به نام خدا

طراحی کامپیوتری سیستم‌های دیجیتال - پاییز 1404

پروژه ۴: **طراحی Hash Generator RTL**

طراحان: سارا گیتی– شهزاد ممیز– هاتف رضایی

هدف پروژه:

در این بخش می‌خواهیم نسخه ی ساده شده تمرین اول (بر اساس تغییرات پروژه سوم) را با ابزار سنتز کنیم سپس نتیجه بدست آمده را با تمرین سوم مقایسه کنیم.

صورت مسئله:

مسئله ی این تمرین ساده شده ی تمرین اول (تولید کننده‌ی هش) است. برای ساده سازی عرض داده ها به 4 کلمه ی 4 بیتی کاهش یافته و همچنین به جای استفاده از فانکشن round از یک ضرب کننده استفاده می شود که نیمه اول داده را در نیمه دوم آن ضرب می کند. برای این پروژه باید به جای تابع rotate باید از عمل گر "*" استفاده شود. در نتیجه حاصل خروجی تمرین سوم و چهارم یکسان شود.

مراحل اجرا:

در این پروژه از ابزار yosys استفاده می‌شود. با استفاده از lut2synth.ys و lut2mapper.py باید کد وریلاگ خود را به ماژول‌های Actel مپ کنید. برای این کار باید قسمت TODO از کد lut2mapper.py را تکمیل کنید. برای سازگار سازی yosys و ماژول های Actel مراحل زیر را طی کنید:

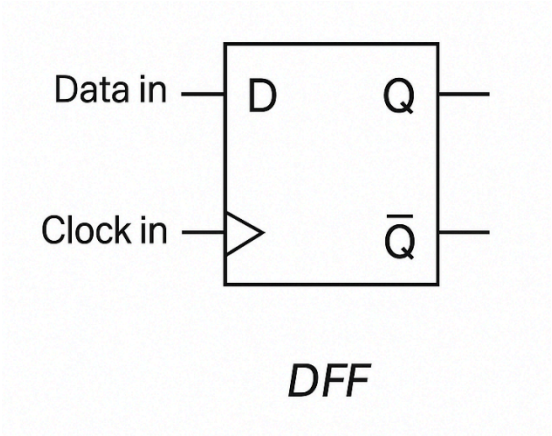
۱. در لینوکس yosys را با دستور sudo apt install yosys نصب کنید.
۲. ساده‌سازی‌های ذکر شده را بر روی پروژه ۱ اعمال کنید.
۳. اسم فایل های وریلاگ خود را در فایل lut2synth.ys در قسمت read_verilog بنویسید و آن را با دستور زیر اجرا کنید.

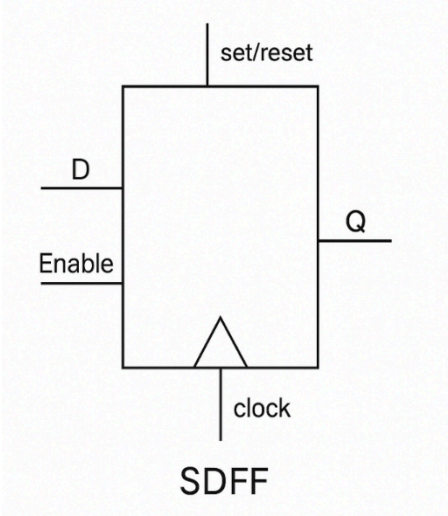
```
Yosys -s lut2synth.ys
```

این دستور پروژه شما را سنتر می‌کند و یک کد وریلاگ به شما می‌دهد که تنها از lut2 (یک lut با دو input و SRAM با ۴ مقدار)، lut1 (یک lut با یک input و SRAM با ۲ مقدار) برای قسمت combinational کد و از SDFF , DFF برای قسمت sequential کد استفاده کرده است.

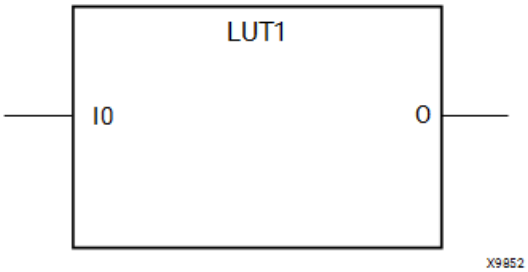
برای اینکه yosys بتواند پروژه را سنتز کند، باید قسمت‌هایی که مقادیر متفاوت را از یک فایل می‌خوانید را حذف کنید. برای استفاده از مقادیر متفاوت حافظه از یک آرایه در کد خود استفاده کنید.

۴. حال باید قسمت‌های TODO کد پایتون lut2mapper.py را تکمیل کنید. این کد به صورت کلی کد تولید شده پس از اجرا lut2synth.ys را به عنوان ورودی می‌گیرد و کد های خروجی که تنها از ماژول‌های Actel در آن استفاده شده را خروجی می‌دهد. قسمت‌هایی که باید تکمیل کنید مشخص می‌کنند هر متغیر lut2, lut1, SDFF, DFF چگونه به ماژول‌های Actel مپ شوند.

