



به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخ نامه 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- (۴ نمره) پردازنده ای داریم که بدون استفاده از خط لوله با نرخ clock برابر ۲,۵ گیگاهرتز هر دستور را به طور متوسط در ۴ چرخه اجرا می کند. همین پردازنده به یک پردازنده خط لوله با پنج مرحله ارتقا یافته است، اما به دلیل تاخیر داخلی خط لوله، سرعت clock به ۲ گیگاهرتز کاهش یافته است. فرض کنید در خط لوله هیچ تاخیری وجود ندارد. نرخ افزایش سرعت (تسریع) چند است؟

پاسخ:

اجرای هر دستور بدون خط لوله ۴ چرخه ساعت و با خط لوله یک چرخه ساعت طول می کشد.

$$Exec_{NonPipeline} = \frac{4}{2.5 \times 10^9} = 1.6 \times 10^{-9} s$$

$$Exec_{Pipeline} = \frac{1}{2 \times 10^9} = 0.5 \times 10^{-9} s$$

$$Speedup = \frac{1.6 \times 10^9}{0.5 \times 10^9} = 3.2$$

۲- (۴ نمره) یک خط لوله با ۴ مرحله داریم که به ترتیب ۸۰۰ و ۵۰۰ و ۴۰۰ و ۳۰۰ پیکوثانیه زمان اجرا دارند. مرحله اول از این خط لوله را با دو مرحله که عملکرد مشابهی دارند جابه جا می کنیم که هر کدام به ترتیب ۶۰۰ و ۳۵۰ پیکوثانیه زمان اجرا دارند. تسریع حاصل چقدر است؟

پاسخ:

چرخه ساعت در هر دو خط لوله برابر با طولانی ترین زمان اجرا است که در خط لوله اول ۸۰۰ و در خط لوله دوم ۶۰۰ پیکوثانیه است. با توجه به این که اجرای هر دستور در خط لوله به اندازه یک چرخه ساعت طول می کشد، تسریع حاصل به این صورت حساب می شود:

$$Speedup = \frac{800 \times 10^9}{600 \times 10^9} = 1.33$$

۳- (۴ نمره) خط لوله D1 با ۵ مرحله که هر کدام به ترتیب ۳، ۲، ۴، ۳ و ۲ نانوثانیه زمان اجرا دارند موجود است. همچنین یک خط لوله با نام D2 داریم که ۸ مرحله اجرا دارد که هر کدام ۲ نانوثانیه زمان اجرا دارند. نسبت زمان اجرای N دستور را در دو خط لوله به دست آورید و توضیح دهید کدامیک سریع‌ترند و چرا.

پاسخ:

زمان اجرای N دستور در هر خط لوله با K مرحله که هر مرحله T ثانیه طول می‌کشد، از رابطه زیر به دست می‌آید:

$$ExecTime = KT + (N - 1)T = NT + (K - 1)T$$

با توجه به این که زمان اجرای هر مرحله در خط لوله اول ۴ و در خط لوله دوم ۲ نانوثانیه است، نسبت زمان‌های اجرای دو خط لوله به این ترتیب محاسبه می‌شود:

$$\frac{ExecTime_1}{ExecTime_2} = \frac{4N + 4 \times (5 - 1)}{2N + 2 \times (8 - 1)} = \frac{4N + 16}{2N + 14}$$

بنابراین، همیشه زمان اجرای N دستور در خط لوله اول بیشتر طول می‌کشد.

۴- (۸ نمره) یک خط لوله (pipeline) متشکل از ۴ مرحله داریم که زمان اجرای هر کدام به ترتیب ۶۰ و ۵۰ و ۹۰ و ۸۰ نانوثانیه است. همین‌طور یک latch delay به اندازه ۱۰ نانو ثانیه داریم.

