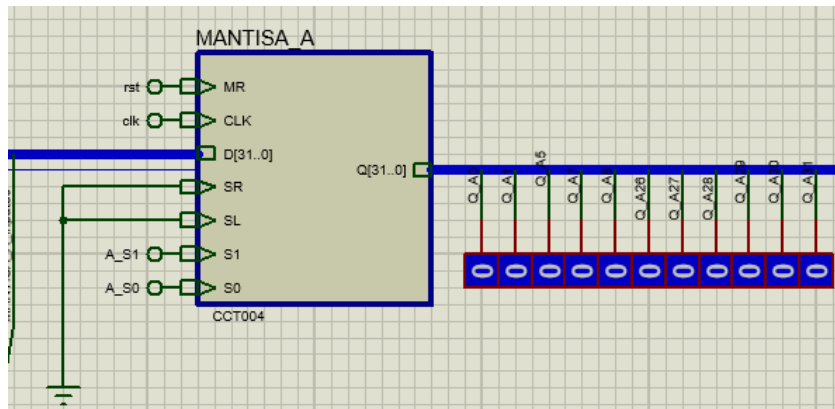


شکل (۱.۱.۲)

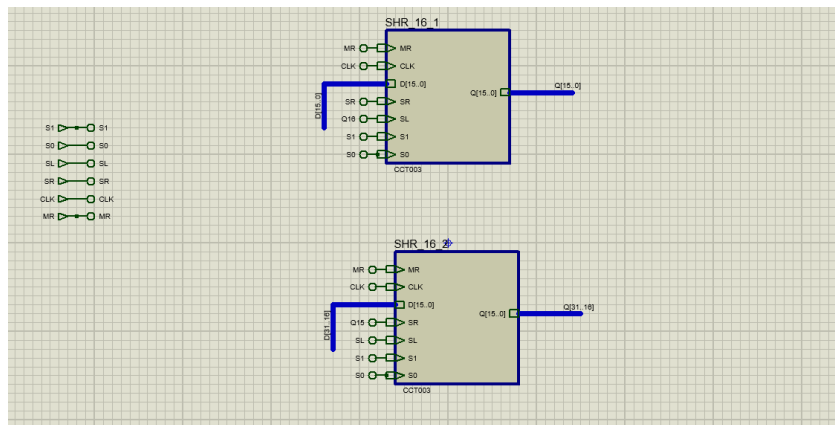


شکل (۲.۱.۲)

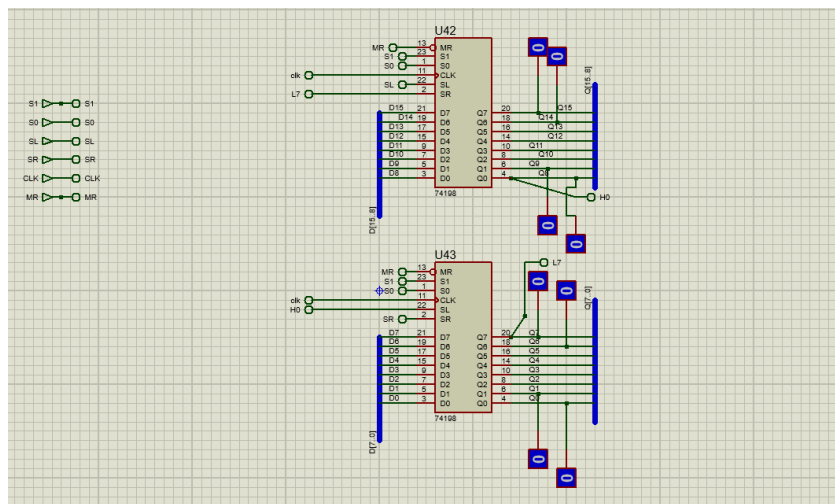
- رجیستر ۳۲ بیتی با قابلیت لود و شیفت چپ و راست:



شکل (۳.۱.۲)

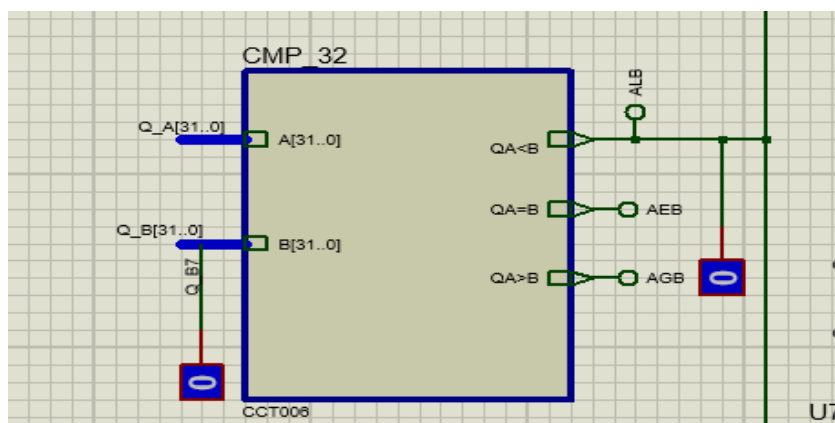


شکل (۴.۱.۲)