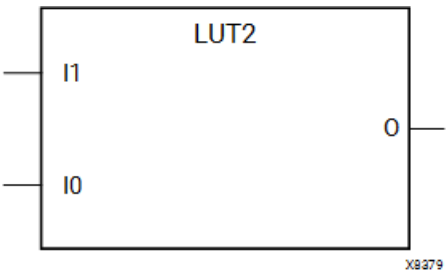




Inputs	Outputs
I0	O
0	INIT[0]
1	INIT[1]
INIT = Binary number assigned to the INIT attribute	



Inputs		Outputs
I1	I0	O
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]
INIT = Binary equivalent of the hexadecimal number assigned to the INIT attribute		



شکل ۱.سلول های مربوط به ماژول های SDFF ، LUT2 ، LUT1 و DFF

نکته: هر lut1 , lut2 را می‌توان با یک ماژول C1 یا C2 پیاده‌سازی کرد. شما باید تنها این پیاده‌سازی های متفاوت را در کد کامل کنید.
همچنین می‌توانید DFF را با S1 و SDFF را با S2 پیاده‌سازی کنید. تنها باید نحوه پیاده‌سازی های متفاوت را در کد تکمیل کنید.

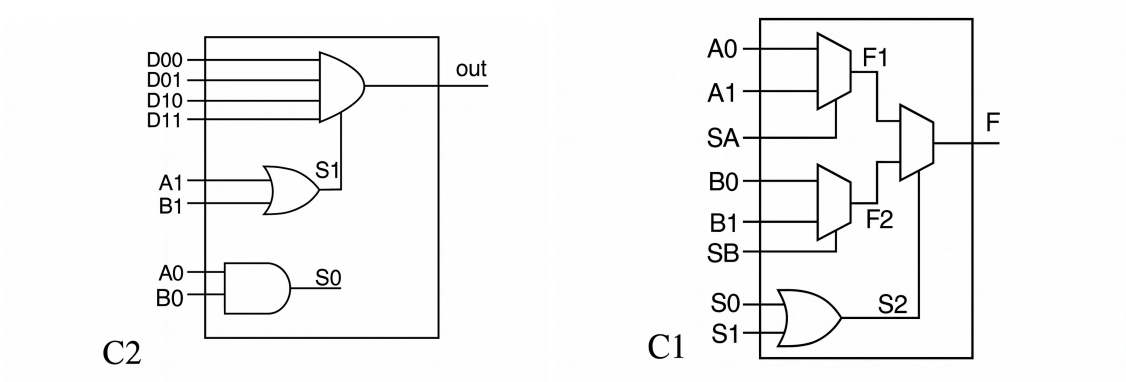
۵. حال کد lut2mapper.py را اجرا کنید.

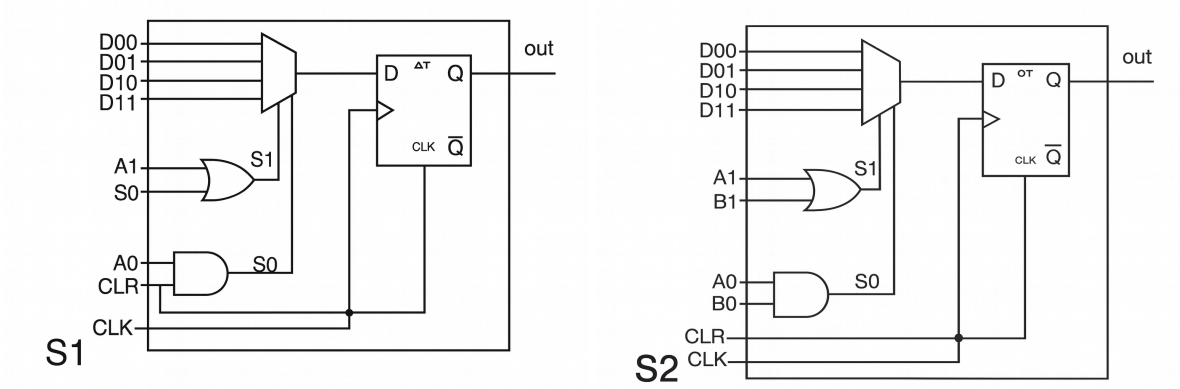
```
Python lut2mapper.py <output_code_of_lut2synth.v> < --cell c1 or ––cell c2 (optional)>
--count-cells
```

با استفاده از

```
< –cell c1 or –cell c2 >
```

می‌توانید مشخص کنید که lut2, lut1 به c1 یا c2 مپ شوند و که به صورت پیش فرض c2 انتخاب می‌شود.





شکل ۲.سلول های قابل برنامه ریزی مربوط به ماژول پایه

۱. تحلیل نتایج

کد های خروجی lut2mapper.py را همانند پروژه ۳ اجرا کنید تا تعداد سل‌های Actel استفاده شده مشخص شود و درستی اجرا کد خود را بررسی کنید و تفاوت تعداد گزارش شده در را با پروژه ۳ مقایسه کنید.

۲. گزارش

طریقه مپ کردن lut1, lut2, SDFF, DFF به ماژول‌های Actel را بکشید و به همراه کد های خود آنها را آپلود کنید.

۳. امتیازی (۲۰ نمره)

در صورت ارائه ابزاری که بتواند کد شما را با ماژول‌های Actel سنتز کند و عملکرد کد همچنان درست باشد، به شما نمره امتیازی تعلق خواهد گرفت.

سایر نکات

- انجام این تمرین به صورت گروه های دونفره خواهد بود.
- فایل ها و گزارش خود را تا قبل از موعد تحویل، با نام CAD_HW4_<SID>.zip در محل مربوطه در صفحه درس آپلود کنید.
- در صورت هرگونه ابهام می توانید از طریق ایمیل با طراحان در ارتباط باشید.
- نام گذاری صحیح متغیرها، تمیزی کد و توضیحات و پارامتری بودن ورودی‌های ماژول‌ها می تواند تا حدودی کاستی‌های کد را در بخش‌های دیگر جبران کند.
- هدف این تمرین یادگیری شماست! در صورت کشف تقلب، مطابق با قوانین درس برخورد خواهد شد.

موفق باشید