به نام خدا

اعضای گروه :

- فرزام کوهی رونقی ۴۰۱۱۰۶۴۰۳
- ثنا بابایان ونستان ۴۰۱۱۰۵۶۸۹
- مریم شیران ۴۰۰۱۰۹۴۴۶

شرح آزمایش چهارم :

در این آزمایش مدار یک جمع/تفریق کننده ممیز شناور را با استاندارد IEEE-754 ۳۲ بیتی طراحی می کنیم.

بیت های مدار :

- A : ورودی ۳۲ بیتی اول
- B : ورودی ۳۲ بیتی دوم
- OUT : خروجی ۳۲ بیتی
- S : برای شروع shifter اولیه است.
- ST : برای شروع normalizer است.
- ADD/SUB : مشخص کننده جمع و تفریق که ۰ یعنی جمع و ۱ یعنی تفریق.
- E : نشان دهنده اتمام پروسه shifter اولیه است.
- END : با اتمام پروسه جمع و تفریق این بیت ۱ می شود.
- OVERFLOW : با overflow شدن حاصل، بیت OVERFLOW یک می شود.
- UNDERFLOW : با underflow شدن حاصل، بیت UNDERFLOW یک می شود.

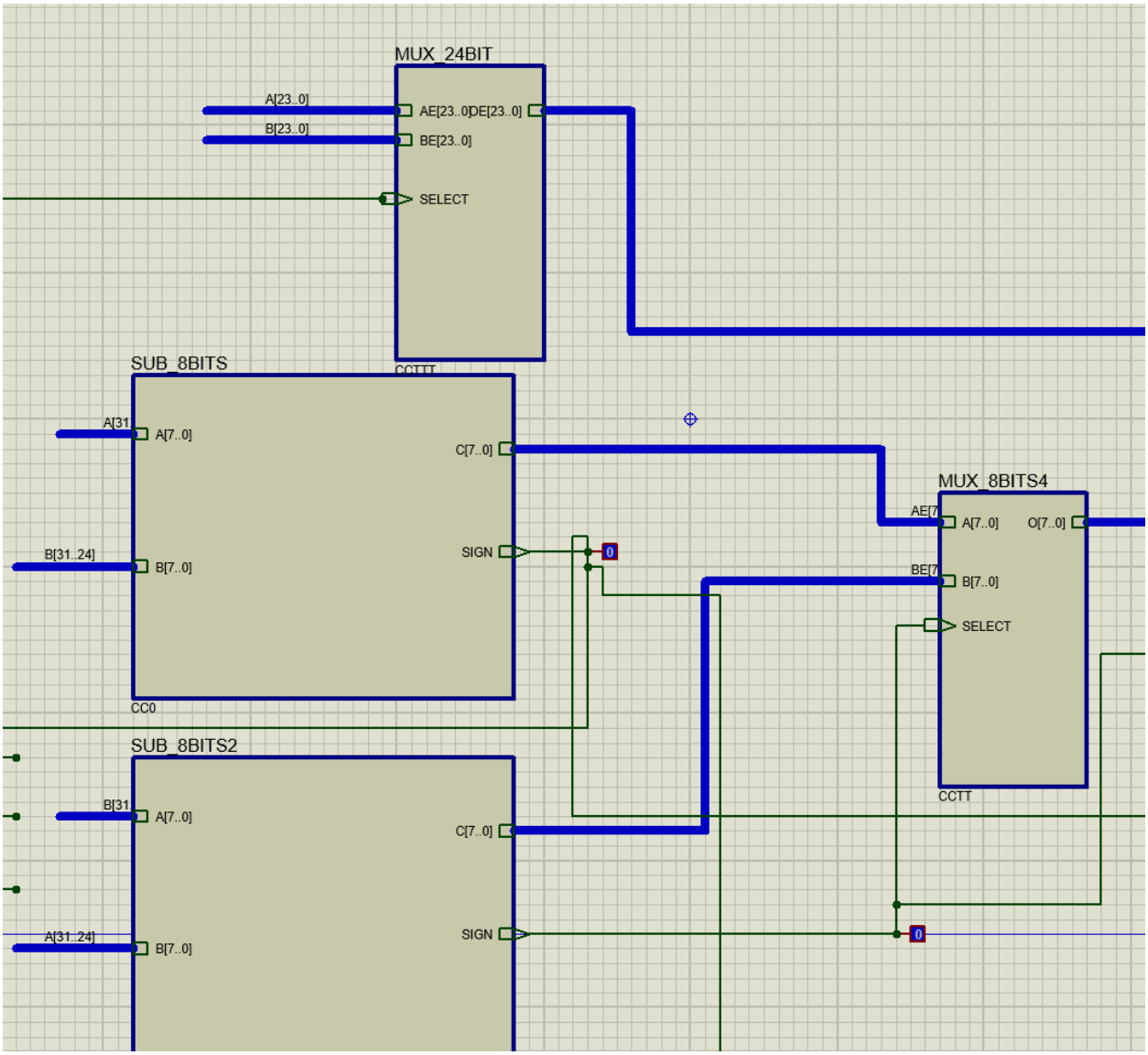
در واقع هر کدام از A, B, OUT ۳۳ بیت دارند زیرا آن ۱ پشت ممیز را هم به عنوان ورودی در نظر گرفتیم.(بیت ۲۳)

نحوه کار با مدار :

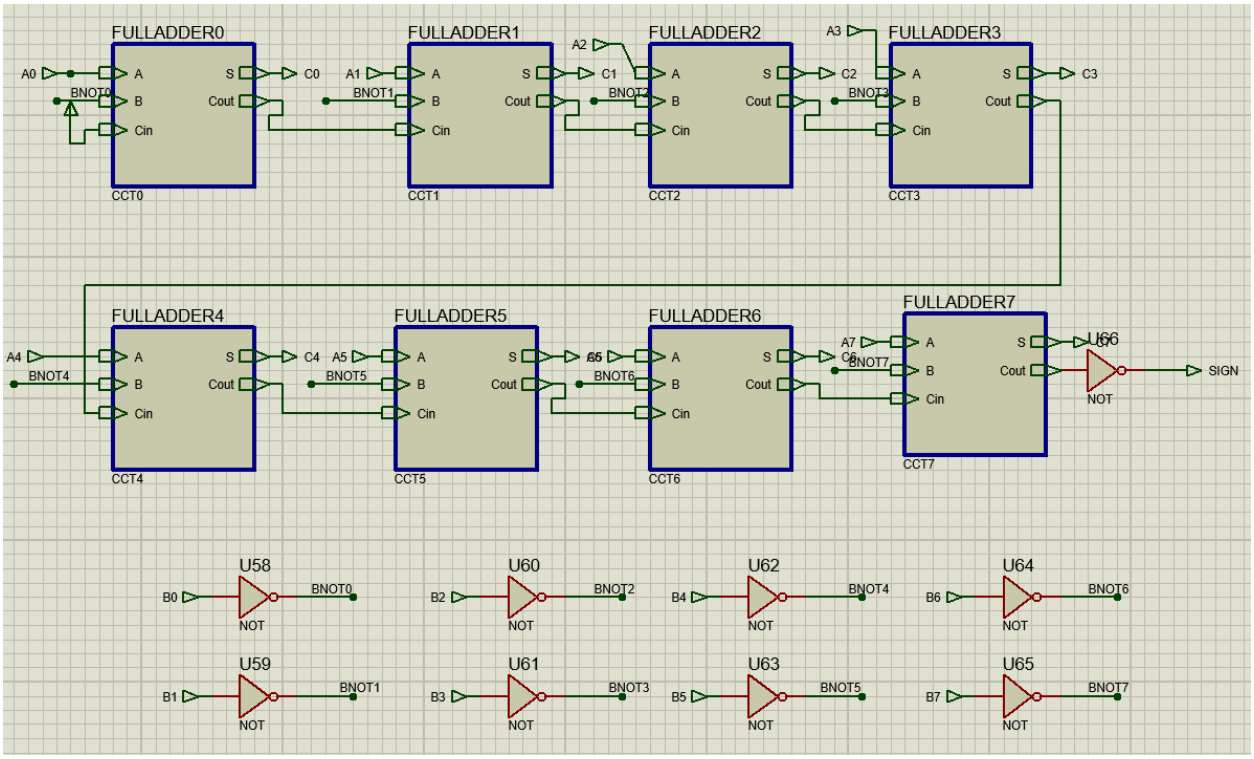
در حد دو ثانیه بیت S را ۱ و سپس ۰ میکنیم و سپس منتظر میمانیم که کار shifter تمام شود که یعنی ۱ شدن بیت E. سپس بیت ST را باز هم در حد دو ثانیه ۱ و سپس ۰ می کنیم. بعد از ۱ شدن بیت END ما جواب مورد نظر را به دست آورده ایم.

نحوه عملکرد مدار :

ابتدا اختلاف دو نما را به دست می آوریم بدین صورت که هم A – B و هم B – A را محاسبه می کنیم و سپس با استفاده از یک multiplexer تصمیم می گیریم که کدام را به عنوان قدرمطلق اختلاف دو نما به شیفتر ورودی دهیم.

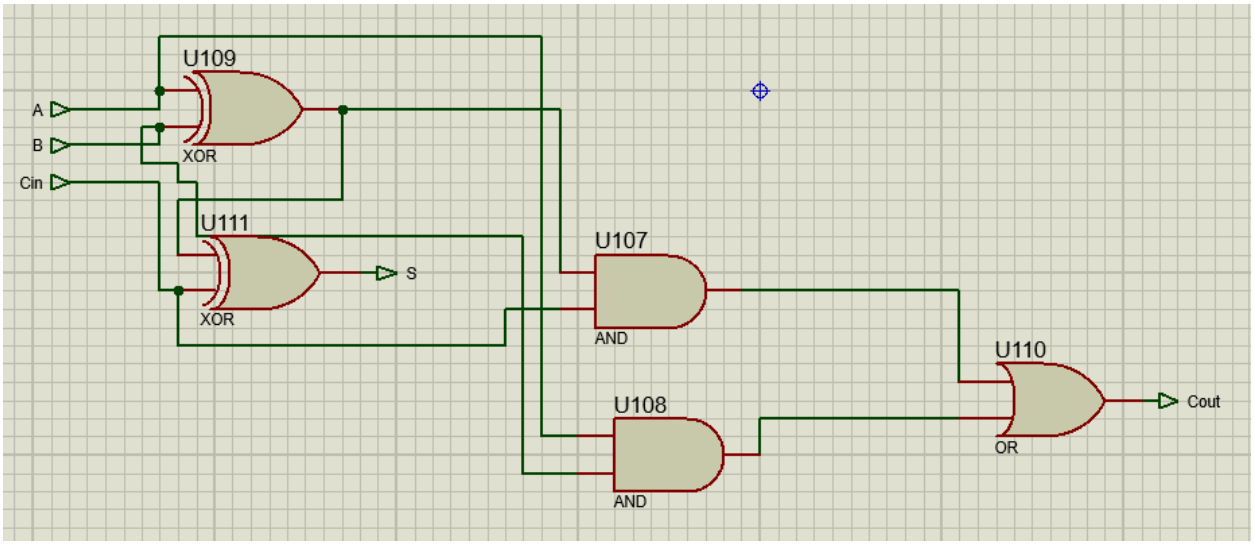


شکل ۱ .دو منها کننده و یک ماکس برای به دست آوردن قدرمطلق اختلاف دو نما .



