



تمرین سری سوم

- پرسش‌های خود را در سامانه CW و تالار مربوط به تمرین مطرح نمایید.
- پاسخ سوالات را تایپ نمایید.
- پاسخ تمرین را به صورت یک فایل زیپ با فرمت HW1\_401234567.zip آپلود کنید. فایل زیپ باید به صورتی باشد که پس از باز کردن آن بدون هیچ پوشه‌ای فایل‌های زیر با ساختار زیر قرار گرفته باشند:

```
1 .
2 |-- HW1T_401234567.pdf
3 |-- practical
4   |-- HW1P_401234567_401234568.pdf
5   |-- schematic.circ
6   |-- verilog
7       |-- circuit
8       |   |-- main.v
9       |   |-- ...
10      |-- gates
11      |   |-- ...
12      |-- toplevel
13      |   |-- ...
```

- در صورت عدم تطابق فایل آپلود شده با فرمت بالا، تمرین شما تصحیح نخواهد شد.
- پاسخ سوالات تئوری و گزارش تمرین‌های عملی باید به فرمت pdf باشد.
- هر دانشجو می‌تواند حداکثر سه تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.
- تمرینات عملی به صورت گروه‌های دو نفر تحویل داده شود.
- هر دو عضو گروه موظف هستند تمرینات خود را بارگذاری کنند.
- عواقب عدم تطابق بین پاسخ دو عضو گروه برعهده خودشان است.
- تحویل تمرین به صورت انگلیسی مجاز نیست. در صورت تحویل تمرین به صورت انگلیسی (حتی بخشی از تمرین) نمره تمرین موردنظر صفر در نظر گرفته می‌شود.
- در صورت مشاهده تقلب برای بار اول نمره هر دو طرف صفر می‌شود. در صورت تکرار نمره کل تمرینات صفر خواهد شد.
- استفاده از ابزارهایی مانند ChatGPT به منظور ابزار کمک آموزشی مجاز است به شرط آن که به خروجی آن اکتفا نشود.
- توجه شود که پروژه نهایی درس در گروه‌های چهار نفر تحویل گرفته می‌شود.
- سوالات با عنوان اختیاری نمره‌ای ندارند اما جواب دادن به آن‌ها کمک به‌سزایی در یادگیری درس می‌کند.

## تمارین تئوری

۱. فرض کنید یک ضرب‌کننده booth داریم که برای عملیات‌های جمع و تفریق آن از CSA استفاده شده است. می‌خواهیم از این ضرب‌کننده برای ضرب اعداد ۳۲ بیتی علامت‌دار استفاده کنیم. فرض کنید که تاخیر هر بخش از آن به صورت زیر باشد:

$$D_{FA} = 2ns, D_{MUX} = 3ns, D_{gate} = 1ns, D_{shift} = 2ns$$

(آ) اندازه بلوک‌های CSA را به گونه‌ای مشخص کنید که تاخیر این ضرب‌کننده حداقل باشد. (دقت کنید که اندازه بلوک‌های CSA می‌توانند متفاوت از یکدیگر باشند)

(ب) در این حالت، حداقل و حداکثر تاخیری که این ضرب‌کننده به ازای ورودی‌های متفاوت می‌تواند داشته باشد چقدر است؟ یک نمونه از ورودی‌هایی که باعث حداقل و حداکثر تاخیر می‌شوند را بنویسید.

۲. درباره جمع‌کننده carry-skip تحقیق کنید و به سوالات زیر پاسخ دهید.

(آ) شیوه کار این جمع‌کننده را توضیح دهید.

(ب) آیا بهبود زمانی نسبت به جمع‌کننده ripple-carry در این جمع‌کننده تضمین شده است؟ توضیح دهید.

(ج) منطق بخش skip را برای یک بلوک جمع‌کننده carry-skip ۳ بیتی نشان دهید. (می‌توانید از multiplexer استفاده کنید.)

(د) فرض کنید هر بلوک جمع‌کننده carry-skip دارای n جمع‌کننده یک بیتی است. یعنی هر بلوک دو عدد n بیتی را جمع می‌کند. احتمال رد شدن<sup>۱</sup> بیت نقلی در این بلوک چقدر است؟ محاسبات خود را نشان دهید.

می‌خواهیم با استفاده از الگوریتم booth دو عدد را در هم ضرب کنیم. تاخیر مدارهای مختلف برابر:

۳. ۱. تاخیر هر عمل جمع 12ns است.

