

- * سه مورد از مسائلی که در ادامه آمده است را به انتخاب خود حل کنید.
- * اگر بیش از سه مورد حل کنید، موارد اضافه به انتخاب مدرس درس و دستیار آموزشی حذف خواهد شد.
- * پس از آزمون تعدادی از دانشجویان به شکل تصادفی انتخاب شده و باید راه حل و طرح خود را در یک جلسه‌ی مصاحبه‌ی مجازی شرح دهند.
- * نمره‌ی در نظر گرفته شده برای حل تمام مسائل یکسان است.
- * دقت شود که برای حل مسائل نیاز است که دانشجویان مطالب مربوطه را مورد مطالعه قرار دهند. به عنوان مثال برای حل مسئله‌ی ۴ نیاز به مطالعه‌ی روش فشرده‌سازی LPC وجود دارد. این کار با آگاهی از شرایط دانشجویان انجام شده است و اینکه برخی از مطالب را بسیاری از دانشجویان از پیش نمی‌دانند. چنانکه تقریباً در هر پروژه‌ی مهندسی در دنیای واقعی مطالعه‌ی اولیه از لازمه‌های کار است.

(۱) پردازنده‌ای که کد آن در منبع "A very simple 8-bit RISC processor for FPGA" نوشته‌ی S de Pablo و دیگران آورده شده است را در نظر بگیرید. برای این پردازنده (که کد Verilog آن در منبع ذکر شده موجود است) یک برنامه‌ی زبان ماشین بنویسید که یک لیست پیوندی از اعداد صحیح با حداقل ۱۰ عنصر ۸ بیتی را دریافت نموده و بزرگترین مقدار موجود در لیست را به عنوان خروجی در یک خانه‌ی حافظه ثبت کند. سپس این پردازنده را به همراه کد نوشته شده شبیه‌سازی کنید. برای اجرای یک برنامه باید هم کد زبان ماشین برنامه و هم داده‌ای که نیاز است توسط آن پردازش شود در واحد حافظه‌ی پردازنده قرار داده شود که این کار را در Test bench انجام دهید. در ضمن ده عضو لیست پیوندی را در خانه‌هایی از حافظه قرار دهید که مجاور هم نباشند تا بدینوسیله تفاوت لیست پیوندی با آرایه حتماً در نظر گرفته شود.

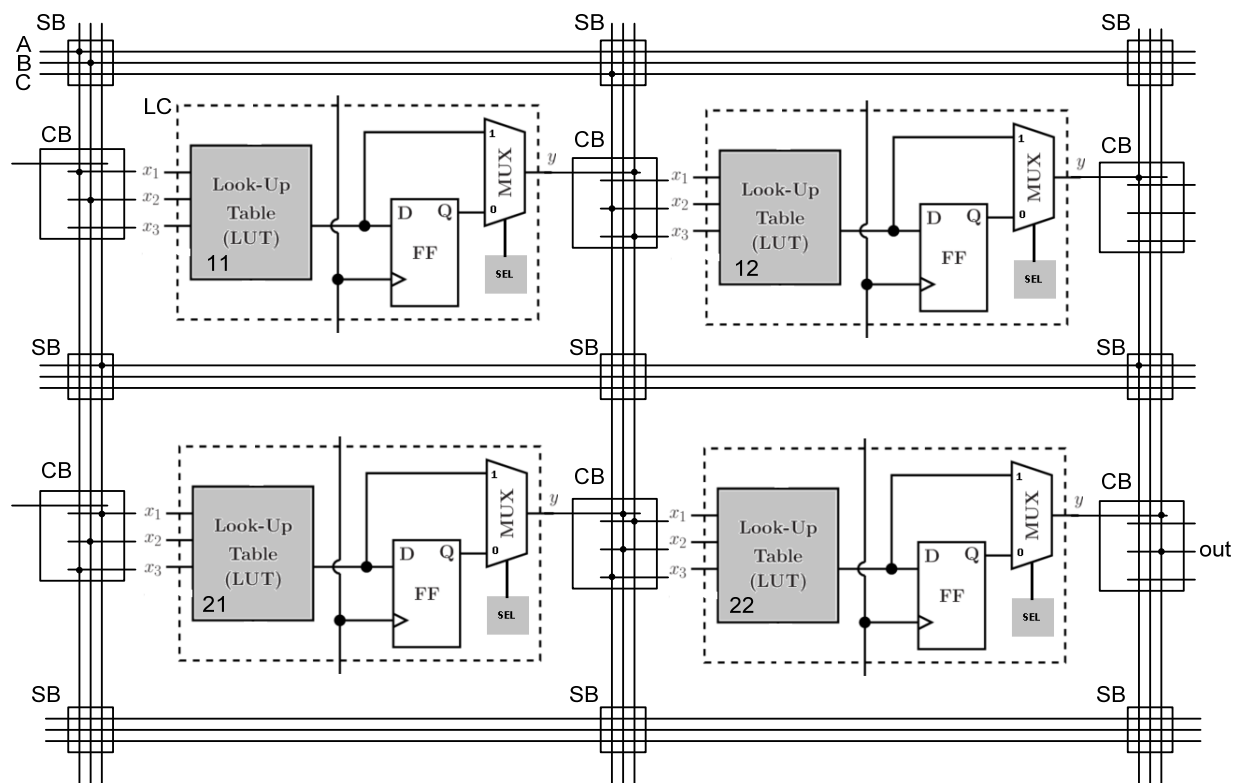
(۲) یک کد Verilog پارامتری و قابل سنتز برای واحد FFT (تبدیل فوری‌ی سریع) بنویسید که در آن پارامتر n معرف تعداد نقطه در FFT باشد. در انجام این کار از روش تجزیه کردن به واحدهای نیمه (decomposition into half-size) استفاده کنید. با سنتز مدار برای پنج ساینز متوالی توسط یک ابزار سنتز دلخواه برای FPGA به نام Spartan 7 بیان کنید که افزایش تعداد نقاط چه تأثیری بر حجم سخت‌افزار دارد. دقت کنید که منظور از سنتز فقط همان عمل سنتز منطقی است و منظور قرار دادن مدار داخل FPGA نیست.

