

معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دکتر اسدی
بهار ۱۴۰۳

مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶



تمرین ۶

سوال ۱

۱. جدول زیر زمان اجرای مراحل مختلف انواع دستورات روی یک پردازنده MIPS را به نانوثانیه نشان می دهد:

Type	I Cache	Decode	ALU	PC Update	D Cache	R Write
R-Type	0.9	0.8	0.8	-	-	0.9
Load	0.9	0.8	0.8	-	1	0.9
Store	0.9	0.8	0.8	-	1	-
Branch	0.9	0.8	0.8	0.2	-	-

فرض کنید setup time برابر 0.05ns و hold time برابر 0.05ns است.

آ) فرض کنید این پردازنده single cycle است. با این فرض throughput اجرای دستورات را محاسبه کنید.

ب) فرض کنید این پردازنده multi cycle است و خطهای قرمز جدول بالا، لبه های clock را نشان می دهند. -throughput اجرای هر کدام از انواع دستورات را محاسبه کنید.

آ) در طراحی سینگل سایکل، دوره ی تناوب کلاک با اندازه ی زمان طولانی ترین دستور + setup time + hold time است. پس در این پردازنده، دوره تناوب کلاک برابر با ۴/۵ns است. پس throughput اجرای دستورات این پردازنده $\frac{1}{4/5}$ دستور در هر نانوثانیه است.

ب) برای هر دستور تعداد کلاک را حساب کرده و throughput را با فرمول زیر حساب می کنیم

$$\text{throughput} = \frac{1}{\text{clock-rate} \times \text{number-of-clock}}$$

در طراحی مولتی سایکل، دوره ی تناوب کلاک با اندازه ی زمان طولانی ترین بازه + setup time + hold time است که برای این پردازنده برابر است با: ۱/۱ns

R-type این دستور به ۴ کلاک نیاز دارد پس $\text{throughput} = \frac{1}{1/1 \times 4}$ دستور در نانوثانیه

Load این دستور به ۵ کلاک نیاز دارد پس $\text{throughput} = \frac{1}{1/1 \times 5}$ دستور در نانوثانیه

Store این دستور به ۴ کلاک نیاز دارد پس $\text{throughput} = \frac{1}{1/1 \times 4}$ دستور در نانوثانیه

