پاسخ سوال ۱)

برای بدست آوردن مقدار سیگنالهای کنترلی کافی است بدانیم در سیکل‌های مورد نظر، چه دستوری در چه طبقه‌ای از خط لوله قرار گرفته است. سپس با داشتن دستور می‌توان مقدار سیگنالهای کنترلی را بدست آورد.

هنگامیکه دستورها زیاد باشد (مثل این سوال)، ترسیم نمودار چند سیکل ساعتی خط لوله برای محاسبه دستورات درون خط لوله برای یک سیکل خاص، بسیار زمان‌بر است. راه دیگر محاسبه است. واضح است که اگر پرش، انشعاب، استثنا یا تعلیق رخ ندهد، دستورها پشت سرهم اجرا خواهند شد و در هر سیکل یک دستور وارد خط لوله می‌شود و در عوض یک دستور از خط لوله خارج می‌گردد. در سیکل اول دستور lw $1,0($0) در طبقه Fetch از خط لوله قرار دارد، در سیکل دوم همین دستور در طبقه Decode و دستور lw $2,4($0) در طبقه Fetch قرار می‌گیرد. اگر هیچ مشکلی پیش نیاید، در سیکل هشتم، دستور هشتم در طبقه Fetch قرار دارد، پس دستور هفتم در طبقه Decode است و ... . یعنی در سیکل ۸، نمودار تک سیکل ساعتی خط لوله به صورت زیر است:

lui $10,0x7FFF	sw $1,12($0)	addi $1,$1,15	add $1,$1,$3	add $1,$1,$2
Fetch	Decode	Execute	Memory	WriteBack

قبل از آنکه مقدار سیگنالهای کنترلی را برای این سیکل بدست آوریم، ابتدا ثابت می‌کنیم فرض ما مبنی بر عدم مشکل درست است.

- **پرش و انشعاب:** در کل این کد هیچ دستور پرش یا انشعاب وجود ندارد
- **استثنا:** اولاً همه دستورها، دستورهای شناخته شده پردازنده MIPS هستند، پس هیچگاه استثنا برای «دستورالعمل تعریف نشده» نداریم. مقدار رجیسترهای $1 و $2 و $3 نیز به اندازه‌ای بزرگ نیستند که حاصل جمع آنها باعث سرریز شود. پس استثنا برای سرریز نیز اتفاق نمی‌افتد.
- **تعلیق:** چون دستور پرش و انشعاب نداریم، مخاطره کنترلی نیز نداریم و در نتیجه تعلیق به دلیل دستورهای پرش و انشعاب نخواهیم داشت. همچنین پردازنده مجهز به پیش‌فرستادن می‌باشد. پس تمامی مخاطرات داده‌ای به جز خواندن رجیستری که می‌خواهیم از حافظه مقداری را درون آن بریزیم، تعلیقی را ایجاد نمی‌کنند. اگر دقت کنید بین بارگذاری $1 و استفاده از آن دو دستور، و بین بارگذاری هر یک از رجیسترهای $2 و $3 و استفاده از آنها یک دستور فاصله وجود دارد. هنگامیکه پیش‌فرستادن داشته باشیم، وجود فاصله یک دستوری برای رفع تعلیق این دستور کافی است. پس هیچ تعلیقی نخواهیم داشت.

مقدار سینگالها در سیکل ۸ به صورت زیر است:

سیگنال	مقدار	دلیل
ForwardA	10	دستورهای طبقه MEM و WB هر دو باید در رجیستر $1 مقدار بنویسند، دستور addi که در طبقه EXE است، می‌خواهد مقدار $1 را بخواند. مقدار موجود در طبقه MEM مقدار به‌روزتری است. پس باید این مقدار به طبقه EXE فرستاده شود. پس باید Forward A برابر با 10 باشد. (برای توضیح بیشتر به «شرایط پیش‌فرستادن» در کتاب یا اسلایدهای استاد مراجعه کنید.)
ForwardB	00	عملوند دوم دستور addi یک بلافصل می‌باشد. لذا نیاز به پیش‌فرستادن نیست.

