



۱. (آ) از آنجایی که در کلاک ششم به مقدار R1 و در کلاک یازدهم به مقدار R4 نیاز داریم پس نتایج وضعیت E3 به وضعیت E1 forward می شود. همچنین چون به مقدار R1 در کلاک هفتم نیاز داریم، نتایج وضعیت M به وضعیت E1 forward می شود. همچنین نتیجه وضعیت E3 به condition register ها forward می شود. (به دلیل SUBI و JNZ در کلاک پانزدهم)

(ب) این سیستم از Hardware interlocking بهره می برد چون وابستگی های داده را با استفاده از stall در خط-لوله رفع می کند و از NOP در سطح کد استفاده نمی کند.

(ج) دستور MUL R4, R1, R1 دفعه اول در کلاک سوم واکنشی می شود. بعد از حلقه اول، این دستور در کلاک دوازدهم دوباره واکنشی می شود.

دستور JNZ L1 در مرحله Decode stall می خورد و دستور MUL R4, R1, R1 را به تاخیر می اندازد پس می توان نتیجه گرفت که میان دو دفعه ای که این دستور واکنشی می شود، ۱۰ چرخه فاصله است. از آنجایی که  $R1 = 1024$  پس حلقه باید ۸ بار اجرا شود.

$$T = 12 + 7 \times 10 = 82$$

(د) از آنجایی که حلقه باید ۸ دفعه اجرا شود و هر حلقه شامل ۶ دستور است و ما دو دستور قبل از حلقه داریم، با در نظر گرفتن دستوری که در زمان T نیز واکنشی می شود:

$$N = 2 + 8 \times 6 + 1 = 51$$

(ه) حلقه باید ۹ دفعه تکرار شود و در نتیجه تعداد دستورات به صورت زیر محاسبه می شود:

$$2 + 6 \times 9 + 1 = 57$$

و تعداد stall ها به صورت زیر محاسبه می شود:

$$1 + 4 \times 9 = 37$$

پس تعداد کل چرخه ها برابر با

$$57 + 37 + 6 = 100$$

می شود.

۲.

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

if m approaches infinity, then:

$$throughput = \frac{1}{T_p} = \frac{1}{\frac{20}{n} + \frac{n}{20}}$$

با مشتق‌گیری و • گذاشتن می‌یابیم بهترین n برابر 20 است و بهترین گذردهی برابر است با:

$$5 \times 10^8 s^{-1}$$

۳. خطوط لوله به صورت زیر می‌باشند.

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
				W	M	X	W	M	X	D	F	sub \$2, \$3, \$1
	W	M	X	D	d*	d*	D	d*	d*	F		lw \$5, 0(\$2)
W	M	X	D	F			F					addi \$4, \$5, 1
												add \$5, \$3, \$1

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
						W	W	M	X	D	F	sub \$2, \$3, \$1
				W	M	X	M	X	D	F		lw \$5, 0(\$2)
		W	M	M	X	D	D	d*	F			addi \$4, \$5, 1
							F					add \$5, \$3, \$1

$$speed - up = \frac{12 - 9}{9} \times 100 = 33\%$$

۴. خطوط لوله به صورت زیر است:

(آ) بخش الف

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
	WB	WB MEM	MEM EX	EX ID	WB ID IF	WB MEM IF	WB MEM EX	MEM EX ID	EX ID IF	ID IF	IF	lw \$1, 40(\$6) beq \$2, \$3, Label2 (T) beq \$1, \$2, Label1 (NT) sw \$2, 20(\$4) and \$1, \$1, \$4

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
WB	WB MEM	MEM EX	EX ID	WB ID IF	WB MEM IF	WB MEM EX	WB MEM EX ID	MEM EX ID IF	EX ID IF	ID IF	IF	add \$1, \$5, \$3 sw \$1, 0(\$2) add \$2, \$2, \$3 beq \$2, \$4, Label1 (NT) add \$5, \$5, \$1 sw \$1, 0(\$2)

(ب) بخش ب

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
WB	MEM	WB EX	WB MEM ID	MEM EX IF	WB EX ID	WB MEM *** ***	WB MEM EX ID IF	MEM EX ID IF	EX ID IF	ID IF	IF	lw \$1, 40(\$6) beq \$2, \$3, Label2 (T) add \$1, \$6, \$4 beq \$1, \$2, Label1 (NT) sw \$2, 20(\$4) and \$1, \$1, \$4

