

آزمایش دهم

هدف آزمایش: تفريق کننده ممیز شناور

نکته: میتواند با جمع کننده تطابق داشته باشد از منظر تعداد بایاس و ...

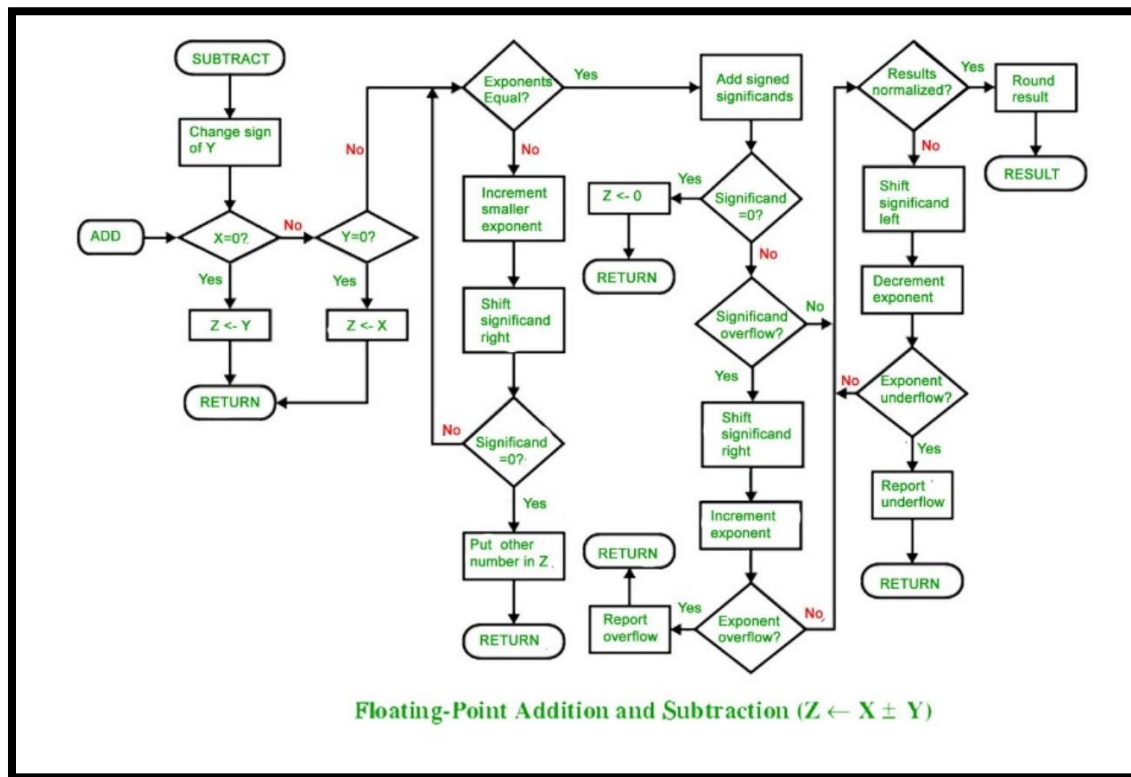
شرح آزمایش:

برای ساخت این ماژول ابتدا اقدام به تعریف Port ها میکنیم. پورت A و B دو عدد Floating Point ما هستند که از ساختار زیر پیروی میکنند:

- مجموعاً ۳۲ بیت هستند
- بیت ۳۱ برای علامت است
- بیت ۳۰ تا ۲۳ برای Exponent در بایاس ۱ است
- بیت ۲۲ تا ۰ هم برای Fraction است

یک تک بیت Ready (برای فهمیدن آنکه جواب آماده است یا نه) و همچنین یک تک بیت Overflow برای تشخیص سرریز (یا همان خطای محاسبه در اعداد اعشاری) استفاده میکنیم. و در نهایت خروجی که یک عدد ممیز شناور با همان فرمت مشابه A و B است.

