به نام خداوند بخشنده مهربان

معماری کامپیوتر

دانشکده مهندسی کامپیوتر

دکتر اسدی
بهار ۱۴۰۳

مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶

تمرین ۸ تئوری

سوال ۱

۱. یک پردازنده دارای ۴ گروه دستورالعمل نوع الف تا د می‌باشد. نسبت وقوع این دستور العمل‌ها در یک برنامه محک در جدول مقابل نشان داده شده است که هر گروه از دستور العمل‌ها نیاز به چه مراحلی در اجرا دارند و زمان اجرای هر مرحله چه مقدار است. نسبت افزایش سرعت اجرای این برنامه در صورت پیاده‌سازی خطلوله نسبت به عدم پیاده‌سازی خطلوله چقدر است؟

مراحل اجرای دستور	FE	DE	EXE	MEM	WB		
زمان اجرای مرحله	10ns	7ns	10ns	12ns	7ns	درصد وقوع	نوع دستور
	✓	✓	✓	✓	✓	٪۲۰	الف
	✓	✓	✓	-	✓	٪۴۰	ب
	✓	✓	✓	✓	-	٪۲۰	ج
	✓	✓	✓	-	-	٪۲۰	د

بدون پیاده سازی خط لوله، می توان از دو نوع single cycle و multi cycle استفاده کرد، در نوع اول، دوره تناوب برابر با طولانی ترین دستور است که برابر با $۴۶ns$ است و CPI آن برابر یک است.
برای نوع دوم نیز دوره ی تناوب کلاک برابر است با طولانی ترین مرحله ی اجرا که برابر با $۱۲ns$ است. و برای CPI نیز داریم:

$$\text{CPI} = ۰/۲ * ۵ + ۰/۴ * ۴ + ۰/۲ * ۴ + ۰/۲ * ۳ = ۴$$

و در معماری پایپ لاین دوره تناوب کلاک برابر است با کلاک در طراحی multi cycle یا همان $۱۲ns$ است. و CPI آن در حالتی که هازارد نداشته باشیم برابر است با ۱. پس speed up در مقابل نوع اول برابر است با:

$$SU_{SC} = \frac{IC \times CPI_{SC} \times CT_{SC}}{IC \times CPI_{PL} \times CT_{PL}} = \frac{۴۶}{۱۲}$$

و در مقابل نوع دوم:

$$SU_{MC} = \frac{IC \times CPI_{MC} \times CT_{MC}}{IC \times CPI_{PL} \times CT_{PL}} = \frac{۴}{۱}$$

سوال ۲

۲. همان طور که در درس توضیح داده شد، زمانی که یک CPU دارای خط‌لوله به یک دستور پرش می‌رسد، دو کار می‌تواند انجام دهد. یا در خط‌لوله bubble قرار دهد تا نتیجه‌ی پرش مشخص شود یا پیش‌بینی کند که taken branch (یا not taken branch) رخ می‌دهد و با این فرض دستورات بعدی را در خط‌لوله قرار دهد. سپس در صورتی که اشتباه پیش‌بینی شده بود، خط‌لوله را خالی کند. حال به سوالات زیر جواب دهید.

