



درس مدارهای منطقی

تکلیف کامپیوتری سوم: پیاده‌سازی محاسبه‌گر آماری

دانشکده‌گان فنی دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

دکتر بیژن علیزاده

نیم‌سال دوم سال تحصیلی ۱۴۰۲-۰۳

دستیار آموزشی: زینب پورقیصری

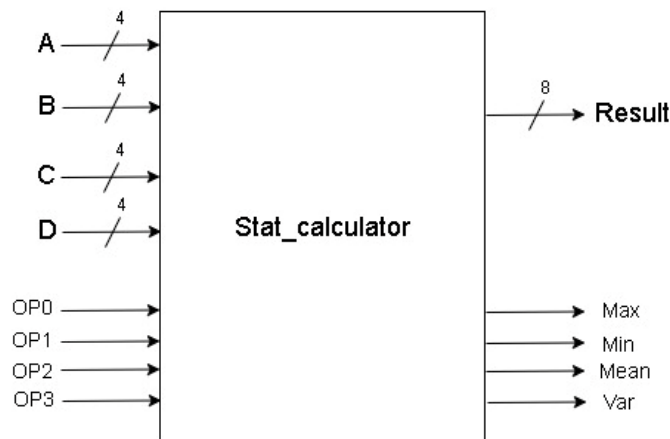
z.pourgheisari@gmail.com

۱. مقدمه

در علم آمار و احتمالات، محاسبه و یافتن مقادیر کمینه، بیشینه، میانگین و واریانس از اهمیت بسیاری برخوردار است. این مقادیر نه تنها به ما کمک می‌کنند تا داده‌ها را بهتر درک کنیم، بلکه در تصمیم‌گیری‌ها و پیش‌بینی‌های مختلف نیز نقش اساسی دارند. در این تکلیف کامپیوتری، قصد داریم سیستمی طراحی کنیم که قادر به محاسبه‌ی این چهار عملیات بر روی اعداد ورودی باشد.

۲. توصیف سیستم

این سیستم چهار سیگنال با نام‌های OP0 تا OP3 دارد. هر یک از این سیگنال‌ها متناظر با انجام یک عملیات آماری روی ورودی‌های سیستم است. ورودی‌های سیستم، چهار عدد ۴ بیتی هستند که بر اساس مقدار ورودی OP، یکی از عملیات‌های محاسبه‌ی بیشینه، کمینه، میانگین یا واریانس روی آن‌ها انجام شده و خروجی ۸ بیتی تولید می‌شود. سیگنال‌های OP نسبت به یکدیگر دارای اولویت هستند و اگر چند سیگنال به طور همزمان فعال باشند، عملیات مربوط به سیگنال با اولویت بالاتر انجام می‌شود. برای نمایش اینکه خروجی مربوط به کدام عملیات آماری است، چهار خروجی تک بیتی با نام‌های Max, Min, Mean, Var برای سیستم در نظر می‌گیریم. شکل ۱ بلوک دیاگرام این سیستم را نشان می‌دهد. جدول ۱ نیز بیانگر ارتباط میان ورودی‌ها و خروجی‌ها است. (واضح است که در عملیات‌هایی که حاصل ۴ بیتی دارند، ۴ بیت پر ارزش خروجی برابر صفر هستند).



شکل ۱- دیاگرام کلی سیستم

جدول ۱- ارتباط ورودی‌ها و خروجی‌ها

{ OP3, OP2, OP1, OP0 }	Result	Max	Min	Mean	Var
xxx1	Max (A, B, C, D)	1	0	0	0
xx10	Min (A, B, C, D)	0	1	0	0
x100	Mean (A, B, C, D)	0	0	1	0
1000	Var (A, B, C, D)	0	0	0	1

۳. بلوک‌های محاسباتی

برای پیاده‌سازی سیستم نهایی لازم است برای هر عملیات بلوک جداگانه‌ای تعریف کنیم. هر بلوک چهار عدد ۴ بیتی به عنوان ورودی دریافت کرده و حاصل عملیات را در خروجی‌اش قرار می‌دهد. در صورت لزوم، بلوک‌ها می‌توانند ورودی‌ها و خروجی‌های دیگری برای ارتباط با یکدیگر داشته باشند.

➤ بلوک محاسبه‌کننده‌ی بیشینه و کمینه:

بلوک‌های min_calculator و max_calculator را در سطح گیت و به صورت ساختاری پیاده‌سازی کنید. طراحی خود را به صورت سلسه مراتبی انجام دهید. برای مثال، در صورت نیاز به استفاده از مقایسه‌کننده، ابتدا یک مقایسه‌کننده‌ی تک‌بیتی با نام comparator_1bit در سطح گیت طراحی کرده و سپس با اتصال تعداد مناسبی از این مقایسه‌کننده‌ها، یک مقایسه‌کننده با ابعاد موردنیاز بسازید.

کد Verilog ماژول‌های لازم برای طراحی این دو بلوک را بنویسید. در یک تست‌بنچ عملکرد هر دو بلوک بیشینه و کمینه را با حداقل سه دسته ورودی مختلف بررسی کنید. بلوک دیاگرام آن‌ها را نیز در گزارش خود رسم کنید.

➤ بلوک محاسبه‌کننده میانگین و واریانس:

بلوک‌های `var_calculator` و `mean_calculator` را به صورت رفتاری توصیف و پیاده‌سازی کنید. واحدهای محاسباتی نظیر جمع‌کننده و ضرب‌کننده، دو ورودی هستند و استفاده از تقسیم‌کننده نیز مجاز نیست. توجه داشته باشید که لازم است تعداد بیت‌های محاسبات میانی را به طور مناسبی تنظیم کنید تا بیت‌های نقلی از دست نروند. با توجه به محدودیت‌های موجود، کد Verilog این دو بلوک را بنویسید و عملکرد آن دو را در یک تست‌بنچ و با حداقل سه دسته ورودی مختلف بررسی کنید.

۴. پیاده‌سازی سیستم نهایی

در این بخش با استفاده از بلوک‌هایی که در بخش قبل طراحی کرده‌اید، ماژول `stat_calculator` که همان سیستم نهایی است را پیاده‌سازی می‌کنید. ابتدا بلوک دیاگرام این سیستم را رسم کنید. بلوک‌های اضافی که در سیستم نهایی لازم دارید را مشخص کرده و کد Verilog آن‌ها را در سطح رفتاری بنویسید. کد Verilog سیستم کلی `stat_calculator` را با اتصال بلوک‌های موجود بنویسید. تست‌بنچ مناسبی برای بررسی عملکرد بخش‌های مختلف سیستم طراحی کرده و کد آن را بنویسید. حداقل به ازای دو سری ورودی مختلف، هر چهار عملیات سیستم را تست کنید. محاسبات لازم برای صحت‌سنجی عملکرد را همراه با شکل موج خروجی در گزارش کار قرار دهید.

با آرزوی بهترین‌ها برای شما