

پاسخ تمرین شماره ۵ درس معماری کامپیوتر

امیر حسین عاصم یوسفی
۹۶۱۱۰۳۲۳

۲۴ خرداد ۱۳۹۸

سوال ۱

با توجه به این طولانی ترین تاخیر مربوط به Mem می باشد که برابر با ۲ نانوثانیه می باشد و این که تاخیر رجیستر های میانی ۰/۱ نانوثانیه هستند پس برای Clock Time جدید داریم :

$$\text{Clock Time new} = 2\text{ns} + 0.1\text{ns} = 2.1\text{ns}$$

برای این که تاثیر stall را در نظر بگیریم باید CPI را برای زمانی که stall داریم طبق فرمول زیر به دست آوریم :

$$\text{CPI} = \text{IdealCPI} + \text{stall cycles}$$

در این جا $\text{IdealCPI} = 1$ در نظر می گیریم و چون به ازای هر ۴ دستور یک stall داریم پس

$$\text{stall cycles} = 1/4 = 0.25\text{ns}$$

بنابراین طبق فرمول گفته شده $\text{new CPI (with Stall)} = 1 + 0.25 = 1.25\text{ns}$ تا به حال مقادیر CT و CPI را برای معماری Pipeline به دست آوردیم حال Speedup را طبق فرمول زیر به دست می آوریم :

$$\text{SpeedUp} = \frac{\text{Single cycle EXE time}}{\text{Pipeline EXE time}}$$
$$\text{Execution Time} = \text{CPI} \cdot \text{IC} \cdot \text{Cycle time}$$

حال با توجه به بالا داریم :

$$\text{SpeedUp} = \frac{\text{IC} \times 1 \times 7}{\text{IC} \times 1.25 \times 2.1} = \frac{7}{2.625} = 2.67 \text{ ns}$$

سوال ۲

برای این سوال با توجه به قانون Amdahl داریم :

$$\text{SpeedUp} = \frac{1}{\frac{P}{N} + S}$$

که در این جا P نشان دهنده بخشی است که به صورت Parallel نوشته شده و N نشان دهنده تعداد پردازنده ها و S بیانگر بخشی است که به صورت موازی نوشته نشده است . پس داریم :

$$\text{SpeedUp} = \frac{1}{\frac{x}{4} + (1-x)} = 2 \rightarrow x = \frac{2}{3} \Rightarrow x = 66\%$$

بنابراین باید ۶۶ درصد از برنامه را به صورت موازی نوشت تا اجرای برنامه بر روی این ۴ پردازنده ۲ برابر سریع تر شود .

سوال ۳

با استفاده کردن از حافظه نانو که در Nano Programming مورد استفاده قرار می گیرد که از یک Control Storage Organization دو سطحی استفاده میکند که سطح بالاتر Micromemory از نوع عمودی یا (Vertical) می باشد که آدرس مورد نیاز سطح پایین تر خود یا همان Nano Memory را تامین می کند . این سطح وظیفه تولید کردن سیگنال های کنترلی را دارند و ویژگی آن ها صرفه جویی در حجم ریز برنامه واحد کنترل می باشد . بدون وجود این نوع حافظه اگر بخواهیم به صورت یک سطحی Single Layer جلو برویم حجم ریز برنامه به صورت زیر است :

$$\# \text{ of control memory bits} = \# \text{ of control signals} * \# \text{ of microinstructions} = 50 * 180 = 9000 \text{ bit}$$

ولی اگر با حافظه های نانو جلو برویم با توجه به این که از یک چارت عملیاتی با ۱۱۴ نقطه کنترلی استفاده می کنیم داریم :

$$\text{First layer (Microstore)} = 180 * 8 = 1440 \text{ bit}$$

$$\text{Second layer (NanoMemory)} = 114 * 50 = 5700 \text{ bit}$$

بنابراین در صورت استفاده از حافظه نانو حداقل $9000 - (5700 + 1440) = 1860 \text{ bit}$ در حجم ریز برنامه صرفه جویی می شود .

سوال ۴

سوال ۵

با توجه به این که Full Forwarding داریم بنابراین مراحل به شکل زیر می باشد :

IF Stage	IF/ID stage	ID Stage	ID/EX Stage	EX Stage	EX/Mem Stage	M Stage	M/WB Stage	WB Stage
----------	-------------	----------	-------------	----------	--------------	---------	------------	----------

با توجه به این مراحل و ترتیب دستورات ذکر شده در صورت سوال می توان فهمید که 6\$ که در عملیات تفریق جزو عملوند ها می باشد در مرحله **M/WB** تولید می شود و عملوند دوم این عملیات یعنی 8\$ در مرحله **EX/M** تولید می شود .

