درس معماری کامپیوتر

تمرین سری ششم **۱۴۰۴/۰۳/۰۲**

متین باقری **۴۰۲۱۰۵۷۲۷**

---

۱) آ)

$$40\% * 6 + 20\% * 6 + 30\% * 2 + 10\% * 2 = 4.4\ CPI$$

ب)

$$4.4 + 4 = 8.4\ CPI$$

ج)

استفاده از P2 بهینه تر است.

$$\frac{8.4}{2} = 4.2 \qquad ===> speedup = \frac{4.4}{4.2} = 1.047619$$

د) مورد دوم تسریعی ندارد. مورد اول CPI را از 4.4 به 4 می‌رساند و تسریع دارد. پس مورد اول را انتخاب می‌کنیم.

.1

$$40\% * 6 + 20\% * 6 + 30\% * 1 + 10\% * 1 = 4\ CPI$$

.2

$$40\% * 3 + 20\% * 12 + 30\% * 2 + 10\% * 2 = 4.4\ CPI$$

۲) آ) مسیر داده تغییری نمی‌کند. در واحد کنترلی ALU باید این دستور اضافه شده و در صورت تشخیص دادن آن یک سیگنال جدید با ALU بدهد تا ALU یک محاسبه جدید (max) را انجام دهد.

ب) دو مرحله اول که در همه دستورات مشترک است. در مرحله EX دو رجیستر خوانده شده وارد ALU شده و خروجی ALU به این صورت است: rd = (rs >= rt) ? rs : rt

نحوه محاسبه max را می‌توان اینگونه در نظر گرفت که یک max داریم با ورودی صفر rs و ورودی یک rt، بیت سلکتور آن متصل است به MSB حاصل rs – rt .

از آنجا که این دستور از نوع R-type در نظر گرفته می‌شود، مرحله mem ندارد (یا اگر مجبور باشیم فرض کنیم باید در ۵ کلاک اجرا شود، در این مرحله هیچ کاری انجام نمی‌شود.) در مرحله WB داده موجود در ALUout در rd نوشته می‌شود.

cycle	IF	ID	EX	WB
PCWriteCond	x (0)	0	0	0
PCwrite	1	0	0	0
IorD	0	x	x	x
MemRead	1	0	0	0
MemWrite	0	0	0	0
MemtoReg	x	x	x	0
IRWrite	1	0	0	0
PCSource	00	x	x	x
ALUop	00 (add)	00 (add)	10 (R-type)	xx
ALUSrcB	01	11	00	xx
ALUSrcA	0	0	1	x
RegWrite	0	0	0	1
RegDst	x	x	x	1

ج) اسمبلر la را به lui و ori تبدیل کرده که به ترتیب ۳ و ۴ کلاک زمان می‌برند.

$$\text{cycles} : 3+4+5+5+4+4+3+4+4 = 36 \quad , \text{instruction} = 9 \quad , \text{CPI} = 4$$

---

۳) آ) single cycle:

CPI همیشه 1 است.

R-type:

$$200 + 50 + 100 + 50 = 400$$

lw:

$$200 + 50 + 100 + 200 + 50 = 600$$

sw:

$$200 + 50 + 100 + 200 = 550$$

beq:

$200 + 50 + 100 = 350$ با فرض اینکه نوشتن در PC تاخیری ندارد

$200 + 50 + 100 + 50 = 400$ با فرض اینکه نوشتن در PC تاخیر دارد

J:

200 با فرض اینکه نوشتن در PC تاخیری ندارد

$200 + 50 = 250$ با فرض اینکه نوشتن در PC تاخیر دارد

از آنجا که طول کلاک باید ثابت باشد، هر کلاک ۶۰۰ واحد زمانی طول میکشد و همه دستورات نیز ۶۰۰ واحد زمانی طول میکشند.

multi cycle:

در اینجا طول کلاک برابر با ۲۰۰ (بیشترین زمان مورد نیاز در یک کلاک) است.

R-type: CPI = 4

4 * 200 = 800

---

lw: CPI = 5

5 * 200 = 1000

---

sw: CPI = 4

4 * 200 = 800

---

beq: CPI = 3

3 * 200 = 600

J: CPI = 3

3 * 200 = 600

ب)

single cycle CPI = 1

multi cycle CPI = $25\% \times 5 + 10\% \times 4 + 52\% \times 4 + 11\% \times 3 + 2\% \times 3 = 4.12$

ج) در multi cycle تاخیر یک کلاک 200 است، در single cycle 600 است. بنابراین clock rate در multi cycle ۳ برابر است. میزان تسریع multi cycle نسبت به single cycle در اجرای برنامه قسمت ب:

speedup = 600 / (4.12 * 200) = 0.728155339

بنابراین زمان اجرا در single cycle کمتر multi cycle است و single cycle عملکرد بهتری دارد.