		توجه کنید که در شکل تفاوتی بین بلافصل و مقدار رجیستر rt قائل نشده است.
EX/MEM.RegisterRd	0x1	مقدار این سیگنال (که خود یک رجیستر خط لوله است)، آدرس رجیستر مقصد دستور موجود در طبقه MEM را مشخص می‌کند. این دستور add $1,$1,$3 می‌باشد پس مقدار این سیگنال 0x1 می‌باشد.
MEM/WB.RegisterRd	0x1	همانند سیگنال بالایی است با این تفاوت که برای طبقه WB است. در این سیکل دستور add $1,$1,$3 در این طبقه وجود دارد. پس مقدار این سیگنال 0x1 می‌باشد.
IF/ID.Regwrite	0	مشخص می‌کند که دستور فعلی طبقه ID نیاز به نوشتن در رجیستر فایل دارد یا نه. چون دستور sw است، نیاز به نوشتن در رجیستر فایل ندارد، پس مقدار سیگنال یک است.
ID/EX.Regwrite	1	همانند سیگنال بالاست با این تفاوت که برای طبقه EXE می‌باشد. دستور addi است پس نوشتن در رجیستر فایل دارد.
EX/MEM.Regwrite	1	مشابه بالا
MEM/WB.Regwrite	1	مشابه بالا
PC	0x0000301C	مقدار PC آدرس دستوری را مشخص می‌کند که در مرحله Fetch قرار دارد. یعنی آدرس دستور lui .
PcWrite	1	هنگامی این سیگنال صفر است که نیاز به تعلیق داشته باشیم. چون تعلیق نداریم پس مقدار آن باید یک باشد.
ID.Flush	0	هر گاه لازم به خالی کردن خط لوله باشد، مقدار این سیگنال یک می‌شود. خط لوله در دو صورت خالی می‌شود: انشعاب و استثنا. در اینجا هیچکدام رخ نداده است، پس مقدار سیگنال صفر است.
IF.Flush	0	مشابه بالا
EX.Flush	0	مشابه بالا
EPC	بدون تغییر	چون استثنا رخ نداده است، پس مقدار این رجیستر بدون تغییر می‌ماند.

در سیکل ۹ نیز شرایط همانند سیکل ۸ است. نه تعلیق داریم، نه استثنا، نه پرش و نه انشعاب. نمودار تک سیکل ساعتی خط لوله برای سیکل ۹ به صورت زیر است:

ori $10,$10,0xFFFF	lui $10,0x7FFF	sw $1,12($0)	addi $1,$1,15	add $1,$1,$3
Fetch	Decode	Execute	Memory	WriteBack

با توجه به نمودار بالا می‌توان مقدار سیگنالها را برای سیکل ۹ به صورت زیر حساب کرد:

سیگنال	مقدار	دلیل
ForwardA	00	دستور sw در طبقه EXE قرار دارد. برای محاسبه آدرس حافظه لازم است مقدار رجیستر $0 در ورودی اول ALU قرار گیرد یعنی نیاز به پیش‌فرستادن نداریم.
ForwardB	00	برای محاسبه آدرس حافظه به بلافصل نیز نیاز داریم.
EX/MEM.RegisterRd	0x1	دستور addi در طبقه MEM قرار گرفته است. این دستور I-Type می‌باشد و فیلد Rd برای آن بی معنی است اما توجه داشته باشید که رجیستر

		EX/MEM.RegisterRD همواره مقدار رجیستر مقصد دستوری که در مرحله MEM قرار دارد را نگه‌داری می‌کند. یعنی بسته به دستور مقدار آن با فیلد Rd یا Rt برابر است.
MEM/WB.RegisterRd	0x1	مشابه سیکل ۸
IF/ID.Regwrite	1	lui نیاز به نوشتن در رجیستر فایل دارد
ID/EX.Regwrite	0	sw نبار به نوشتن در رجیستر فایل ندارد
EX/MEM.Regwrite	1	addi نیاز به نوشتن در رجیستر فایل دارد
MEM/WB.Regwrite	1	add نیاز به نوشتن در رجیستر فایل دارد
PC	0x00003020	مشابه سیکل ۸
PcWrite	1	مشابه سیکل ۸
ID.Flush	0	مشابه سیکل ۸
IF.Flush	0	مشابه سیکل ۸
EX.Flush	0	مشابه سیکل ۸
EPC	بدون تغییر	مشابه سیکل ۸

