گزارش پروژه 4

CAD

امیرحسین علی خانی و صدرا مدائنی اول

810102479 و 810102564

طبق پروژه ۳، وزن هر ماژول به صورت زیر است:

- **C2**: ۱۱ گیت
- **C1**: ۱۰ گیت
- **S1**: ۱۳ گیت
- **S2**: ۱۳ گیت

بر اساس خروجی اسکریپت lut2mapper.py تعداد سلول‌های استفاده شده به شرح زیر است:

```
PS C:\CAD_CA4> python lut2mapper.py synth_output.v --cell c2 --count-cells
Found 2 LUT1 and 555 LUT2 instances (out of 557 $lut)
Total number of c2: 557
Total number of S1: 6
Total number of S2: 14
Created mapped design: mapped_synth_output.v
Created modules file: modules.v
PS C:\CAD_CA4> python lut2mapper.py synth_output.v --cell c1 --count-cells
Found 2 LUT1 and 555 LUT2 instances (out of 557 $lut)
Total number of c1: 557
Total number of S1: 6
Total number of S2: 14
Created mapped design: mapped_synth_output.v
Created modules file: modules.v
PS C:\CAD_CA4>
```

- تعداد **C2یا C1**: ۵۵۷ عدد
- تعداد **S1**: ۶ عدد
- تعداد **S2**: ۱۴ عدد
- که با توجه به فایل پروژه 3 اگر هزینه ها را همان مقدار در نظر بگیریم می شود 6378 گیت.

