تمرین چهارم **تاریخ تحویل: ۲۴ خرداد**

۱. طراحی جمع/تفریق‌کننده اعداد ممیز شناور[1]

هدف از این تمرین، آشنایی با استاندارد IEEE 754 برای نمایش اعداد ممیز شناور و طراحی یک سیستم کامل شامل مسیر داده[2] چندمرحله‌ای و واحد کنترل[3] است. شما باید با استفاده از قطعات پایه، عملیات جمع/تفریق دو عدد اعشاری را به صورت سخت‌افزاری و ترتیبی پیاده‌سازی کنید.

مشخصات ورودی‌ها و خروجی‌ها

ورودی‌ها:

- `Clk`: سیگنال ساعت
- `Rst`: سیگنال بازنشانی[4]
- `Start`: سیگنال شروع عملیات (روشن به مدت یک چرخه ساعت به ازای هر بار محاسبه)
- `A`: عملوند اول (۳۲ بیت)
- `B`: عملوند دوم (۳۲ بیت)
- `Sub`: انتخاب میان جمع و تفریق

خروجی‌ها:

- `Result`: حاصل‌جمع نهایی (۳۲ بیت)
- `Done`: سیگنال پایان عملیات

منطق پیاده‌سازی

در این تمرین، اعداد ورودی مطابق قالب تک‌دقت استاندارد IEEE 754[5] در نظر گرفته می‌شوند. هر عدد ۳۲ بیتی از سه بخش علامت، نما و مانتیس تشکیل شده است. بیت علامت تعیین‌کننده‌ی مثبت یا منفی بودن عدد، بخش نما شامل نمای بایاس‌شده[6] و بخش مانتیس شامل بیت‌های اعشاری بعد ممیز[7] است. قالب کلی این نمایش در جدول ۱ آمده است.

[1] Floating-Point Adder
[2] Datapath
[3] Control Unit
[4] Reset
[5] Single-Precision Floating-Point
[6] Biased Exponent
[7] Mantissa/Fraction

جدول ۱: قالب عدد ممیز شناور تک‌دقت در استاندارد IEEE 754

31	**30-23**	**22-0**
Sign	Exponent	Fraction
1 bit	8 bits	23 bits

در اعداد نرمال‌شده، بیت پنهان 1 در ابتدای مانتیس در نظر گرفته می‌شود؛ بنابراین مقدار مؤثر مانتیس به صورت 1.Fraction خواهد بود. واحد جمع/تفریق باید ابتدا این سه بخش را از ورودی‌های A و B جدا کرده و سپس مراحل الگوریتم را روی آن‌ها اعمال کند.

الگوریتم جمع ممیز شناور دارای چهار مرحله اصلی است که واحد کنترل باید آن را به صورت یک ماشین حالت[8] مدیریت کند:

۱. **مقایسه نماها**: ابتدا نمای دو عدد مقایسه می‌شود. کسر عددی که نمای کوچکتری دارد باید به سمت راست شیفت داده شود تا زمانی که نماها برابر گردند.

۲. **جمع کسرها**: پس از هم‌ارزش شدن نماها، کسرها (همراه با بیت پنهان ۱) با یکدیگر جمع می‌شوند.

۳. **نرمال‌سازی حاصل‌جمع**[9]: نتیجه جمع ممکن است نرمال نباشد.

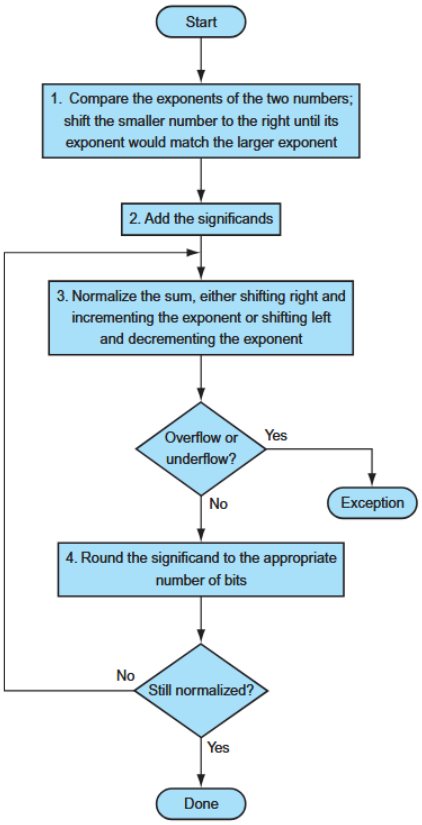
- اگر در جمع سرریز رخ دهد، کسر یک واحد به راست شیفت داده شده و نما افزایش می‌یابد.
- اگر صفرهای بعد ممیز[10] به وجود آمده باشد، کسر به چپ شیفت داده شده تا به فرم نرمال درآید و نما کاهش می‌یابد.
- در این مرحله، در صورت خروج نما از محدوده مجاز، خطای Overflow یا Underflow رخ می‌دهد.

