تمرین دوم **تاریخ تحویل: ۸ خرداد**

۱. طراحی واحدهای پردازش بیتی پیشرفته[1]

هدف از این تمرین، درک عمیق ساختار سخت‌افزاری عملیات پردازش بیتی در پردازنده است. برخلاف زبان‌های سطح بالا، در سخت‌افزار حلقه‌ی تکرار وجود ندارد. در این تمرین شما باید با استفاده از روش‌های «درختی» و «لایه‌ای» در محیط شماتیک[2]، واحدهای شمارش صفر و شیفت‌دهنده ۳۲ بیتی را بدون استفاده از عناصر[3] سطح بالا طراحی کنید.

مشخصات ورودی‌ها و خروجی‌ها

واحد شمارشگر (CLZ، CTZ، CLO، CTO)

ورودی‌ها:

- `Data`: داده ورودی برای واحد شمارشگر (۳۲ بیت)
- `Count_Op`: انتخاب نوع عملیات شمارش (۲ بیت)

خروجی‌ها:

- `Count`: تعداد بیت‌های شمارش شده توسط واحد شمارشگر (۶ بیت — عددی بین ۰ تا ۳۲)

واحد شیفت‌دهنده (Barrel Shifter)

ورودی‌ها:

- `Data`: داده ورودی برای واحد شیفت‌دهنده (۳۲ بیت)
- `Shift_Amt`: مقدار شیفت (۵ بیت)
- `Shift_Op`: سیگنال کنترلی نوع شیفت (۲ بیت)

خروجی‌ها:

- `Result`: خروجی شیفت داده شده توسط واحد شیفت‌دهنده (۳۲ بیت)

[1] CLZ, CTZ, Barrel Shifter
[2] Schematic
[3] Component

منطق پیاده‌سازی

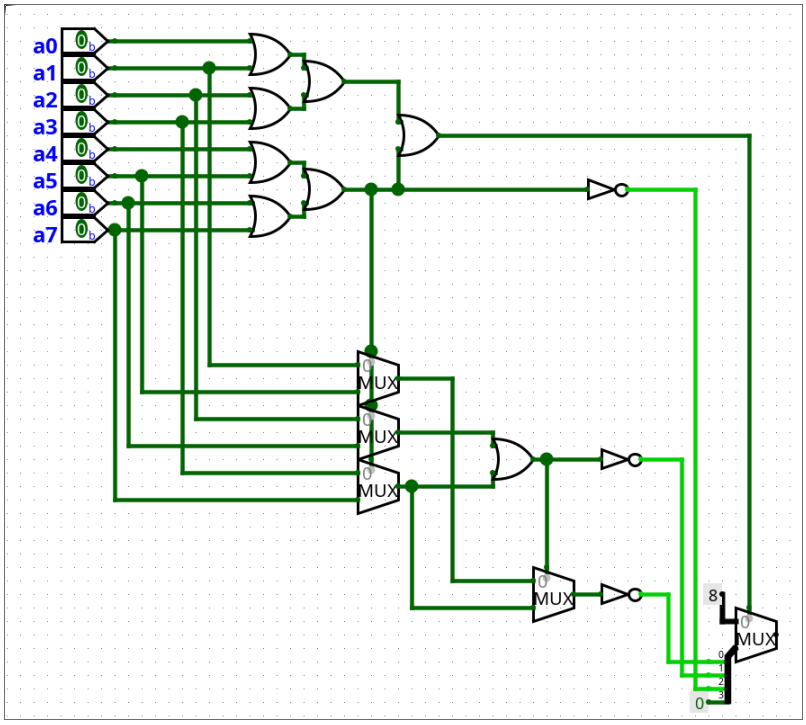
واحد شمارشگر (CLZ، CTZ، CLO، CTO)

این پیمانه باید عملیات‌های شمارش صفرهای ابتدایی CLZ[4]، شمارش صفرهای انتهایی CTZ[5]، شمارش یک‌های ابتدایی CLO[6] و شمارش یک‌های انتهایی CTO[7] را پشتیبانی کند.

این پیمانه باید با روش درختی و بدون استفاده از بلوک‌های آماده مانند Bit Finder یا Priority Encoder طراحی شود. پیشنهاد می‌شود ابتدا یک مدار پایه برای محاسبه CTZ بسازید و سپس حالات دیگر (CLZ، CLO، CTO) را با تغییر شکل ورودی آن و بدون هزینه اضافی طراحی کنید. برای پیاده سازی خود CTZ نیز باید الگوریتم جست و جوی دودویی را پیاده سازی کنید: در هر مرحله بازه کاندیدا را به ۲ نیم تقسیم کنید و بررسی کنید که آیا بازه سمت راست تماما ۰ است یا نه، اگر نبود پس جواب در همان سمت است و اگر بود در سمت دیگر. جدول۱ عملیات متناظر با هر کد کنترلی را نشان می‌دهد. برای درک بهتر شکل ۱ را ببینید.