(۳) شکلی که در ادامه ترسیم شده است بخش کوچکی از یک FPGA فرضی را نشان می‌دهد. این FPGA اتصال منظمی از واحدهایی با نام LC مخفف Logic cell، CB مخفف Connection block و SB مخفف Switch block است. چگونگی اتصالات در هر واحد SB با ۹ بیت و چگونگی اتصالات در هر واحد CB با ۱۲ بیت مشخص می‌شود. در این سؤال جهت سادگی، چگونگی اتصالات در واحدهای SB و CB به جای اینکه با آرائه‌ی بیت‌های مربوطه (۹ بیت برای SB و ۱۲ بیت برای CB) مشخص شود با گذاشتن علامت دایره مشخص شده است به این شکل که هر جا دایره هست اتصال برقرار است و هر جا دایره نیست اتصال برقرار نیست. هر واحد LC دارای یک LUT است که منطق عملکرد آن را مشخص

می‌کند و همچنین یک واحد یک بیتی SEL دارد که مشخص می‌کند که از فلیپ‌فلاپ استفاده شود یا خیر. فرض کنید برای این واحدهای LC وضعیت LUT و SEL به صورتی باشد که در ادامه آمده است.

LC 11		LC 12		LC 21		LC 22	
$x_3x_2x_1$	y	$x_3x_2x_1$	y	$x_3x_2x_1$	y	$x_3x_2x_1$	y
000	0	000	0	000	0	000	1
001	1	001	0	001	1	001	0
010	1	010	1	010	1	010	0
011	0	011	1	011	0	011	1
100	0	100	1	100	1	100	0
101	1	101	1	101	0	101	1
110	1	110	0	110	0	110	1
111	0	111	0	111	1	111	0
SEL	0	SEL	0	SEL	1	SEL	0

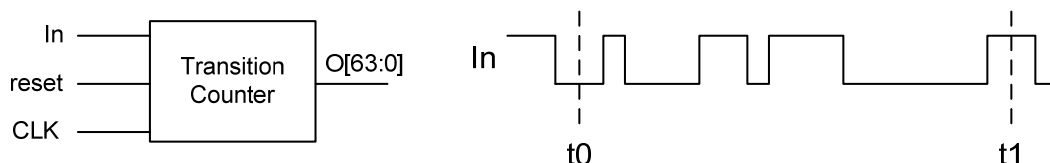
با فرض اینکه برای این بخش از FPGA سیگنال‌های A B C ورودی باشند و سیگنال out خروجی باشد عملکرد این مدار را مشخص کنید. اگر مدار ترکیبی است عملکرد آن را با ترسیم جدول صحت مشخص کنید و اگر مدار ترتیبی است عملکرد آن را با ترسیم یک نمودار حالت مشخص کنید.