سوال ۶

برنامه به صورت زیر می باشد :

۱. sub \$a3,\$a2,\$a1

۲. and \$v1,\$a3,\$v4

۳. or \$v2,\$v5,\$a3

۴. lw \$v4,100(\$a3)

۵. sw \$v2,100(\$a3)

الف

با توجه به شماره گذاری های بالا به مخاطرات زیر می رسم :

۱

بین خطوط ۲ و ۱ که بر سر رجیستر \$a3 با یک دیگر **Data Hazard** دارند .

۲

بین خطوط ۴ و ۱ بر سر رجیستر \$a3 با یک دیگر **Data Hazard** دارند .

۳

بین خطوط ۵ و ۳ بر سر \$v2 با یک دیگر **Structure Hazard** دارند .

۴

خطوط ۵ و ۱ بر سر \$a3 با یک دیگر **Data Hazard** دارند .

۵

بین خطوط ۳ و ۱ بر سر \$a3 با یک دیگر **Data Hazard** دارند .

۶

بین خطوط ۵ و ۴ بر سر \$a3 با یک دیگر **Structure Hazard** دارند .

ب

با توجه به قسمت قبل پردازنده برای دستور دوم باید ۲ کلاک متوقف شود بنابراین برای این که دستورات به درستی اجرا شوند در کل ۲ Stall داریم که به صورت زیر می باشد :

	1	2	3	4	5	6	7	8	9	10	11
1	IF	ID	EX	Mem	WB						
2		IF	stall	stall	ID	EX	Mem	WB			
3					IF	ID	EX	Mem	WB		
4						IF	ID	EX	Mem	WB	
5							IF	ID	EX	Mem	WB

بنابراین اجرای دستورات ۱۱ کلاک طول می کشد . با فرض این که پردازنده ۵ مرحله بوده با این توقف اجرای دستورات ۶ Clock Cycle بیشتر طول می کشد .

سوال ۷

الف

با توجه به این که معماری Single Cycle است بنابراین اجرای هر دستور یک کلاک طول می کشد بنابراین طول کلاک باید به اندازه ای باشد که طولانی ترین دستور نیز در یک کلاک انجام شود بنابراین برای این معماری دوره کلاک به صورت زیر است :

$$\text{Clock Cycle Time} = 400 \text{ ps} + 200 \text{ ps} + 300 \text{ ps} + 500 \text{ ps} + 200 \text{ ps} = 1600 \text{ ps}$$

ب

با توجه به این که معماری Multi Cycle می باشد بنابراین به جای این که تمام مراحل اجرای یک دستور در یک کلاک انجام شود ، هر مرحله اجرا در یک کلاک انجام می شود بنابراین کلاک آن باید طوری باشد که طولانی ترین مرحله نیز به درستی انجام شود بنابراین دوره کلاک به صورت زیر است :

$$\text{Clock Cycle Time} = \text{MAX}(400\text{ps} , 200\text{ps} , 300\text{ps} , 500\text{ps} , 200\text{ps}) = 500 \text{ ps}$$

سوال ۸

الف

با توجه به این که معماری Single Cycle می باشد بنابراین برای تمام دستورات $\text{CPI} = 1$ پس برای انجام شدن تمام دستورات Load داریم :

$$\begin{aligned}\# \text{ of Clock Cycles} &= \text{CPI} * \text{IC} \\ \text{IC} &= 1000000 * 0.25 = 250000 \\ \# \text{ of Clock Cycles} &= 1 * 250000 = 250000\end{aligned}$$

بنابراین برای انجام شدن تمام دستورات Load به ۲۵۰۰۰۰ کلاک نیاز داریم .
با توجه به معماری داریم :

$$\text{Clock Cycle Time} = 400 \text{ ps} + 200 \text{ ps} + 300 \text{ ps} + 500 \text{ ps} + 200 \text{ ps} = 1600 \text{ ps}$$

بنابراین زمان اجرا برای این معماری به صورت زیر است :

$$\text{Execution Time} = \text{IC} * \text{CPI} * \text{Clock Cycle Time} = 250000 * 1 * 1600\text{ps} = 400000000 \text{ ps} = 0.0004 \text{ s}$$

بنابراین برای اجرای تمام دستورات Load در این معماری نیاز به ۰/۰۰۰۴ ثانیه داریم .