باز هم تا سیکل ۱۲ هیچ رخداد خاصی وجود ندارد. در سیکل ۱۲ نمودار تک سیکل ساعتی خط لوله به صورت زیر است:

`add $1,$2,$3`	`sw $1,12($0)`	`add $1,$10,$10`	`ori $10,$10,0xFFFF`	`lui $10,0x7FFF`
Fetch	Decode	Execute	Memory	WriteBack

در این سیکل نیز از پرش، انشعاب و تعلیق خبری نیست. اما دستور در مرحله EXE دستور add می‌باشد که ممکن است باعث بروز استثنا شود اینکه دو دستور lui و ori باعث بارگذاری مقدار 0x7FFFFFFF در $10 می‌شوند و add رجیستر $10 را با خودش جمع می‌کند. مقدار 0x7FFFFFFF بزرگترین عدد مثبت ۳۲ بیتی است، پس حاصل جمع با خودش حتماً سرریز خواهد کرد. پس در این سیکل استثنا رخ می‌دهد.

هنگامیکه در سیکل ۱۲ هستیم، واحد کنترل وجود سرریز را تشخیص می‌دهد اما در سیکل ۱۲ خط لوله خالی نمی‌شود بلکه در سیکل ۱۳ خالی می‌شود. لذا مقدار سیگنالها در سیگنال ۱۲ به صورت زیر است:

سیگنال	مقدار	دلیل
ForwardA	10	$10 باید به عنوان ورودی اول ALU قرار گیرد. دستورهای مرحله MEM و WB هر دو در این رجیستر می‌نویسند اما مقدار صحیح در مرحله MEM قرار دارد.
ForwardB	01	بازهم $10 باید به عنوان ورودی دوم ALU قرار گیرد. پس باید از مرحله MEM مقدار صحیح را بگیریم. لذا مقدار 01 مناسب است. (طبق شکل صورت سوال این مقدار انتخاب شد. متاسفانه کتاب رویکرد ثابتی در ارائه Data Path ندارد و بسته به شکل، مقدار سیگنالهای کنترلی ممکن است تغییر کند)
EX/MEM.RegisterRd	0xA	دستور ori باید در رجیستر $10 بنویسد.
MEM/WB.RegisterRd	0xA	دستور lui باید در رجیستر $10 بنویسد (با وجود اینکه عدد ۱۰ در فیلد Rt این دستور قرار دارد)

IF/ID.Regwrite	0	sw نیاز به نوشتن در رجیستر فایل ندارد
ID/EX.Regwrite	1	add نبار به نوشتن در رجیستر فایل دارد
EX/MEM.Regwrite	1	ori نیاز به نوشتن در رجیستر فایل دارد
MEM/WB.Regwrite	1	lui نیاز به نوشتن در رجیستر فایل دارد
PC	0x0000302C	مشابه سیکل ۸
PcWrite	1	توجه داشته باشید که استثنا رخ داده اما استثنا نیاز به تعلیق ندارد و باید بلافاصله به روتین مدیریت استثنا پرش کند.
ID.Flush	1	استثنا رخ داده پس باید خط لوله خالی شود.
IF.Flush	1	استثنا رخ داده پس باید خط لوله خالی شود.
EX.Flush	1	استثنا رخ داده پس باید خط لوله خالی شود.
EPC	بدون تغییر	آدرس دستور بعد از دستوری که موجب استثنا شده است در پشت رجیستر EPC قرار دارد اما تا لبه بالارونده کلاک اتفاق نیفتد، این مقدار درون EPC بارگذاری نمی‌شود.

اکنون لبه بالارونده کلاک اتفاق می‌افتد و وارد سیکل ۱۳ می‌شویم. پردازنده به آدرس روتین مدیریت استثنا برای سرریز یعنی 0x80000180 پرش می‌کند و خط لوله خالی می‌شود. توجه کنید که طبق مثال کتاب، مرحله WB خالی نمی‌شود!!!