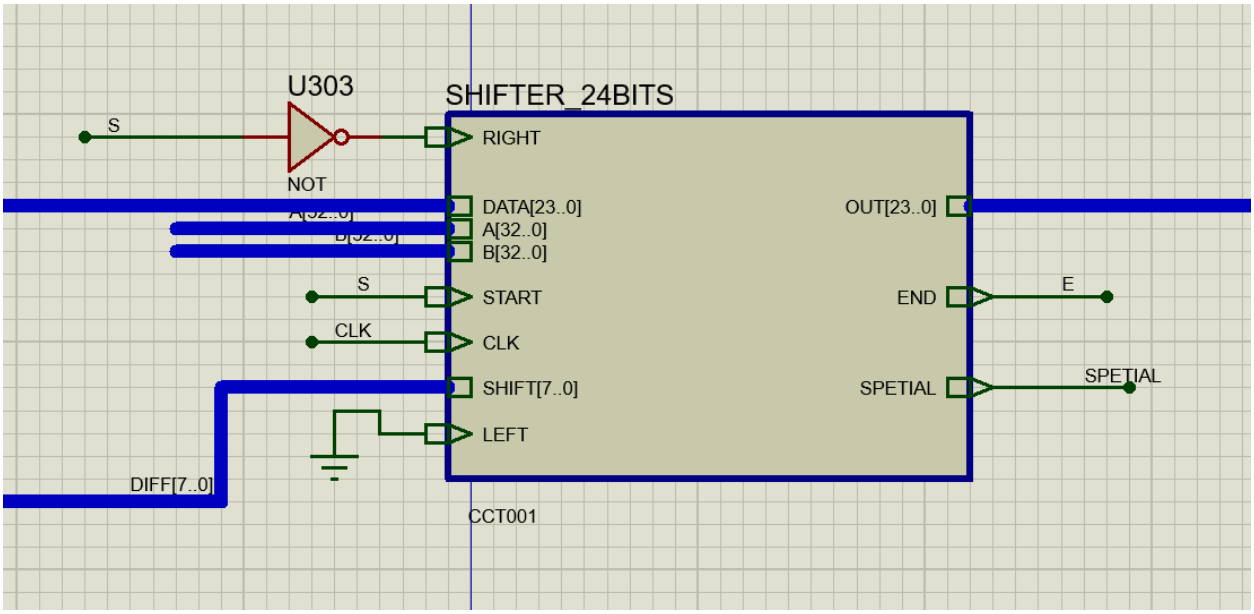
شکل ۲ . ساختار درونی منهاکننده ۸ بیتی.

برای منها کننده باید ورودی دوم را مکمل دو کنیم و با عدد اول جمع کنیم. برای این کار ابتدا بیت های عدد دوم را نات می کنیم و سپس با عدد اول جمع می کنیم البته با carry in ۱ برای full adder اول و خب بقیه پروسه جمع را داریم.



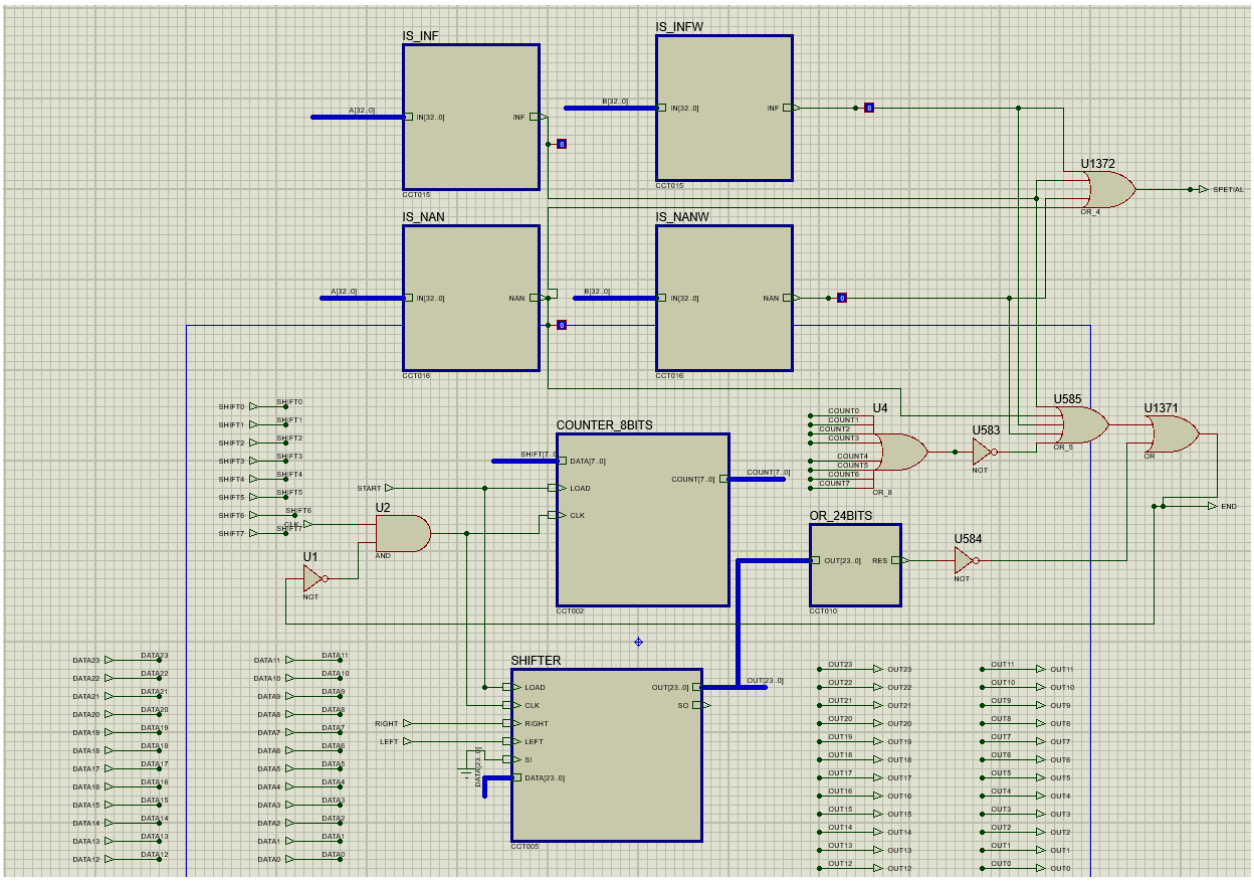
شکل ۳ . ساختار درونی یک full adder .

حال یک ماژول شیفتر ۲۴ بیتی داریم که در شکل زیر نشان می دهیم :



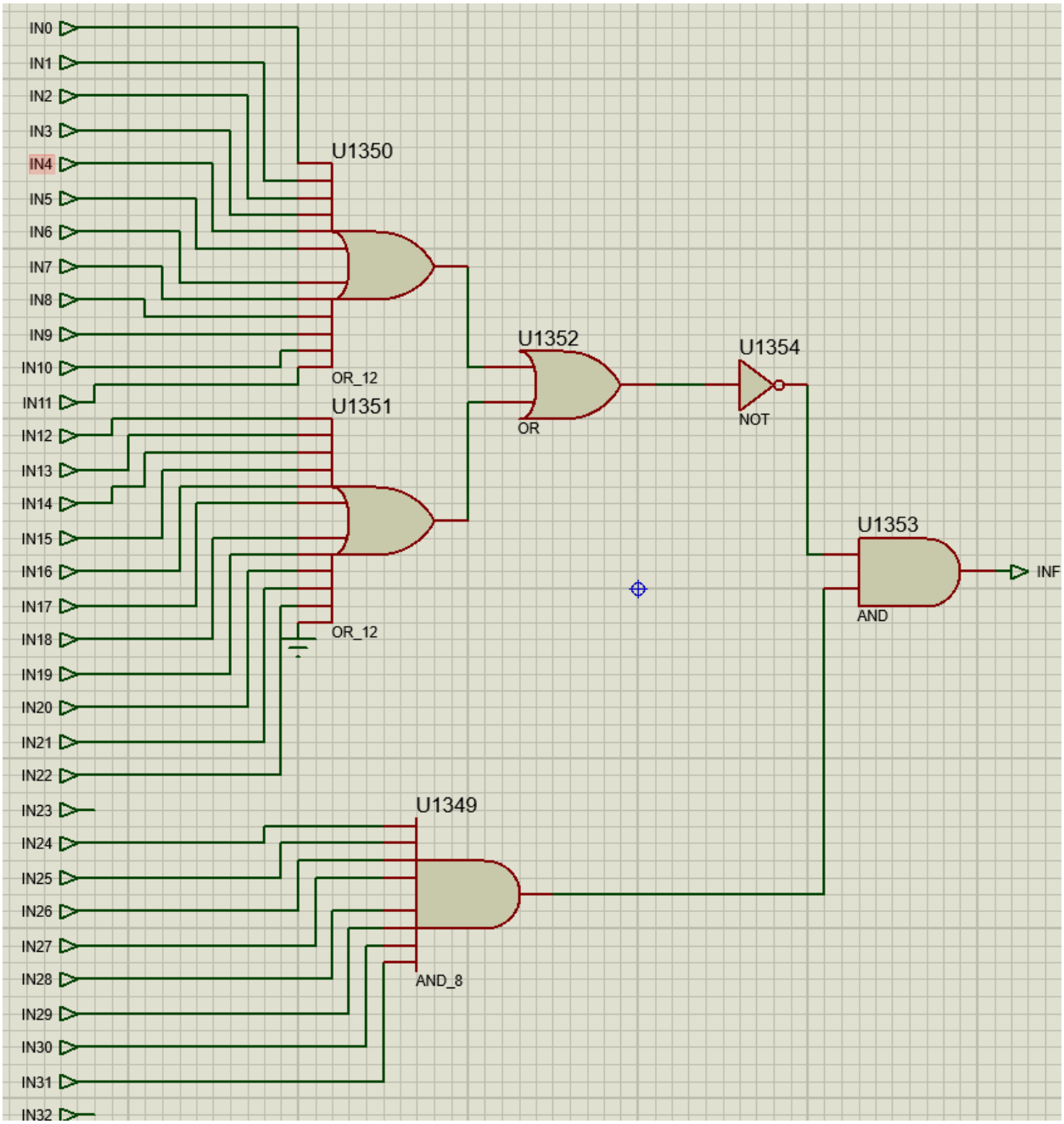
شکل ٤ نمای بیرونی شیفتر ٢٤ بیتی.

در این آزمایش از Bus استفاده کردیم زیرا کار با تعداد سیم زیاد یکم مشکل بود. ورودی LEFT را ۰ به ground وصل کردیم که یعنی همیشه ۰ باشد زیرا در اینجا تنها به شیفت راست نیاز داریم.