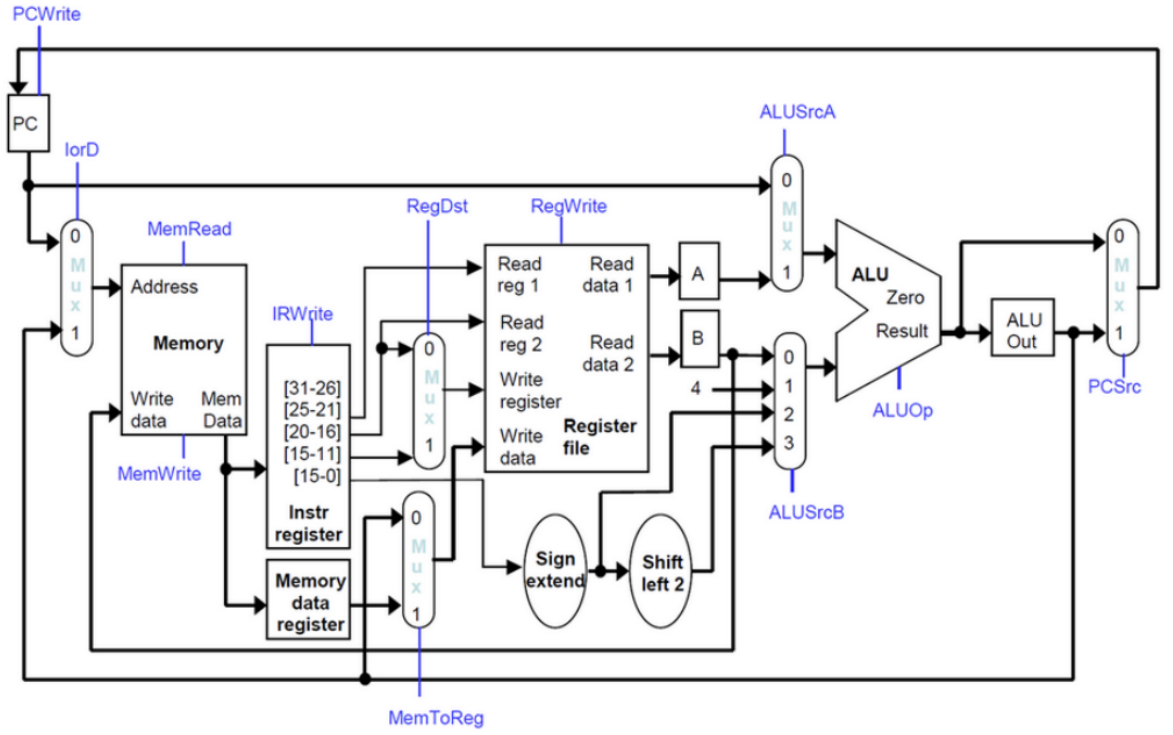
آ) درباره تاثیر branch predictorsها بر عملکرد پردازنده‌ها تحقیق کنید.

ب) در رابطه با نحوه کارکرد branch predictor تک‌مرحله‌ای و دومرحله‌ای را تحقیق کنید.

(آ) با اعمال کردن مکانیزم branch prediction در صورت موفقیت آمیز بودن آن، باعث کاهش CPI و افزایش کارایی پردازنده می شود. همچنین stall ها را نیز کاهش می دهد، و اگر به درستی حاصل پرش را پیش بینی نکند، نیاز است به منطق پاکسازی آثاری که این پیشبینی غلط به جا گذاشته، مانند flush کردن خط لوله و بازگردانی هرگونه write ای که صورت گرفته است.

(ب) در branch predictor تک مرحله ای، از یک شمارنده برای نگه داشتن نتیجه ی پرش های گذشته استفاده می شود، و بر اساس آنها، پرش های آینده را نیز پیش بینی می کند. اینگونه که با هر پرش به این شمارنده اضافه می شود و در غیر این صورت، با عبور از دستور برنچ، از این شمارنده کم می شود و بیت علامت آن معیار پیش بینی پرش است و برای branch predictor دو مرحله ای، اگر فرض کنیم این predictor بر اساس n استیت پرش پیشین باشد، یک جدول با 2^n خانه داریم که هر کدام احتمال پرش بعدی را بر اساس n پرش قبلی دارد.

۳. پردازنده Multi-Cycle MIPS که در شکل زیر آماده است را در نظر بگیرید. می‌خواهیم دستور lws را به آن اضافه کنیم.
lws rd, rs, rt (rd = MEM[rt+(4*rs)])



به عنوان مثال اگر مقدار a0 برابر ۱۰۰۰ بوده و مقدار a1 برابر ۱۰ باشد، دستور lws $t0,$a1,$a0 مقداری که درون آدرس ۱۰۴۰ قرار دارد را درون ثبات t0 قرار می‌دهد.

آ) تغییرات لازم در مسیرداده برای این که امکان استفاده از این دستور را داشته باشیم، مشخص کنید.