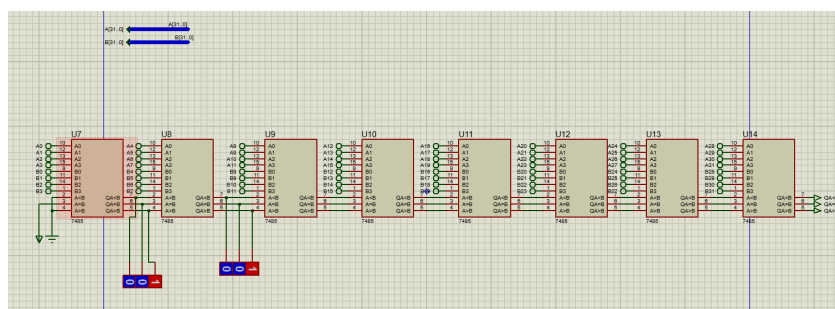


شکل (۵.۱.۲)

- مقایسه کننده ۳۲ بیتی:

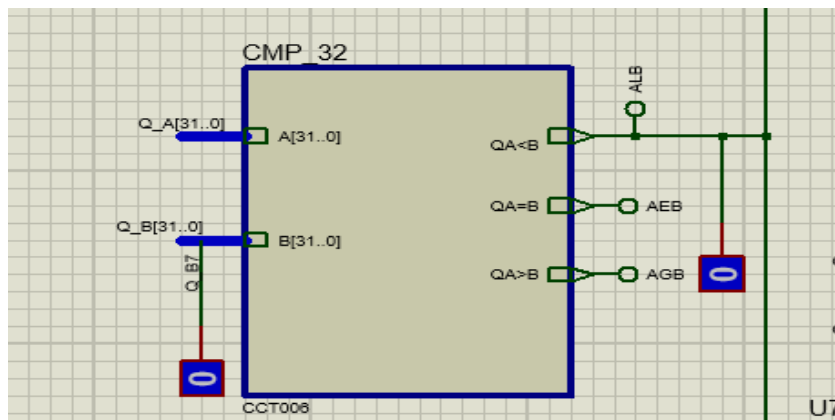


شکل (۶.۱.۲)

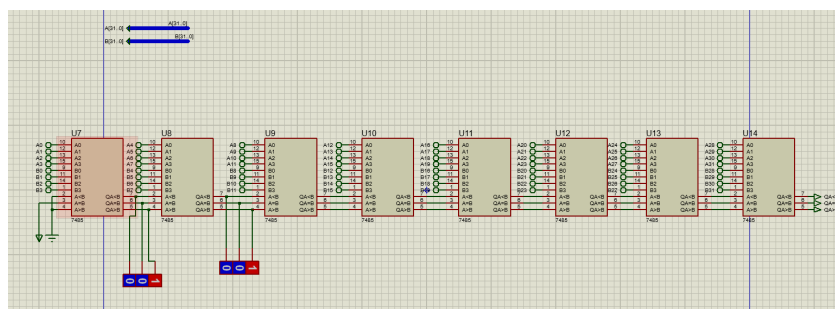


شکل (۷.۱.۲)

- جمع کننده ۳۲ بیتی:



شکل (۸.۱.۲)

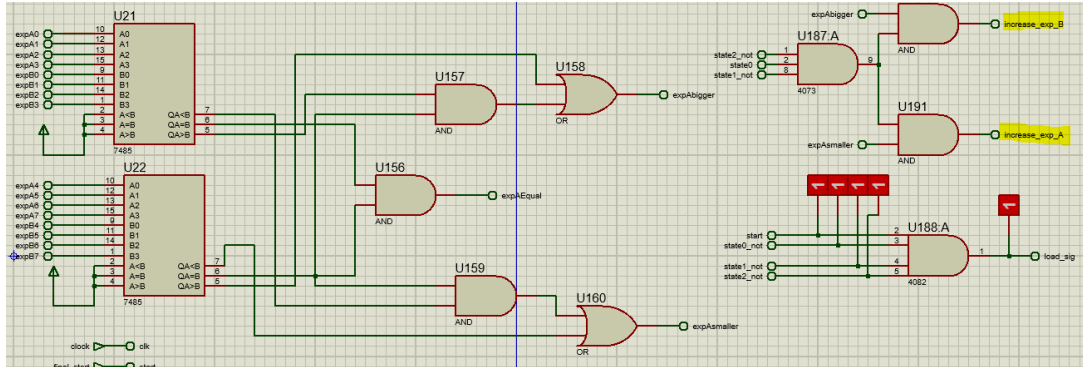


شکل (۹.۱.۲)

## ۲.۲ مرحله اول الگوریتم (هم نما کردم ورودی ها)

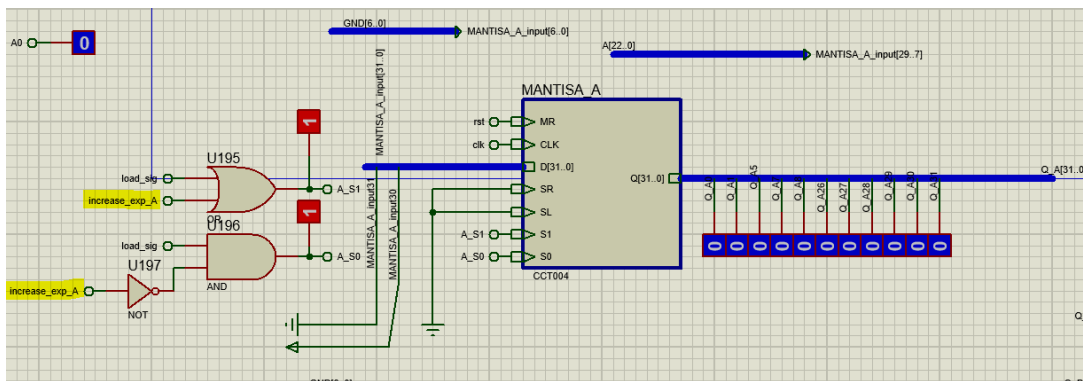
در این مرحله و بعد از لود شدن ورودی ها هر بار که نمای عددی از عدد دیگر بزرگ تر باشد، عدد با نمای کوچکتر یک واحد به راست شیفت میخورد و نمای آن یک واحد افزایش نیابد تا زمانی که هر دو نما با هم برابر شوند. برای ساخت چنین چیزی از رجیستر ۳۲ بیتی که بالا ساختیم استفاده میکنیم و در هر مرحله با سیگنال های  $increase\_exp\_B$  و  $increase\_exp\_A$  یکی از دو عدد را به چپ شیفت میدهیم.