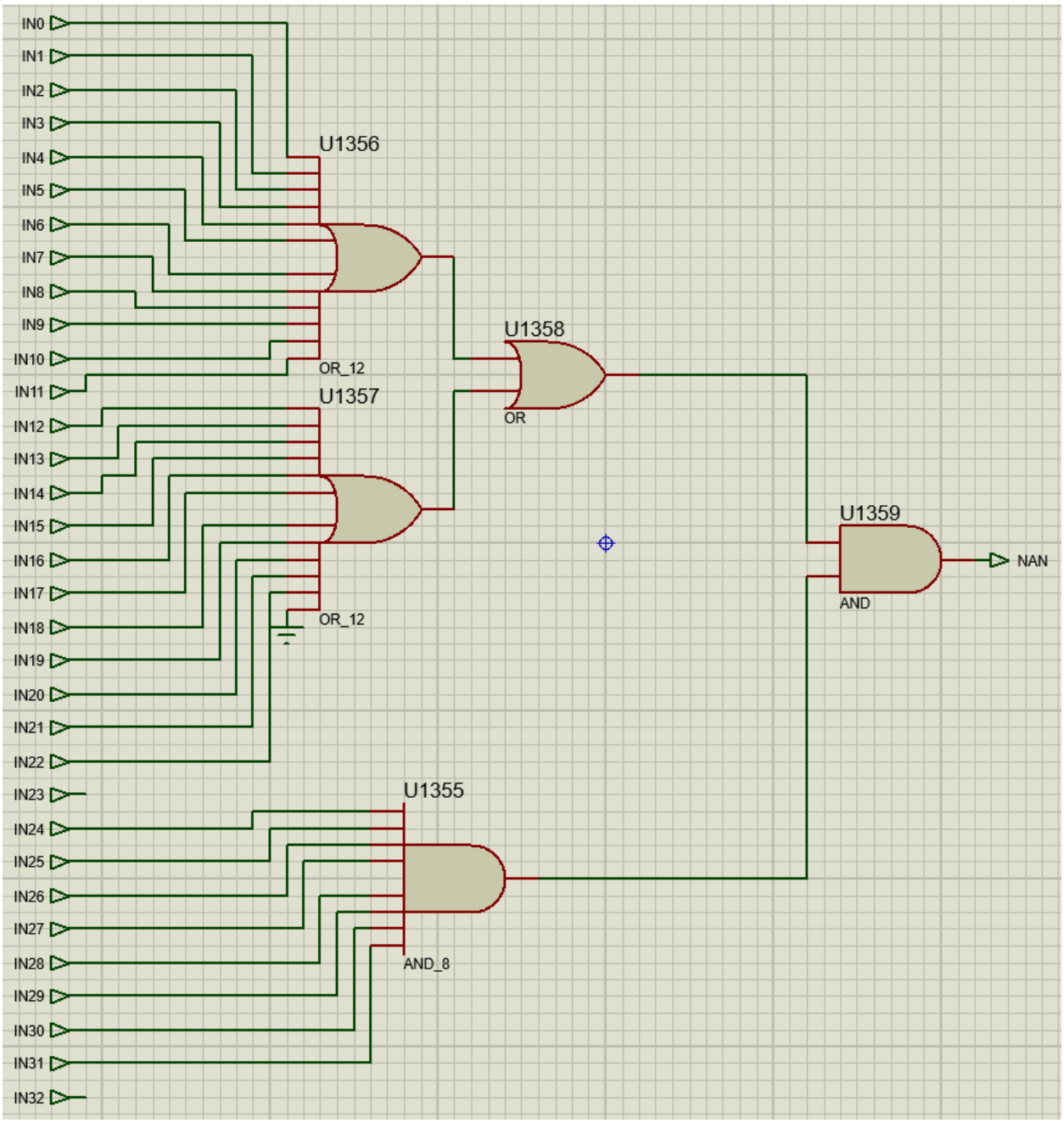


شکل ٥ نمای درون شیفتر ٢٤ بیتی.

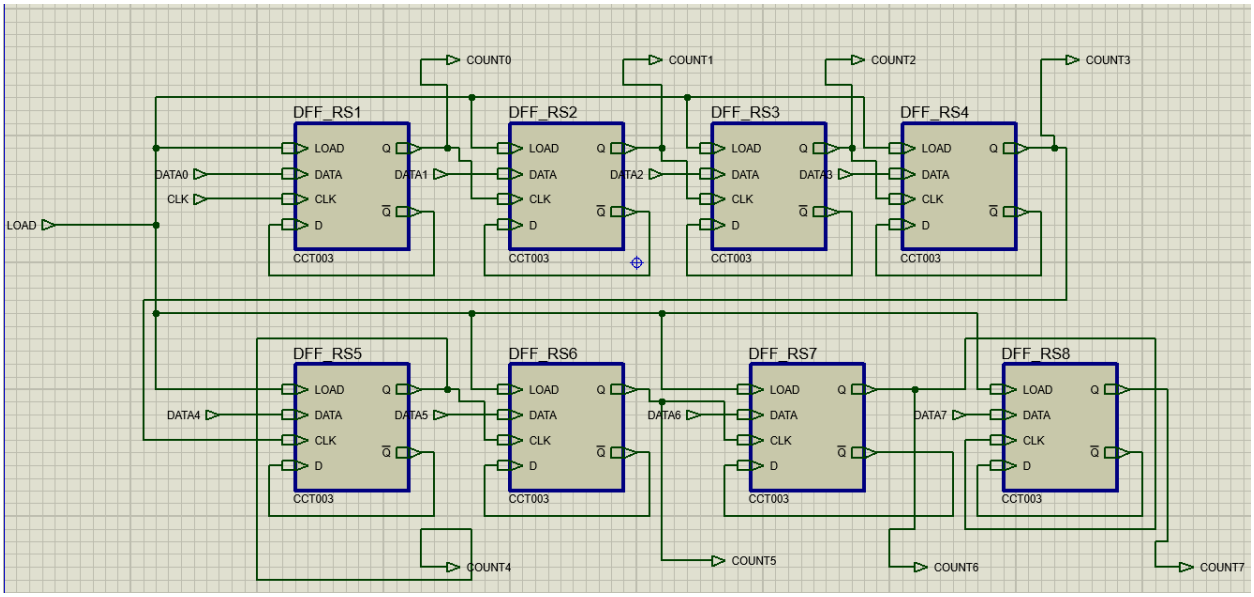
در این ماژول ورودی ها را برای INF, NAN بودن بررسی می کنیم تا اگر هر کدام باشد دیگر وقت را با شیفتر هدر ندهیم. ساختار درونی هر کدام را در ادامه نشان می دهیم.



شکل ٦ تشخیص دهنده INF که یعنی همه بیت های نما ۱ باشد و همه بیت های مانتیس ۰ .

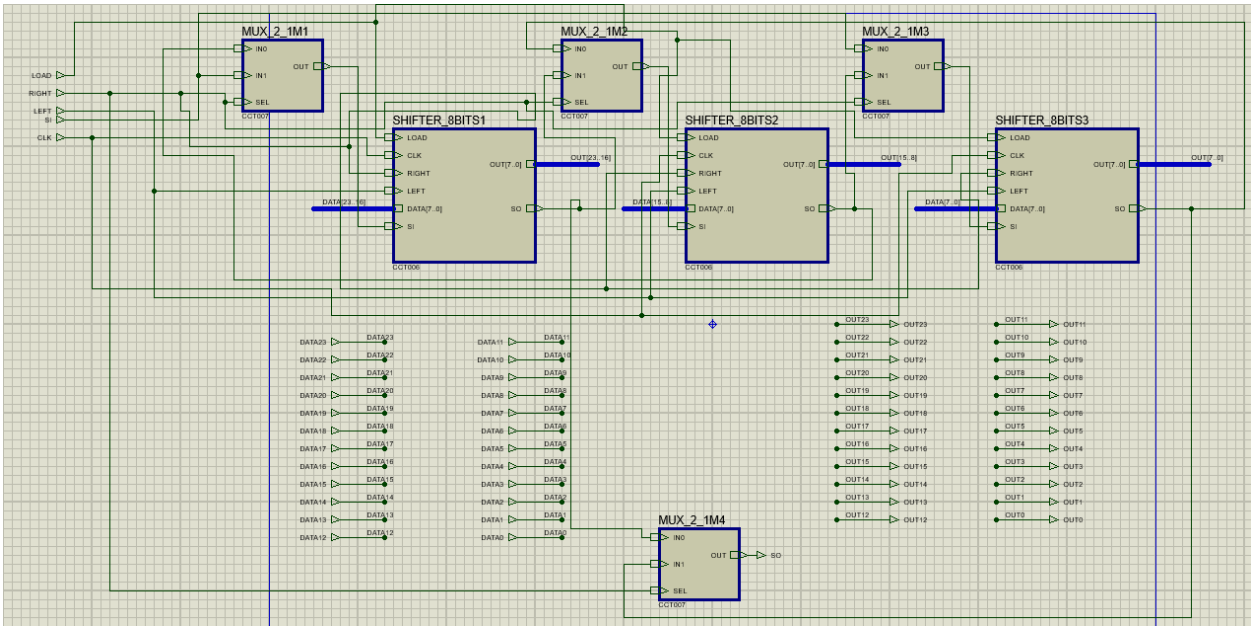


شکل ۷. تشخیص دهنده NAN که همه بیت های نما ۱ باشد و حداقل یکی از بیت های مانتیس غیر صفر باشد.

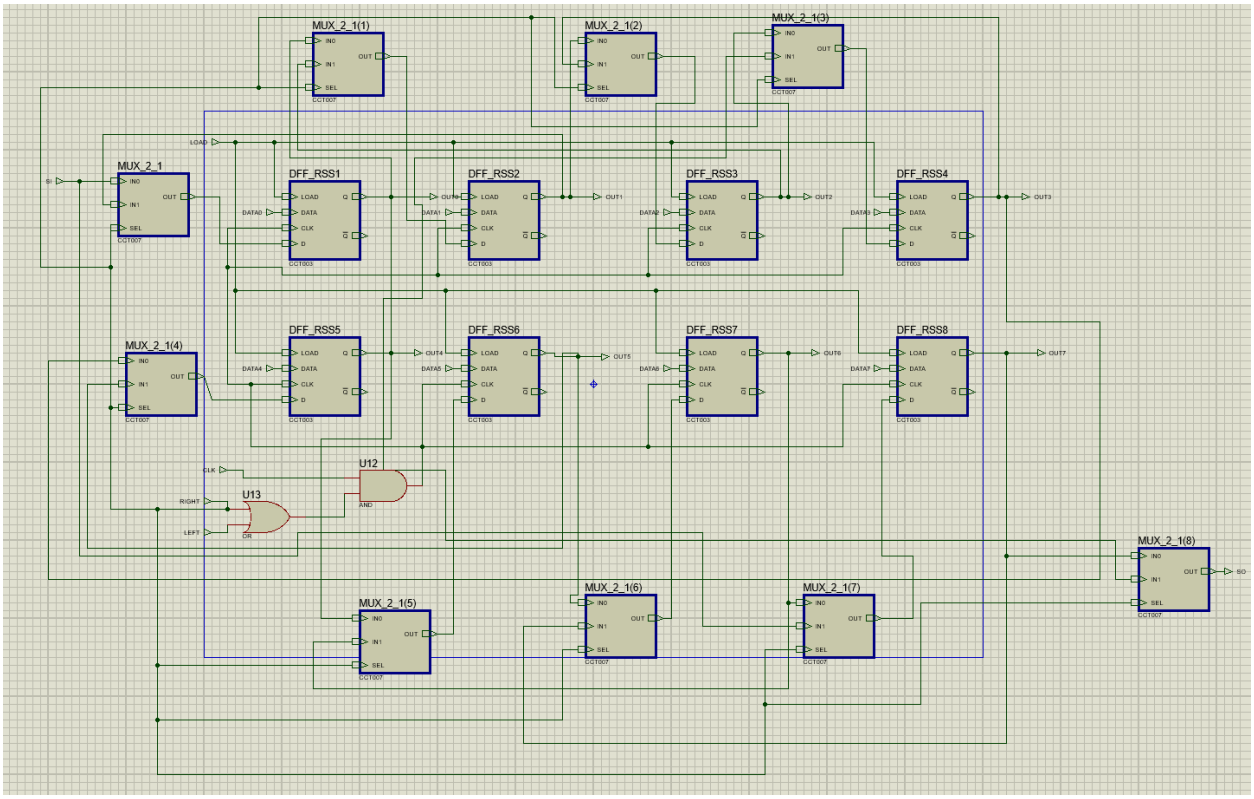


شکل ۸. نمای درون شمارنده ۸ بیتی.

