



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 نام گذاری 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- (۸ نمره) یک مسیر داده تک چرخه‌ای در اختیار داریم که هر چرخه آن در زمان $T = 20 \text{ ns}$ اجرا می‌شود. معمار سخت‌افزار مسیر داده را با یک خط لوله n مرحله‌ای جایگزین کرده است که زمان اجرای هر مرحله آن برابر است با $T_p = T/n + 0.05n$. بیشینه گذردهی (throughput) بر حسب n و با فرض این که دنباله طولانی از دستورات وارد مسیر داده می‌شود، چقدر است؟

پاسخ:

گذردهی یا Throughput برابر با نسبت تعداد دستورات به زمان کل اجرای همه دستورات است. برای m دستور، فرمول گذردهی بدین شکل خواهد بود:

$$\text{throughput} = \frac{m}{(m-1)T_p + nT_p}$$

زمانی که m به سمت بی‌نهایت میل کند، گذردهی برابر با $\frac{1}{T_p}$ می‌شود. در این حالت می‌خواهیم گذردهی بیشینه شود. بنابراین مشتق گذردهی را برابر صفر قرار می‌دهیم:

$$\frac{1}{T_p} = \frac{1}{\frac{20}{n} - \frac{n}{20}} \Rightarrow \frac{2n(20n) - 20(400 + n^2)}{(20n)^2} = 0$$

پاسخ این معادله $n = 20$ است، بنابراین گذردهی بیشینه برابر است با:

$$\frac{1}{T_p} = \frac{1}{\left(\frac{20}{20} + \frac{20}{20}\right) \times 10^{-9}} = 5 \times 10^8$$

۲- (۸ نمره) در یک پردازنده از یک خط لوله ۱۳ مرحله‌ای برای واکشی و اجرای دستورات استفاده می‌شود. اگر ۲۵٪ دستورات برنامه پرش باشند، و با فرض زیاد بودن دستورات، بیشینه و کمینه تسریع قابل دستیابی توسط این پردازنده نسبت به پردازنده مشابه غیرلوله‌ای چقدر خواهد بود؟ (فرض کنید وابستگی‌های ساختاری و داده‌ای وجود ندارد.)

پاسخ:

با فرض این که در این برنامه n دستور داشته باشیم و زمان اجرای هر مرحله از خط لوله T_p باشد، تسریع از رابطه زیر محاسبه می‌شود:

$$\text{Speedup} = \frac{13nT_p}{\text{pipeline execution time}}$$

بیشینه تسریع وقتی رخ می‌دهد که همه دستورات پرش بدون تعلیق اجرا شوند. در این صورت، زمان اجرا از رابطه زیر به دست می‌آید:

$$\min(\text{pipeline exec time}) = (n - 1)T_p + 13T_p = (n - 12)T_p$$

بنابراین:

$$\text{Speedup}_{\max} = \lim_{n \rightarrow \infty} \frac{13nT_p}{(n - 12)T_p} = 13$$

کمینه تسریع وقتی رخ می‌دهد که برای اجرای هر دستور پرش به ۱۲ تعلیق نیاز داشته باشیم. در این صورت یک چهارم دستورات در ۱۳ مرحله و بقیه در یک مرحله اجرا می‌شوند، بنابراین زمان متوسط اجرای هر دستور به این صورت محاسبه می‌شود:

$$T_{\text{instr}} = 0.75 \times T_p + 0.25 \times 13 \times T_p = 4T_p$$

و در نتیجه داریم:

$$\text{Speedup}_{\min} = \lim_{n \rightarrow \infty} \frac{13nT_p}{(n - 1)4T_p + (13 + 13/4)T_p} = 13/4 = 3.25$$

با استدلال دیگری می‌توانیم به نتیجه مشابهی برسیم:

اگر هر دستور پرش نیاز به ۱۲ تعلیق داشته باشد، مثل این است که به ازای هر دستور پرش ۱۳ دستور عادی اجرا می‌کنیم، بنابراین به جای اجرای n دستور، $4n = (0.25n + 0.75 \times 13n)$ دستور اجرا می‌کنیم. تسریع حاصل را می‌توانیم این طور محاسبه کنیم:

$$\text{Speedup}_{\min} = \lim_{n \rightarrow \infty} \frac{13nT_p}{(4n - 1)T_p + 13T_p} = 13/4 = 3.25$$

۳- (۴۴ نمره) قطعه کد زیر را در نظر بگیرید. در ابتدای اجرای این قطعه کد، دو ثبات \$3 و \$5 حاوی آدرس ابتدای دو آرایه درون حافظه هستند و محتوای ثبات \$2 صفر است.

```
loop:  lw $1, $2($3)      # $1 ← Mem[$2+$3]
      lw $4, $2($5)      # $4 ← Mem[$2+$5]
      add $6, $1, $4
      sw $6, $2($3)      # Mem[$2+$3] ← $6
      add $2, $2, 4
      bne $2, 400, loop
      lw $6, 100         # $6 ← Mem[100]
```

الف) توضیح دهید این قطعه کد چه کاری انجام می‌دهد؟

پاسخ:

این قطعه برنامه محتوای نظیر به نظیر خانه‌های دو آرایه را که در آدرس‌های ذخیره شده در \$3 و \$5 قرار دارند، با هم جمع می‌کند و نتیجه را آرایه اول ذخیره می‌کند. این کار را برای ۱۰۰ خانه متوالی تکرار می‌کند و در آخر هم عدد ۱۰۰ را در ثبات \$6 قرار می‌دهد.