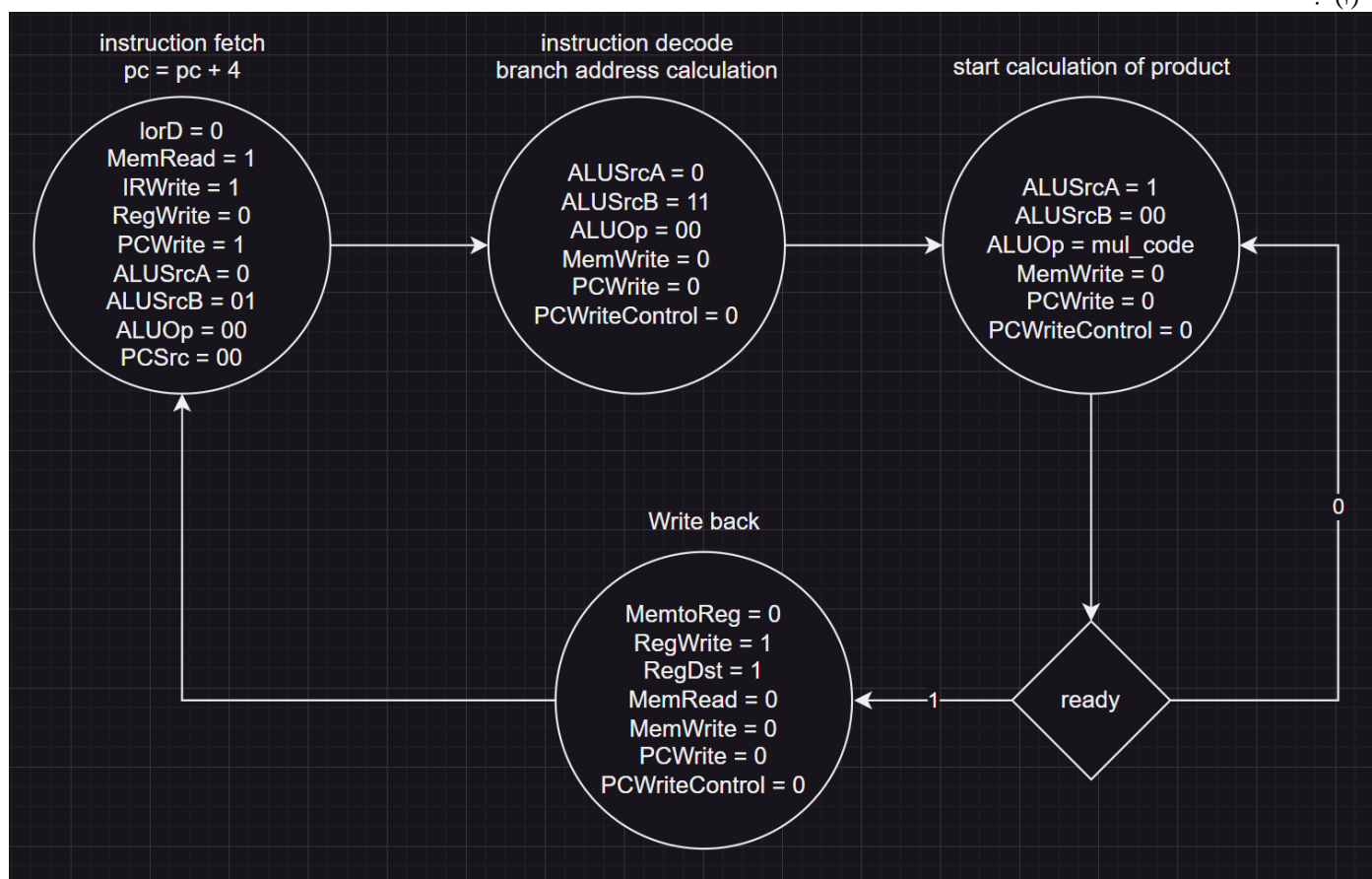
Branch این دستور به ۳ کلاک نیاز دارد پس $\text{throughput} = \frac{1}{1/1 \times 3}$ دستور در نانوثانیه

۲. مقدار سیگنال‌های کنترلی مورد نیاز در هر کلاک را برای هریک از موارد زیر بنویسید.

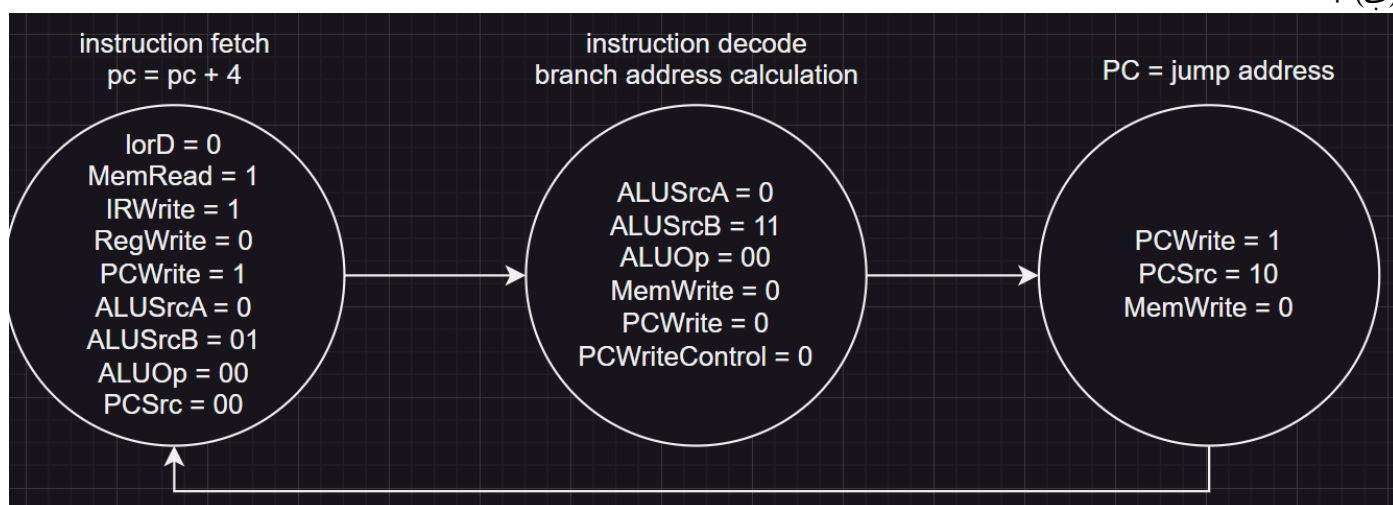
mul \$t8, \$t9, \$s0 (آ)