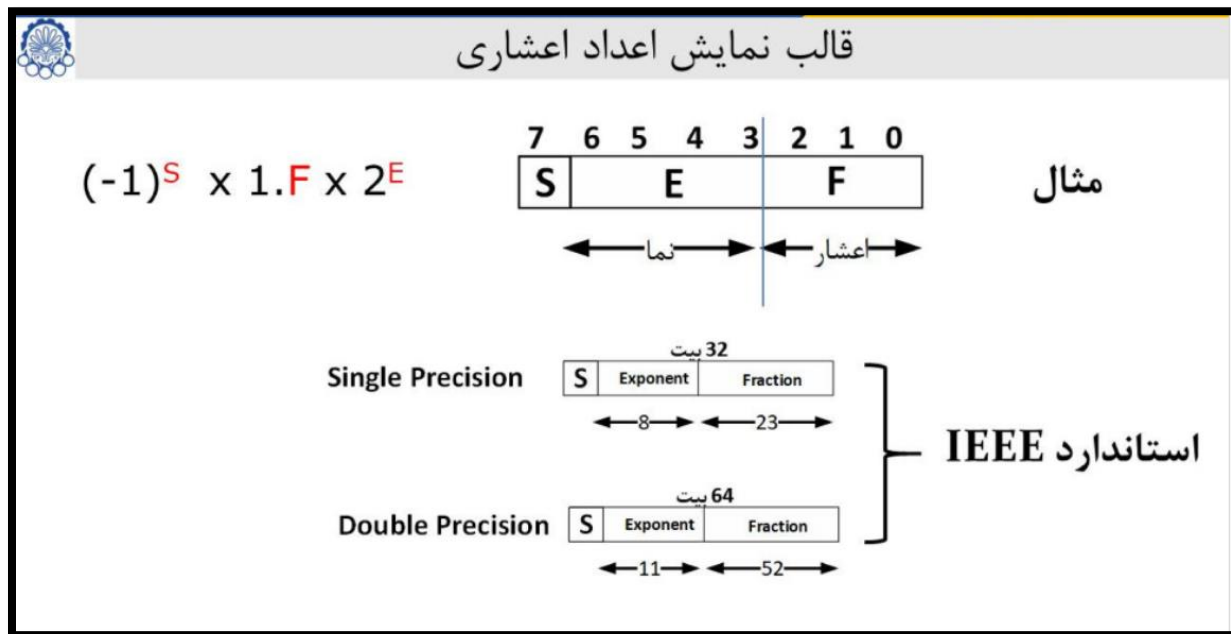
حال وارد بخش اصلی میشویم که بیس فلوچارت زیر طراحی شده است:



خب ابتدا چند متغیر تعریف میکنیم که هرکدام آنها را در بخش خود توضیح خواهیم داد.

چندین حالت را بررسی میکنیم:

- حالت ۱: در این حالت A و B برابر یکدیگر اند بنابراین حاصل تفاضل آنها صفر خواهد شد، نکته ای که وجود دارد آن است که خروجی باید ۳۲ بیت تماما صفر باشد (قرارداد در درس برای نمایش عدد صفر). درنهایت بیت Ready را یک میکنیم
- حالت ۲: یکی از اعداد A یا B صفر باشند در این صورت جواب اگر A صفر باشد قرینه B و در غیر این صورت خود A میشود
- حالت ۳: ابتدا بررسی میکنیم که عدد A کوچکترین عدد هست یا خیر (اینکار را با استفاده از بیت های ۳۰ تا ۲۳ مربوط به Exponent انجام میدهیم. اگر A کوچکترین بود. حال نوبت به temp میرسد که پیشتر آن را تعریف کرده بودیم (بخش Fraction کانکت میشود با ۱ که نماینگر بخش صحیح اعشار است - چرا که ما در Fraction فقط و فقط خود اعشار را نگه میداریم) - یادآوری:



- حال در صورتی که A کوچکتر باشد، شیفت راست می‌دهیم A temp را به اندازه تفاضل Exponent، B و A. در غیر این صورت همینکار را برای temp B انجام می‌دهیم اما به اندازه تفاضل Exponent عکس آن یعنی A از B
- حال ۳ حالت را بررسی می‌کنیم: اگر بیت ۳۱ ام (یا همون علامت) برابر باشند و هر دو منفی اگر temp A که در مرحله قبل حساب کردیم بزرگتر از temp B بود، temp Result Sub برابر با تفاضل temp A از temp B خواهد بود. عکس آن نیز برای حالت دیگر اعمال می‌شود.
- در حالت بزرگتر بیت علامت خروجیمان منفی و در غیر این صورت مثبت است
- اگر A و B هر دو مثبت باشند و همچنین temp A بزرگتر از temp B باشد تفاضل را انجام می‌دهیم و در temp Result Subtract قرار می‌دهیم. (عکس آن برای حالت دوم)
- علامت را بر اساس مدل تفاضل تعیین می‌کنیم اگر A بزرگتر بود قطعا نتیجه مثبت خواهد بود و در غیر این صورت منفی
- حالت آخری که در اینجا بررسی می‌شود حالتی که دو علامت یکسان نباشند در این صورت حاصل را در temp Result Sum میریزیم که شمال جمع temp A (کانکت شده با یک صفر) و temp B (کانکت شده با یک صفر) است. اینجا Operation جمع است.
- در نهایت بیت علامت برابر است با بیت ۳۱ ام

- حال وارد بخش آخر بررسی و تحلیل خود میشویم. در این حالت ابتدا چک میکنیم آیا فرایندی که رخ داده جمع است یا نه