۲. تاخیر هر عمل مکمل‌گیری 4ns است.

۳. تاخیر هر عمل انتقال 8ns است.

برای ضرب دو عدد ۶ بیتی 011001 و 101000:

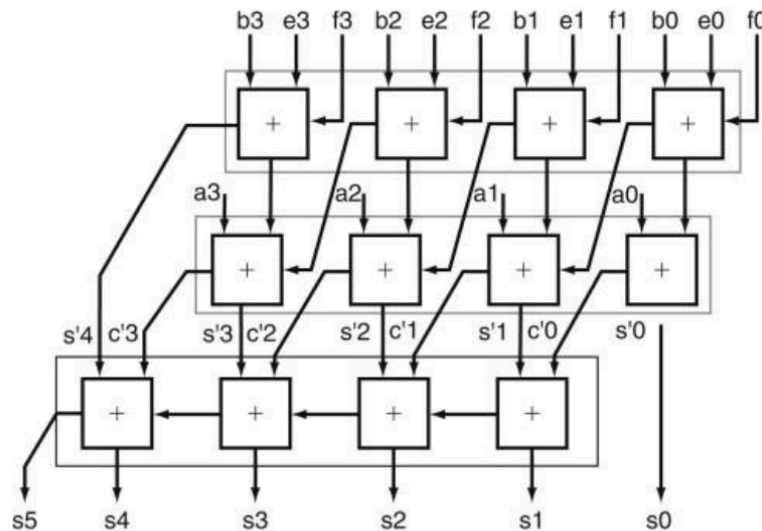
(آ) عملیات‌های ضرب را به صورت کامل شرح دهید. (این عملیات ضرب را باید دوبار انجام دهید و هر بار مضروب<sup>۲</sup> را تغییر دهید.)

(ب) تاخیرهای هر عملیات را محاسبه کرده و با هم مقایسه نمایید.

۴. ساختار جمع‌کننده carry select را در نظر بگیرید. با فرض اینکه تاخیر هر دو خروجی جمع‌کننده کامل، و همچنین تاخیر مالتیپلکسر ۱ واحد زمانی باشد، حداکثر تعداد بیتی را که می‌توان در k واحد زمانی جمع زد بدست آورید. همچنین به صورت مختصر جواب خود را توجیه کنید. (راهنمایی ۱: می‌توانید از چند لایه مالتیپلکسر استفاده کنید.) (راهنمایی ۲: جواب شما می‌تواند به فرم یک رابطه بازگشتی باشد. در صورتی که پاسخ شما اینگونه بود، حالت پایه را فراموش نکنید.)

۵. یک روش برای جمع کردن چندین عدد با هم، استفاده از Carry Save Adder است.

(الف) فرض کنید می‌خواهیم ۴ عدد را طبق شکل زیر با هم جمع کنیم. در مورد این روش تحقیق کنید و عملکرد آن را توضیح دهید.



ب) اگر در مرحله‌ی آخر به جای جمع‌کننده‌ی عادی از یک Carry Look Ahead استفاده کنیم، تاخیر این روش را محاسبه کنید.  
فرض کنید:

$$d_{AND} + d_{OR} = d_{XOR} = d$$

۶. سوال اختیاری - فرض کنید A و B دو عدد n بیتی باشند. می‌خواهیم این دو عدد را با استفاده از الگوریتم ضرب Shift-and-Add در یکدیگر ضرب کنیم. A و B با احتمال یکسان، هر عدد n بیتی‌ای می‌توانند باشند.

(آ) امید ریاضی تعداد عملیات‌های جمع n بیتی، تعداد عملیات‌های تفریق n بیتی، و تعداد عملیات‌های انتقال<sup>۳</sup> را به ازای ورودی‌های مختلف ممکن برای A و B بیابید.

(ب) فرض کنید تاخیر هر عمل جمع 10ns، هر عمل تفریق 15ns و هر عمل انتقال 5ns باشد. متوسط زمان تاخیر را برای این دو عدد n بیتی محاسبه کنید.

۷. سوال اختیاری - در این سوال می‌خواهیم جمع‌کننده‌های دیگری که می‌توانند از جمع‌کننده انتخابی بهتر عمل کنند را مورد بررسی قرار دهیم و با هم مقایسه کنیم.