در شمارنده که به پایین می شمارد ابتدا عدد مورد نظر را در شمارنده load می کنیم و سپس شروع به شمارش رو به پایین می کند.



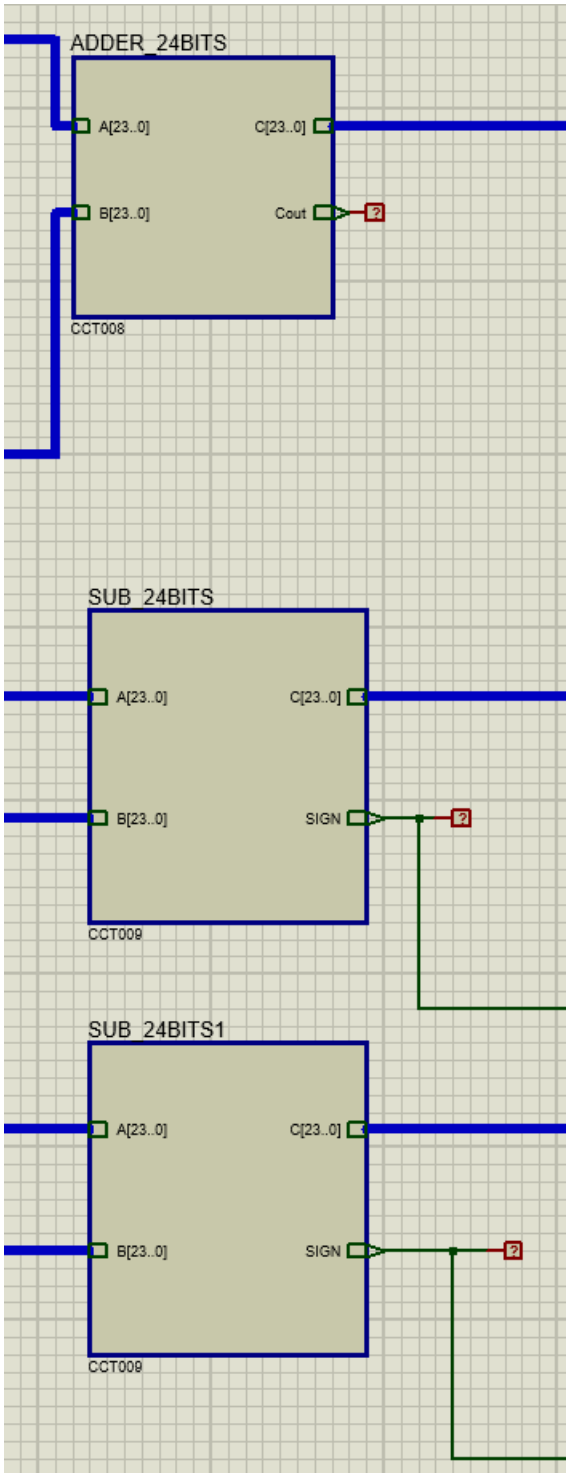
شکل ۹ . نمای درون شیفتر ۲۴ بیتی.



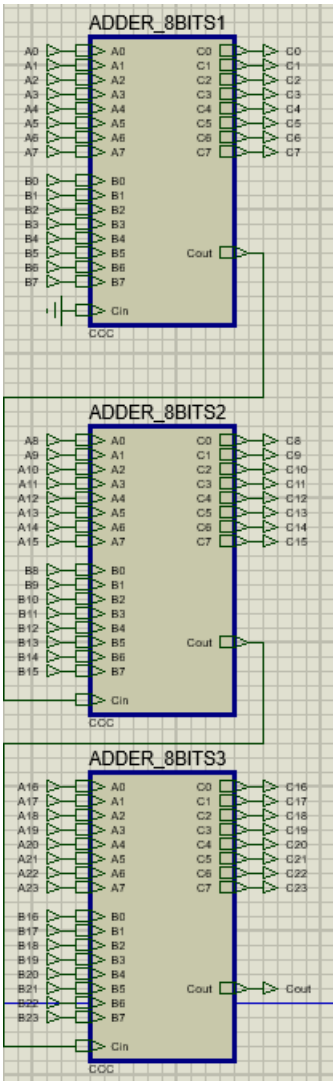
شکل ۱۰ . نمای درون شیفتر ۸ بیتی.

زمانی بیت E یک می شود که یا عدد شمارنده صفر شده باشد یا همه بیت های مانتیس صفر باشند و یا یکی از دو حالت INF, NAN برای یکی از ورودی ها پیش آمده باشد.

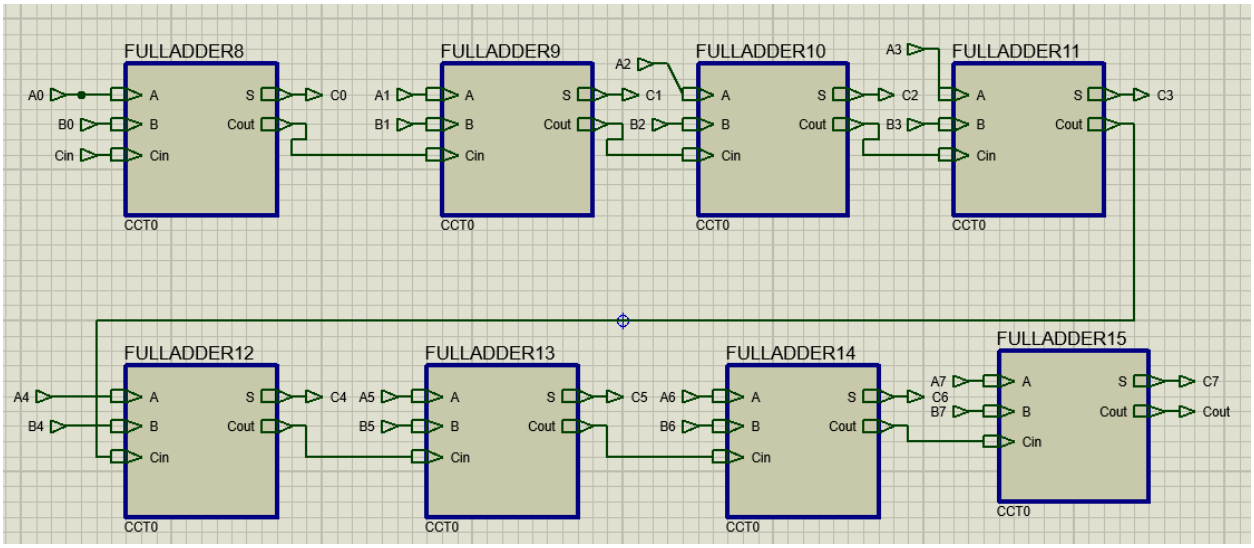
برای مانتیس ما همیشه سه مقدار A + B, A – B, B – A را محاسبه می کنیم که با توجه به ساین یکی را در ماکس انتخاب می کنیم.



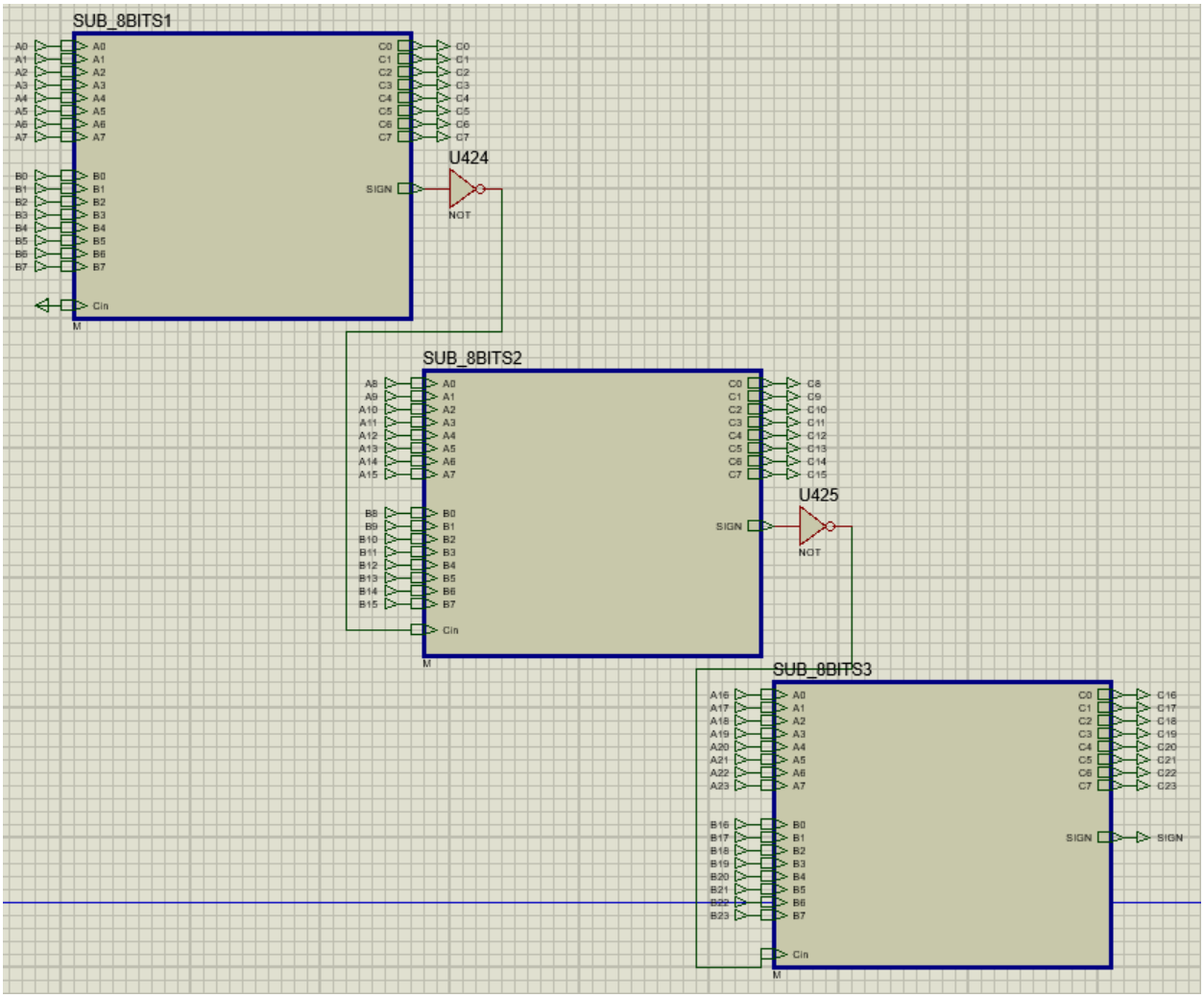
شکل ۱۱ . محاسبه A + B, A – B, B – A برای مانتیس های دو ورودی.



شکل ۱۲ . نمای درونی جمع کننده ۲۴ بیتی.

