

به نام خدا



آزمایشگاه معماری کامپیوتر

گزارش کار سوم

جمع/تفریق کننده شناور

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

پاییز ۱۴۰۱

استاد:

حمید سربازی آزاد

دستیار آموزشی:

عطیه یونسی

نویسندگان:

احمد رضا خناری

نگار عسکری

علی نظری

فهرست

۲	مقدمه
۳	گزارش آزمایش
۳	بخش ۱. طراحی اصلی
۸	بخش ۲. تست مدار
۱۰	نتیجه گیری

مقدمه

در این آزمایش یک جمع/تفریق کننده ی شناور طراحی می کنیم. اعداد ورودی و خروجی این مدار در قالب IEEE۷۵۴ در نظر گرفته شده اند. در این قالب و با توجه به پهنای ۱۲ بیتی که در نظر گرفته شده است، ۴ بیت به توان می رسد و ۷ بیت به بخش اعشاری و ۱ بیت هم که برای علامت است. در صورتی که حداقل یکی از چهار بیت توان نا صفر باشد، مقدار عددی که این قالب نمایش می دهد برابر با مقدار زیر است

$$(-1)^{sign} * 2^{e_3e_2e_1e_0-7} * 1.f_7f_6...f_0$$

که به این اعداد نرمال می گوییم و اگر هر ۴ بیت توان صفر باشند مقدار عددی برابر است با

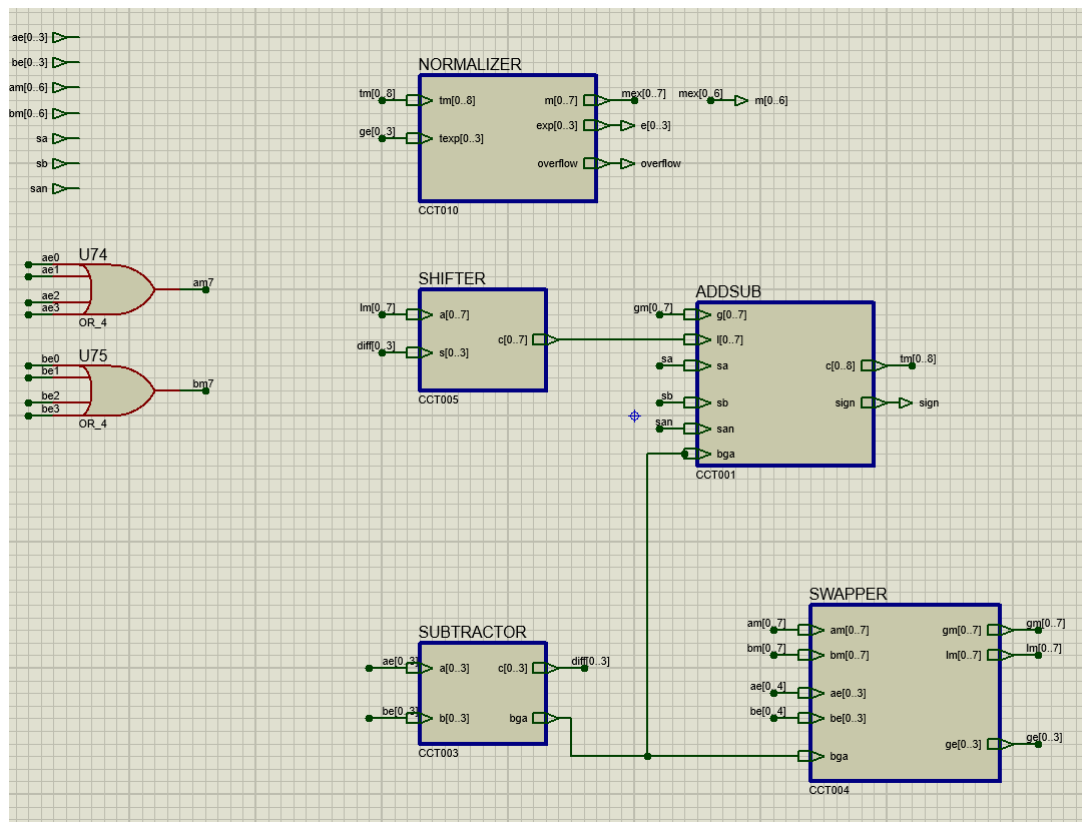
$$(-1)^{sign} * 2^{-6} * 0.f_7f_6...f_0$$