(آ) فرض کنید می‌خواهیم ۲ عدد ۱۲۸ بیتی را با هم جمع کنیم به گونه‌ای که کمترین تاخیر ممکن را داشته باشیم، برای این کار می‌خواهیم یک جمع‌کننده انتخابی طراحی کنیم. فرض کنید تاخیر جمع‌کننده‌های کامل با مالتیپلکسرها برابر باشد. بهترین جمع‌کننده انتخابی را طراحی کنید و گروه بندی و سائز هر گروه را نیز تعیین کنید همچنین مقدار تاخیر را نیز بر حسب D محاسبه کنید.

(ب) حال فرض کنید می‌خواهیم این ۲ عدد را به گونه‌ای جمع کنیم که کمترین هزینه ممکن را مصرف کنیم برای این کار از جمع‌کننده انتخابی استفاده می‌کنیم و به شرطی که تاخیر جمع‌کننده از  $(9 \times \log_2(n) + 2)D$  که n تعداد بیت‌های ورودی هر عدد است بیشتر نشود. بهترین جمع‌کننده انتخابی را طراحی کنید و گروه بندی و سائز هر گروه را نیز تعیین کنید همچنین مقدار هزینه نهایی را نیز بر حسب C محاسبه کنید. (فرض کنید که هزینه هر جمع‌کننده کامل تک بیتی به صورت C باید و هزینه هر مالتیپلکسر با هر سائزی به صورت 4C باشد، برای تاخیر می‌توانید از فرض قسمت (آ) استفاده کنید.)

(ج) در مورد جمع‌کننده Brent-Kung تحقیق کنید و پیچیدگی تاخیر و هزینه را بر حسب O بنویسید و به صورت خلاصه در مورد کاربردهای آن توضیح دهید. در نهایت این جمع‌کننده را با جمع‌کننده‌های (آ) و (ب) بر حسب تاخیر و هزینه به صورت کلی مقایسه کنید.

(د) درمورد جمع کننده Kogge-Stone تحقیق کنید و پیچیدگی تاخیر و هزینه را بر حسب O بنویسید و به صورت خلاصه درمورد کاربردهای آن توضیح دهید. در نهایت این جمع کننده را با جمع کننده‌های (آ) و (ب) بر حسب تاخیر و هزینه به صورت کلی مقایسه کنید.

## تمارین عملی

۱. در این تمرین قصد داریم یک ALU طراحی کنیم.

### Inputs:

- a (32 bits)
- b (32 bits)
- aluop (4 bits)
- output\_inverted
- output\_inc
- clk
- rst

### Outputs:

- res\_low (32 bits)
- res\_high (32 bits)
- done

لازم است این واحد محاسباتی، از عملیات زیر با شماره مشخص شده، پشتیبانی کند.

- ADD = 0
- SUB = 1
- MUL = 2
- DIV = 3
- AND = 4
- OR = 5
- XOR = 6
- CLO = 7
- CLZ = 8
- SLL = 9
- SRL = 10
- SRA = 11
- ROTR = 12

در طراحی خود لازم است که به نکات زیر توجه کنید:

- برای پیاده‌سازی عملیات تقسیم لازم است از مازول طراحی شده در تمرین ۲ استفاده کنید.
- عملیات ضرب به گونه‌ای باید پیاده‌سازی شود که در ۱۶ کلاک نتیجه آن در خروجی قرار بگیرد.
- برای انجام عملیات ضرب لازم است که در ابتدا یک carry-select-adder چهار قطعه بسازید. سپس به واسطه آن و با طراحی یک carry-save-adder به طوری که ۳ عدد ۳۲ بیتی را جمع کند، طراحی ضرب کننده را کامل کنید.

- سیگنال شروع ضرب کننده و تقسیم کننده باید در اولین چرخه که aluop برابر عملیات مورد نظر شد ۱ شود و پس از آن ۰ شود.
- در عملیات های ترکیبی دیگر باید سیگنال done مقدار ۱ شود.
- CL(O/Z) : شمارش تعداد بیت ۱/۰ مقدم ( leading one/zero ) .
- پس از انجام عملیات مشخص شده، ابتدا نتیجه در صورت ۱ بودن بیت output\_inverted وارون میشود و سپس با output\_inc جمع میشود.
- در طراحی خود به جز موارد ذکر شده می‌توانید از مازول‌های آماده در logisim-evolution استفاده کنید.

نحوه داوری این سوال به صورت زیر خواهد بود:

```
1 ./synth_valid.sh schematic.circ HW3/tb1.v
```