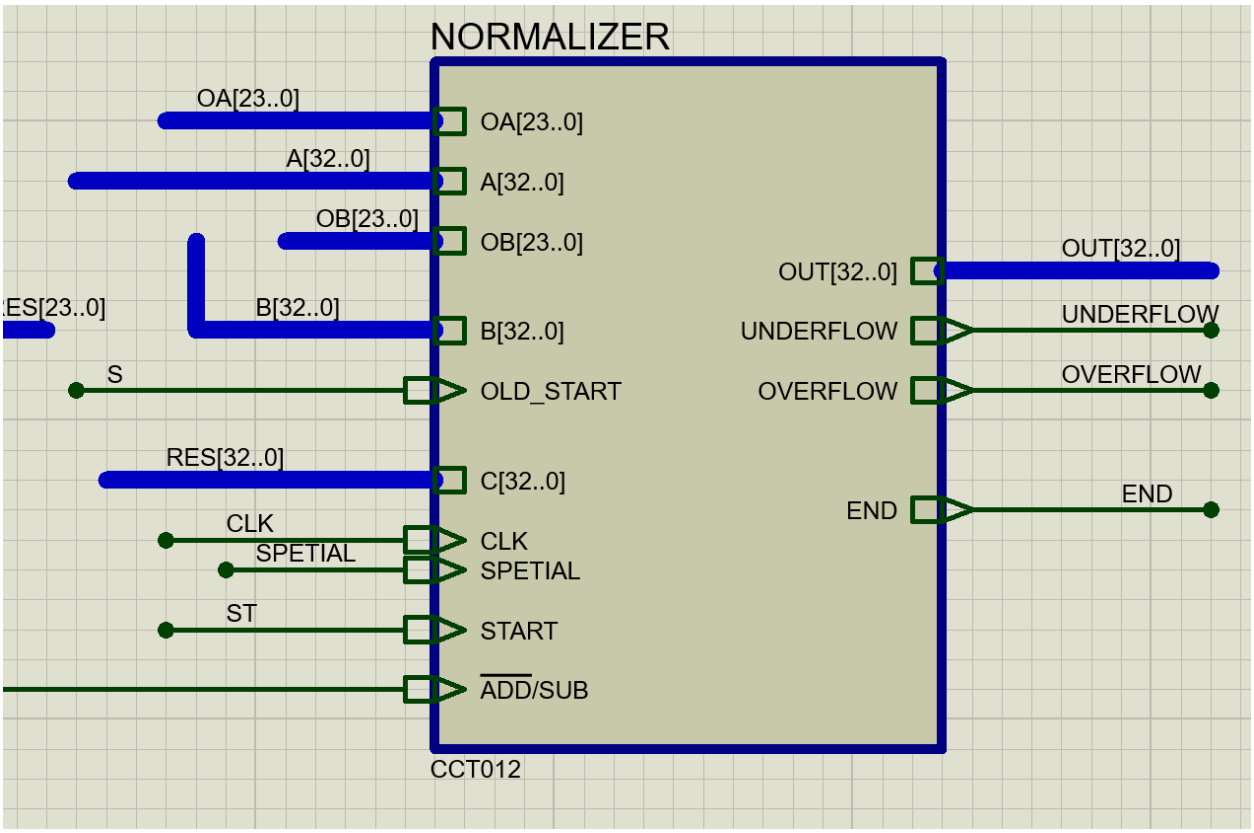


شکل ۱۳ . نمای درونی جمع کننده ۸ بیتی.

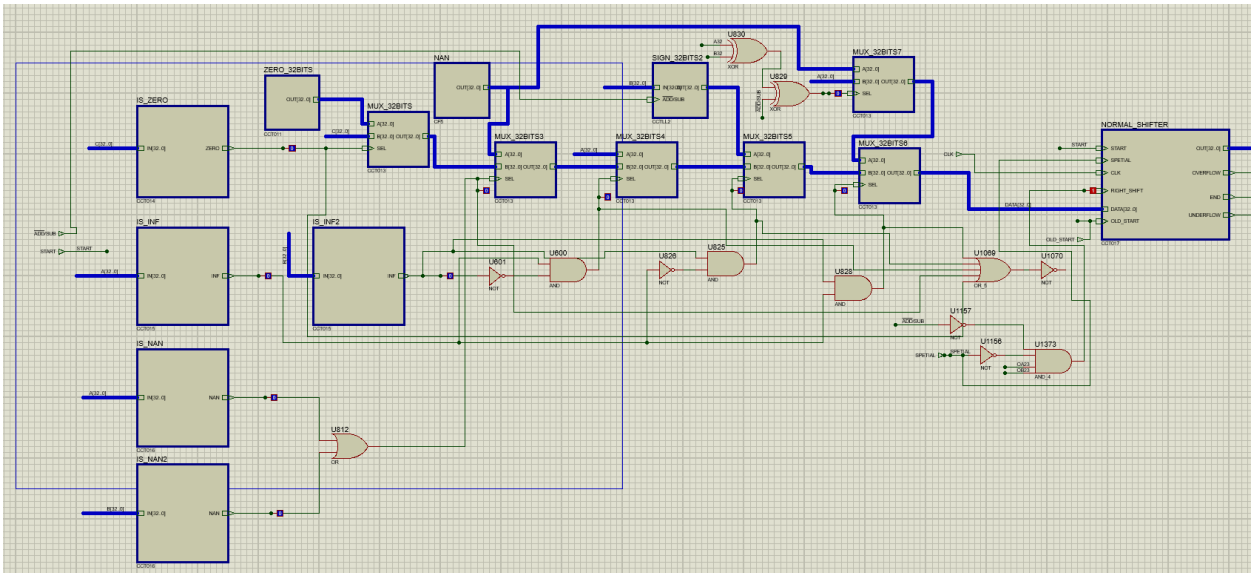


شکل ۱۴ . نمای درونی تفریق کننده ۲۴ بیتی.

و حال نوبت به قسمت نرمالایزر رسیده است.

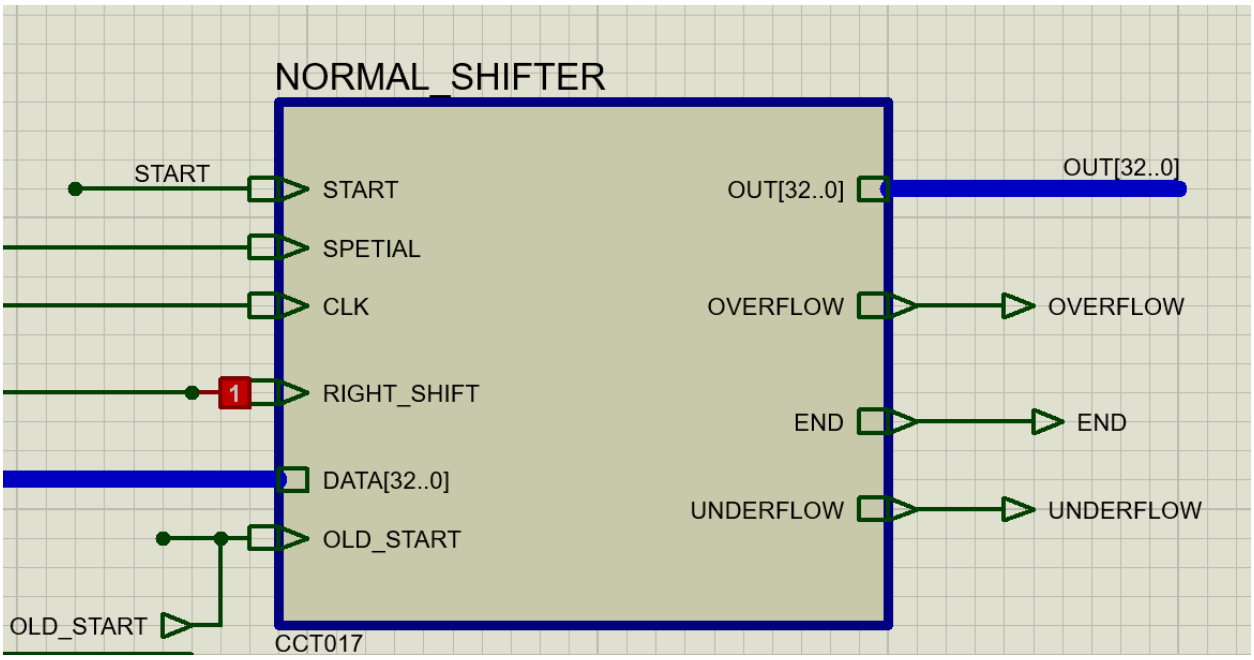


شکل ۱۵ . نمای بیرونی واحد normalizer.

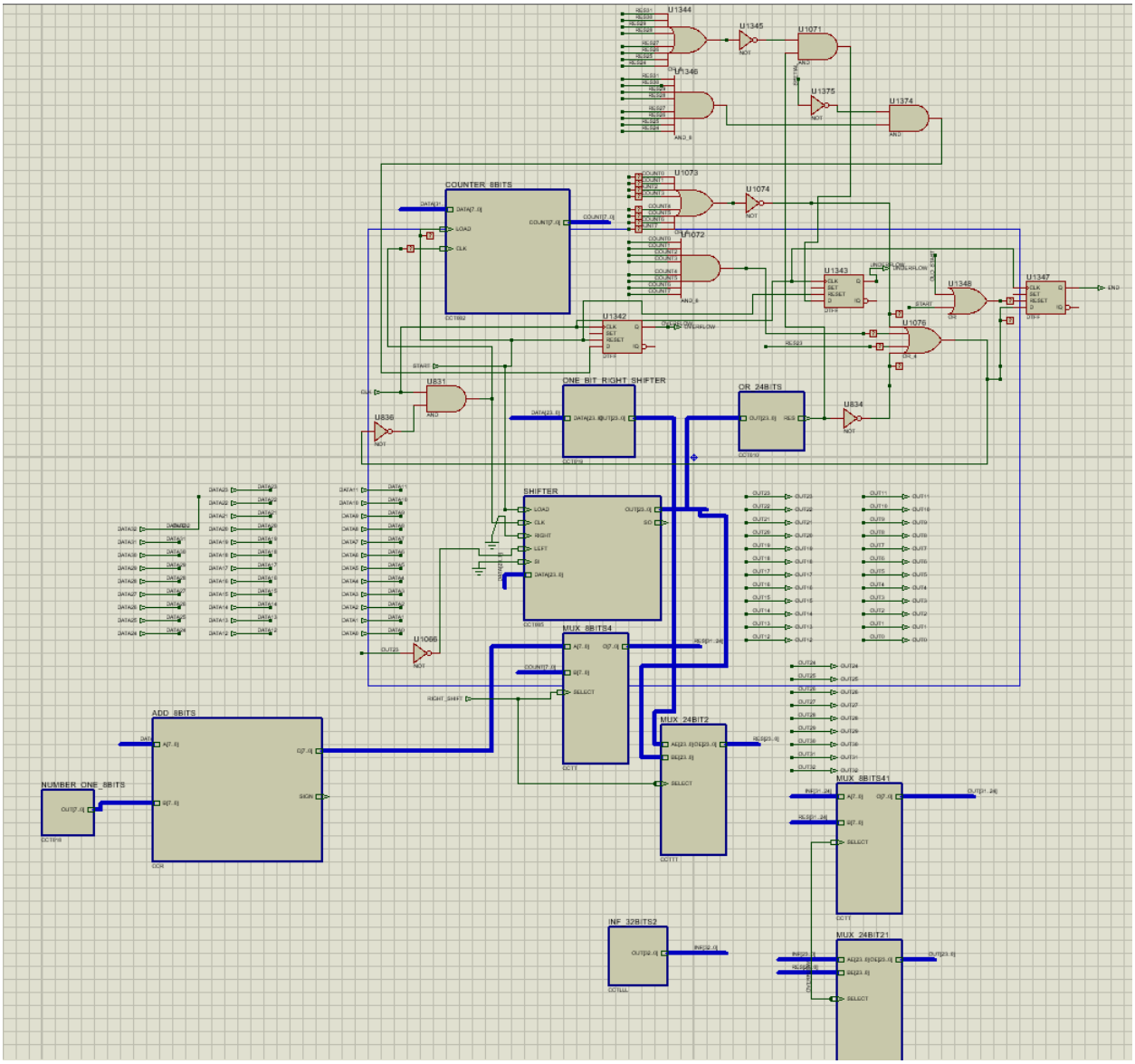


شکل ۱۶ . نمای درونی normalizer.