که این اعداد subnormal نامیده می شوند. به این ترتیب برای عدد صفر باید ۴ بیت توان و ۷ بیت اعشار صفر باشند. در این آزمایش مدار را به شکل ترکیبی طراحی می کنیم بنابراین نیازی به start و end نداریم. ورودی های مدار شامل ۴ بیت توان A و B که آن ها را با ae[0...3] و be[0...3] نشان می دهیم. و ۷ بیت هم هر کدام برای بخش اعشار داریم که با am[0...6] و bm[0...6] نمایش می دهیم و بیت های علامت را هم با sa و sb نمایش می دهیم و نهایتاً یک سیگنال san هم داریم که در صورت یک بودن یعنی تفریق باید بکنیم و در صورت صفر بودن یعنی جمع باید بکنیم. خروجی هم دقیقاً شامل بخش توان و اعشار و علامت است. و یک بیت هم برای نمایش داشتن overflow است.

گزارش آزمایش

بخش ۱. طراحی اصلی

حال کمی در مورد الگوریتمی که پیاده سازی خواهیم کرد صحبت می کنیم. الگوریتم به این شکل است که ابتدا عدد با تعداد بیت کمتر را پیدا می کنیم و اعشار آن را تا جایی به راست شیفت می دهیم که بتوان دو عدد یکی شود. سپس عملیات جمع و تفریق را به صورت کامل انجام می دهیم که برای انجام عملیات به صورت کامل و بدون داشتن سرریز نیاز به ۸ بیت مقدار و یک بیت علامت داریم. سپس به ۹ بیت حاصل نگاه می کنیم، اگر پر ارزش ترین بیت یک بود یعنی باید به توان یکی اضافه شود و مقدار اعشار یکی به راست شیفت داده شود. در غیر این صورت باید تا جایی که به اولین یک برسیم یا توان به صفر برسد اعشار را به راست شیفت دهیم. همچنین در صورت رفتن از حالت subnormal به حالت نرمال باید توان را از صفر به یک ببریم. این عملیات را شمای کلی طراحی به شکل زیر است:

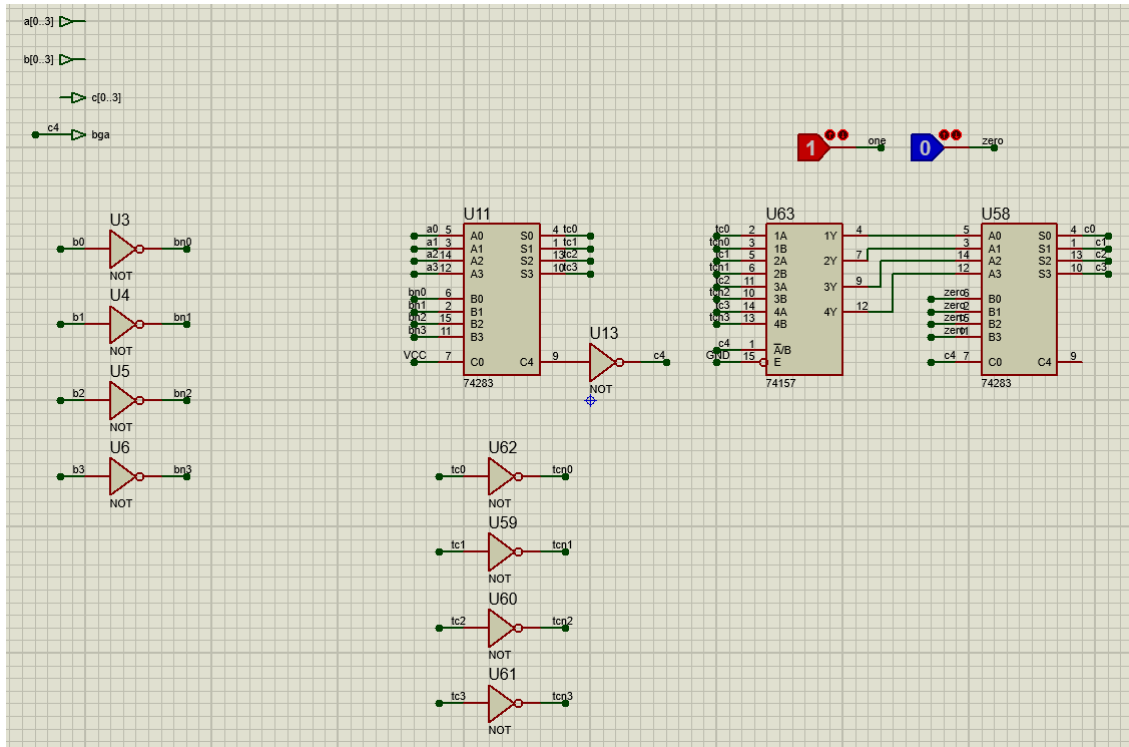


شکل ۱: Main Diagram

همانطور که مشخص است، ابتدا بخش subtractor قدر مطلق تفاضل بین توان ها را محاسبه کرده و همچنین از طریق سیگنال bga مشخص می کند که a بزرگتر است یا b بزرگتر است. سپس ماژول swapper بخش اعشاری عدد با توان بزرگ تر را در gm و بخش اعشاری عدد با توان کوچک تر را در lm و توان بزرگتر را در ge می ریزد. سپس ماژول shifter بخش اعشاری عدد

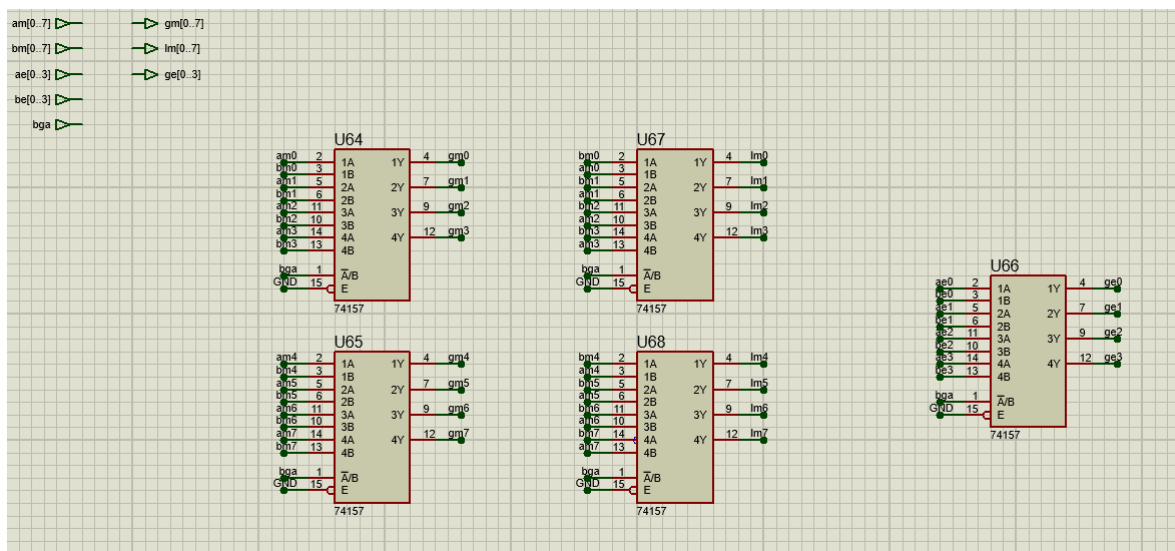
با توان کوچک تر را به اندازه مناسب به راست شیفت می دهد و پس از آن عملیات روی دو عدد توسط ماژول AddSub انجام می شود. نهایتاً هم ماژول normalizer کار نورمالایز کردن را انجام می دهد.