??	nop	nop	nop	ori $10,$10,0xFFFF
Fetch	Decode	Execute	Memory	WriteBack

به جای ?? اولین دستور روتین مدیریت استثنا یعنی دستور موجود در آدرس 0x80000180 قرار می‌گیرد:

سیگنال	مقدار	دلیل
ForwardA	00	برای nop نیاز به Forwarding نداریم
ForwardB	00	برای nop نیاز به Forwarding نداریم
EX/MEM.RegisterRd	0x0	دستور nop در رجیستر صفر می‌نویسد.
MEM/WB.RegisterRd	0xA	دستور ori در رجیستر ۱۰ می‌نویسد
IF/ID.Regwrite	0	nop نیاز به نوشتن ندارد.
ID/EX.Regwrite	0	nop نیاز به نوشتن ندارد.
EX/MEM.Regwrite	0	nop نیاز به نوشتن ندارد.
MEM/WB.Regwrite	1	ori نیاز به نوشتن دارد.
PC	0x80000180	آدرس دستور فعلی
PcWrite	1	nop موجب هیچ عملی نمی‌شود پس تعلیق ندارد.
ID.Flush	0	استثنا رخ داده و خط لوله خالی شده. دیگر نیاز به خالی کردن خط لوله نیست.
IF.Flush	0	مشابه بالا
EX.Flush	0	مشابه بالا
EPC	0x00003028	دقت کنید که دستور add که در آدرس 0x00003024 قرار دارد موجب استثنا شده است اما همواره آدرس دستور بعدی در EPC قرار می‌گیرد و وظیفه کم کردن عدد ۴ بر عهده روتین مدیریت استثنا است.

پاسخ سوال ۲)

در حالت غیرخط لوله پردازنده می‌تواند تک سیکلی (Single-Cycle) یا چند سیکلی (Multi-Cycle) باشد. برای هنگامیکه پردازنده چند سیکلی باشد به اطلاعات بیشتری از جمله واحدهای مورد نیاز در اجرای هر دستور نیاز داریم. پس با توجه به اطلاعات صورت سوال، مقصود حالت تک سیکلی است. در این حالت تمام قسمت‌های پشت سر هم قرار گرفته و طول سیکل برابر تا مجموع تاخیر آنها خواهد بود:

$$Single - Cycle\ CPU\ Cycle = ۳۶ + ۳۹ + ۲۳ + ۲۸ + ۶۴ = ۱۹۰$$

اگر تعداد دستورها را n فرض کنیم. زمان اجرای تک سیکلی برابر است با:

$$Single - Cycle\ CPU\ Time = IC.CPI.Period = n \times ۱ \times ۱۹۰$$

دقت کنید که در پردازنده‌های تک سیکلی، *CPI* برای تمامی دستورها برابر یک است. چون طبق تعریف، معماری تک سیکل معماری است که تمامی دستورها در یک سیکل اجرا شوند. پس *CPI* برای همه دستورها یک است.

در حالت خط لوله، طول سیکل برابر است با بزرگترین تاخیر بین همه قسمت‌ها بعلاوه تاخیر ثبات‌های خط لوله. پس:

$$Pipeline\ CPU\ Cycle = ۶۴ + ۱ = ۶۵$$

تعداد دستورها n و تعداد طبقات خط لوله ۵ است. لذا:

$$Pipeline\ CPU\ Time = \left(n + ۵ - ۱\right) \times ۶۵$$

در نتیجه تسریع برابر است با:

$$\text{تسریع} = \frac{۱۹۰n}{۶۵n + ۴ \times ۶۵}$$

حداکثر تسریع هنگامی اتفاق می‌افتد که تعداد دستورها بسیار بزرگ باشد $(n \to \infty)$ و تعلیق ایجاد نشود (یعنی هیچ‌گاه در خط لوله، حباب نداشته باشیم. پس حدکثر تسریع برابر است با:

$$\text{حداکثر تسریع} = \lim_{n\to\infty} \text{تسریع} = \frac{۱۹۰}{۶۵} = ۲/۹۲$$

پس گزینه «الف» صحیح است.

