معماری کامپیوتر

نیم‌سال دوم ۱۴۰۳-۱۴۰۲

دانشکده مهندسی کامپیوتر

تمرین دو – محاسبات **مهلت تحویل ساعت ۲۴ روز جمعه ۱۱ اسفند**

به موارد زیر توجه کنید:

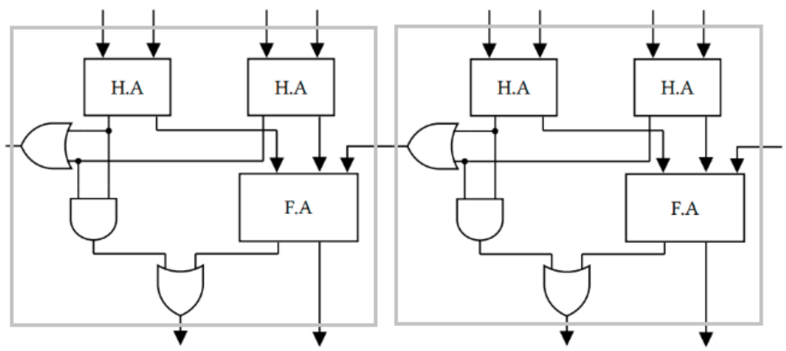
۱- حتما نام و شماره دانشجویی خود را روی پاسخنامه بنویسید.

۲- کل پاسخ تمرینات را در قالب یک فایل pdf با شماره دانشجویی خود نام‌گذاری کرده در سامانه CW بارگذاری کنید.

۳- این تمرین ۶۰ نمره دارد که معادل ۰٫۶ نمره از نمره کلی درس است.

۴- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر کل نمره این تمرین را از دست خواهند داد.

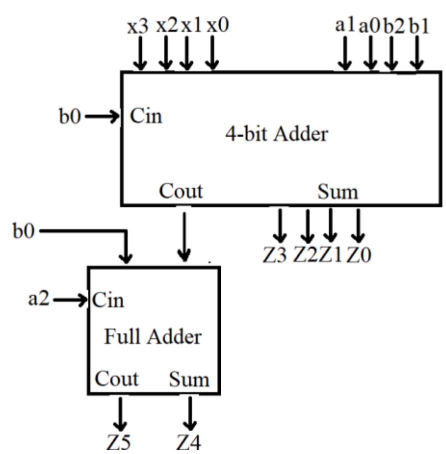
۱- (۱۰ نمره) در شکلِ زیر دو سلول که برای انجام محاسبهٔ خاصی طراحی شده، در کنار هم قرار گرفته است. اگر دو سلول دیگر را به همین ترتیب به آن اضافه کنیم، تاخیر محاسبهٔ نتیجهٔ نهایی چه تغییری می‌کند؟



پاسخ:

تاخیر در انجام عملیات هیچ تغییری نمی‌کند. تنها همبستگی این سلول‌ها از طریق OR کردن دو خروجی HAها به‌دست می‌آید. دقیقا در همان زمانی که برای تولید ورودی نقلی از سلول اول به سلول دوم صرف می‌شود، سایر ورودی‌های نقلی نیز تولید می‌شوند. (دقت کنید که در اینجا منظور از نقلی، بیت carry در جمع نیست بلکه منظور بیتی است که بین دو سلول جابجا می‌شود.)

۲- (۱۰ نمره) فرض کنید نمایش bitwise سه عدد X=x3x2x1x0 و A=a2a1a0 و B=b2b1b0 را داریم. حال آنها را به این صورت وارد این مدار شکل زیر میکنیم. با syntax یکی از زبان‌های برنامه‌نویسی سطح بالا مثل C یا Python، بگویید حاصل این مدار (Z=Z5Z4Z3Z2Z1Z0) چگونه از A و B و X ساخته می‌شود. اجازهٔ استفاده از همهٔ عملیات منطقی و محاسباتی زبان سطح بالا (مثل / و ٪ و .... در C) را دارید.