اگر کمی دقیق تر بخواهیم توضیح دهیم، داخل ماژول subtractor حاصل تفاضل توان ها محاسبه می شود و اگر تفاضل منفی شود، بیت ۴ هم منفی شده که در این صورت دوباره این تفاضل را از صفر کم می کنیم تا مثبت شود. به این ترتیب خود بیت ۴ می تواند عامل تعیین بزرگ تر بودن توان ها باشد از دیگری. شمای داخل subtractor در شکل زیر آمده است.



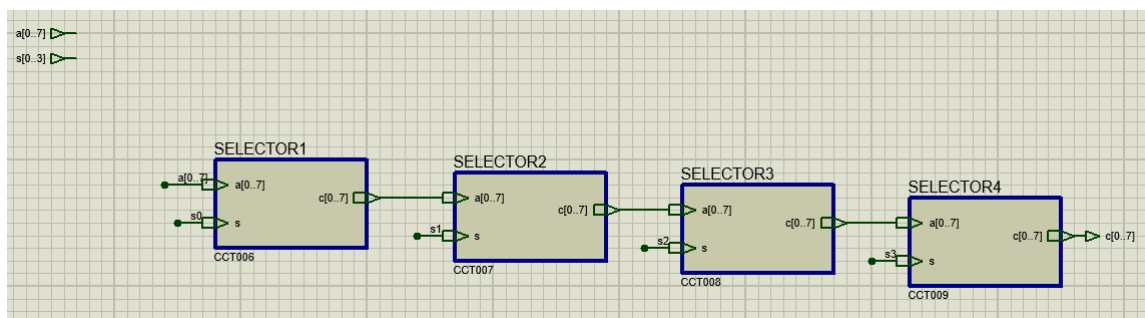
شکل ۲: Subtractor

داخل ماژول swapper هم به شکل زیر است:



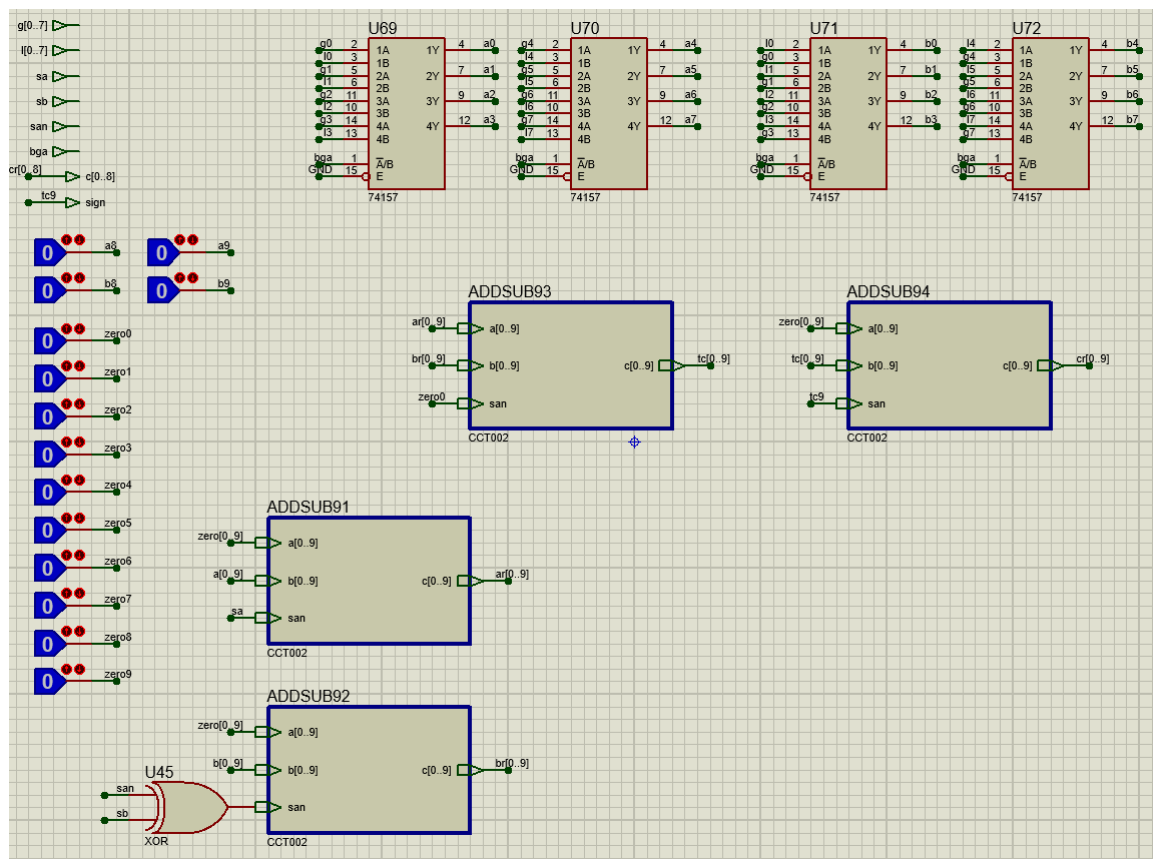
شکل ۳: Swapper

در این ماژول تنها خروجی های gm و lm و ge مشخص می شود.
داخل ماژول shifter هم به شکل زیر است:



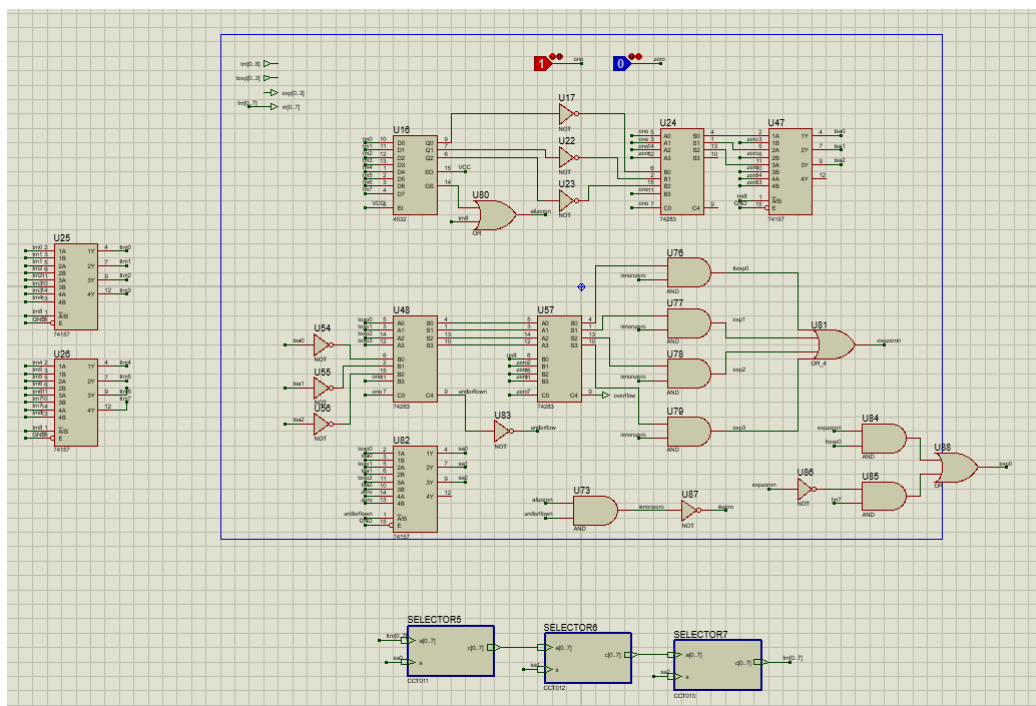
شکل ۴: Shifter

پس برای شیفต์ دادن lm از shifter barrel استفاده می کنیم. در selector ها هم که صرفاً بر اساس اینکه s صفر است یا یک، یکی از ورودی ها را انتخاب می کنیم. این selector ها هم طراحی خیلی ساده ای دارند و از ۷۴۱۵۱ ساخته شده اند. در بخش اصلی جمع یا تفریق که بخش اعشاری جمع یا تفریق می شود به مانند زیر طراحی انجام شده است:



شکل ۵: AddSub

در این بخش خروجی را حتماً unsigned در نظر می‌گیریم یعنی اگر خروجی منفی باشد حتماً آن را از صفر کم می‌کنیم. و علامت را در بیت sign خروجی می‌دهیم. درون جمع یا تفریق کننده‌های داخل این ماژول هم پیاده‌سازی ساده‌ای وجود دارد و از قطعه ۷۴۲۸۱ استفاده می‌کنیم. حال به ماژول نورمالایزر می‌رسیم:



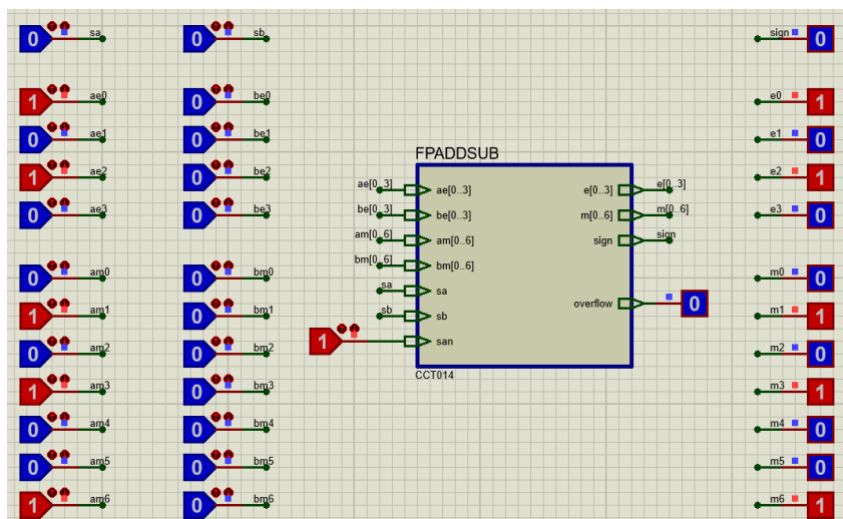
شکل ۶: Normalizer

این ماژول با توجه به توان و اولین بیت یک پر ارزش در حاصل اعشاری ۹ بیتی تعداد شیفت به چپ و راست را تعیین می کند و همچنین توان جدید را نیز محاسبه می کند. در محاسبه توان جدید این نکته مهم است که اگر حاصل subnormal بوده یعنی توان ۴ بیت صفر بوده اما بیت هشتم حاصل یک باشد، باید توان را به یک تبدیل کنیم تا خروجی نرمال شود. درون عکس، نحوه محاسبه شیفت دادن و به دست آوردن توان جدید در بخش بالایی مدار است. و عملیات شیفت دادن به راست یا چپ هم در بخش پایینی مدار انجام می شود. دقت کنید که در این بخش از CMOS ۴۵۳۲ استفاده شده که یک Encoder Priority است و مکان اولین بیت یک از سمت پر ارزش را مشخص می کند.

بخش ۲. تست مدار

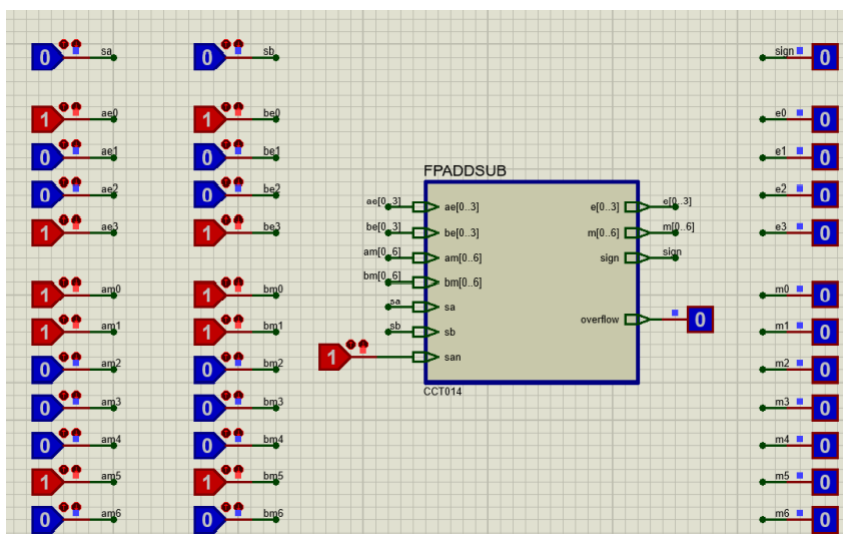
چند نمونه را به مدار می دهیم:

$$1.1001010 * 2^{-2} - 0 = 1.1001010 * 2^{-2}$$



شکل ۷: Example

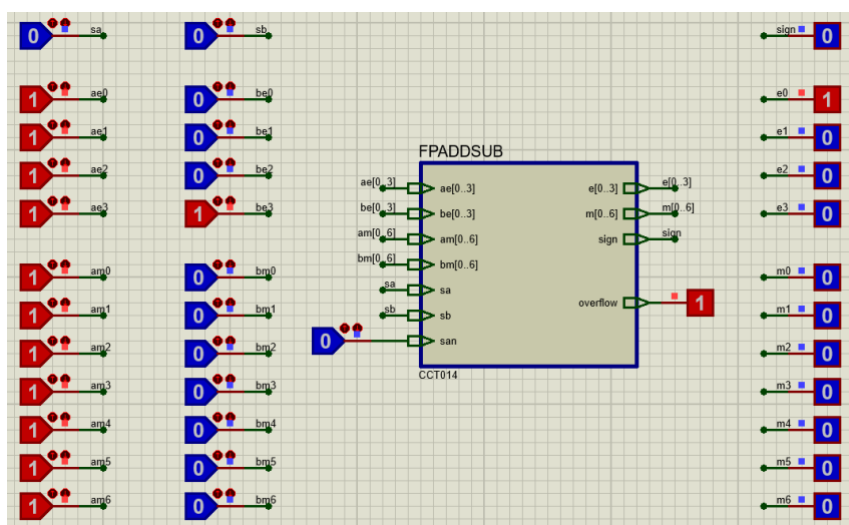
$$1.0100011 * 2^2 - 1.0100011 * 2^2 = 0$$



شکل ۸: Example

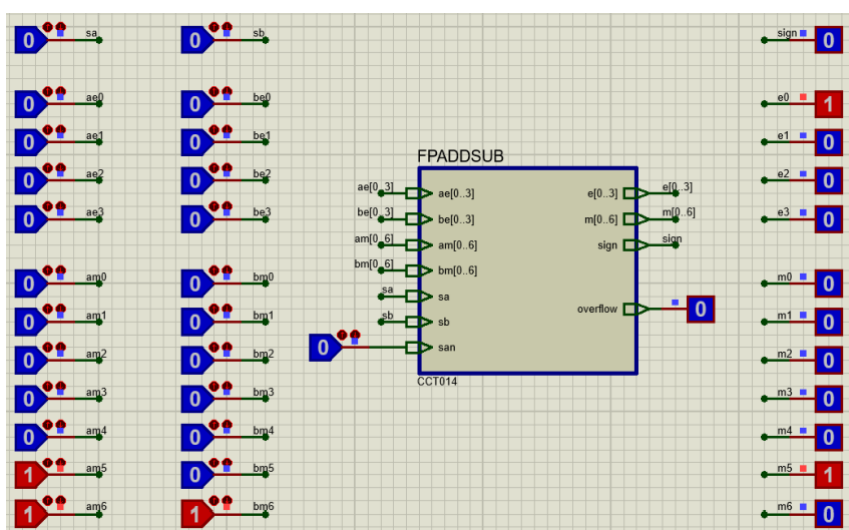
مثال زیر دچار overflow است

$$1.1111111 * 2^8 + 2^1 = 2^9$$



شکل ۹: Example

در مثال زیر هم دو عدد subnormal را جمع می کنیم ولی جمع نهایی normal است:
 $0.11 * 2^{-6} + 0.1 * 2^{-6} = 1.01 * 2^{-6}$



شکل ۱۰: Example

نتیجه گیری

در نتیجه ما با کمک الگوریتم گفته شده و طراحی ترکیبی ای که داشتیم موفق شدیم مدار جمع یا تفریق کننده اعداد ممیز شناور را بسازیم و در تست هایی که از آن گرفتیم ۷ خروجی به درستی نمایش داده شد.