پاسخ سوال ۳)

راه اول:

طبق صورت سوال رویکرد اتخاذ شده برای انشعاب رویکرد تعلیق است. هرجا رویکرد تعلیق اختیار شود به این معنی است که می‌توان به تعداد کافی nop درج کرد تا دیگر لازم نباشد تعلیق به صورت سخت‌افزاری انجام شود (در این صورت تعداد سیکل‌های لازم برای اجرای هر یک از دستورها، تعداد طبقات خط لوله می‌باشد). پس از هر ۱۹ دستور معمولی، یک انشعاب وجود دارد. کافی است پس از هر انشعاب به جز دستور انشعاب آخر، ۷ دستور nop قرار دهیم. پس تعداد nopهای اضافه شده برابر است با:

$$\text{تعداد } nop \text{ اضافه شده} = ۷ \times \left(\frac{۱۰۰}{۲۰} - ۱\right) = ۲۸$$

عدد $\frac{۱۰۰}{۲۰}$ تعداد دستورهای انشعاب است. پس تعداد کل دستورها پس از افزودن nop برابر است با:

$$\text{تعداد دستور} = ۱۰۰ + ۲۸ = ۱۲۸ \xrightarrow{\text{زمان سیکل یک} = ۱۰ns} \text{اجرا زمان} = \left(۱۲۸ + ۸ - ۱\right) \times ۱۰ = ۱۳۵۰\boldsymbol{ns}$$

راه دوم:

اگر رویکرد اتخاذ شده برای یک دستور خاص، تعلیق باشد، آن دستور مرزی است که دستورهای قبل و بعد آنرا از نظر زمان اجرایی از هم مستقل می‌کند. یعنی در این سوال هر دستور انشعاب باعث می‌شود که دستورهای قبل و بعد آن از نظر زمان اجرایی همانند دو برنامه مجزا عمل کنند. پس زمان اجرای برنامه برابر است با تعداد دستورهای انشعاب ضرب‌در زمان اجرای بخش‌های مستقل. تعداد دستورهای انشعاب $\frac{۱۰۰}{۲۰}$ است و زمان اجرای هر ۲۰ دستور $\left(۲۰ + ۸ - ۱\right) \times ۱۰ = ۲۷۰ns$ است. پس زمان اجرا برابر است با:

$$\text{زمان اجرا} = ۲۷۰ \times ۵ = ۱۳۵۰\boldsymbol{ns}$$

پس گزینه «ج» صحیح است.

پاسخ سوال ۴)

حداکثر تسریع در صورت عدم وجود تعلیق رخ می‌دهد برابر است با تعداد طبقات خط لوله. در صورت سوال ذکر شده که مشکلات وابستگی داده و دسترسی حافظه نداریم. معنی آن عدم تعلیق است. لذا حداکثر تعلیق ۱۱ است.

حداقل هنگامی حاصل می‌شود که از تعلیق برای پرش‌ها استفاده می‌کنیم. یعنی در صورت برخورد با پرش تا خروج آن از خط لوله، حباب وارد خط لوله کنیم. این بدان معنی است که دستور پرش تکه‌هایی در برنامه ایجاد می‌کند که از نظر زمان اجرا مستقل هستند.

فرض می‌کنیم k طبقه خط لوله داشته باشیم و n تعداد کل دستورها و m تعداد دستورهای پرش باشد. همچنین فرض می‌کنیم پرش اول بعد از $(n_۱ - ۱)$ دستور قرار گرفته باشد. پرش دوم بعد از $(n_۲ - ۱)$ دستور و ... . تعداد سیکل کل برابر است با مجموع سیکل‌های لازم برای هر مجموعه مستقل:

$$\text{تعداد سیکل کل خط لوله} = (n_۱ + k - ۱) + (n_۲ + k - ۱) + \cdots + (n_m + k - ۱)$$

$$= \sum_{i=۱}^{m} (n_i + k - ۱) = n + m(k - ۱)$$

پس زمان اجرا برای پریود T برابر است با:

$$\text{زمان اجرا خط لوله} = (n + m(k - ۱))T$$

در این سوال ده درصد دستورها پرش هستند . پس $m = ۰/۱n$ با جایگذاریم داریم:

$$\text{زمان اجرا خط لوله} = (n + ۰/۱n(۱۱ - ۱))T = ۲nT$$

زمان اجرای تک سیکل نیز برابر است با

$$\text{زمان اجرا تک سیکل} = ۱۱nT$$

$$\text{حداقل تسریع} = \frac{۱۱nT}{۲nT} = ۵/۵$$

پس گزینه «ب» صحیح است.