هر یک از مقادیر زیر را حساب کنید:

الف- چرخه ساعت (cycle time) این خط لوله

ب- زمان اجرای هر دستور بدون خط لوله

ج- ضریب افزایش سرعت با استفاده از خط لوله

د- زمان اجرای ۱۰۰۰ دستور متوالی و بدون وابستگی در این خط لوله

پاسخ؛

الف- چرخه ساعت برابر با ماگزیمم تاخیر هر مرحله است. در این سوال چون تاخیر latch موردنیاز بین هر دو مرحله جداگانه داده شده، باید ماگزیمم زمان اجرا را با تاخیر latch جمع کنیم، بنابراین چرخه ساعت ۱۰۰ نانوثانیه است.

ب- زمان اجرای هر دستور بدون خط لوله برابر با حاصل جمع زمان اجرای هر مرحله است. (در این مورد نیازی به latch‌های بین هر دو مرحله نداریم.)

$$Exec_{NonPipeline} = 60 + 50 + 90 + 80 = 280 \text{ ns}$$

ج-

$$Speedup = \frac{Exec_{NonPipeline}}{Exec_{Pipeline}} = \frac{280 \times 10^9}{100 \times 10^9} = 2.8$$

د-

$$ExecTime = KT + (N - 1)T = NT + (K - 1)T = (1000 + (4 - 1)) \times 100 = 100300 \text{ ns}$$

۵- (۲۰ نمره) فرض کنید یک پردازنده MIPS مبتنی بر خط لوله پنج مرحله‌ای داریم. (IF و ID در نیمه دوم و WB در نیمه اول هر چرخه عمل می‌کنند) فرض کنید در ابتدا در این خط لوله forwarding نداریم و نتیجه پرش در مرحله EX مشخص می‌شود. قطعه کد زیر را در نظر بگیرید.

```
xor $t1,$t1,$t1
addi $t0,$0,100
loop:
add $t1,$t1,$t0
sub $t0,$t0,1
bne $t0,$0,loop
```

الف- وابستگی‌های داده‌ای آن را پیدا کنید.

پاسخ:

دستور سوم به دستور اول و دوم

دستور چهارم به دستور دوم

دستور پنجم به دستور چهارم

ب- اجرای این قطعه کد چند چرخه (Cycle) طول می‌کشد؟ دیاگرام ساده‌ای از خط لوله آن را رسم کنید.

پاسخ:

xor	IF	ID	EX	MEM	WB											
addi		IF	ID	EX	MEM	WB										
add			-	-	IF	ID	EX	MEM	WB							
sub						IF	ID	EX	MEM	WB						
bne							-	-	IF	ID	EX	MEM	WB			
add										-	-	IF	ID	EX	MEM	WB

در هر بار اجرای حلقه، دو تعلیق برای اجرای bne داریم (چون به دستور قبلی وابسته است) و دو تعلیق برای اجرای add داریم (بار اول چون دستور add به دو دستور قبلی وابسته است و بارهای بعدی چون اجرای دستور add با فاصله بعد از bne اجرا می‌شود. بنابراین مثل این است که در حلقه ۷ دستور داریم.

کل حلقه صد بار تکرار می‌شود و دو دستور هم ابتدای حلقه داریم، بنابراین در مجموع ۷۰۲ دستور باید اجرا شود که اولی ۵ چرخه و بقیه هر کدام یک چرخه طول می‌کشند و در مجموع برنامه در ۷۰۶ چرخه اجرا می‌شود. اگر فرض کنیم، در انتهای این قطعه برنامه دستور دیگری هم داریم که باید پس از پایان یافتن حلقه اجرا شود، این دستور بعد از ۷۰۸ چرخه اجرا می‌شود، چون آدرس دستور بعد از bne بعد از دو حلقه در PC قرار می‌گیرد.

ج- اکنون با فرض اینکه نتیجه پرش در پایان مرحله ID مشخص شود، دوباره به قسمت ب پاسخ دهید.