ابتدا جواب حالات خاص مانند INF + INF = INF یا INF – INF = NAN و ... را محاسبه کرده و خروجی می دهد. یا مثلا اگر NAN در یکی از ورودی ها باشد حاصل حتما NAN خواهد بود. سپس حاصل را به عنوان ورودی به ماژول NORMAL_SHIFTER می دهد.



شکل ۱۷ . نمای بیرون ماژول NORAML_SHIFTER.



شکل ۱۸ . نمای درون ماژول NORMAL_SHIFTER.

درون این ماژول ما شیفتر داریم که یا حالا شیفت به راست است و یا چپ. تنها حالتی که ما شیفت به راست داریم حالتی است که عبارت ما جمع باشد و البته با نماهای یکسان و در غیر این حالت تا زمانی که یک ۱ بیفتد سمت چپ، به چپ شیفت می دهیم و همزمان به کمک شمارنده ای که داریم، نما هم هندل می شود.

امتحان کردن چند ورودی :

Normal Case: Basic addition/subtraction of normalized floating-point number

Input A: 3.5 (0x40600000)

Binary: 0b01000000011000000000000000000000

Input B: 2.25 (0x40100000)

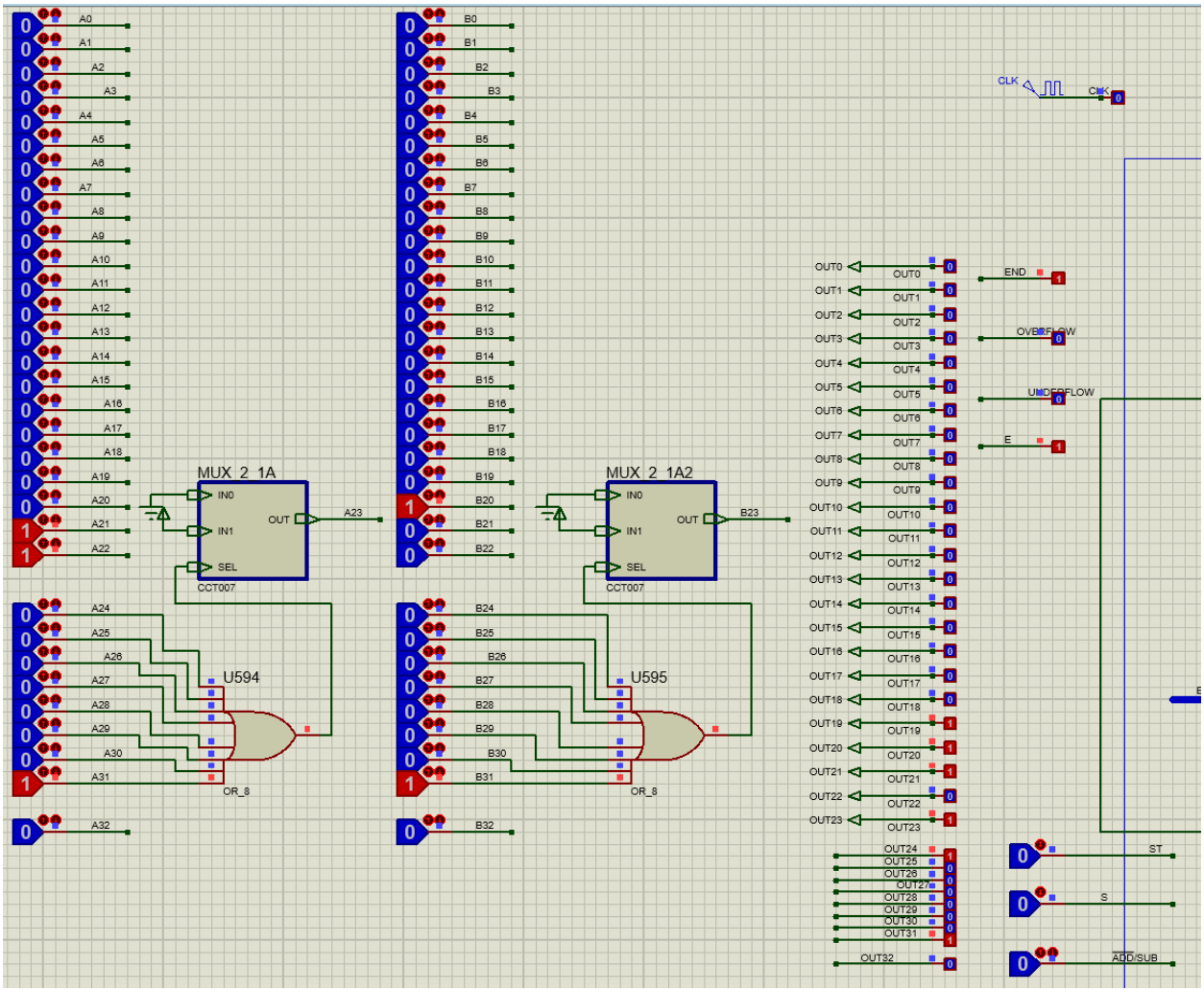
Binary: 0b01000000000100000000000000000000

Addition Output C: 5.75 (0x40B80000)

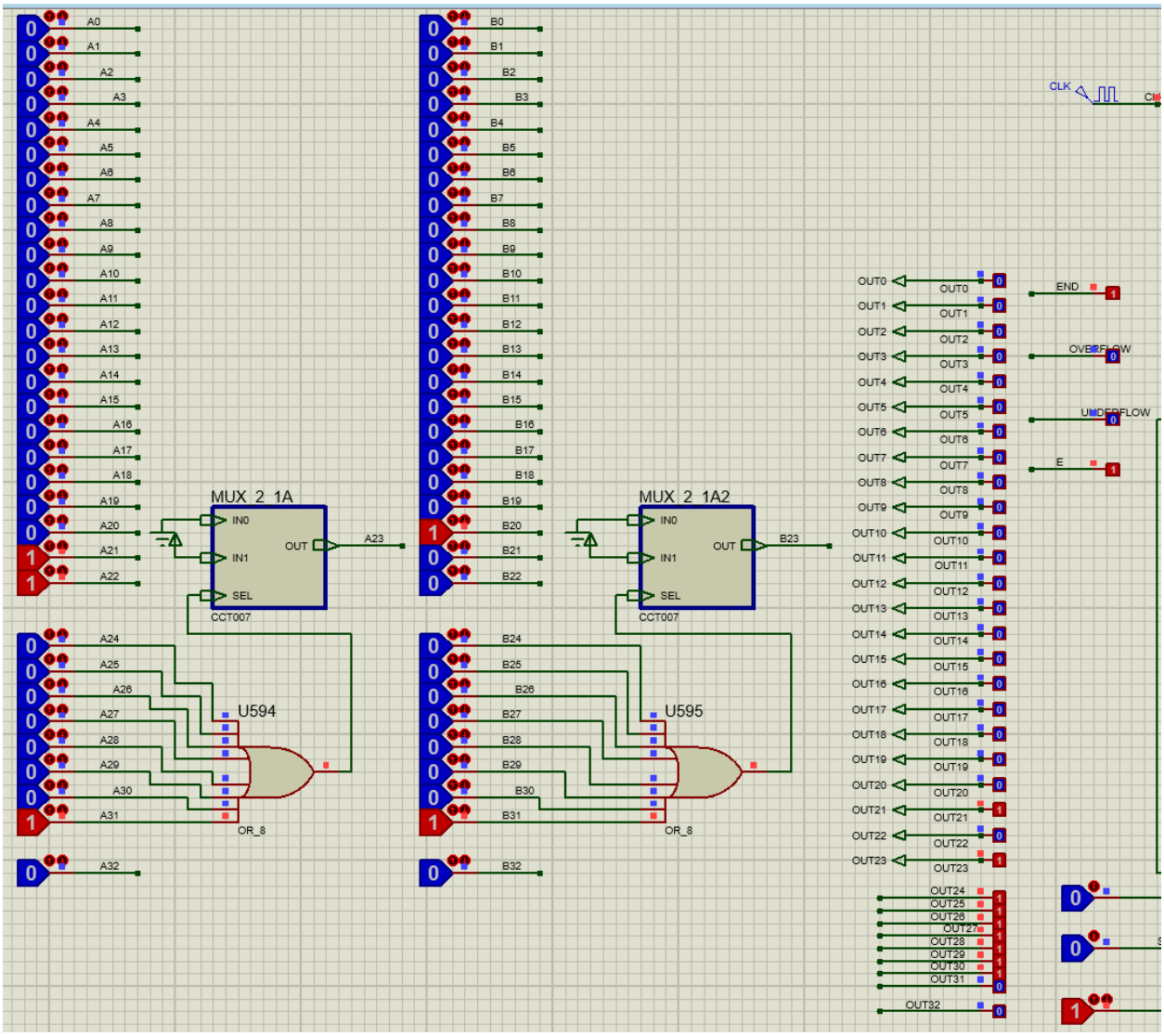
Binary: 0b01000000101110000000000000000000

Subtraction Output C: 1.25 (0x3FA00000)

Binary: 0b00111111101000000000000000000000



شکل ۱۹ . جواب جمع.



شکل ۲۰ . جواب تفریق.

Case 1: Addition resulting in an overflow

Input A: 3.4e38 (0x7F7FC99E)