- ابتدا سیگنال های  $increase\_exp\_A$  و  $increase\_exp\_B$  را میسازیم:  
(کافی است مقادیر نمای هر دو عدد را مقایسه کنیم تا بفهمیم کدام بزرگ تر است و سیگنال مان را از روی آن بسازیم)



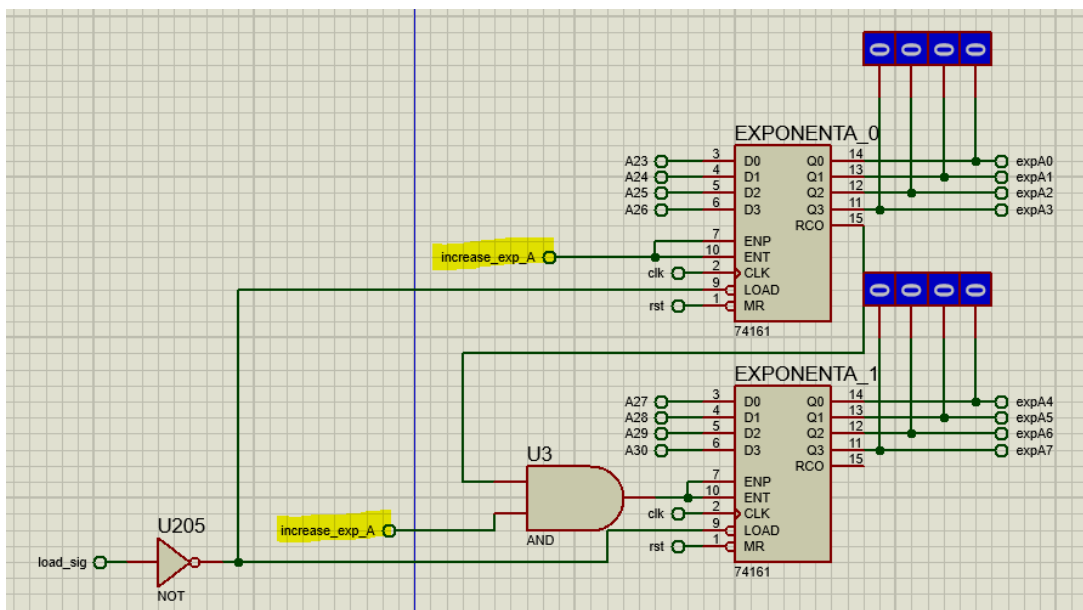
شکل (۱.۲.۲)

- سپس با استفاده از این سیگنال ها مقادیر رجیستر های حاوی مانتیسا را شیفت میدهیم:



شکل (۲.۲.۲)

- همچنین مقدار نمای متناظر رجیستر ای که شیفت داده شد را یکی زیاد میکنیم:



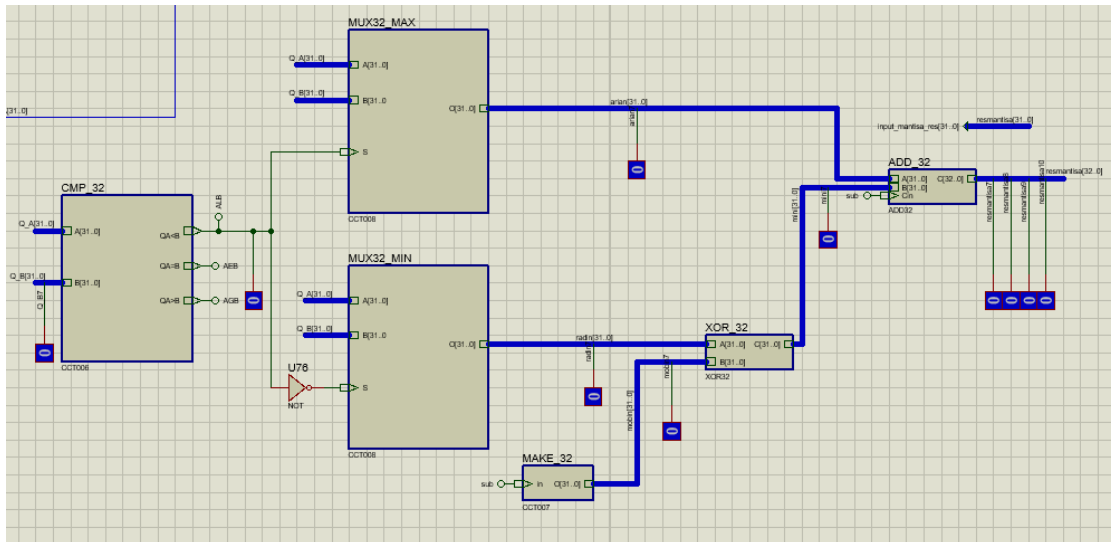
شکل (۳.۲.۲)