ب

با توجه به اسلاید های درس داریم :

$$\begin{aligned}\text{ALU} &= 4 \text{ Clock Cycle} \\ \text{Load} &= 5 \text{ Clock Cycle} \\ \text{Store} &= 4 \text{ Clock Cycle} \\ \text{Branches} &= 3 \text{ Clock Cycle}\end{aligned}$$

ابتدا مقدار CPI را به دست می آوریم :

$$\text{CPI} = (4 * 0.5) + (3 * 0.15) + (5 * 0.25) + (4 * 0.1) = 4.1$$

حال مانند حالت قبل برای تعداد کلاک های لازم برای اجرا شدن تمام دستورات Load داریم :

$$\# \text{ of Clock Cycles} = \text{IC} * \text{CPI} = 250000 * 4.1 = 1025000$$

بنابراین تعداد کلاک های لازم برابر است با ۱۰۲۵۰۰۰
با توجه به معماری داریم :

$$\text{Clock Cycle Time} = \text{MAX}(400\text{ps} , 200\text{ps} , 300\text{ps} , 500\text{ps} , 200\text{ps}) = 500 \text{ ps}$$

و مانند حالت قبل برای زمان اجرا داریم :

$$\text{Execution Time} = \text{IC} * \text{CPI} * \text{Clock Cycle Time} = 250000 * 4.1 * 500\text{ps} = 512500000 \text{ ps} = 0.0005125$$

بنابراین اجرای تمام دستورات Load در این معماری ۰/۰۰۰۵۱۲۵ ثانیه طول می کشد .

سوال ۹

```

1    lw    $t2, 12($s0)
2    la    $t0, player_x
3    lw    $t1, 0($t0)
4    addi  $t1, $t1, 16
5    blt   $t1, $t2, no_collide
6    li    $v0, 1
7 no_collide:
8    jr    $ra

```

الف

با توجه به بالا داریم :

۱

خطوط ۱ و ۴ با یک دیگر **Structure Hazard** دارند . به این دلیل که در زمانی که دستور lw می خواهد مقداری را بخواند دستور addi می خواهد Fetch بشود که چون یک حافظه داریم این موضوع باعث به وجود آمدن Structure Hazard می شود .

۲

خطوط ۳ و ۶ با یک دیگر **Structure Hazard** دارند . مانند قبل در زمانی که دستور lw می خواهد از حافظه بخواند در همان زمان دستور li می خواهد Fetch شود که این به دلیل داشتن یک حافظه غیر ممکن است و باعث به وجود آمدن Structure Hazard می شود.

۳

خطوط ۲ و ۳ بر سر \$t0 با یک دیگر **Data Hazard** دارند .

۴

خطوط ۳ و ۴ بر سر \$t1 با یک دیگر **Data Hazard** دارند.

۵

خطوط ۴ و ۵ بر سر \$t1 با یک دیگر **Data Hazard** دارند.

۶

خطوط ۱ و ۵ بر سر \$t2 با یک دیگر **Data Hazard** دارند . این Hazrd ممکن است به خاطر stall های موجود به وجود نیاید . ولی احتمال به وجود آمدنش صفر نیست .

۷

بعد از دستور blt یک **Control Hazard** به وجود می آید .

ب

در این حالت فقط باید برای مرحله MEM متوقف شویم که نمودار آن به صورت زیر می باشد :

	1	2	3	4	5	6	7	8	9	10	11	12
1 lw \$t2, 12(\$s0)	IF	ID	EX	MEM	WB							
2 la \$t0, player_x		IF	ID	EX	MEM	WB						
3 lw \$t1, 0(\$t0)			IF	ID	EX	MEM	WB					
4 addi \$t1, \$t1, 16				ID	ID	STALL	EX	MEM	WB			
5 blt \$t1, \$t2, no_collide					IF	STALL	ID	EX	MEM	WB		
6 li \$v0, 1							IF	ID	EX	MEM	WB	
jr \$ra								IF	ID	EX	MEM	WB

بنابراین همان طور که می توان در نمودار دید در این حالت اجرای برنامه ۱۲ سیکل طول می کشد . Forward ها با فلش مشخص شده اند .