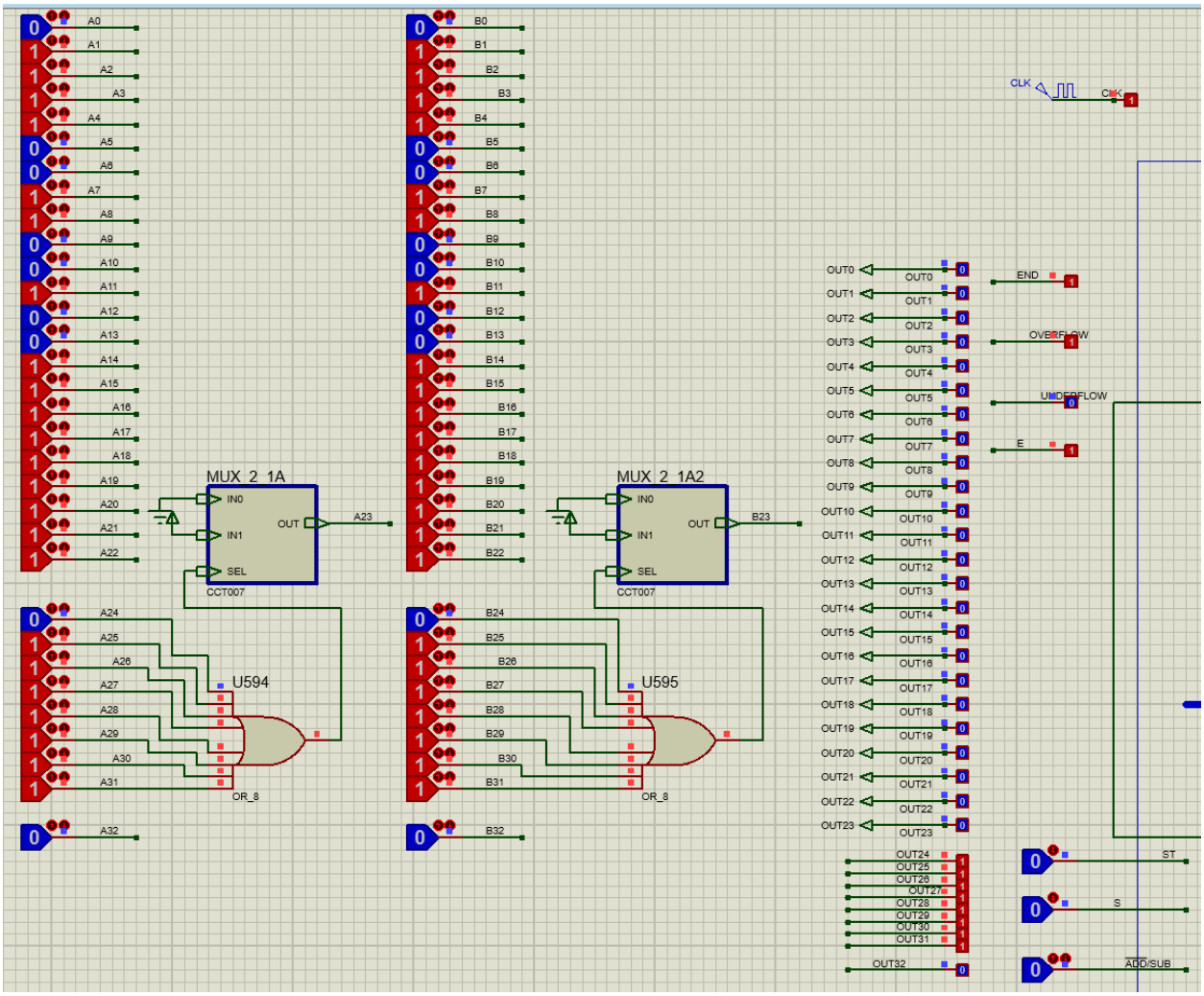
Binary: 0b01111111011111111100100110011110

Input B: 3.4e38 (0x7F7FC99E)

Binary: 0b01111111011111111100100110011110

Addition Output C: Overflow/Inf (0x7F800000)

Binary: 0b01111111100000000000000000000000



شکل ۲۱ . جواب جمع که نشان‌دهنده overflow می باشد.

Case 2: Subtraction resulting in an underflow

Input A: 1.4e-45 (0x00000001)

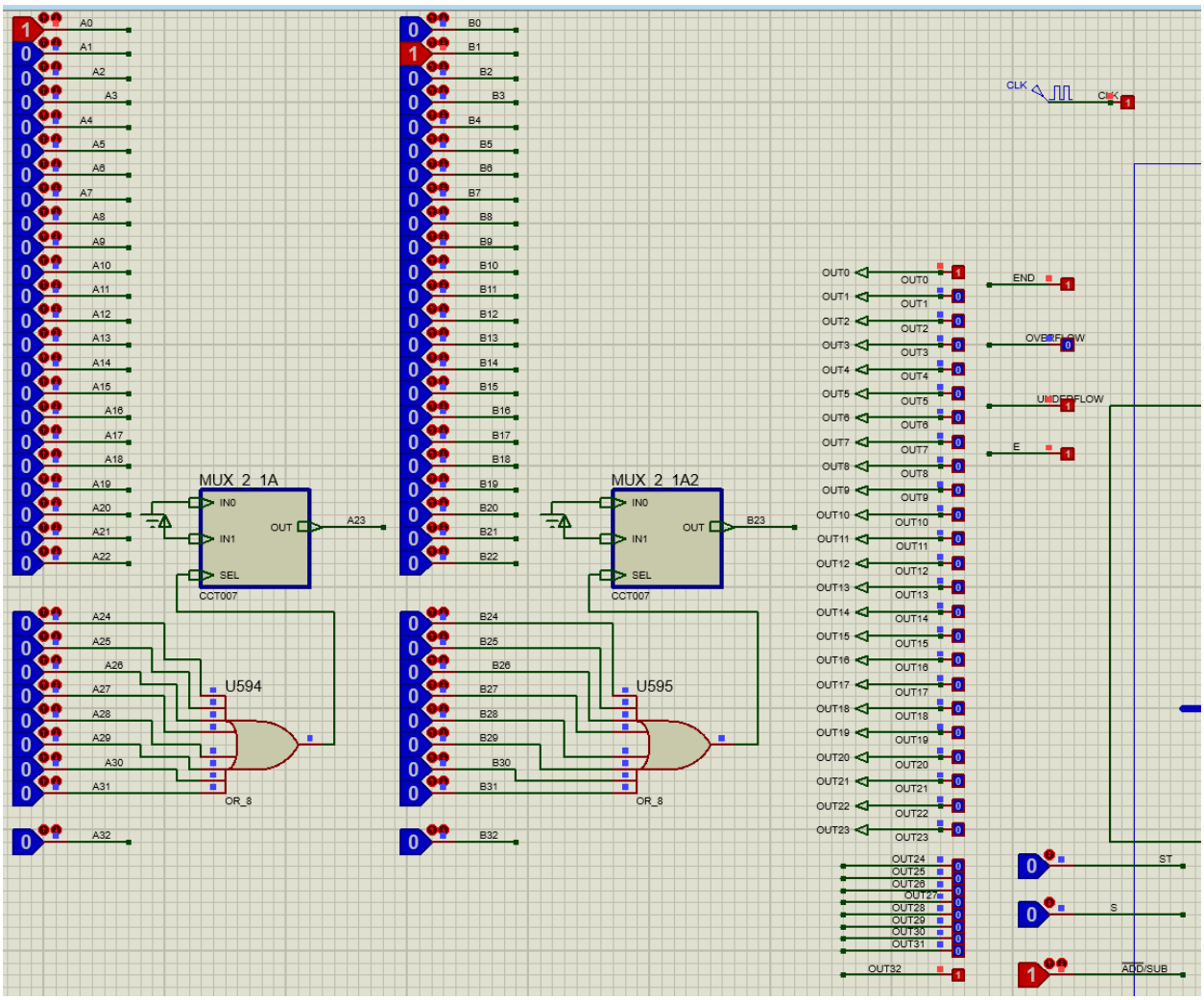
Binary: 0b00000000000000000000000000000001

Input B: 2.8e-45 (0x00000002)

Binary: 0b00000000000000000000000000000010

Subtraction Output C: Underflow

Binary: 0b10000000000000000000000000000000



شکل ۲۲ . جواب تفریق که underflow است.

Case 3: Addition of very small numbers

Input A: 1.4e-45 (0x00000001)

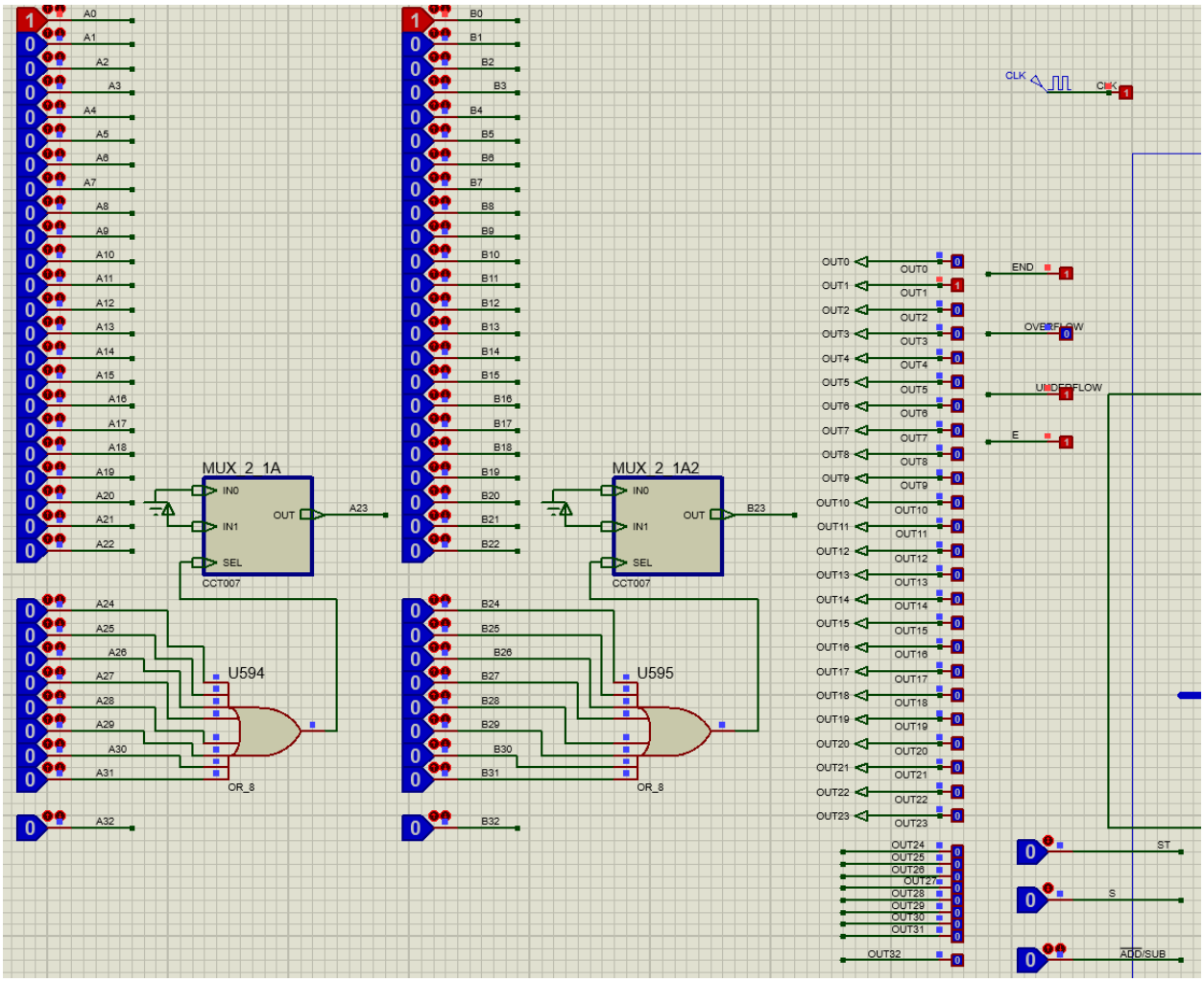
Binary: 0b00000000000000000000000000000001

Input B: 1.4e-45 (0x00000001)

Binary: 0b00000000000000000000000000000001

Addition Output C: 2.8e-45 (0x00000002)

Binary: 0b00000000000000000000000000000010



شکل ۲۳ . جواب جمع ۱ + ۱ = ۲ است.

