



طراحی کامپیوتری سیستم‌های دیجیتال  
دانشکده مهندسی برق و کامپیوتر  
پاییز ۱۴۰۲

پروژه دوم: مروری بر logic design و مقدمه سنتز بر روی FPGA

دستیاران آموزشی: هادی بابالو، مریم بحرینی‌زاد، علی عطااللهی

توضیحات پروژه

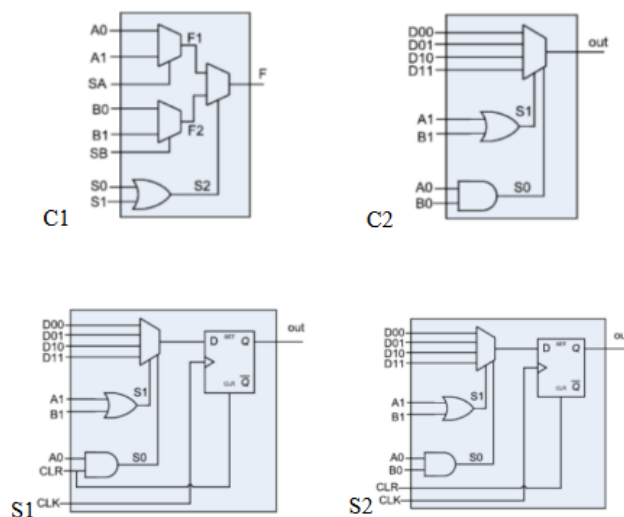
در تمرین‌های یک و دو، شبکه MaxNet با چهار نورون را در جهت پیدا کردن عدد با بیشترین مقدار در میان چهار عدد را پیاده‌سازی کرده‌اید. بنابراین:

الف) ابتدا کد وریلگ ماژول‌های نشان داده شده در شکل 1 بنویسید و از عملکرد صحیح آن مطمئن شوید.

ب) سپس مسیر داده و کنترلر مدار تمرین قبل را بر روی کاغذ و در سطح ماژول‌های نشان داده شده در شکل-1 طراحی کنید (فقط مجاز هستید از همین ماژول‌ها استفاده کنید).

ج) سپس با استفاده از instantiate-کردن ماژول‌های وریلگ قسمت الف مدار طراحی شده در قسمت ب را با استفاده از زبان وریلگ پیاده‌سازی و سپس شبیه‌سازی کنید.

برای یادآوری و درک بهتر پروژه، توصیه می‌شود که توضیحات مربوط به پروژه اول و نحوه کارکرد این شبکه MaxNet را حتما مرور کنید.



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

هدف این تمرین، پیاده‌سازی و سنتز طراحی بر روی بخش‌های قابل برنامه‌ریزی یک FPGA است. بنابراین ضروری است تا مسیرهای داده و کنترل را ابتدا به صورت ساختاری یا همان structural طراحی کنید و سپس طرح را بر روی سلول‌های منطقی قابل برنامه‌ریزی معرفی‌شده، سنتز کنید. سلول‌های منطقی در دسترس در شکل-۱ نمایش داده شده‌اند. دقت کنید که برای طراحی مسیر داده و کنترلر (FSM) شما تنها مجاز به استفاده از سلول‌های S1، C2، C1 و S2 هستید.

در واقع در این پروژه برای آشنایی با روند سنتز کردن طرح بر روی FPGA، باید component-های طراحی استفاده شده در مسیر داده مربوط به پروژه یک را با بلوک‌های برنامه‌پذیر معرفی‌شده جایگزین کنید. همچنین کنترلر هم باید فقط توسط همین بلوک‌ها طراحی و ساخته شود.

برای تحویل این تمرین تا شنبه ۲۵ آذر ماه فرصت دارید. چنانچه عملکرد ماژول ذکر شده صحیح باشد، نمره کامل به شما تعلق خواهد گرفت و بهینه‌سازی و استفاده از کمترین ماژول نمره امتیازی خواهد داشت.

### پیاده‌سازی جمع‌کننده‌ها و ضرب‌کننده‌ها برای اعداد floating point:

۱. اعداد ورودی و وزن‌های شبکه به صورت «اعداد ۵ بیتی fixed point» در اختیار شما قرار می‌گیرد و به این صورت ذخیره می‌شوند (در پروژه قبل ۳۲-بیت فلوٹ بودند). به این صورت که ۳ بیت سمت راست نشان‌دهنده‌ی بخش اعشاری عدد خواهند بود. همچنین فرمت representation تمام اعداد به صورت مکمل دو می‌باشد. به طور مثال، عدد ۵ بیتی 11110 معادل ۰.۲۵- دسیمال خواهد بود. همچنین

