بسمه تعالی

امتحان میان‌ترم درس طراحی سیستم‌های دیجیتال — دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف

* شروع امتحان از ۲۴ اردیبهشت ۱۴۰۰ است و پاسخ‌ها باید پیش از ساعت ۱۲ شب ۳۱ اردیبهشت برای دستیار آموزشی ارسال گردد.

* از میان سؤال‌های داده شده تعداد سه عدد را به دلخواه انتخاب نموده و پاسخ دهید و به بیش از سه سؤال پاسخ ندهید. برای پاسخ دادن به هر سؤال نیازمند کسب اطلاعات و مطالعه در حوزه‌ی مربوطه خواهید بود (مثلاً برای پاسخ به سؤال ۵ نیاز دارید که روش نمایش اعداد به شکل Fractional را مورد مطالعه قرار دهید) که این بخشی از حل مسئله محسوب می‌شود. مطالعه و یادگیری بخشی رایج و مهم در امتحان‌های take-home است و چنین نیست که دانشجو انتظار داشته باشد که تمام مطالب مورد نیاز برای حل مسئله را از قبل بداند.

۱- می‌دانیم که توصیف ASM ترکیبی از متن و گراف را به کار می‌گیرد. یک قرارداد ارائه کنید که بتوانیم با استفاده از آن توصیف ASM را کاملاً به شکل Text نمایش دهیم و آن را در فایل txt. ذخیره کنیم. قرارداد خود را شرح دهید و syntax آن را مشخص کنید. سپس یک قطعه کد Python بنویسید که ورودی ASM را با قالب قراردادی شما و در فایل txt. دریافت کرده و آن را تبدیل به کد Verilog کند. این روش را برروی یک نمونه کد ASM که قبلاً داشتیم (مانند تقسیم کننده) به کار بگیرید و خروجی Verilog آن را شبیه‌سازی کنید. برای بررسی کد Verilog باید Testbench بنویسید و Testbench را کد Python شما تولید نمی‌کند بلکه باید جداگانه به زبان Verilog نوشته شود.

۲- توصیف Verilog برای یک واحد UART (شامل فرستند و گیرنده) را بنویسید. در Testbench مربوطه نشان دهید که چگونه با اتصال دو واحد ارسال و دریافت انتقال داده میان دو بخش انجام می‌شود. در کد خود مجاز به استفاده از توصیف dataflow نیستید و فقط مجاز به استفاده از آنچیزی هستید که تاکنون در درس ارائه شده است. دقت کنید که در Testbench سیگنال ساعت بین دو بخش فرستنده و گیرنده مشترک نباشد و همچنین مقداری اختلاف فاز بین آن‌ها وجود داشته باشد تا نشان دهد که طراحی شما به این مورد حساس نیست.

۳- پردازنده‌هایی وجود دارند که به آن‌ها OISC گفته می‌شود. این پردازنده‌ها در مجموعه دستورات زبان ماشین خود فقط دارای یک عدد دستور هستند ولی این دستور دارای خاصیتی است که می‌توان هر برنامه‌ای را با آن نوشت و از این نظر دستوری کامل است. یک نمونه از این دستورها که می‌تواند در چنین پردازنده‌ای به کار گرفته شود دستور subleq است (مخفف subtract and branch if less than or equal to zero). این دستور به این شکل کار می‌کند که subleq a,b,c ابتدا برروی حافظه‌ی کامپیوتر عمل MEM[b]=MEM[b]-MEM[a] را انجام می‌دهد (تفریق در اینجا به معنای جمع با مکمل ۲ است). سپس اگر MEM[b] کوچکتر یا برابر 0 باشد به آدرس c پرش انجام می‌شود. توصیف Verilog چنین پردازنده‌ای را بنویسید. سپس برنامه‌ی زبان ماشین مرتب‌سازی یک آرایه‌ی ۱۰ عنصری را با این پردازنده بنویسید و در Testbench به اجرا درآورید و نشان دهید که پردازنده درست کار می‌کند. لازم به ذکر است که بر اساس آنچه در درس ساختار کامپیوتر دیده‌اید هم برنامه و هم داده‌های مورد پردازش درون حافظه قرار دارند که باید این کار را در Testbench انجام دهید.