پاسخ ۵)

در صورت سوال مشخص نشده که رویکرد اتخاذ شده برای دستورهای پرش چیست. این سوال را برای رویکرد تعلیق و فرض عدم پرش حل می‌کنیم. حل برای رویکردهای دیگر مخاطرات کنترلی نیاز به اطلاعات بیشتری دارد که در سوال موجود نیست.

رویکرد (همیشه) تعلیق:

در این رویکرد چه پرش انجام شود چه نشود، حباب وارد خط لوله می‌شود. لذا فرقی ندارد که پرش انجام شود یا نه. زمان اجرای بدون خط لوله برابر است با

$$\text{زمان اجرا بدون خط لوله} = \left(۲۰ + ۸ + ۲۰ + ۱۲\right) \times n = ۶۰n$$

طبق سوال قبل زمان اجرای خط لوله از رابطه $\left(n + m\left(k - ۱\right)\right)T$ بدست می‌آید که در آن $k = ۴$ و $m = ۰/۱\text{n}$ است. پس

$$\text{زمان اجرا خط لوله} = \left(n + ۰/۱n\left(۴ - ۱\right)\right) \times ۲۰ = ۲۶n$$

پس تسریع برابر است با:

$$\text{تسریع} = \frac{۶۰\text{n}}{۲۶n} = ۲/۳$$

پس گزینه «الف» صحیح است.

رویکرد فرض عدم پرش (تعلیق در صورت پرش)

زمان اجرای بدون خط لوله همانند رویکرد قبل برابر با $۶۰n$ است. فرمول زمان اجرای خط لوله همانند رویکرد قبل است با این تفاوت که چون در نیمی از موارد پرش اتفاق می‌افتد، تعداد دستورهای پرش که انجام می‌شوند برابر با $m = \frac{۰/۱}{۲}n = ۰/۰۵n$ است. لذا:

$$\text{زمان اجرا خط لوله} = \left(n + ۰/۰۵n\left(۴ - ۱\right)\right) \times ۲۰ = ۲۳n$$

پس تسریع برابر است با:

$$\text{تسریع} = \frac{۶۰n}{۲۳n} = ۲/۶$$

پس گزینه «ج» صحیح است.

پاسخ سوال ۶)

مشخص نیست که مقصود سوال از پیاده‌سازی معمولی تک سیکلی است یا چند سیکلی. این سوال را برای هر دو حالت حل می‌کنیم.

تک سیکلی:

$$\text{زمان اجرا بدون خط لوله} = \left(۱۰ + ۷ + ۱۰ + ۱۲ + ۷\right) \times n = ۴۶n$$

$$\text{زمان اجرا خط لوله} = \left(n + ۵ - ۱\right) \times ۱۲ = ۱۲n + ۴۸$$

$$\text{تسریع} = \frac{۴۶n}{۱۲n + ۴۸} \quad \overset{n \to \infty}{=} \quad ۳/۸$$

که در هیچ‌یک از این گزینه‌ها نیست.

چند سیکلی:

در این حالت زمان اجرای دستورات متفاوت است. زمان اجرای بدون خط لوله برابر است با:

$$\text{زمان اجرا بدون خط لوله} = ۴۶ \times ۰/۲n + ۳۴ \times ۰/۴n + ۳۹ \times ۰/۲n + ۲۷ \times ۰/۲n = ۳۶n$$

چون خط لوله با تعداد طبقات متفاوت داریم، زمان اجرای نوع‌های مختلف دستور با هم متفاوت است (در MIPS خط لوله تک با تعداد طبقات ثابت داشتیم)

گروه	درصد وقوع (درصد)	زمان اجرا
الف	۲۰	$۱۲\left(۵ + ۰/۲n - ۱\right)$
ب	۴۰	$۱۲\left(۴ + ۰/۴n - ۱\right)$
ج	۲۰	$۱۲\left(۴ + ۰/۲n - ۱\right)$
د	۲۰	$۱۲\left(۳ + ۰/۳n - ۱\right)$
زمان اجرای کل:		$۱۹۲ + \left(n - ۴\right) \times ۱۲$

در نتیجه تسریع برابر است با:

$$\text{تسریع} = \frac{۳۶n}{۱۹۲ + \left(n - ۴\right) \times ۱۲} \quad \overset{n \to \infty}{=} \quad ۳$$

پس گزینه «د» صحیح است.

موفق باشید

گروه حل تمرین