j loop (ب)

(آ)



(ب)



۳. زمان‌های واحدهای اصلی در یک مسیره داده را به صورت زیر در نظر بگیرید:

۱. Memory access (read and write) = 25 ps

۲. ALU = 20 ps

۳. Register file (read and write) = 15 ps

حداقل زمان چرخه ساعت (min clock cycle time)، میانگین CPI و میانگین زمان اجرای دستورالعمل را برای مسیره داده single-cycle و multi-cycle زیر را تعیین کنید.
10% lw, 10% sw, 40% register-type, 20% branch, 20% jump

instruction	IF	Decode & R-Read	ALU	D-cache	R-Write
jump	25ps	15ps	-	-	-
branch	25ps	15ps	20ps	-	-
R-type	25ps	15ps	20ps	-	15ps
sw	25ps	15ps	20ps	25ps	-
lw	25ps	15ps	20ps	25ps	15ps

در طراحی single cycle حداقل زمان چرخه ساعت برابر است با طول طولانی ترین دستور، که برابر است با $100ps$ ، میانگین CPI برای همه ی دستورات ۱ است و میانگین زمان اجرای دستورالعمل ها نیز همگی یکسان و $100ps$ است.
در طراحی multi cycle حداقل زمان چرخه برابر است با زمان مورد نیاز برای انجام شدن طولانی ترین ریز تسک که در اینجا برابر است با عملیات memory access که برابر است با $25ps$ ، میانگین CPI نیز به این صورت حساب می شود:

$$CPI = 0.1 \times 5 + 0.1 \times 4 + 0.4 \times 4 + 0.2 \times 3 + 0.2 \times 3 = 3.7$$

و میانگین زمان اجرای دستورالعمل نیز برابر است با $CPI \times \text{clock-rate}$ که برابر است با $92.5ps$

۴. پردازنده Multi-cycle MIPS را در نظر بگیرید. فرض کنید می‌خواهیم دستور زیر را به آن اضافه کنیم:

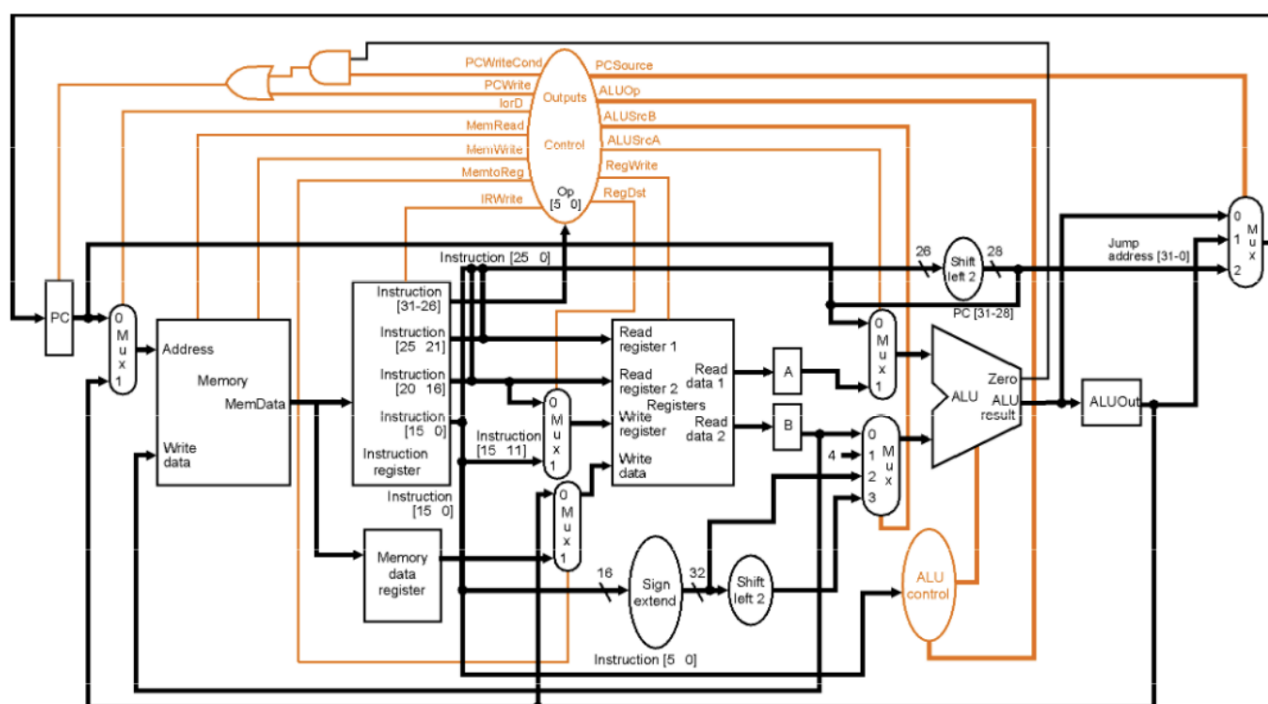
addm rt, rd, rs

که در این دستور

$$rd = rs + \text{Mem}[rt]$$

این دستور عملاً برخلاف دستورهای رایج R-Format که ساختار Register-Register دارند، ساختار Register-Memory داشته و هم حافظه و هم ثبات‌ها در عملیات جمع دخیل هستند.

آ) در مسیرهاده زیر تغییرات لازم را برای ساخت این دستور اعمال کنید و علت تغییر مدار خود را توضیح دهید. توجه کنید که حق تغییر در واحدهای حافظه، ثبات‌ها و ALU را ندارید ولی سایر بخش‌های مدار را می‌توانید تغییر بدهید.



ب) حالت‌های مربوط به این دستور را به FSM پردازنده اضافه کنید.

آ) به این منظور در مسیر داده تغییراتی می‌دهیم، ابتدا یک انشعاب از سیم rt با ماکس ورودی Memory می‌دهیم، با استفاده از این مسیر می‌توانیم در کلاک سوم، $\text{Mem}[rt]$ را در Memory data register داشته باشیم. سپس از این رجیستر به ماکس ALUSrcB یک سیم می‌کشیم تا بتوانیم عملیات جمع را نیز انجام دهیم و تمام.

۵. شرکتی یک ISA شامل سه دسته دستور دارد و طی سه نسل برای آن سه معماری متفاوت ایجاد کرده است:

- معماری اول: دستورات نوع A و B و C به ترتیب در 2, 4, 8 چرخه ساعت اجرا می‌شوند.
- معماری دوم: دستورات نوع A به میزان 62.5% و دستورات نوع B به میزان ۲۵% و دستورات نوع C به میزان ۵۰% سریعتر شده‌اند.
- معماری سوم (نسبت به معماری اول): دستورات نوع A به میزان 87.5% و دستورات نوع B و نوع C هر کدام ۵۰% سریعتر شده‌اند.