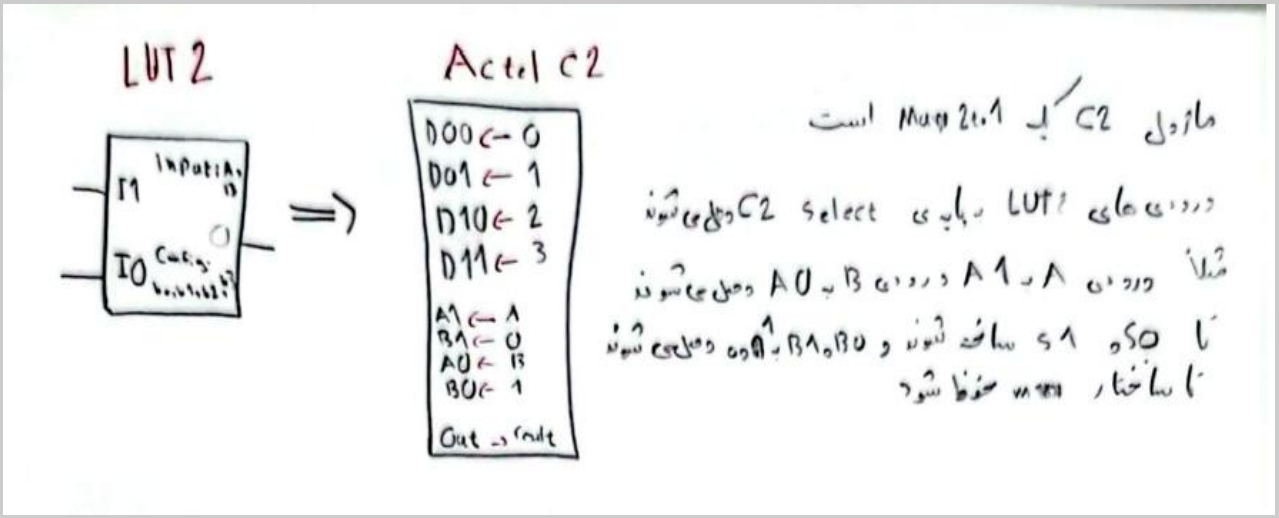
جدول مقایسه پروژه ۳ و ۴

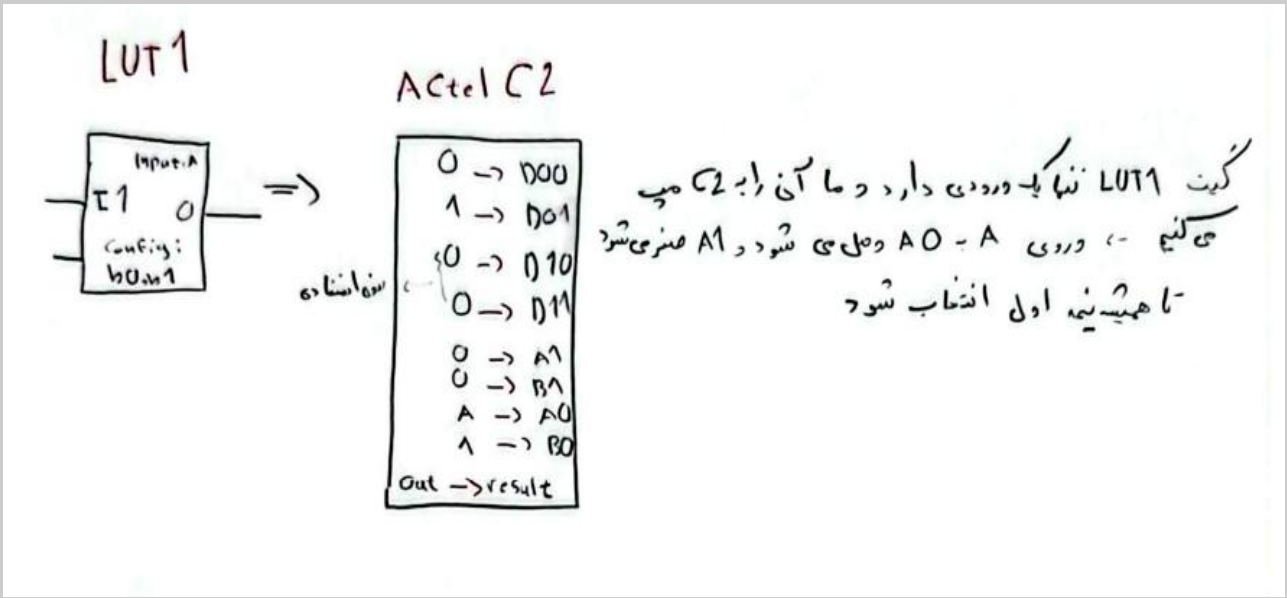
معیار	پروژه ۳	پروژه ۴	تحلیل
نوع طراحی	Structural (دستی)	Synthesis (با Yosys)	طراحی دستی معمولاً بهینه‌تر است.
تعداد کل گیت‌ها	**۳۷۵۴**	**۶۳۸۷**	**افزایش حدود ٪۷۰** در حالت اتوماتیک.
تعداد ماژول‌های منطقی (C2)	(کمتر بوده)	۵۵۷	ابزار سنتز برای هر گیت منطقی یک C2 مصرف کرده است.
تعداد ماژول‌های حافظه (S1/S2)	۲۰	۲۰	تعداد فلیپ‌فلاپ‌ها در هر دو روش یکسان و بهینه است.

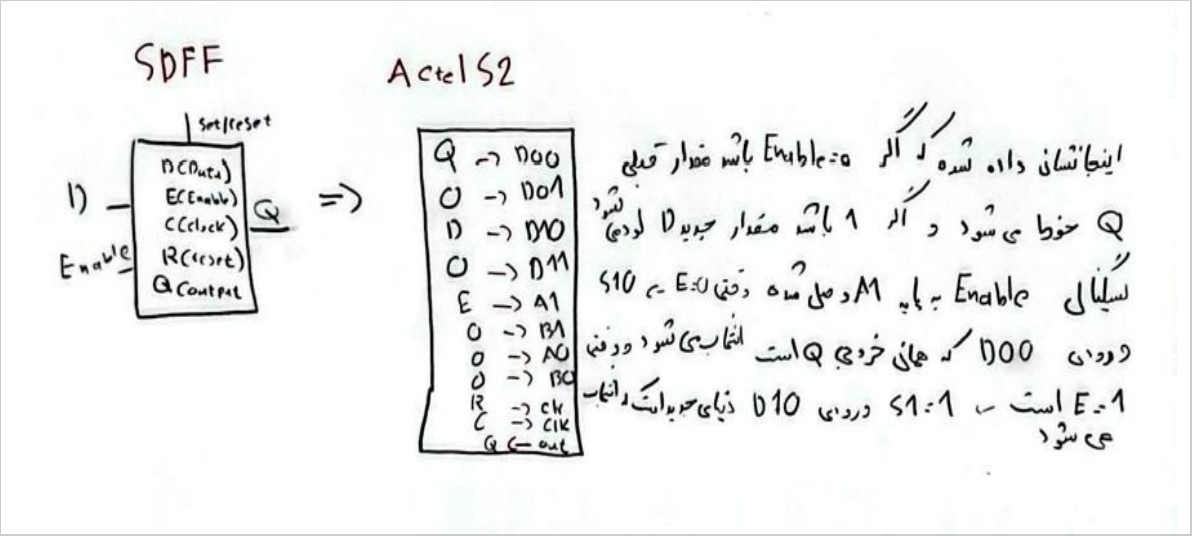
مقایسه نشان می‌دهد که تعداد گیت‌های معادل در روش سنتز اتوماتیک (پروژه ۴) نسبت به روش طراحی دستی (پروژه ۳) افزایش یافته است. دلایل تفاوت این است کهدر پروژه ۴، از دستور synth -lut 2 استفاده شد که مدار را به صورت کورکورانه به LUTهای ۲ ورودی می‌شکند و همه آن‌ها به سلول **C2** (با هزینه ۱۱) مپ می‌شوند. ابزار سنتز در این روش نمی‌تواند مثل یک طراح انسان، از قابلیت‌های خاص سلول‌ها (مثل استفاده از C1 با هزینه ۱۰ یا ترکیب چند گیت در یک ماژول) استفاده کند. و ابزارهای سنتز عمومی (مثل Yosys بدون کتابخانه اختصاصی) معمولاً نمی‌توانند ساختارهای خاص الگوریتم (مثل جمع‌کننده‌ها یا ضرب‌کننده‌ها) را به اندازه طراحی دستی فشرده‌سازی کنند، مگر اینکه از تکنیک‌های پیشرفته‌تر (Techmapping دقیق) استفاده