- تضمین می‌شود که ۴ عدد ورودی برنامه (۴ مقدار اولیه نوروها) در بازه  $0 \leq x \leq 1$  می‌باشد. در نتیجه ورودی‌های PU-ها ۵ بیتی می‌باشند.
2. ضرب کننده‌هایی که تعریف می‌کنید، ۲ عدد ۵ بیتی به عنوان ورودی دریافت می‌کنند و خروجی ۱۰ بیتی را ایجاد می‌کنند (۶ بیت سمت راست آن نشان‌دهنده‌ی بخش اعشاری می باشد).
3. جمع‌کننده‌های لایه اول PU-ها، ورودی ۱۰ بیتی بدست‌آمده از ضرب‌کننده‌ها را دریافت می‌کنند و خروجی ۱۱ بیتی تولید می‌کنند. این خروجی‌ها وارد جمع‌کننده لایه بعدی PU می‌شوند و در نهایت اعداد ۱۲ بیتی خروجی داده می‌شود. این اعداد ۱۲ بیتی در مرحله بعد وارد Activation Function می‌شوند. خروجی شما باید مجدداً عددی ۵ بیتی باشد. به این علت که بعد از چک‌کردن شرط خاتمه عملیات و در صورت نیاز، باید دوباره برای انجام دوره بعدی محاسبات وارد ساختارهای PU که نیاز به ورودی ۵ بیتی دارند بشود.
4. در صورتی که ورودی Activation Function عددی منفی بود خروجی Activation Function برابر صفر خواهد بود. در صورت مثبت‌بودن ورودی، باید ۳ بیت پرارزش بخش اعشاری، ۱ بیت کم ارزشترین بیت بخش صحیح و ۱ بیت پر ارزش ترین بیت بخش صحیح (بیت علامت) از عدد ۱۲ بیتی را به عنوان خروجی Activation Function و در قالب یک عدد ۵ بیتی خروجی بدهید.
5. همچنین دقت کنید که مقدار اپسیلون که در پروژه اول برابر ۰.۲ در نظر گرفته شده بود. در این پروژه برای سادگی محاسبات برابر ۰.۲۵ در نظر بگیرید.

#### مواردی که در حین پیاده‌سازی باید در نظر بگیرید:

6. نحوه گرفتن ورودی‌ها و خروجی دادن برنامه، باید دقیقاً مطابق با پروژه اول درس باشد.
7. رجیسترها را متناسب با نیاز خود، با حداقل بیت مورد نیاز تعریف کنید.
8. اکیدا توصیه می‌شود طراحی به صورت سلسله مراتبی انجام شود. یعنی در ابتدا هر یک از بلوک‌های C1, S2, S1, C2 را به صورت یک ماژول تعریف کرده و سپس با استفاده از آنها سایر ماژول‌ها را به صورت سلسله مراتبی پیاده‌سازی و استفاده کنید.
9. تا جای ممکن کنترلر خود را ساده تعریف کنید تا در ساده کردن جدول کارنوی مربوطه و سپس پیاده‌سازی آن با ماژول‌های تعریف شده به مشکل نخورید.

#### مواردی که باید تحویل دهید:

- طراحی کنترلر و مسیر داده به صورت دستی
- کدهای مربوط به زبان ورپلاگ (تمامی ماژول‌ها)

- 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.

## نکات پایانی

- برای تمرین دوم، لازم است فایل‌های HDL و testbench خود را مطابق ساختار توضیح داده شده در trunk/doc در subdirectory های trunk آپلود کنید. همچنین، اطمینان حاصل کنید که با اجرای trunk/sim/sim\_top.tcl تست‌بنچ شما اجرا می‌شود. برای اجرای این اسکریپت می‌توانید از دستور زیر در Modelsim استفاده کنید:

```
>> do <sim_file>
```

- لازم است فرمت خروجی مدار شما دقیقاً مطابق ساختار مطرح شده برای ورودی باشد. توجه کنید که صحت کارکرد مدار شما با تست‌های آماده بررسی خواهد شد.
- فایل‌ها و گزارش خود را تا قبل از موعد تحویل هر فاز، با نام CAD\_HW2\_<SID>.zip در محل مربوطه در صفحه درس آپلود کنید.
- هدف از این تمرین، یادگیری شماسست! در صورت کشف تقلب، مطابق با قوانین درس با آن برخورد خواهد شد.
- در صورت داشتن هرگونه سوال یا ابهام از طریق ایمیل‌های زیر با دستیاران آموزشی در ارتباط باشید.

[m.h.babalu@gmail.com](mailto:m.h.babalu@gmail.com)

[aliataollahi40@gmail.com](mailto:aliataollahi40@gmail.com)