جدول ۱: دستورات شمارش بیت‌ها

کد کنترلی[8]	نام دستور	شرح عملیات
00	**CTZ**	شمارش تعداد بیت‌های ۰ در سمت کم‌ارزش عدد
01	**CTO**	شمارش تعداد بیت‌های ۱ در سمت کم‌ارزش عدد
10	**CLZ**	شمارش تعداد بیت‌های ۰ در سمت پرارزش عدد
11	**CLO**	شمارش تعداد بیت‌های ۱ در سمت پرارزش عدد



شکل ۱: نمونه پیاده‌سازی CLZ هشت‌بیتی

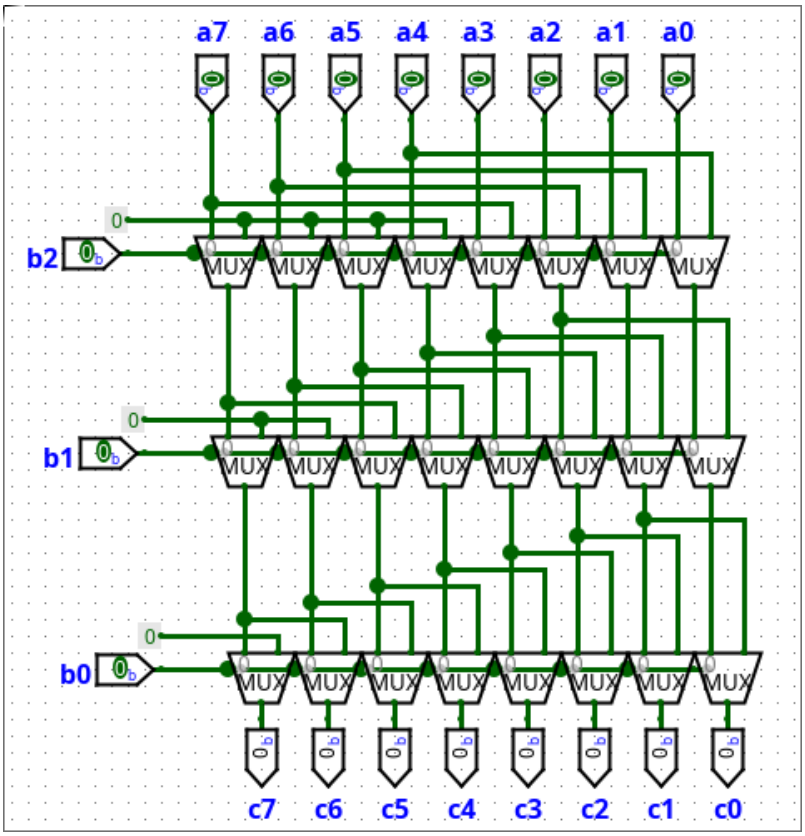
[4] Count Leading Zeros
[5] Count Trailing Zeros
[6] Count Leading Ones
[7] Count Trailing Ones

واحد شیفت‌دهنده (Barrel Shifter)

این پیمانه باید با ساختار لایه‌ای و بدون استفاده از عنصر شیفت دهنده طراحی شود. هر لایه از مالتی‌پلکسرها[9] متناظر با یک بیت از `Shift_Amt` است و برای پیاده‌سازی همزمان شیفت چپ و راست می‌توان از تکنیک Bit_Reverser استفاده کرد. جدول ۲ عملیات متناظر با هر کد کنترلی را نشان می‌دهد. برای درک بهتر الگوریتم شکل ۲ را ببینید.

جدول ۲: دستورات شیفت‌دهنده

کد کنترلی[10]	نام دستور	شرح عملیات
00	**SLL**	شیفت منطقی به چپ (پر کردن بیت‌های خالی با صفر)
01	**SRL**	شیفت منطقی به راست (پر کردن بیت‌های خالی با صفر)
10	**SRA**	شیفت حسابی به راست (پر کردن بیت‌های خالی با تکرار بیت علامت[11])



شکل ۲: نمونه پیاده‌سازی شیفت‌دهنده ۸ بیتی

خروجی‌های مورد انتظار برای تحویل

- فایل طراحی مدار در محیط Logisim (نام فایل: HW2-STUDENT_ID.circ).
- گزارش کاملی از نحوه طراحی و اجرای آزمون ارائه کنید (نام گزارش: HW2-STUDENT_ID.pdf). توجه داشته باشید که نمره‌دهی تنها بر اساس نتایج همین آزمون انجام می‌شود.

[9] Multiplexer