پاسخ:

```
Z = X + (4*(A%4)) + (B/2) + (B%2)*(1+16) + (A/4)*16
```

۳- (۱۰ نمره) تاخیرِ یک جمع‌کنندهٔ ۱۶ بیتی را که از دو لایه (Carry-look-ahead) CLA استفاده می‌کند با تاخیر یک جمع‌کنندهٔ Carry Select که از چهار لایه هر کدام با ۴ تمام‌افزا (Full-Adder) تشکیل شده مقایسه کنید.
در محاسباتِ خود تاخیر هر مالتی‌پلکسر را DMUX، تاخیر هر گیت XOR را DXOR و تاخیر هر گیت AND یا OR را DG بنامید و فرض کنید همیشه DG<DXOR<2DG و DMUX=DXOR باشد.
پاسخ:
تاخیر یک جمع‌کنندهٔ ۱۶ بیتی که از دو لایه CLA استفاده می‌کند، مطابق با نمودار زیر، برابر با $\boldsymbol{2DXOR+6DG}$ می‌شود.

$$a_i,b_i \xrightarrow{\max(XOR,AND)=DXOR} p_i,g_i \xrightarrow{AND+OR=2DG} P_k,G_K \xrightarrow{AND+OR=2DG} C_k \xrightarrow{AND+OR=2DG} c_i \xrightarrow{XOR=DXOR} s_i$$

در یک جمع‌کنندهٔ Carry Select که چهار مرحلهٔ مساوی ۴ بیتی دارد، بیت‌های نقلی همهٔ مراحل بعد از $8DG$ حاضر می‌شود ولی در هر مرحله انتخاب بیت نقلی خروجی باید پس تعیین بیت نقلی مرحلهٔ قبل انجام شود که به اندازهٔ تاخیر یک مالتی‌پلکسر طول می‌کشد. بنابراین تاخیر کل، مطابق با نمودرا زیر، برابر است با $\boldsymbol{8DG+3DMUX}$

$$Block_0 \xrightarrow{8DG} Block_1 \xrightarrow{DMUX} Block_2 \xrightarrow{DMUX} Block_3 \xrightarrow{DMUX} Block_4$$

با توجه به مفروضاتِ سوال، تاخیر جمع‌کنندهٔ دومی بیشتر است.

۴- (۱۰ نمره) می‌خواهیم یک جمع‌کنندهٔ (Carry Select Adder) CSA بسازیم که یک جمعِ ۶۴ بیتی را در k مرحلهٔ نامساوی انجام دهد. فرض کنید تاخیر تمام‌افزا و مالتی‌پلکسر برابر است. توضیح دهید تعداد بیت هر مرحله چند باشد که نتیجهٔ نهایی با کمترین تاخیر به دست آید؟ تاخیر نهایی را حساب کنید.
پاسخ:
وقتی تعدادِ مراحل نامساوی است، کمترین تاخیر وقتی رخ می‌دهد که نتیجهٔ جمع در هر مرحله همزمان با نتیجهٔ بیتِ نقلیِ مرحلهٔ قبل آماده شود. اگر تاخیر تمام‌افزا و مالتی‌پلکسر برابر باشد، این شرایط وقتی رخ می‌دهد که تعداد جمع‌کننده‌های دو مرحلهٔ اول مساوی باشد و تعداد جمع‌کننده‌ای مراحل سه به بعد، هر کدام یک بیت بیشتر از مرحلهٔ قبل باشد. بنابراین، اگر تعداد جمع‌کننده‌های مرحلهٔ اول و دوم را x و تعداد کل مراحل را n بنامیم، باید معادلهٔ زیر را حل کنیم:

$x+x+(x+1)+(x+2)+\cdots+(x+n-2)=64$

$nx+(1+2+\cdots+n-2)=64$

در این صورت، اگر تاخیر تمام‌افزا و مالتی‌پلکسر برابر باشد، کل تاخیر برابر مجموعِ تاخیر یک جمع‌کنندهٔ $x+n-2$ بیتی و یک‌مالتی‌پلکسر خواهد بود که برابر است با $x+n-1$ تمام‌افزا. اگر بخواهیم این تاخیر کمینه شود، یکی از دو ترکیب زیر پاسخ ما خواهد بود که در حالت دوم $x=4,n=9$ تاخیر برابر با $12D_{FA}$ می‌شود.