Case 4: Subtraction of very large numbers

Input A: 1.0e38 (0x7E967699)

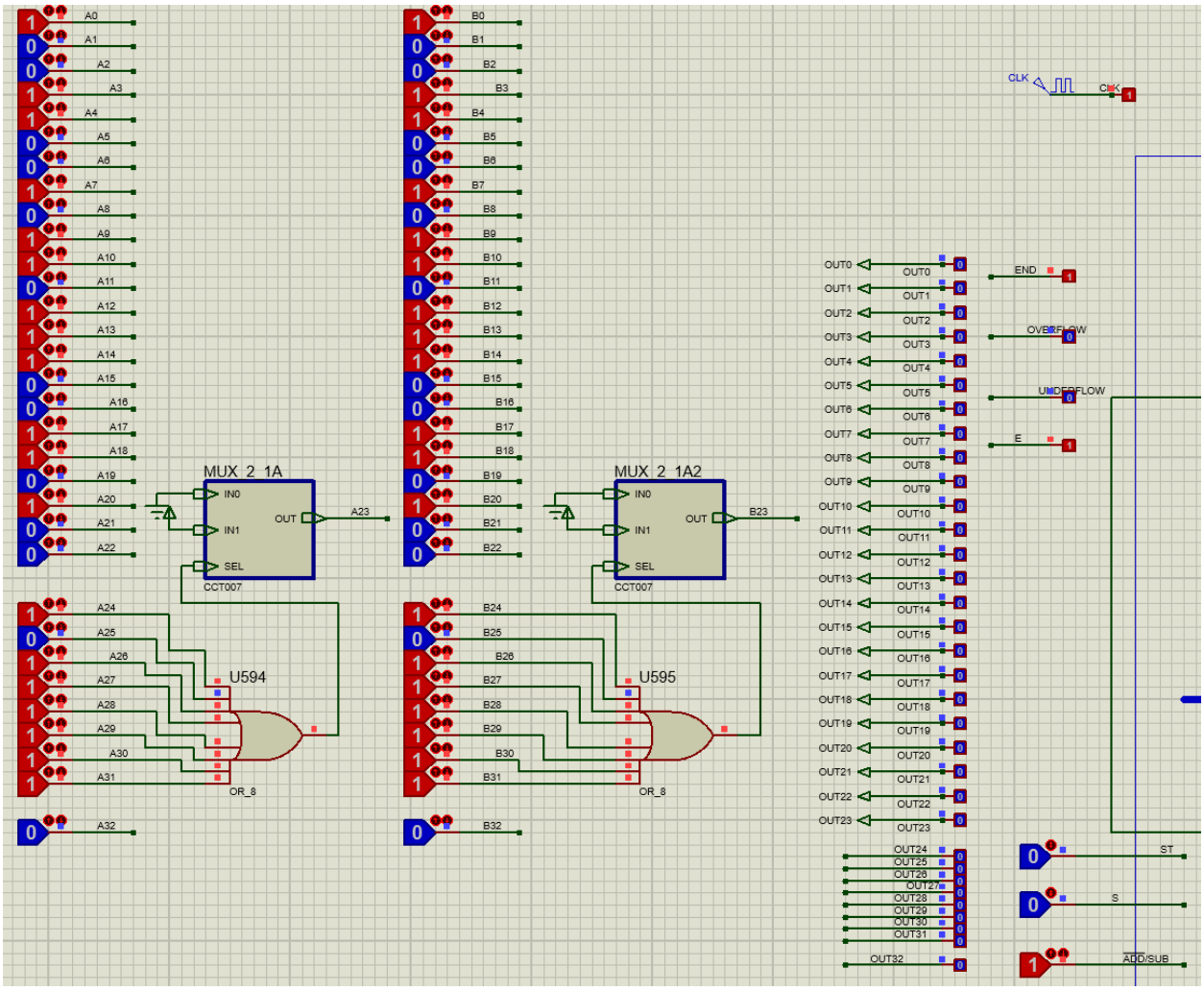
Binary: 0b01111110100101100111011010011001

Input B: 1.0e38 (0x7E967699)

Binary: 0b01111110100101100111011010011001

Subtraction Output C: 0 (0x00000000)

Binary: 0b00000000000000000000000000000000



شکل ۲۴ . جواب تفریق دو عدد بزرگ که حاصلی برابر با صفر به ما می دهد.

Case 5: Addition/subtraction involving zero

Input A: 0 (0x00000000)

Binary: 0b00000000000000000000000000000000

Input B: 2.25 (0x40100000)

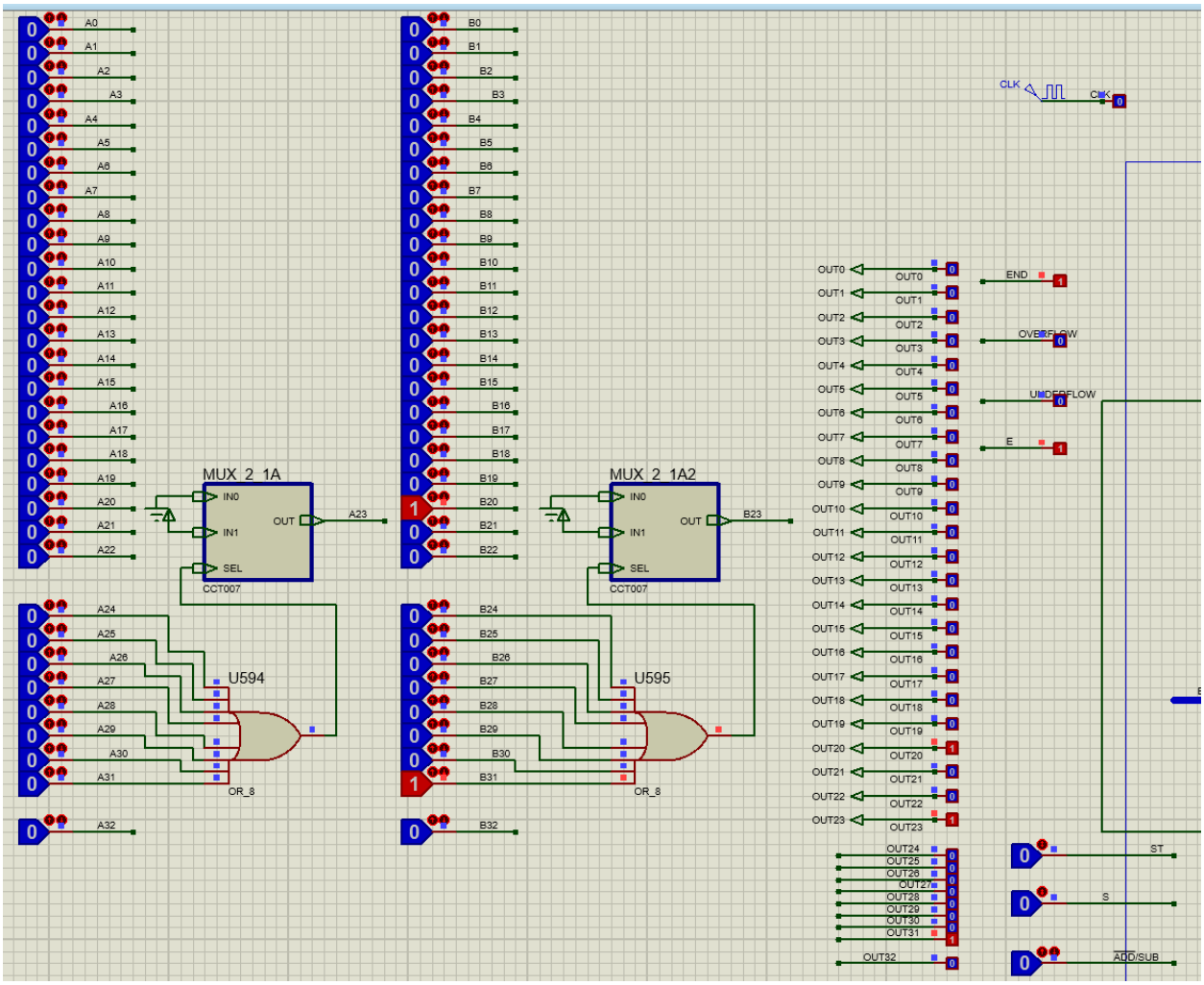
Binary: 0b01000000000100000000000000000000

Addition Output C: 2.25 (0x40100000)

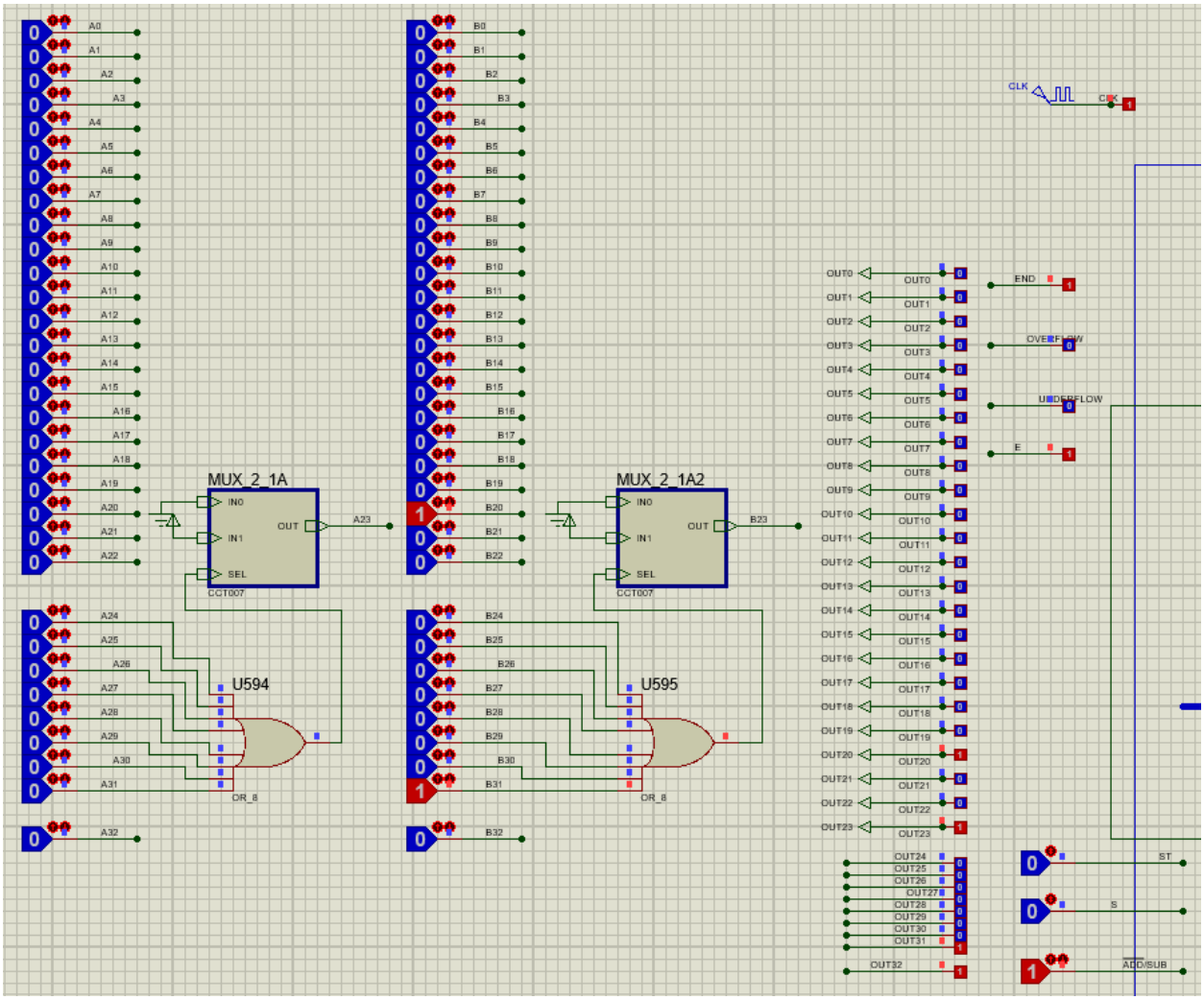
Binary: 0b01000000000100000000000000000000

Subtraction Output C: -2.25 (0xC0100000)

Binary: 0b11000000000100000000000000000000



شکل ۲۵ . جواب جمع با صفر.



شکل ۲۶ . جواب کم کردن از ۰ که همون یعنی منفی کردن.