۴) می‌خواهیم یک مدار قابل سنتز داشته باشیم که با استفاده از روش LPC (Linear prediction code) اجازه‌ی فشرده‌سازی سیگنال‌های صوتی را می‌دهد. از چنین فشرده‌سازی می‌توان در کاربردهای انتقال صوت و به شکل real-time استفاده نمود (real-time در اینجا به این معنی است که همان موقع که صوت تولید می‌شود در سمت فرستنده فشرده شده و در سمت گیرنده از فشرده‌گی خارج شده و دریافت می‌شود. به عنوان مثال تلفن‌های دیجیتال یا ابزارهایی مانند شبکه‌های اجتماعی که قابلیت تماس صوتی دارند می‌توانند از این روش استفاده کنند). فرض کنید سیگنال‌های صوتی مورد نظر mono بوده و نمونه‌های آن (sampleها) ۱۶ بیتی هستند. مدار CODEC قابل سنتز برای انجام این کار را ارائه دهید (دقت کنید که CODEC به معنای Coder + Decoder است). یک فایل wav نوع mono و 16 بیتی با ضبط کردن صدای خودتان ایجاد کنید و در Test Bench همان داده‌ها را به CODEC خود بدهید تا شبیه‌سازی بر روی آن انجام شود. صوت شما چند درصد فشرده‌سازی می‌شود؟ قابلیت سنتز کد خود را حتماً با یک ابزار سنتز واریسی کنید.

۵) با استفاده از ASM مداری طراحی کنید که اعداد تصادفی ۳۲ بیتی با توزیع نرمال (گوسی) ایجاد کند. شکل مدار را بطور کامل ترسیم کنید. ذکر کنید که توزیع نرمالی که توسط مدار شما ایجاد می‌شود چه میانگین و واریانس دارد. راهنمایی: الف) برای تولید اعداد تصادفی با توزیع نرمال می‌توان ابتدا اعداد با توزیع یکنواخت ایجاد کرد که با مداری ساده به نام LFSR قابل انجام است. سپس با جمع کردن (متوسط گرفتن) از اعدادی که توزیع یکنواخت دارند می‌توان طبق قضیه‌ی حد مرکزی اعداد با توزیع نرمال ایجاد نمود. ب) یک متغیر با توزیع نرمال در واقع می‌تواند مقادیر از $-\infty$ تا $+\infty$ را داشته باشد ولی در عمل احتمال آنکه مقدار آن از حدی که با مثلاً $-b$ تا $+b$ مشخص می‌شود فراتر برود بسیار بعید است. به همین دلیل می‌توان آن را با تعداد بیت‌های محدود (در اینجا ۳۲ بیت) نمایش داد و نگرانی وجود ندارد. پ) با توجه به علامت‌دار بودن اعداد از روش مکمل ۲ برای نمایش اعداد علامت‌دار استفاده کنید. در این سوال فقط کد ASM و شکل کامل مدار مورد نظر است و جداول شبیه‌سازی از شما خواسته نشده است ولی این بدان معنا نیست که مدار شما می‌تواند باگ داشته باشد.

۶) شکل زیر یک مدار شمارنده‌ی تعداد تغییرات برای ورودی In است. مثلاً بافرض اینکه برای شکل موج زیر در زمان t_0 شمارنده reset شده باشد در زمان t_1 باید در خروجی مقدار ۷ بدهد.



الف- ورودی In نسبت به CLK در این مدار باید دارای چه شرطی باشد تا این مدار بتواند تعداد تغییرات را به درستی بشمارد؟ استدلالی که شرط بر اساس آن بدست می‌آید را بیان کنید.

ب- یک توصیف رفتاری قابل سنتز برای این مدار ترتیبی بنویسید. در این توصیف باید واریسی کنید که اگر هنگام شبیه‌سازی شرط بند الف رعایت نشود روی صفحه‌ی شبیه‌ساز خطا چاپ کند ولی طبعاً این قسمت هنگام سنتز بی‌اثر بوده و سنتز نگردد.

۷) یک برنامه‌ی Python بنویسید که هر کد Verilog که توصیف آن dataflow قابل سنتز است را به عنوان ورودی به آن بدهیم، در خروجی به ما معادل رفتاری قابل سنتز آن را بدهد. راهنمایی: به ازای هر assign در توصیف dataflow یک بدنه‌ی always ایجاد کنید. صحت عملکرد کد خود را با استفاده از یک ابزار شبیه‌ساز و یک ابزار سنتز واریسی کنید.

موفق باشید

اجلالی