در نتیجه بعد از تعدادی کلاک نمای دو عدد برابر خواهد بود و آماده‌ی جمع شدن باهم میشوند. حال به قسمت جمع/تفریق کردن دو عدد میرسیم:

### ۳.۲ مرحله دوم الگوریتم (جمع کردن دو عدد)

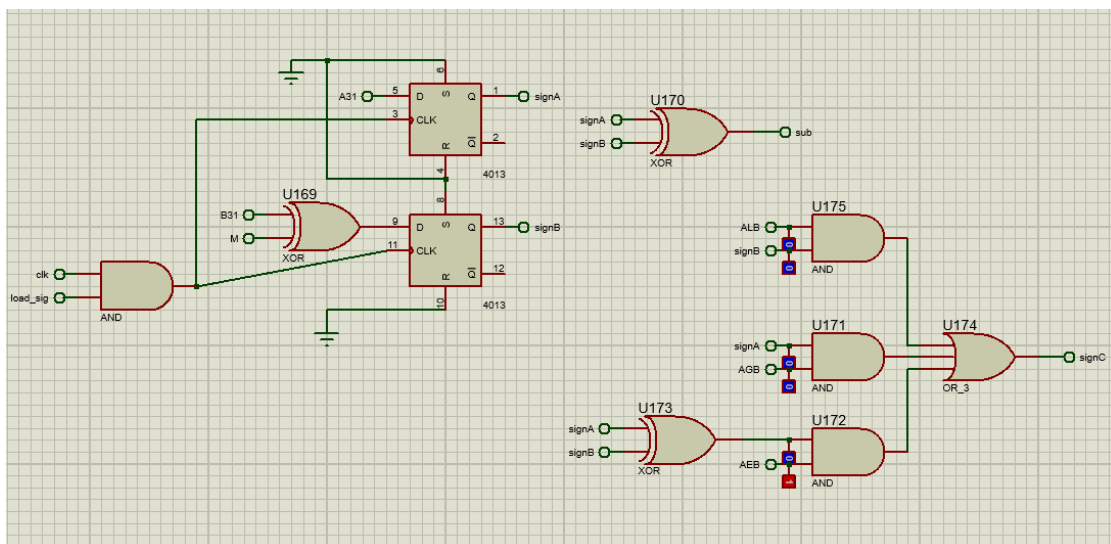
در این مرحله به منظور راحت تر شدن مدار اگر دو عدد  $A$  و  $B$  هم علامت بودن (سیگنال  $M$  که برای منها است را نیز جزو علامت  $B$  حساب میکنیم) کافی است

دو عدد را باهم جمع کنیم در غیر این صورت باید اختلاف این دو عدد را حساب کنیم و مقدار قدر مطلق جواب نهایی را به دست آوریم: به این منظور با استفاده از مقایسه کننده ۳۲ بیتی عدد بزرگ تر و عدد کوچک تر (از لحاظ قدر مطلق) را حساب میکنیم، اگر علامت دو عدد یکسان بود به سادگی آنها را با ماژول Add ۳۲ بیتی جمع میکنیم و در غیر این صورت با استفاده از یک ماژول XOR ۳۲ بیتی عدد کوچک تر مون رو ۱'s complement میکنیم و سپس در هنگام جمع با دادن  $Cin = 1$  در ماژول جمع ۳۲ بیتی به ۲'s complement تبدیل میکنیم



شکل (۱.۳.۲)

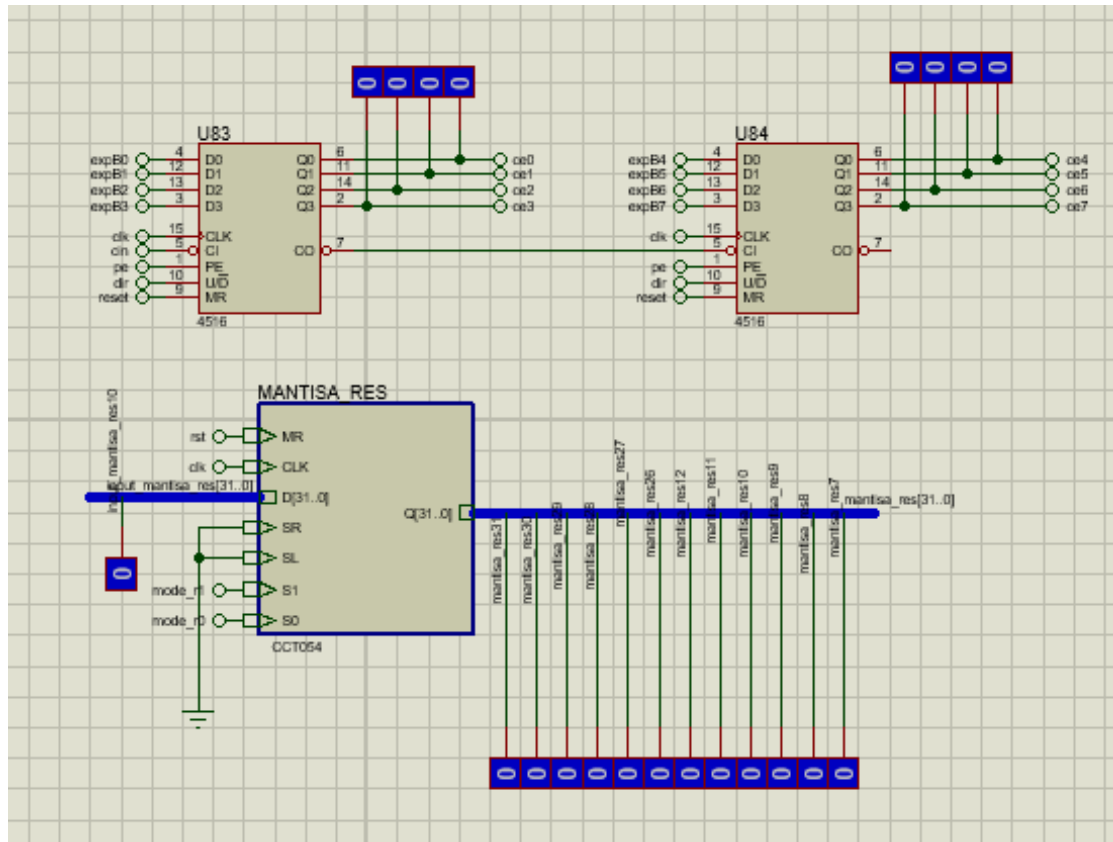
همچنین در نهایت با استفاده از نتیجه مقایسه دو عدد و علامت های آنها میتوانیم علامت جواب را به این شکل حساب کنیم:



شکل (۲.۳.۲)

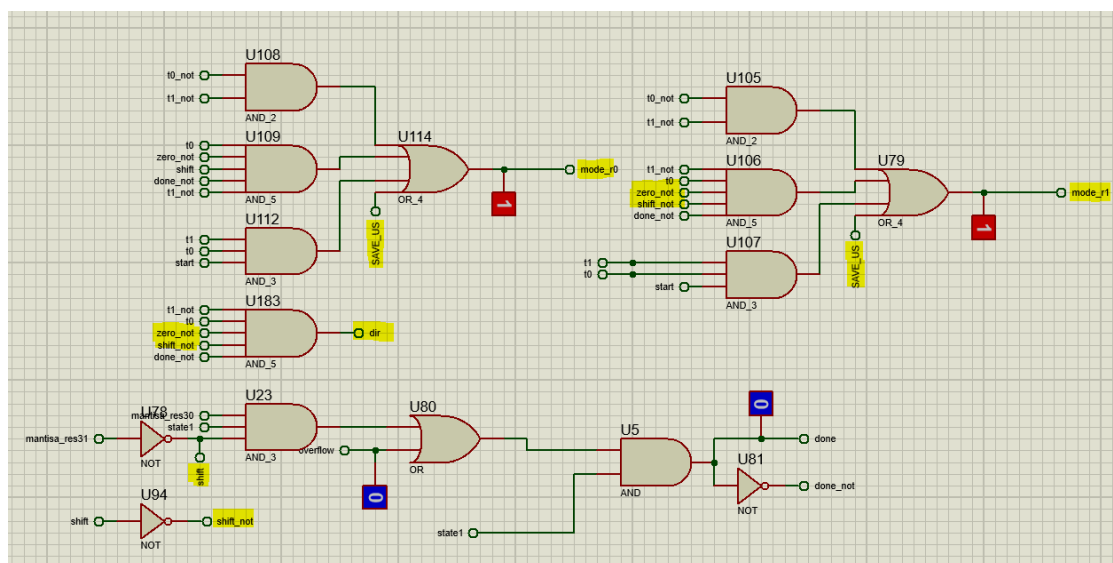
## ۴.۲ مرحله سوم الگوریتم (نرمال سازی جواب)

در این مرحله ابتدا جواب آماده شده از جمع کننده مرحله قبل را در رجیستر مانتیسا جواب لود میکنیم و همچنین مقدار نمای برابر شون رو هم در یک شمارنده ۸ بیتی (از ترکیب دو تا شمارنده ۴ بیتی میسازیمش) لود میکنیم، برامون مهم است که این شمارنده توانایی بالا شمردن و پایین شمردن داشته باشد زیرا ممکن است بسته به شرایط مجبور به شیفت چپ و یا شیفت راست شویم که به این معنی است که نمای جواب مان ممکن است بیشتر یا کمتر شود.



شکل (۱.۴.۲)

همچنین لازم داریم که بدونیم در این لحظه اگر نیاز به شیفت هست و هنوز نرمالایز تموم نشده است، باید به چه سمتی شیفت بدهیم، برای کنترل کردن این نکته از سیگنال *dir* استفاده میکنیم که از روی سیگنال *shift\_not* و *shift* ساخته شده است که آن نیز به بیت ۳۱ مانتیسا جواب نگاه میکند و در صورتی که ۱ باید مقدار آن نیز ۱ خواهد بود. همچنین سیگنال های *mode\_r1* و *mode\_r0* نیز برای دادن سیگنال مطابق رجیستر ها است.