$x=9,n=6 \ or \ x=4,n=9$

۵- (۱۰نمره) تقسیمِ ۲۰۳ بر ۱۷ را طبق نمودار و سخت‌افزارِ اسلایدِ ۳۹ انجام دهید و پاسخ خود را در جدولی مشابه با جدول زیر وارد کنید. مقسوم را ۱۰ بیتی و مقسوم‌علیه را ۵ بیتی فرض کنید.
دو سطر اول جدول برای آشنایی شما با روند کار پر شده است.

	remainder	divisor	rem-div (< or >)	quotient
0	0011001011	1000100000	<	0
1	0011001011	0100010000	<	00
2	0011001011	0010001000	>	001
3	0001000011	0001000100	<	0010
4	0001000011	0000100010	>	00101
5	0000100001	0000010001	>	**001011**
6	**0000010000**			

با توجه به نتیجهٔ سوالِ قبل، خارج‌قسمت و باقی‌ماندهٔ تقسیم‌های زیر را به دست آورید. در هر مورد توضیح دهید علامت‌های خارج‌قسمت و باقی‌مانده را بر چه اساسی به دست آورده‌اید.

$(-203)/(-17)$ Quoteint=+11 Remainder=-16
$(-203)/(+17)$ Quoteint=-11 Remainder=-16
$(+203)/(-17)$ Quoteint=-11 Remainder=+16

باقی‌مانده با مقسوم‌علیه و خارج‌قسمت با حاصل‌ضرب علامت‌های مقسوم و مقسوم‌علیه هم‌علامت است.

۶- (۱۰ نمره) در یک روشِ نمایشِ اعدادِ اعشاری آنها به صورت یک آرایه ۳۲ بیتی نمایش داده می‌شوند که بیت اول بیت علامت است و بعد از آن x بیت به نما (exponent) و y بیت به بخشِ کسری (fraction) تخصیص داده می‌شود. این نحوهٔ نمایش مشابه نمایش Single Precision Floating Point است که در درس با آن آشنا شدید، البته با x و y متفاوت. ضمنا در این روش نمایش نما هم به صورت biased نمایش داده می‌شود، اما bias آن هر مقداری می‌تواند باشد. می‌دانیم نمایش عدد ۷/۷- با این روش معادلِ $(BBD99999)_{16}$ است. مقادیر x و y و bias را بیابید.

پاسخ:

$(7)_{10} = (111)_2$ $(0.7)_{10} = (1\overline{0110})_2$ $\Rightarrow (7.7)_{10} = (111.1\overline{0110})_2 = (1.111\overline{0110})_2 \times 2^2$

$(BBD99999)_{16} = (\mathbf{1}\ 0111011110110011001100110011001)_2$

می‌بینیم که چهار بیت ۰۱۱۰ باید چندین بار در بخش کسری تکرار شده باشد و پیش از آن باید ۱۱۱ داشته باشیم، بنابراین می‌توانیم شروع و خاتمه بخش کسری را پیدا کنیم.

$(BBD99999)_{16} = (\mathbf{1}\ 011101\ \mathbf{1110110011001100110011001})_2$

بنابراین $\boldsymbol{y = 25}$ و $\boldsymbol{x = 6}$ است و نما برابر با ۰۱۱۱۰۱ است.

$(011101)_2 = (29)_{10} = \text{bias} + 2 \Rightarrow \mathbf{bias = 27}$