ب) خط لوله ۵ مرحله‌ای IF, ID, EX, MEM و WB را در نظر بگیرید. فرض کنید فایل ثبات‌ها (register file) امکان خواندن و نوشتن همزمان در یک چرخه ساعت را ندارد. همچنین فرض کنید قابلیت forwarding و

پیش‌بینی پرش هم در خط لوله وجود ندارد و هیچ سخت‌افزار اضافه هم برای مقایسه ثبات‌ها و محاسبه مقصد پرش تعبیه نشده است. بنابراین، دستوری که باید پس از دستور پرش اجرا شود در مرحله EX تعیین می‌شود. مراحل اجرای یک مرحله از حلقه بالا را در این خط لوله نشان دهید. (طبعاً نیازی به نمایش مراحل اجرای دستور آخر که خارج از حلقه است، نیست) همچنین تعداد چرخه‌های لازم را برای اجرای کلی برنامه به دست آورید. در تمامی قسمت‌های سوال زمانی که دستور بعدی قابل اجرا نیست یک دستور NOP اجرا کنید و توجه کنید که اجرای گسسته بخش‌های خط لوله ممکن نیست. یعنی اجرا به شکل زیر غیرمجاز است.

```
if | id | ex | mem | wb
    | if |    | id  | ex  | mem | wb
```

پاسخ:

اجرای یک مرحله از حلقه طبق جدول زیر انجام می‌شود و ۱۷ چرخه طول می‌کشد:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
lw	IF	ID	EX	MEM	WB														
lw		IF	ID	EX	MEM	WB													
add			-	-	-	IF	ID	EX	MEM	WB									
sw							-	-	-	IF	ID	EX	MEM	WB					
add											IF	ID	EX	MEM	WB				
jne												-	-	-	IF	ID	EX	MEM	WB
lw																		IF	ID

چون حلقه ۱۰۰ بار تکرار می‌شود و هر بار اجرای آن ۱۷ چرخه طول می‌کشد، اجرای کل برنامه ۱۷۰۰ چرخه برای حلقه و ۵ چرخه برای دستور آخر طول خواهد کشید، یعنی در مجموع ۱۷۰۵ چرخه.

ج) فرض کنید فایل ثبات‌ها امکان خواندن و نوشتن همزمان در یک چرخه ساعت را دارند، به این ترتیب که نوشتن در لبه بالا رونده و خواندن در لبه پایین رونده انجام می‌شود. یک بار دیگر به سوال بند ب پاسخ دهید.

پاسخ:

این بار، هر بار اجرای حلقه سه چرخه کمتر طول می‌کشد چون مراحل WB و ID دو دستور متفاوت را می‌توانیم همزمان انجام دهیم.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
lw	IF	ID	EX	MEM	WB														
lw		IF	ID	EX	MEM	WB													
add			-	-	IF	ID	EX	MEM	WB										
sw						-	-	IF	ID	EX	MEM	WB							
add									IF	ID	EX	MEM	WB						
jne										-	-	IF	ID	EX	MEM	WB			
lw															IF	ID	EX	MEM	WB

چون حلقه ۱۰۰ بار تکرار می‌شود و هر بار اجرای آن ۱۴ چرخه طول می‌کشد، اجرای کل برنامه ۱۴۰۰ چرخه برای حلقه و ۵ چرخه برای دستور آخر طول خواهد کشید، یعنی در مجموع ۱۴۰۵ چرخه.

د) فرض کنید خط لولهٔ بند ج، امکان forwarding از مراحل wb و mem به مرحلهٔ ex را هم دارد. به سوال بند ب دوباره پاسخ دهید.

پاسخ:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
lw	IF	ID	EX	MEM	WB														
lw		IF	ID	EX	MEM	WB													
add			-	IF	ID	EX	MEM	WB											
sw					IF	ID	EX	MEM	WB										
add						IF	ID	EX	MEM	WB									
bne							IF	ID	EX	MEM	WB								
lw								-	-	IF	ID	EX	MEM	WB					

چون حلقه ۱۰۰ بار تکرار می‌شود و هر بار اجرای آن ۹ چرخه طول می‌کشد، اجرای کل برنامه ۹۰۰ چرخه برای حلقه و ۵ چرخه برای دستور آخر طول خواهد کشید، یعنی در مجموع ۹۰۵ چرخه.

ه) فرض کنید سازوکار پیش‌بینی پرش هم به خط لولهٔ بند د اضافه شده است و فرض کنید که این خط لوله پیش‌بینی می‌کند که همهٔ پرش‌ها انجام می‌شود. به این ترتیب که دستور مقصد دستور پرش پس از پایان مرحلهٔ ID وارد خط لوله می‌شود. اگر پیش‌بینی غلط باشد، این دستور و همه دستوراتی که پس از آن وارد خط لوله شده‌اند تخلیه شده و دستور درست وارد می‌شود. یک بار دیگر به سوال بند ب پاسخ دهید.

پاسخ:

این بار تفاوت در این است که دستور بعد از دستور پرش بلافاصله بعد از آن اجرا می‌شود، بنابراین خود حلقه در ۸ چرخه کامل می‌شود و زمانی که پرش انجام شود (یعنی ۱۰۰ بار اول) پیش‌بینی درست بوده، بنابراین اجرای کل حلقه ۸۰۰ چرخه طول می‌کشد، اما دستور آخر با یک چرخه تاخیر اجرا خواهد شد، چون پیش‌بینی اشتباه است و یک بار باید خط لوله خالی شود. بنابراین کل برنامه در ۸۰۶ چرخه اجرا می‌شود.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
lw	IF	ID	EX	MEM	WB														
lw		IF	ID	EX	MEM	WB													
add			-	IF	ID	EX	MEM	WB											
sw					IF	ID	EX	MEM	WB										
add						IF	ID	EX	MEM	WB									
bne							IF	ID	EX	MEM	WB								
lw								IF	ID	EX	MEM	WB							