



دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

طراحی کامپیوتری سیستم‌های دیجیتال
بهار ۱۴۰۱

CA3: Review on logic design and introduction to FPGA bit-stream generation

مقدمه

در تمرین‌های یک و دو، ماژولی برای پیاده‌سازی یک تابع encoder را طراحی و پیاده‌سازی کردید. در این تمرین قصد داریم برای یک مدل ساده از آن، مسیر داده و کنترل را در سطح گیت، طراحی و پیاده‌سازی کنیم. در این تمرین ابتدا مدار خود را بر روی کاغذ طراحی می‌کنید و سپس مدار طراحی شده را با استفاده از وریلاگ پیاده‌سازی کرده و شبیه‌سازی می‌کنید.

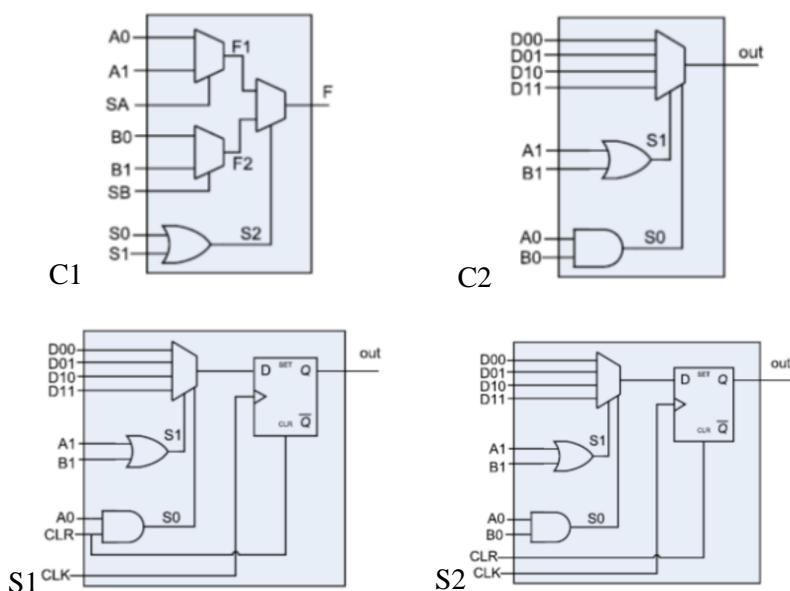
مهلت تحویل این تمرین ۱۰ روز در نظر گرفته شده است.

توضیحات پروژه

همان‌طور که به یاد دارید، تابع encoder شامل ۵ تابع colParity, rotate, permute, reevaluate و addRC بود که در تمرین اول تابع permute را طراحی و پیاده‌سازی کردید. برای این تمرین نیز، تابع permute را در نظر گرفته‌ایم. تمام توضیحات این تابع به تفصیل در تمرین اول گفته شده است. تعداد تکرار این تابع همانند قبل ۲۴ مرتبه است. سایر جزئیات تابع تغییر نمی‌کند.

همانند تمرین قبل باید داده‌ها را از فایل “input_i.txt” بخوانید و خروجی را در فایل “output_i.txt” بنویسید.

هدف این تمرین، پیاده‌سازی طراحی نهایی تابع *permute* بر روی بخش‌های قابل برنامه‌ریزی FPGA می‌باشد. بنابراین ضروری است تا مسیرهای داده و کنترل را در سطح گیت طراحی کرده و آن‌ها را بر روی سلول‌های منطقی قابل برنامه‌ریزی یک FPGA، سنتز کنید. سلول‌های منطقی در شکل ۱ نشان داده شده‌اند.



شکل ۱: سلول‌های منطقی قابل برنامه‌ریزی مربوط به ماژول منطقی Actel

دقت کنید که برای طراحی مسیرهاده و کنترلر (FSM) شما تنها مجاز به استفاده از سلول‌های C1، C2، S1 و S2 هستید.

چنانچه عملکرد ماژول ذکر شده صحیح باشد، نمره کامل به شما تعلق خواهد گرفت. (البته عملکرد صحیح در نهایت با استفاده از شبیه‌سازی تمرین سنجیده خواهد شد). با این حال برای گروه‌هایی که مدار را از نظر حاصل ضرب مساحت^۱ در زمان اجرا، بهینه کنند، نمره امتیازی در نظر گرفته خواهد شد.

برای بدست آوردن مساحت کل کافی است مساحت تمام ماژول‌ها را با یکدیگر جمع کنید. مساحت هر ماژول در جدول زیر نشان داده شده است. برای بهینه کردن طرح، تا حد امکان باید از ماژول‌ها بیشترین استفاده را بکنید، بنابراین سعی کنید با توجه به معماری ماژول‌ها، به طور کارآمدی از این ماژول‌ها استفاده کنید.

Module	C1	C2	S1	S2
Area	7	8	15	15

برای بدست آوردن زمان اجرا، کافی است براساس رابطه‌ی زیر عمل کنید.

تعداد کلاک مورد نیاز برای اجرای $\text{permute} * \text{زمان مسیر بحرانی} = \text{زمان اجرا}$

در جدول زیر می‌توانید تاخیرهای مربوط به هر یک از اجزای مدار را مشاهده کنید.

Module	Gates	Mux2:1	Mux4:1	FF
Delay	0.5	1	2	0

مواردی که در حین پیاده‌سازی باید درنظر بگیرید:

- ۱- رجیسترها را متناسب با نیاز خود، با حداقل بیت مورد نیاز تعریف کنید.
- ۲- بهتر است طراحی به صورت سلسله‌مراتبی انجام شود. (ابتدا هر یک از بلوک‌های C1, C2, S1, S2 را به صورت یک ماژول تعریف کرده و سپس با استفاده از آن‌ها سایر ماژول‌ها را به صورت سلسله‌مراتبی پیاده‌سازی کنید).
- ۳- تا جای ممکن کنترلر خود را ساده تعریف کنید تا در ساده کردن جدول کارنوی آن به مشکل نخورید.

مواردی که باید برای طراحی مدار تحویل دهید:

- ۱- طراحی کنترلر و مسیرهاده بر روی کاغذ
- ۲- مسیر بحرانی مدار(می‌توانید از شکل مناسب برای نشان دادن آن استفاده کنید و دنباله‌ی گیت‌هایی که در این مسیر قرار می‌گیرند نیز لیست کنید)

۳- محاسبات مربوط به بهینه‌سازی ذکر شده (مساحت و زمان اجرا) دقت کنید که گزارش این موارد باید به طور دقیق و با جزئیات کامل باشد.

مواردی که باید برای شبیه‌سازی تحویل دهید:

- ۱- کدهای مربوطه به زبان وریلاگ (تمامی ماژول‌ها)
- ۲- یک testbench که تمام فایل‌های ورودی را بخواند، خروجی را در یک فایل txt ذخیره کند. با استفاده از اسکریپت شبیه‌سازی کنید.
- ۳- مشابه تمرین دوم، متناسب با پوشه‌ی trunk، فایل‌های گزارش و کدها و... را در پوشه‌ی متناظر آن قرار دهید.

توجه: انجام این تمرین به صورت گروه‌های دونفره خواهد بود.