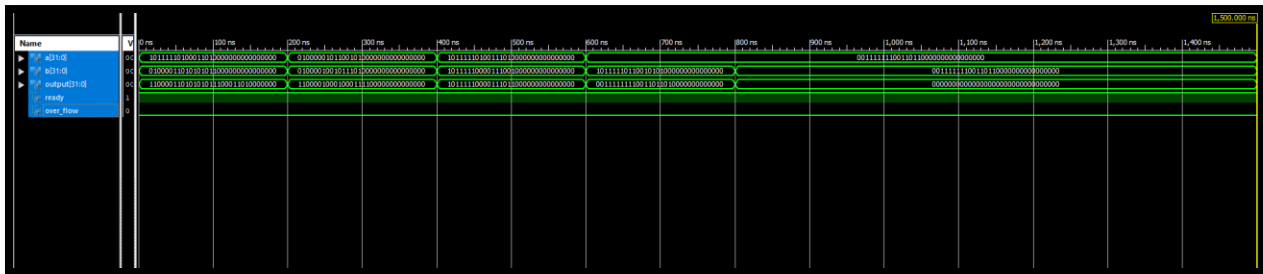
- اگر جمع بود: ۳ حالت رخ میدهد:

- ایندکس ۲۴ ام حاصل جمع موقت برابر با ۱ است: خروجی ما (بیت ۲۲ تا ۰) برابر با جمع موقت خروجی (بیت ۲۳ الی ۱) است. در این حالت به طور واضح Overflow داریم و فقط یک کار کوچک میماند که باید انجام شود آن هم بیت های ۳۰ تا ۲۳ خروجی است (بیت ۳۱ قبلا به دست آمده) که اینکار را بر اساس آنکه آیا A کوچکترین عدد است یا خیر انجام میدهیم. اگر بود کافی است بی علامت B (بیت ۳۰ تا ۲۳) را با ۱ جمع کنیم و حاصل را در خروجی میریزیم. (عکس اینکار برای حالت دیگر) و درنهایت Ready را یک میکنیم
- ایندکس ۲۳ ام حاصل جمع موقت برابر با ۱ است: بیت ۲۲ تا صفر خروجی ما برابر با بیت متناظر آن در حاصل جمع موقت است و بر اساس آنکه A کوچکترین عدد است یا خیر بیت ۳۰ تا ۲۳ خروجی را تعیین میکنیم اگر بود میشود متناظر آن در B در غیر آن صورت متناظر آن در A
- هیچکدام از دو حالت فوق رخ ندهد: این بخش اندکی پیچیده تر از بخش های قبل است برای این مرحله میخواهیم تعداد بیت هایی که از اندیس ۲۲ به بعد حرکت میکنیم تا به اولین ۱ برسیم را بدست آوریم و سپس از آن برای شیفت دادن استفاده کنیم. به همین منظور متغیر صحیح First One Counter را تعریف میکنیم. همچنین First One Flag برای اینکه بفهمیم پیدا شده است یا خیر. بعد از پیدا کردن آن نوبت به همان کار های سابق میرسد، ابتدا خروجی موقت جمع را (مقدار قبلیش را) به تعداد First One Counter شیفت به چپ میدهیم و آن را برابر با بیت ۲۲ تا ۰ متناظرا در خروجی میکنیم. اگر A کوچکتر بود برای بیت ۳۰ تا ۲۳، بیت متناظر آن در B را با نات ۸ بیتی عدد integer Counter مون به علاوه ۱ جمع میکنیم. برای حالت دیگر صرفا به جای بیت متناظر در B از بیت متناظر در A استفاده میکنیم.
- بیت Ready را ۱ میکنیم

- در غیر این صورت دو حالت را بررسی خواهیم کرد:

- بیت ۲۳ ام خروجی موقت تفاضل ۱ است : این حالت بسیار ساده است تنها کافی است بیت ۲۲ تا صفر خروجی را متناظر با خروجی موقت تفاضل آن قرار دهیم و برای بیت های ۳۰ الی ۲۳ متناظر آن را ست کنیم (براساس آنکه A کوچکتر است یا بزرگتر – اگر کوچکتر بود B در غیر اینصورت A)
- حالت فوق رخ ندهد، از همان استراتژی سابق در شمارش تا زمانی که به اولین ۱ برسیم بهره میگیریم. مدل شیفت و جمع هیچ تفاوتی با مشابه آن در حالت جمع ندارد در این مرحله صرفا با Temp Result Subtract سر و کار داریم.
- درنهایت Ready را یک میکنیم و مراحل به پایان میرسد

در نهایت ۵ مثال در تست بنچ طرح کردیم که بر اساس آن خروجی مطلوب به شکل زیر خواهد بود:



تمامی فایل ها در پیوست موجود است.

موفق باشید