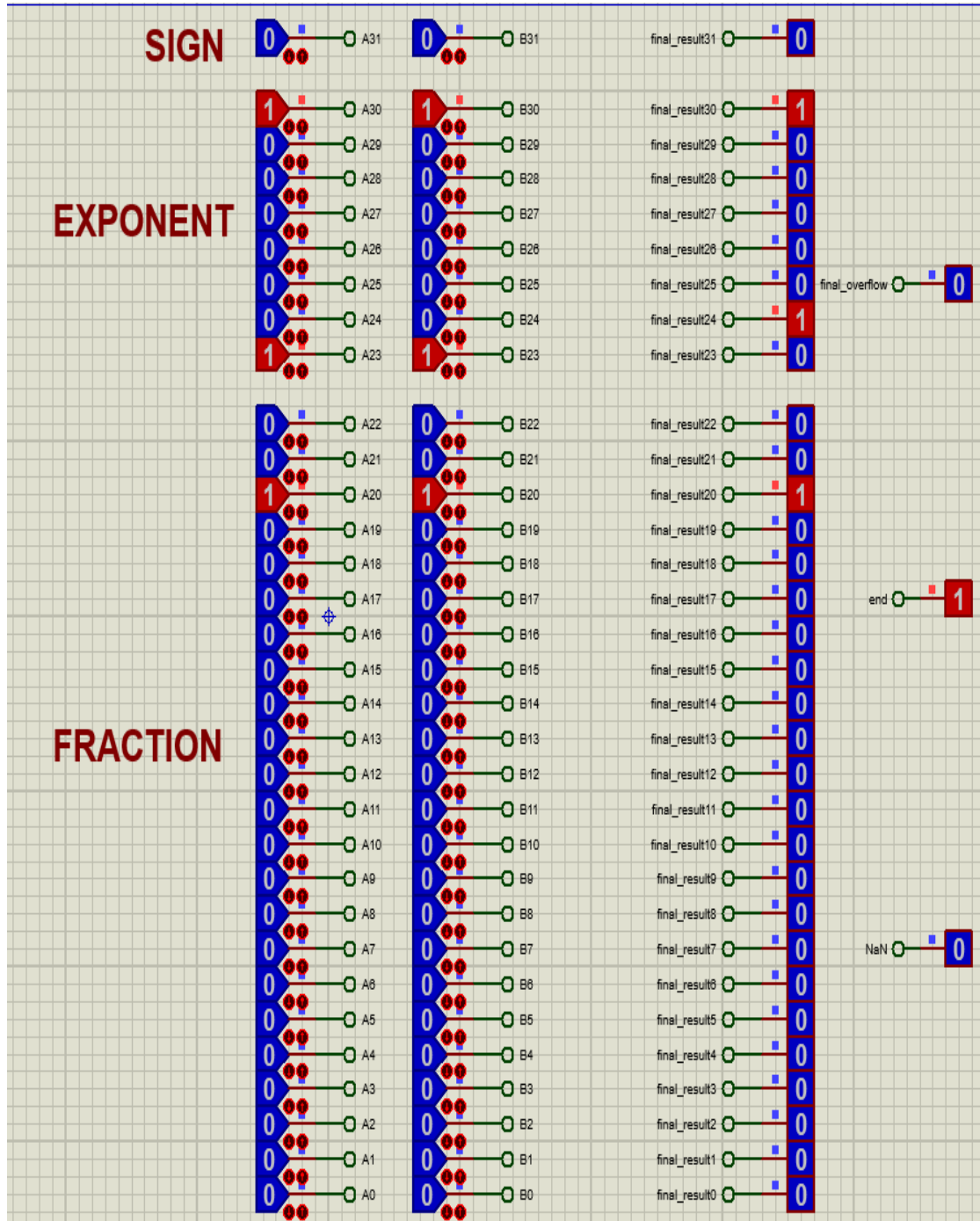
شکل (۲.۴.۲)