پاسخ:

xor	IF	ID	EX	MEM	WB											
addi		IF	ID	EX	MEM	WB										
add			-	-	IF	ID	EX	MEM	WB							
sub						IF	ID	EX	MEM	WB						
bne							-	-	IF	ID	EX	MEM	WB			
add										-	IF	ID	EX	MEM	WB	

در هر بار اجرای حلقه، باز هم دو تعلیق برای اجرای bne داریم، اما برای اجرای add بار اول دو تعلیق و دفعات بعدی یک تعلیق داریم. بنابراین مثل این است که در حلقه ۶ دستور داریم، اما پیش از شروع حلقه باید ۳ دستور اجرا کنیم، پس تعداد کل دستورات ۶+۳=۹ دستور است که اجرای آنها ۹+۶=۱۵ چرخه طول می‌کشد.

اگر فرض کنیم، در انتهای این قطعه برنامه دستور دیگری هم داریم که باید پس از پایان یافتن حلقه اجرا شود، این دستور بعد از ۹ چرخه اجرا می‌شود، چون آدرس دستور بعد از bne بعد از یک حلقه در PC قرار می‌گیرد.

د- اکنون علاوه بر فرض قسمت ج، فرض کنید در ISA پردازنده ما یک دستور به نام LEA وجود داشته باشد. این دستور حاصل جمع یا تفریق را در مرحله ID به دست می‌آورد. (این دستور در MIPS نیست ولی در x86 وجود دارد) با جایگذاری سه دستور جمع و تفریق موجود در کد بالا با LEA و با داشتن forwarding از مرحله ID به ID، به سوال بند ب دوباره پاسخ دهید.

پاسخ:

xor	IF	ID	EX	MEM	WB									
addi		IF	ID	EX	MEM	WB								
add			-	IF	ID	EX	MEM	WB						
sub					IF	ID	EX	MEM	WB					
bne						IF	ID	EX	MEM	WB				
add							-	IF	ID	EX	MEM	WB		

این بار اولین اجرای دستور add یک تعلیق نیاز دارد، چون همچنان به نتیجه دستور xor وابسته است. دستور bne به هیچ تعلیقی نیاز ندارد، چون نتیجه دستور قبلی به موقع آماده می‌شود. دستورات add بعدی همچنان به یک چرخه تعلیق نیاز دارند. بنابراین مثل این است که در حلقه ۴ دستور داریم و پیش از شروع حلقه باید ۲ دستور اجرا کنیم. پس تعداد کل دستورات ۴+۲=۶ دستور است که اجرای آنها ۶+۴=۱۰ چرخه طول می‌کشد.

اگر فرض کنیم، در انتهای این قطعه برنامه دستور دیگری هم داریم که باید پس از پایان یافتن حلقه اجرا شود، این دستور بعد از ۱۰ چرخه اجرا می‌شود، چون آدرس دستور بعد از bne بعد از یک حلقه در PC قرار می‌گیرد.

۶- (۲۰ نمره) فرض کنید می‌خواهیم یک پردازنده بسازیم. برای این کار، دو طراحی داریم که هر دو، ISA یکسانی دارند و روی یک خط لوله پنج مرحله‌ای با مراحل F D X M W پیاده‌سازی می‌شوند. این دو طراحی در موارد زیر متفاوت هستند:

- در طراحی A برای هر دستور ممیز شناور باید ۳ تعلیق (stall) در خط لوله ایجاد کنیم. اما در طراحی B قادریم دستورات ممیز شناور را بدون تعلیق اجرا کنیم.
  - در طراحی A یک مدار ساده برای پیش‌بینی پرش داریم که نتیجه پرش را در مرحله D با احتمال ۸۰٪ درست پیش‌بینی می‌کند. در طراحی B یک مدار پیشرفته پیش‌بینی پرش داریم که نتیجه پرش را در همان مرحله D با احتمال ۹۰٪ درست پیش‌بینی می‌کند. در هر دو طراحی اگر پیش‌بینی غلط بود، باید دو دستور بعد از پرش را از خط لوله پاک کنیم که معادل ۲ چرخه تعلیق است.
  - نرخ ساعت در A برابر ۲.۵ و در B برابر ۱ گیگاهرتز است.
- برای یک برنامه تست با نسبت‌های داده‌شده به سوالات زیر پاسخ دهید:

20% conditional branch instructions  
 20% load instructions  
 10% store instructions  
 50% ALU instructions (of which 20% of those are floating point instructions)

الف- با توجه به ساختار pipeline توضیح دهید چرا سرعت کلاک در طراحی پیشرفته B کمتر شده است و این کاهش چه تاثیری می‌تواند در کارایی داشته باشد؟

ب- با توجه به برنامه تست داده‌شده، مقدار میانگین CPI را برای هر طراحی محاسبه کنید.

ج- مدت زمان اجرای ۱ میلیارد دستور در این دو طراحی را با هم مقایسه کنید. کدام سریع‌تر است؟ چرا؟

پاسخ:

توضیح « در هر دو طراحی اگر پیش‌بینی غلط بود، باید دو دستور بعد از پرش را از خط لوله پاک کنیم که معادل ۲ چرخه تعلیق است.» را می‌توانیم به دو شکل برداشت کنیم، اول اینکه این دو تعلیق علاوه بر تعلیقی است که برای همه پرش‌ها داریم و دوم این که این دو تعلیق شامل همان یک تعلیق هم می‌شود. بنابراین در حل این سوال هر دو پاسخ (۲ یا ۳ تعلیق در پیش‌بینی نادرست پرش) قابل قبول است.

الف- چون در طراحی B واحد محاسباتی (EX) هر دستور ممیز شناور را در یک چرخه ساعت انجام می‌دهد. این در حالی است که در طراحی عادی (A)، این قسمت ۴ چرخه ساعت طول می‌کشد. همین مسئله باعث می‌شود تا مجبور باشیم سرعت کلاک را در طراحی پیشرفته کمتر کنیم. کاهش سرعت کلاک می‌تواند تاثیر مستقیمی بر کاهش کارایی سیستم داشته باشد.

ب- در طراحی A در ۸۰٪ دستورات پرش فقط به یک چرخه تعلیق نیاز داریم و برای بقیه ۳ چرخه تعلیق لازم است. در این طراحی برای همه دستورات ممیز شناور به ۲ (یا به تعبیری ۳) تعلیق نیاز داریم و بقیه دستورات بدون تعلیق (فقط در یک چرخه) اجرا می‌شوند.

$$CPI_A = 0.2 \times (0.8 \times 1 + 0.2 \times 2) + 0.5 \times 0.2 \times 3 + 1 = 0.2 \times 1.2 + 0.3 + 1 = 1.54$$

$$CPI_A = 0.2 \times (0.8 \times 1 + 0.2 \times 3) + 0.5 \times 0.2 \times 3 + 1 = 0.2 \times 1.4 + 0.3 + 1 = 1.61$$

$$CPI_B = 0.2 \times (0.9 \times 1 + 0.1 \times 2) + 1 = 0.2 \times 1.1 + 1 = 1.22$$

$$CPI_B = 0.2 \times (0.9 \times 1 + 0.1 \times 3) + 1 = 0.2 \times 1.2 + 1 = 1.24$$

ج-

$$exec_A = \frac{10^9 \times CPI_A}{clock_A} = \frac{1.54 \times 10^9}{2.5 \times 10^9} = 0.616 \text{ s}$$

$$exec_B = \frac{10^9 \times CPI_B}{clock_B} = \frac{1.22 \times 10^9}{1 \times 10^9} = 1.22 \text{ s}$$

طراحی A سریع‌تر این برنامه تست را اجرا خواهد کرد، چون تاثیر سرعت کلاک بیشتر از تاثیر کاهش تعلیق‌ها شده است و عملاً طراحی پیشرفته B به نتیجه بدتری رسیده است.