حال اگر یک مجموعه دستور داشته باشیم که اجرای آن در معماری دوم ۲ برابر سریعتر از معماری اول و در معماری سوم 1.5 برابر سریعتر از معماری دوم باشد، چند درصد از دستورات آن از نوع A چند درصد از نوع B و چند درصد از نوع C هستند؟

چون تعداد دستورات برای هر معماری یکسان است، نسبت سرعت آنها عکس نسبت CPI آنهاست.

$$\text{ISA}_1: \quad \text{Clock}_A = 8 \quad \text{Clock}_B = 4 \quad \text{Clock}_C = 2$$

$$\text{ISA}_2: \quad \text{Clock}_A = 3 \quad \text{Clock}_B = 3 \quad \text{Clock}_C = 1$$

$$\text{ISA}_3: \quad \text{Clock}_A = 1 \quad \text{Clock}_B = 2 \quad \text{Clock}_C = 1$$

$$a = \text{درصد دستورات A} \quad b = \text{درصد دستورات B} \quad c = \text{درصد دستورات C}$$

با توجه به مطالب ذکر شده این سه معادله را داریم که با حل آنها به جواب می‌رسیم.

$$a + b + c = 100$$

$$3a + 3b + c = \frac{1}{2} \times (8a + 4b + 2c) \rightarrow a = b$$

$$1a + 2b + c = \frac{2}{3} \times (3a + 3b + c) \rightarrow a = \frac{1}{3}c$$

$$\xrightarrow{\text{conclusion}} \frac{5}{3}c = 100 \rightarrow c = 60, a = 20, b = 20$$

۶. به سوالات زیر درباره میکروکدها پاسخ دهید:

- (آ) مفهوم Micro-Code و نقش آن در پردازنده را توضیح دهید.
- (ب) نقش Micro-Code در آسیب‌پذیری‌های مربوط به پردازنده و همچنین روش برطرف کردن آن‌ها را توضیح دهید.
- (ج) توضیح دهید به روزرسانی Micro-code ها از چه راه‌هایی انجام می‌شود و همچنین تحقیق کنید که اگر امکان به روزرسانی Micro-code ها وجود نداشت چه مشکلاتی می‌توانست رخ دهد.

- (آ) مفهوم میکرو کد یا میکرو program یعنی سیگنال‌های کنترلی با استفاده میکرو کد، ریز دستوراتی است که در یک پردازنده استفاده می‌شوند تا با گرفتن ترکیبی از آپکد دستور و استیت ماشین، سیگنال‌های کنترلی آن را مشخص کنند.
- (ب) میکرو کد که مسئول فرستادن سیگنال‌های کنترلی و انجام صحیح دستورات است، آسیب‌پذیری در آن می‌تواند باعث اجرا شدن دستورات مخرب و شناسایی نشده شود. همچنین می‌تواند باعث اجرای غلط دستورات پیچیده شود. و یا مهاجمین می‌توانند از این آسیب‌پذیری استفاده کنند تا برای خود یک راه دسترسی مخفی داشته باشند.
- برای برطرف کردن این مشکلات راه‌هایی وجود دارد. برای مثال چندی یکبار میکروکد ها توسط بایس آپدیت می‌شوند و باگ‌های آنها برطرف می‌شود. همچنین در برخی طراحی‌ها یک محافظ سخت افزاری وجود دارد تا مانع انجام دستورات مخرب شود. و گاهی برای تشخیص میکروکد های صحیح، یک نشانه‌ی تایید در آنها قرار می‌دهند.
- (ج) یکی از روش‌هایی که میکروکد ها بروز می‌شوند، با استفاده از BIOS یا UEFI است. به این صورت که در هر راه اندازی، آنها تغییرات میکرو کد های جدید را اعمال می‌کنند. اگر همچین مکانیزمی وجود نداشت، در صورت پیدا شدن باگی در این میکروکد ها، همه‌ی پردازنده‌های قبل از تصحیح این باگ، آسیب‌پذیر به این خطر می‌شوند و نمی‌توان کاری برای برطرف کردن آن انجام داد.