



## درس مدارهای منطقی و سیستم‌های دیجیتال ۱

### تکلیف کامپیوتری دوم: طراحی و پیاده‌سازی ALU با استفاده از ماژول‌های ترکیبی

دانشکده فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

نیم‌سال اول سال تحصیلی ۱۴۰۳-۰۴

دستیار آموزشی: علیرضا سقائیان، محمدمبین علی‌نژاد

aminalinejad80@gmail.com, a.saghaeian13@gmail.com

#### ۱. مقدمه

واحد محاسبه و منطق (Arithmetic Logic Unit یا ALU) یکی از اجزای اصلی هر سیستم پردازشی است. در این سیستم ALU مسئول انجام عملیات ریاضی (مانند جمع و تفریق) و عملیات منطقی (مانند AND و OR) است. هدف از این پروژه، طراحی و پیاده‌سازی یک ALU ساده است که بتواند ۴ عملیات اصلی زیر را انجام دهد:

۱. جمع دو عدد ۴ بیتی
۲. تفریق دو عدد ۴ بیتی
۳. یافتن مقدار حداقل (min) بین دو عدد
۴. یافتن مقدار حداکثر (max) بین دو عدد

این پروژه شامل طراحی سخت‌افزاری با استفاده از ماژول‌های ترکیبی مانند Carry Look-Ahead Adder و Comparator است و نیازمند استفاده از مفاهیم کدگذاری (Encoder/Decoder) و نمایش خروجی بر روی نمایشگر 7-Segment خواهد بود.

#### ۲. اهداف پروژه

- آشنایی با مفهوم ALU و نحوه عملکرد آن
- پیاده‌سازی مدارهای ترکیبی برای انجام عملیات ریاضی و منطقی
- درک مفاهیم کدگذاری (One-Hot Encoding & Decoding)
- ترکیب ماژول‌های ساخته‌شده در پروژه‌های قبلی با خواسته‌های یک پروژه جدید

### ۳. سیستم ورودی و خروجی

در این پروژه، اعداد ورودی شما باید به صورت One-Hot تعریف شوند و برای انجام محاسبات ریاضیاتی و منطقی در ALU، باید به وسیله یک Encoder این اعداد One-Hot را به Binary تبدیل کنید. لذا در این پروژه در قسمت ورودی از یک 16-to-4 Encoder و در بخش خروجی از یک 4-to-16 Decoder استفاده کنید تا ورودی و خروجی پایانی به صورت One-Hot باشد و محاسبات میانی بر روی اعداد باینری ۴ بیتی انجام شود. توجه داشته باشید که محاسبات ما در ساختار 2's Complement انجام خواهد شد لذا محدوده اعداد باینری شما بین ۸- تا ۷ خواهد بود.

### ۴. واحد محاسباتی

پس از آماده‌سازی سیگنال‌های ورودی و دریافت ۲ عدد ۴ بیتی باینری، اکنون وارد واحد ALU شده تا هر یک از محاسبات خواسته شده را انجام دهیم. در ALU واحدهای محاسباتی متنوعی وجود دارد که بسته به خواست کاربر می‌توان از آنها بهره برد. مشخص کردن نوع محاسبات به وسیله یک سیگنال ۳ بیتی به نام Opcode انجام می‌شود. این سیگنال که نقش یک المان کنترلی را دارد، تعیین می‌کند تا کدام یک از خروجی‌های واحدهای محاسباتی مورد استفاده قرار گیرد و عملاً دستور انجام‌شونده در ALU را تعیین می‌کند. لیست Opcode ها و عملکرد آنها به شرح جدول ۱ است:

جدول ۱- مشخصه هر Opcode و عملکرد آن

Instruction	Opcode	Operation
ADD	001	Result = inp1 + inp2
SUB	010	Result = inp1 - inp2
Min (by adder)	011	Result = min (inp1, inp2)
Max (by adder)	100	Result = max (inp1, inp2)
Min (by comparator)	101	Result = min (inp1, inp2)
Max (by comparator)	110	Result = max (inp1, inp2)
Move	111	Result = inp2

در این پیاده‌سازی ذکر چند نکته حائز اهمیت است که در ادامه آنها را مرور می‌کنیم:

۱. به صورت کلی سیگنال Opcode توسط المان‌های کنترلی در یک سیستم تولید می‌شوند. اما در این پروژه با توجه به اینکه یک سیستم بزرگ و چندجانبه نداریم، وظیفه تولید این سیگنال برعهده شماست. در Test Bench خود باید مقادیر مختلف Opcode را تعریف کرده و نتایج حالات مختلف را بررسی کنید.
۲. برای تشکیل ساختار یک ALU، باید یک جمع‌کننده Carry Lookahead Adder، ۴ بیتی در سطح گیت را پیاده‌سازی کرده و از آن در ماژول ALU خود Instance بگیرید. همچنین این کار را برای یک

مقایسه‌کننده ۴ بیتی نیز انجام داده و ابتدا ساختار آن را به صورت Gate Level پیاده‌سازی کنید. سپس از آن در ماژول ALU خود instance بگیرید.