### ۳ تست کردن مدار شبیه سازی شده

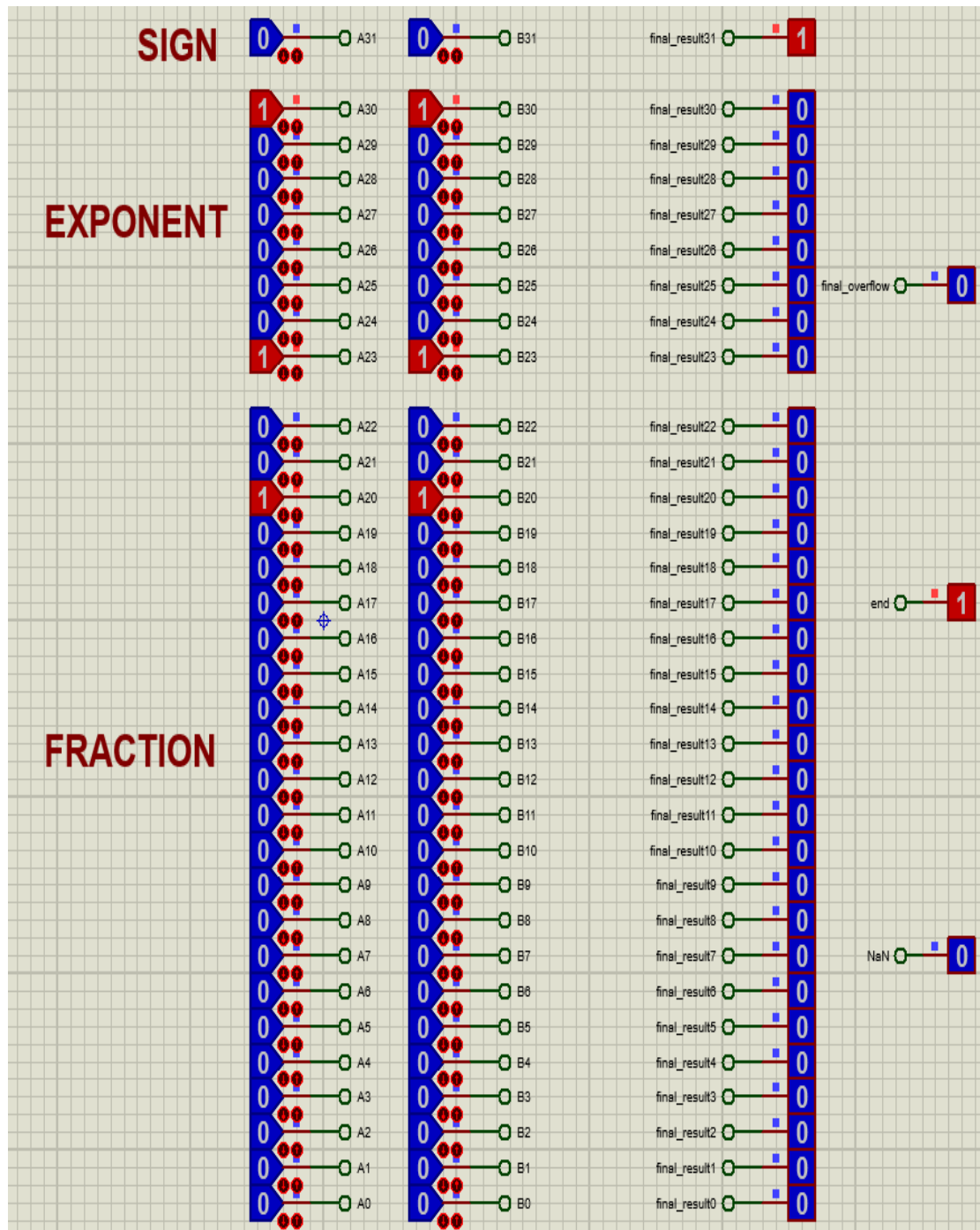
در این مرحله با تعدادی تست کیس عملکرد مدار را بررسی میکنیم:

$$A = 4/5 = 2^2 * 1/0.01, B = 4/5 = 2^2 * 1/0.01 \Rightarrow A + B = 9 = 2^3 * 1/0.01 \bullet$$



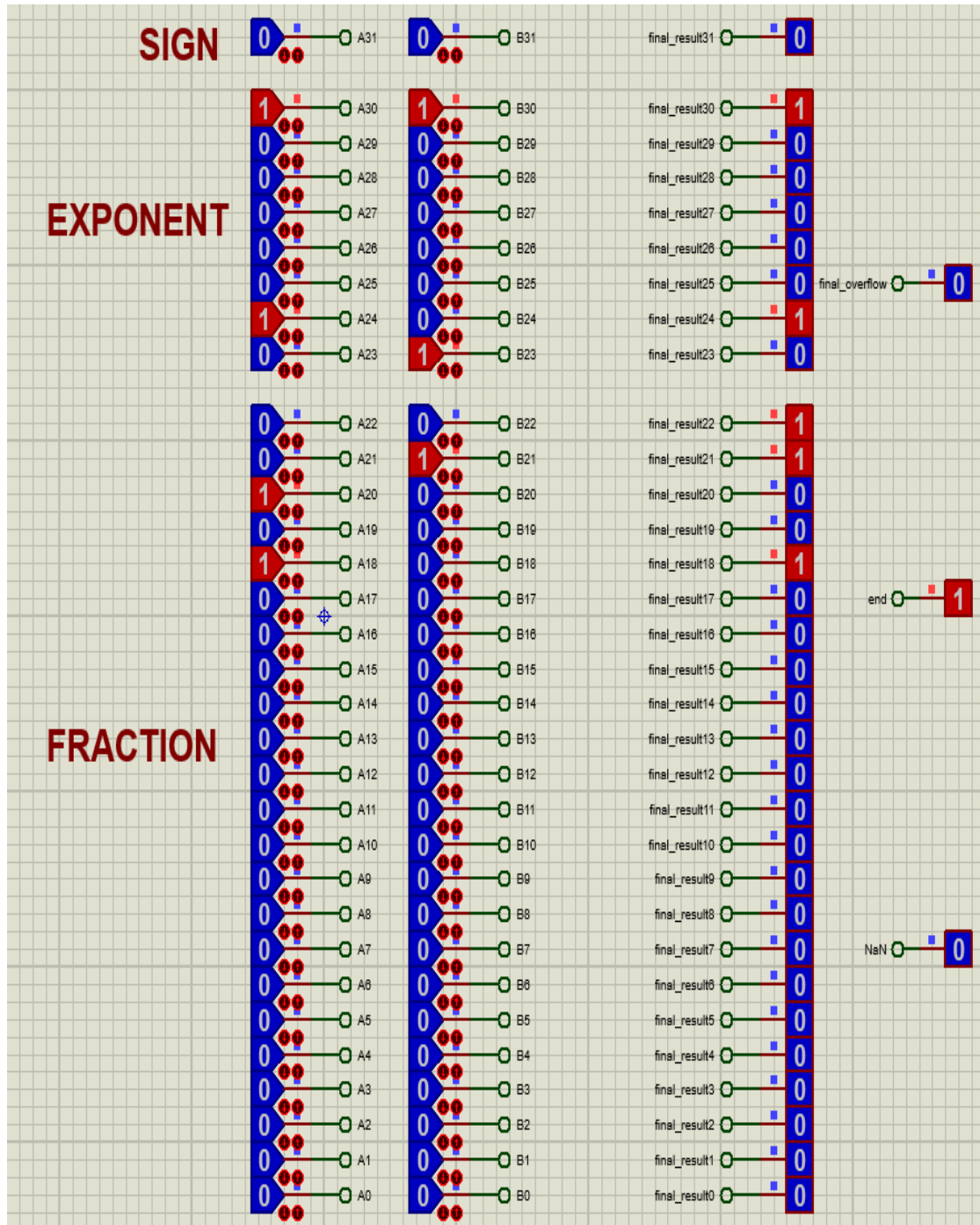
شکل (۱.۳)