۴)

$$1.8 = \frac{5x + 10y + 3z}{3x + 7y + 2z} \times 1.2 ==> x = y$$

$$1.875 = \frac{3x + 7y + 2z}{2x + 5y + 1z} \times \frac{1.5}{1.2} ==> y = z$$

هر یک از سه نوع دستورات، 33.3٪ (یک سوم) از دستورات را تشکیل می‌دهند.

$$T_{limit} < \min\left(\frac{1}{F}, \frac{1}{1.2F}, \frac{1}{1.5F}\right) = \frac{1}{1.5F\ Mega\ Hz} = \frac{2}{3F} micro\ second$$

---

۵) آ) از آنجا که ۱۲ FF داریم، می‌توانیم تا 2^12 = 4096 state مختلف داشته باشیم. 12 + 8 = 20 بیت ورودی میدهیم و به ۳۰ بیت کنترلی و ۱۲ بیت برای حالت بعدی دست پیدا میکنیم. پس اندازه کل حافظه برابر است با

42 Mb = 5.25 MB

ب) اولی 12 + 8 = 20 بیت ورودی و 12 بیت خروجی

2^20 سطر، هر یک دارای 12 بیت ← 12 مگا بیت = 1.5 مگا بایت

دومی دارای 2^12 سطر هر یک شامل 30 بیت ← 122,880 بیت = 120 Kb = 15 KB = ۱۵ کیلو بایت

در مجموع: 1.515 MB

ج) قسمت اول) 2^8 دستور داریم که هریک حداکثر 8 micro-instruction دارد، پس در کل حداکثر 2^11 سطر در rom داریم. هر سطر شامل 30 بیت کنترلی و 11 بیت برای نشان دادن سطر micro-instruction بعدی است.

2^11 * 41 = 82 Kb = 10.25 KB

قسمت دوم) در rom اول، 2^8 دستور داریم که هریک حداکثر 8 micro-instruction دارد، پس در کل حداکثر 2^11 سطر در rom داریم. هر سطر شامل 11 بیت برای نشان دادن سطر micro-instruction بعدی است.

2^11 * 11 = 22 Kb = 2.75 KB

در rom دوم، 2^11 سطر و 30 ستون داریم.

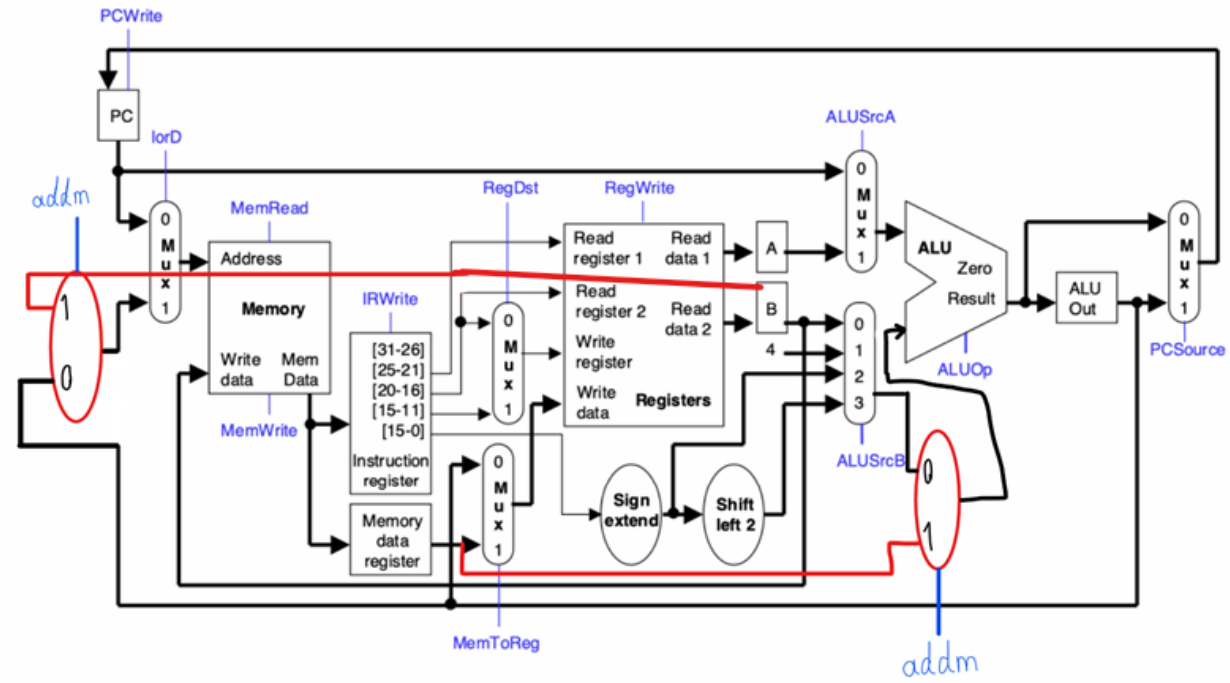
2^11 * 30 = 60 Kb = 7.5 KB

مجموع سایز دو rom:

7.5 + 2.75 = 10.25 KB

---

۶) آ)



پردازنده مشابه تصویر بالا خواهد شد. دو MUX اضافه کردیم که بیت سلکتور آنها سیگنال کنترلی addm است که درصورت addm بودن دستور یک می‌شود. MUX سمت چپ به ما اجازه می‌دهد از حافظه مقدار Mem[rt] را بخوانیم. MUX

سمت راست به ما اجازه می‌دهد مقدار Mem[rt] خوانده شده از حافظه که در memory data register قرار گرفته است را وارد ALU کرده و با مقدار $rs جمع کنیم.

ب) دو سایکل اول در همه دستورات مشترک است و در تصویر درون سوال نیز مشخص شده اند. مقادیر سیگنال ها کنترلی در دیگر سایکل های اجرای دستور addm:

cycle	3 – mem	4 – Exe	5 – WB
PCwrite	0	0	0
IorD	1	x	x
MemRead	1	0	0
MemWrite	0	0	0
MemtoReg	x	x	0
IRWrite	0	0	0
PCSource	x	x	x
ALUop	xxx	010	xx
ALUSrcB	xx	xx	xx
ALUSrcA	x	1	x
RegWrite	0	0	1
RegDst	x	x	1
addm	1	1	1