توجه شود جمع مانتیس‌ها روی جمع‌کننده های به طول دو برابر انجام می‌شود تا عملکرد گرد کردن به درستی انجام شود.

۴. **گرد کردن**[11]: کسر به دست آمده باید به تعداد بیت‌های مجاز (۲۳ بیت) گرد شود. از آنجا که عمل گرد کردن ممکن است عدد را مجدداً از حالت نرمال خارج کند، یک حلقه برگشتی[12] در فلوچارت وجود دارد که در صورت غیرنرمال شدن عدد، به مرحله ۳ بازمی‌گردد.

برای درک بهتر الگوریتم به شکل ۱ توجه کنید.

[8] FSM
[9] Normalize
[10] leading zero
[11] Rounding
[12] Loop



شکل ۱: فلوچارت و الگوریتم ماشین حالت برای جمع اعداد ممیز شناور

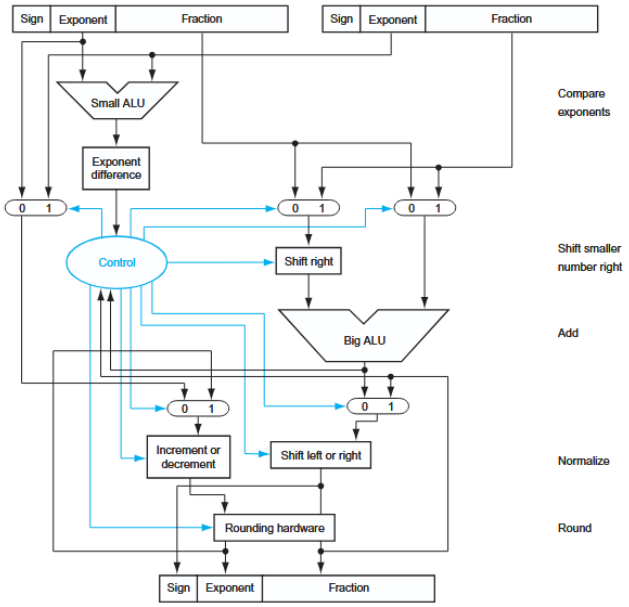
معماری مسیر داده

مسیر داده از اجزای زیر تشکیل شده است که باید با دقت به یکدیگر متصل شوند:

- **مالتی‌پلکسرها**[۱۳]: برای هدایت بیت‌های داده به مسیر مورد نظر، از مالتی پلکسر ها استفاده می‌شود که سیگنال های کنترلی آبی رنگ مسیر را مشخص می‌کنند.
- **Big ALU**: جمع/تفریق‌کننده بزرگ برای جمع/تفریق کسرها.
- **واحدهای نرمال‌سازی**: شامل شیفتر دوطرفه (Shift left or right) و شمارنده بالارونده[۱۴] و پایین‌رونده[۱۵] برای اصلاح نما.
- **سخت‌افزار گرد کردن**: بلوکی که وظیفه گرد کردن مقدار نهایی را بر عهده دارد.

برای درک بهتر الگوریتم به شکل ۲ توجه کنید.

[13] Multiplexer
[14] Increment
[15] Decrement



شکل ۲: معماری سخت‌افزاری مسیر داده برای جمع ممیز شناور

الگوریتم‌ها و پیاده‌سازی این تمرین باید مشابه تمرین دوم انجام شود و می‌توانید از نمونه پیاده‌سازی موجود در این لینک استفاده کنید.

خروجی‌های مورد انتظار برای تحویل

- فایل طراحی مدار در محیط Logisim (نام فایل: HW4-STUDENT_ID.circ).
- گزارش کاملی از نحوه طراحی و اجرای آزمون ارائه کنید (نام گزارش: HW4-STUDENT_ID.pdf). توجه داشته باشید که نمره‌دهی تنها بر اساس نتایج همین آزمون انجام می‌شود.