$$A = \frac{r}{d} = 2^r * \frac{1}{d}, B = \frac{r}{d} = 2^r * \frac{1}{d} \Rightarrow A - B = 0 = 2^r * \dots$$



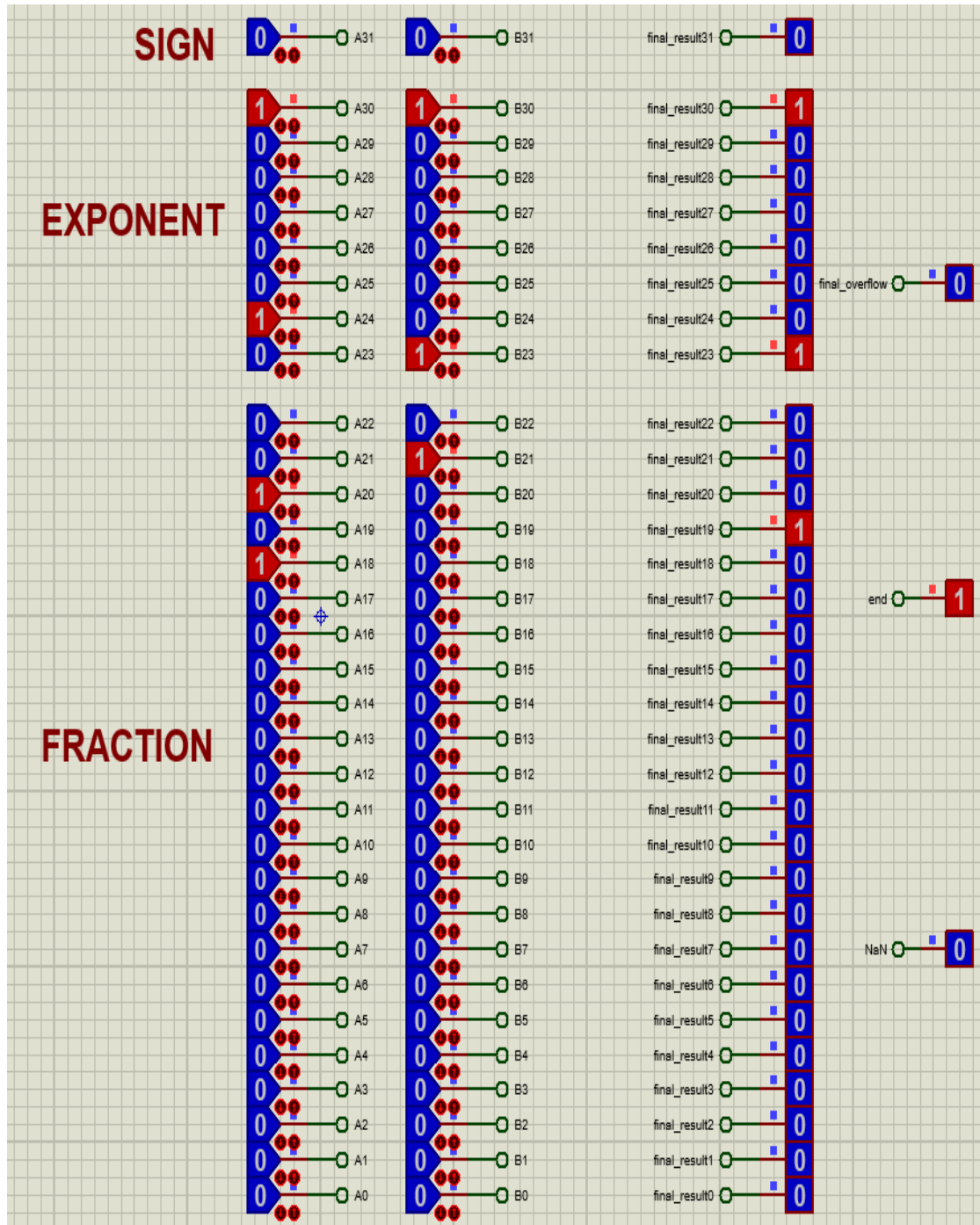
شكل (٢.٣)

$$A = 9/25 = 2^r * 1/00101, B = 5 = 2^r * 1/1000 \Rightarrow A + B = 14/25 = 2^r * 1/11001 \bullet$$



شكل (٣.٣)

$$A = 9/25 = 2^r * 1/0.101, B = 5 = 2^r * 1/0.1000 \Rightarrow A - B = 4/25 = 2^r * 1/0.001 \bullet$$



شكل (٢.٣)