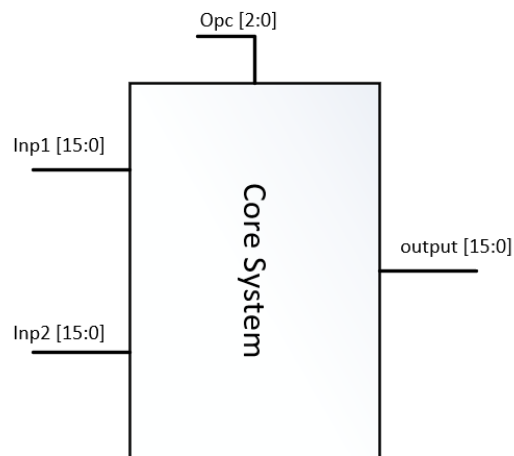
۳. در صورت بروز Overflow در انجام محاسبات، لازم است تا بتوانید آن را شناسایی کرده و پاسخ بدست آمده را براساس Overflow رخ داده اصلاح کنید.

۴. همانطور که در جدول ۱ مشاهده می‌شود، از ۲ روش برای یافتن مقدار کمینه و بیشینه استفاده شده است. در یک حالت شما تنها مجاز به استفاده از ماژول جمع‌کننده موجود در ALU برای یافتن مقدار Extreme هستید. در حالت دیگر تنها امکان استفاده از ماژول مقایسه‌کننده را دارید. در پیاده‌سازی خود حتما به این محدودیت‌ها توجه کنید.

۵. در نهایت و در دستور Move شما باید داده ورودی دوم را مستقیماً بر روی خروجی قرار دهید. توجه داشته باشید که محاسباتی بر روی آن انجام نمی‌شود.

#### ۴. پیاده‌سازی سیستم نهایی

در نهایت در یک ماژول جداگانه که با نام Top\_Module آن را نام‌گذاری می‌کنید باید از تمام المان‌های Encoder، Decoder و ALU خود Instance گرفته و سیستم کلی خود را تشکیل دهید. سیگنال‌های ورودی و خروجی این سیستم در شکل ۱ قابل مشاهده است و شامل ۲ سیگنال ۱۶ بیتی ورودی به صورت One-Hot، یک سیگنال ۳ بیتی برای Opcode و یک سیگنال خروجی ۱۶ بیتی به صورت One-Hot است.



شکل ۱- شماتیک سیستم کلی شامل ورودی‌ها و خروجی

#### ۵. بخش امتیازی\*

با استفاده از 7-Segment طراحی شده در تکلیف کامپیوتری اول و اعمال تغییرات لازم در صورت نیاز، عدد خروجی بدست آمده را بر روی 7-Segment نمایش دهید.

## ۶. خواسته‌های پروژه

۱. طراحی شماتیک سیستم خود را در فایل گزارش کار نمایش داده و توضیحاتی در مورد هر بخش و عملکرد آن ارائه کنید.
۲. کدها به زبان Verilog نوشته شده و ساختار Modular و سلسه مراتبی آن حفظ شود. به طور مثال حتما کد جمع‌کننده CLA را ابتدا به صورت Gate Level در یک ماژول جداگانه تعریف کرده و از آن در ساختار ALU خود Instance بگیرید و در نهایت از ماژول ALU در سیستم نهایی Instance گرفته شود.
۳. با نوشتن یک Test Bench مناسب، تمامی حالات عملکرد ALU را بررسی کرده و حداقل به ازای دو سری ورودی مختلف، صحت هر هفت عملیات سیستم را ارزیابی کنید. محاسبات لازم برای صحت‌سنجی عملکرد را همراه با شکل موج خروجی در گزارش کار قرار دهید.
۴. فایل نهایی شما باید شامل کدهای وریلاگ، طراحی شماتیک انجام‌شده و گزارش کار باشد و نتیجه نهایی به صورت یک فایل zip با فرمت DLD\_CA#2\_StudentNumber.zip تحویل داده شود.

با آرزوی بهترین‌ها برای شما