

## گزارش فاز چهارم پروژه معماری کامپیوتر – گروه Mascarpone

محمد مهدی قیدی – ۹۸۱۰۵۹۷۶

حسین آقایی – ۹۸۱۰۵۶۱۹

میلاد سعادت – ۹۸۱۰۰۴۴۲

پرهام چاوشیان – ۹۸۱۰۰۱۱۸

در این فاز ما قصد داریم تا یک کمک‌پردازنده برای انجام محاسبات ممیز شناور را طراحی کنیم. برای پیاده‌سازی موارد مورد نیاز برای این کمک‌پردازنده از تعدادی ماژول کمکی استفاده می‌کنیم که هر کدام وظیفه انجام یک یا دو تا از محاسبات را دارند که این ماژول‌ها را در ادامه معرفی می‌کنیم و عملکرد آن‌ها را شرح می‌دهیم:

**ماژول جمع/تفریق:** این ماژول در ورودی دو عدد ممیز شناور را به همراه عملیاتی که باید بر روی آن‌ها انجام دهد دریافت می‌کند. برای انجام عملیات مورد نظر، ابتدا عدد با نمای بیشتر را پیدا می‌کنیم و ومانتیس عدد با نمای کوچکتر را به اندازه اختلاف نماها شیفت می‌دهیم. برای انجام عملیات مانتیس‌ها، به کمک یه سیگنال کمکی تعیین می‌کنیم که در اصل چه عملیاتی باید بین مانتیس‌ها رخ دهد و در نهایت با توجه به این سیگنال عملیات را بر روی مانتیس‌ها انجام می‌دهیم. پس از انجام عملیات مانتیس‌ها، در صورت نیاز نمای حاصل را اصلاح می‌کنیم. همچنین به کمک تعدادی دستور شرطی و سیگنال اضافی، حالات خاصی مانند صفر و یا بی‌نهایت را بررسی کرده و در صورت نیاز خروجی را اصلاح می‌کند. همچنین در طی انجام محاسبات برای رسیدن به حالات نهایی، گاهی شیفت‌هایی داده می‌شود که اگر در طی این شیفت‌ها بیت با ارزشی دور ریخته باشیم، خروجی `in_exact` اصلاح می‌شود. در پایان نیز بیت‌های مورد نیاز برای سرریز یا زیرریز تعیین می‌شوند و کار ماژول به پایان می‌رسد. یک ماژول کمکی در کنار این ماژول وجود دارد که از آن برای نرمالایز کردن استفاده می‌کنیم که بسته به اینکه اولین بیت یک از سمت چپ کجا قرار دارد، مقدار مانتیس و مقدار لازم برای شیفت را تعیین را می‌کند تا نما را هم اصلاح کنیم.

**ماژول ضرب:** ابتدا با توجه به علامات اعداد ورودی، علامت عدد خروجی را تعیین می‌کنیم. سپس برای تعیین نمای خروجی، نمای دو عدد ورودی را با یکدیگر جمع می‌کنیم. برای محاسبه مانتیس خروجی نیز، مانتیس‌های دو عدد ورودی را در یکدیگر ضرب می‌کنیم، در صورتی که تعداد بیت‌ها از ۲۴ بیشتر شود، بیت‌ها کم ارزش‌تر را در نظر نمی‌گیریم (این موضوع را در تعیین بیت `in_exact` در نظر می‌گیریم). برای نرمالایز کردن مانتیس نیز احتیاجی به ماژول جداگانه نداریم و تنها باید دو بیت اول را یافتن اولین یک بررسی کنیم و بسته به آن نما و مانتیس را اصلاح کنیم. حالات خاصی مانند صفر و بی‌نهایت را نیز در اینجا با استفاده از تعدادی شرط مدیریت کرده و بر روی خروجی اثر می‌دهیم. در پایان نیز بسته به مقدار نما بیت‌ها سرریز و زیرریز را تعیین می‌کنیم.

**ماژول تقسیم:** عملکرد در این ماژول شباهت زیادی به ماژول ضرب دارد. ابتدا با توجه به علامات اعداد ورودی، علامت عدد خروجی را تعیین می‌کنیم. سپس برای تعیین نمای خروجی، نمای عدد دوم از نمای عدد اول کم می‌کنیم. مانتیس عدد اول را بر عدد خروجی تقسیم می‌کنیم. البته برای جلوگیری از بروز مشکلات کوچک بودن مانتیس اول، ۲<sup>۲۴</sup> برابر مانتیس اول را بر مانتیس دوم تقسیم می‌کنیم. در اینجا نیز حالات خاص به صورت جداگانه مدیریت شده و خروجی‌ها متناظر تعیین می‌شوند. برای بررسی بیت `in_exact` نیز از باقی‌مانده تقسیم استفاده می‌کنیم و با توجه به آن مقدار این بیت را تعیین می‌کنیم. در نهایت با توجه به نمای حاصل، مقدار بیت سرریز یا زیرریز تعیین می‌شود.

**ماژول مقایسه‌کننده:** این ماژول ابتدا بر اساس علامات تصمیم می‌گیرد که به چه صورت عمل کند. اگر اعداد علامات یکسان داشته باشند، ابتدا نماها را مقایسه می‌کند، در صورت یکی نبودن، بر اساس علامت تصمیم می‌گیرد که عدد بزرگتر است. اگر نماها یکسان باشند، به سراغ مانتیس‌ها رفته و آن‌ها را مقایسه می‌کند. مانند مقایسه نماها و بر اساس مقایسه

مانتیس‌ها، تعیین می‌کند که آیا اعداد برابرند یا یکی از دیگری بزرگتر است. حالت خاصی یا بیت `in_exact` برای این ماژول نداریم و عملکرد آن تنها به صورت همین الگوریتم است.

**ماژول معکوس‌گیری:** در این ماژول از ماژول تقسیم استفاده می‌کنیم، چرا که معکوس یک عدد در اصل برابر است با یک تقسیم بر آن عدد، بنابراین در این ماژول به کمک ماژول تقسیمی که قبلاً توضیح داده‌ایم عملیات تقسیم را انجام داده و به کمک سیگنال‌ها خروجی همان ماژول، سیگنال‌های خروجی این ماژول را مقداردهی می‌کنیم.

**ماژول گرد کردن:** در این ماژول عدد ورودی را به یک عدد `integer` ۳۲ بیتی تبدیل می‌کنیم. طبعاً اگر عدد بیش از ۲ به توان ۳۱ یا کوچکتر از منفی ۲ به توان ۳۱ باشد آن را نمیتوانیم نشان دهیم و بیت `overflow` ۱ خواهد بود. پس از شیفت دادن اگر بیت‌هایی در بعد از ممیز قرار داشته باشد `inexact` ۱ خواهد شد و همچنین با توجه به اینکه بیت اول آن‌ها چیست رو به بالا یا پایین گرد خواهیم کرد. و عدد حاصل را در ۳۲ بیت قرار می‌دهیم و در آخر با توجه به علامت مثبت یا منفی بودن عدد را تعیین می‌کنیم.

**ماژول اصلی:** در این ماژول از هر کدام از ماژول‌های قبلی یک واحد قرار می‌دهیم و ورودی‌های آن را تنظیم می‌کنیم سپس با توجه به `func` که دستور ورودی را مشخص می‌کند تصمیم می‌گیرد که خروجی کدام ماژول را در خروجی اصلی قرار دهد دستور‌ها به ترتیب داک پیاده‌سازی شده‌اند و ۰۰۰ برای جمع تا ۱۱۰ برای گرد کردن استفاده شده است.