شود. اگرچه روش اتوماتیک زمان طراحی را بسیار کاهش می‌دهد (چند ثانیه در برابر چند روز)، اما هزینه آن افزایش مساحت مدار (Area) است.

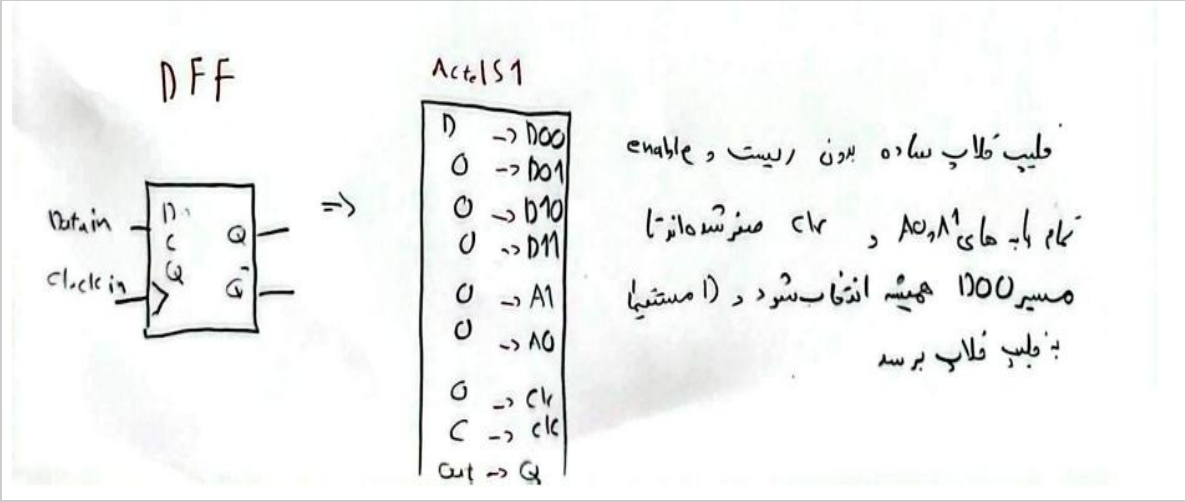
حالا باید طریقه مپ کردن lut1, lut2, SDFF, DFFبه ماژولهای Actelرا بکشیم:







اینجا نشان داده شده که اگر Enable=0 باشد مقدار قبلی Q حفظ می‌شود و اگر 1 باشد مقدار جدید D لود می‌شود
سیگنال Enable به پایه A1 وصل شده وقتی E=0 یعنی S1=0 ورودی D00 که همان خروجی Q است انتخاب می‌شود و وقتی E=1 است یعنی S1=1 ورودی D10 (داده‌ی جدید) انتخاب می‌شود



فلیپ فلاپ ساده بدون ریست و enable
تمام پایه های A0,A1 و clr صفر شده‌اند تا مسیر D00 همیشه انتخاب شود و Q مستقیما به فلیپ فلاپ برسد

امتیازی:

ابزار **Synopsys Synplify Pro** این ابزار با استفاده از فرآیندی به نام **Technology Mapping** کار می‌کند. در این فرآیند، ابزار ابتدا کد را به بلوک‌های منطقی عمومی (Generic Boolean Logic) تبدیل کرده و بهینه‌سازی می‌کند. سپس با استفاده از یک فایل کتابخانه (Technology Library) که مشخصات دقیق سلول‌های Actel مثل تابع منطقی C1 و C2 و هزینه آن‌ها (در آن تعریف شده است، بلوک‌های منطقی را با الگوریتم‌های پوشش‌دهی (Covering Algorithms) به این سلول‌ها نگاشت می‌کند.

همچنین ابزار متن‌باز **Yosys** نیز با استفاده از دستورات dfflibmap(برای فلیپ‌فلاپ‌ها) و abc (برای منطق ترکیبی) قابلیت انجام همین کار را دارد، مشروط بر اینکه فایل توصیف سلول‌ها Liberty) (Fileبه آن داده شود.