۴- توصیف Verilog یک سیستم دیجیتال را بنویسید که از الگوریتم CORDIC استفاده می‌کند و با استفاده از آن سینوس و کسینوس زاویه‌ای که به عنوان ورودی به آن داده می‌شود را محاسبه نموده و با دقت ۴ رقم اعشار (دقت کنید که دقت در مبانی ۱۰ بیان شده است و نه در مبنای ۲) در خروجی ارائه می‌کند.

۵- توصیف ASM یک سیستم دیجیتال را بنویسید که از الگوریتم نصف کردن استفاده می‌کند تا ریشه‌ی دوم ورودی خود را به دست آورد. ورودی و خروجی با روش اعداد Fixed Point از نوع Fractional نمایش داده می‌شوند.

۶- کد Verilog مربوط به اتومات سلولی Conway's Game of Life را بنویسید. ابعاد اتومات باید به شکل پارامتر قابل تعیین باشد. اتومات سلولی باید بدون مرز باشد (یعنی سمت راست آن به سمت چپ و بالای آن به پایین متصل باشد. می‌توان از این نظر به جدول کارنو تشبیه کرد که بدون مرز

هستند). اتومات دارای چهار پایه‌ی یک بیتی clock، command، serial_out و serial_in است. عملکرد پایه‌ی clock همانی است که در تمام سیستم‌های دیجیتال داریم. برای پایه‌ی Command، اگر به آن ۱ بدهیم Game of life اجرا می‌شود ولی اگر به آن 0 بدهیم اتومات در وضعیت خواندن و نوشتن داده قرار می‌گیرد. در وضعیت خواندن و نوشتن داده، اتومات باید مانند یک Shift Register بسیار بزرگ عمل کند. به این شکل که مقادیر درون اتومات را از پایه‌ی serial_out بیرون می‌دهد و از پایه‌ی serial_in داده را وارد اتومات می‌کنیم. در Testbench باید یکی از الگوهای شناخته شده‌ی Game of life را به اتومات خود بدهید (از طریق serial_in)، سپس اتومات را در وضعیت اجرا قرار دهید و بعد از مدت زمانی مشخص محتوای اتومات را (از طریق serial_out) خارج کنید و دیده شود که با رفتار مورد انتظار از Game of life مطابقت دارد.

۷- یک حافظه‌ی TCAM (مخفف Ternary Content addressable memory) را توصیف کنید که ابعاد آن به شکل پارامتریک قابل مشخص کردن باشد. برای بررسی عملکرد آن طبعاً باید Testbench بنویسید.

۸- توصیف رفتاری جمع‌کننده‌ی باینری که پهنای آن به شکل پارامتر قابل مشخص کردن است را بنویسید. سپس با استفاده از این module توصیف ساختاری یک جمع‌کننده‌ی BCD که از الگوریتم Carry Save Addition استفاده می‌کند را بنویسید. در Testbench باید ۲۰ عدد مبانی ۱۰ با طول ۸ رقم را جمع بزنید. برای پیاده‌سازی جمع‌کننده‌ی Carry Save Addition دو روش وجود دارد که یکی مدار ترتیبی ایجاد می‌کند و دیگری مدار ترکیبی که از هر روش که استفاده کنید قابل قبول است.

۹- یک برنامه‌ی Python بنویسید که به آن ابعاد یک ضرب کننده‌ی آرایه‌ای Array multiplier را بدهیم و سپس آن برنامه‌ی پایتون به ما توصیف ساختاری ضرب کننده را به زبان Verilog بدهد. توصیف Verilog باید کاملاً ساختاری باشد و استفاده از توصیف رفتاری یا dataflow غیرمجاز است. با نوشتن Testbench نشان دهید که خروجی‌های کد Python شما سالم بوده و کار می‌کنند.

موفق باشید

اجلالی