۲	۱	۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	Instructions
	WB	MEM	WB EX	WB MEM ID	WB MEM EX IF	WB MEM EX ID	WB MEM EX ID IF	MEM EX ID IF	EX ID IF	ID IF	IF	add \$1, \$5, \$3 sw \$1, 0(\$2) add \$2, \$2, \$3 beq \$2, \$4, Label1 (NT) add \$5, \$5, \$1 sw \$1, 0(\$2)

۵. (آ) با داشتن ۱۰۰ حرف کوچک، هر یک از هشت دستور موجود در حلقه، ۱۰۰ بار اجرا خواهند شد. همچنین سه دستور دیگر برای پردازش null پایانی اجرا می‌شوند. یعنی در مجموع ۸۰۳ دستور اجرا می‌شود.
- (ب) یک پردازنده‌ی تک‌چرخه‌ای، هر دستور را در یک چرخه اجرا می‌کند، بنابراین مقدار CPI برابر با ۱ است. همچنین در مجموع ۸۰۳ دستور اجرا می‌شود، بنابراین با جایگذاری این مقادیر در معادله‌ی زمان اجرای CPU، داریم:

$$t_{\text{exec}} = CPI \times \# \text{ of instructions} \times \text{clock cycle time} = 1 \times 803 \times 8 = 6424ns$$

- (ج) در صورتی که هیچ توقف یا پاک‌سازی رخ ندهد، ۴ چرخه برای پر کردن خط لوله و ۸۰۳ چرخه دیگر برای تکمیل عملکرد است، که در مجموع می‌شود ۸۰۷ چرخه.

کدی که در سوال است، شامل ۳۰۱ دستور انشعاب هست (سه دستور برای هر یک از ۱۰۰ تکرار اول حلقه، و یکی برای انتهای رشته). اگر ۱۰٪ از این انشعاب‌ها اشتباه پیش‌بینی شوند و نیاز به پاک‌سازی یک دستور داشته باشند، در مجموع ۳۰ چرخه تلف شده برای پاک‌سازی خواهیم داشت.

همچنین یک خطر بین دستور lb و beq بلافاصله پس از آن وجود دارد. حتی اگر فرض کنیم ارسال مستقیم در هر جا که ممکن است انجام شود و ثبات‌ها در همان چرخه قابل خواندن و نوشتن باشند، باز هم باید یک توقف دو چرخه‌ای بین این دو دستور وارد کنیم، چون نتیجه‌ی انشعاب در مرحله‌ی ID تعیین می‌شود. ترتیب دستورهای lb و beq در مجموع ۱۰۱ بار اجرا می‌شود، بنابراین در مجموع به ۲۰۲ چرخه توقف نیاز داریم. در نتیجه تعداد کل چرخه‌ها برابر است با:  $807 + 30 + 202 = 1039$

- (د) با در نظر گرفتن ۲ نانو ثانیه برای هر چرخه، ماشین خط‌لوله‌ای به  $1039 \times 2 = 2078ns$  زمان نیاز خواهد داشت. این بهبود عملکرد به اندازه  $3 \approx \frac{6424}{2078}$  برابر است.

۶. (آ) ساختار این پیشبینی کننده به این صورت است که یک شمارنده ۲ بیتی نگه داشته می شود. مقادیر این شمارنده مطابق جدول زیر هستند. با هر بار مشاهده یک برنچ که پرش آن انجام می شود، مقدار این شمارنده افزایش می یابد (مگر اینکه مقدار آن ۳ باشد) و در غیر این صورت مقدار یکی کاهش می یابد (مگر اینکه مقدار آن ۰ باشد). هرکجا نیز نیاز به پیشبینی وجود داشته باشد، اگر مقدار این شمارنده ۰ یا ۱ باشد، پیشبینی می کنیم پرش نخواهیم داشت و اگر ۲ یا ۳ باشد، پیشبینی می کنیم پرش خواهیم داشت.

پیش بینی	مقدار دودویی	حالت
عدم پرش	۰۰	Taken Not Strongly
عدم پرش	۰۱	Taken Not Weakly
پرش	۱۰	Taken Weakly
پرش	۱۱	Taken Strongly

بخش های بعدی این سوال بیشتر جنبه تحقیقاتی برای علاقه مندان داشته و در امتحان از آن ها سوالی نخواهد آمد. به همین دلیل این بخش ها پاسخنامه ندارند.