ب) سیگنال‌های کنترلی در چرخه‌های مختلف برای اجرای این دستور را بنویسید. فرض کنید که ALU امکان انجام عمل جمع با ALUOp=000 و عمل ضرب با ALUOp=100 را دارد. دو چرخه ابتدایی که در همه دستورات مشترک هستند به صورت زیر است:

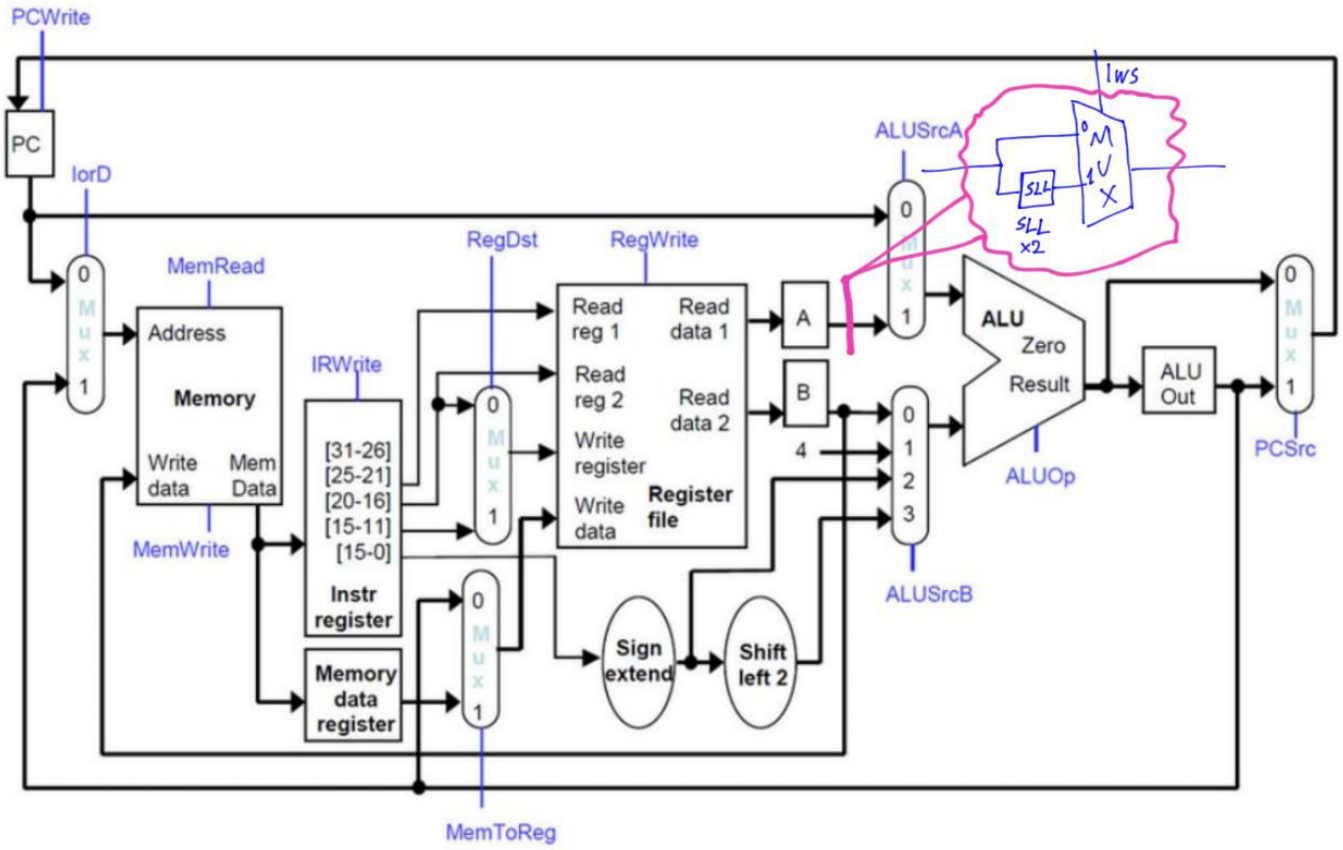
چرخه اول:

IorD=0, MemRead=1, IRWrite=1, ALUSrcA=0, ALUSrcB=01, ALUOp=000, PC-Source=0, PCWrite=1

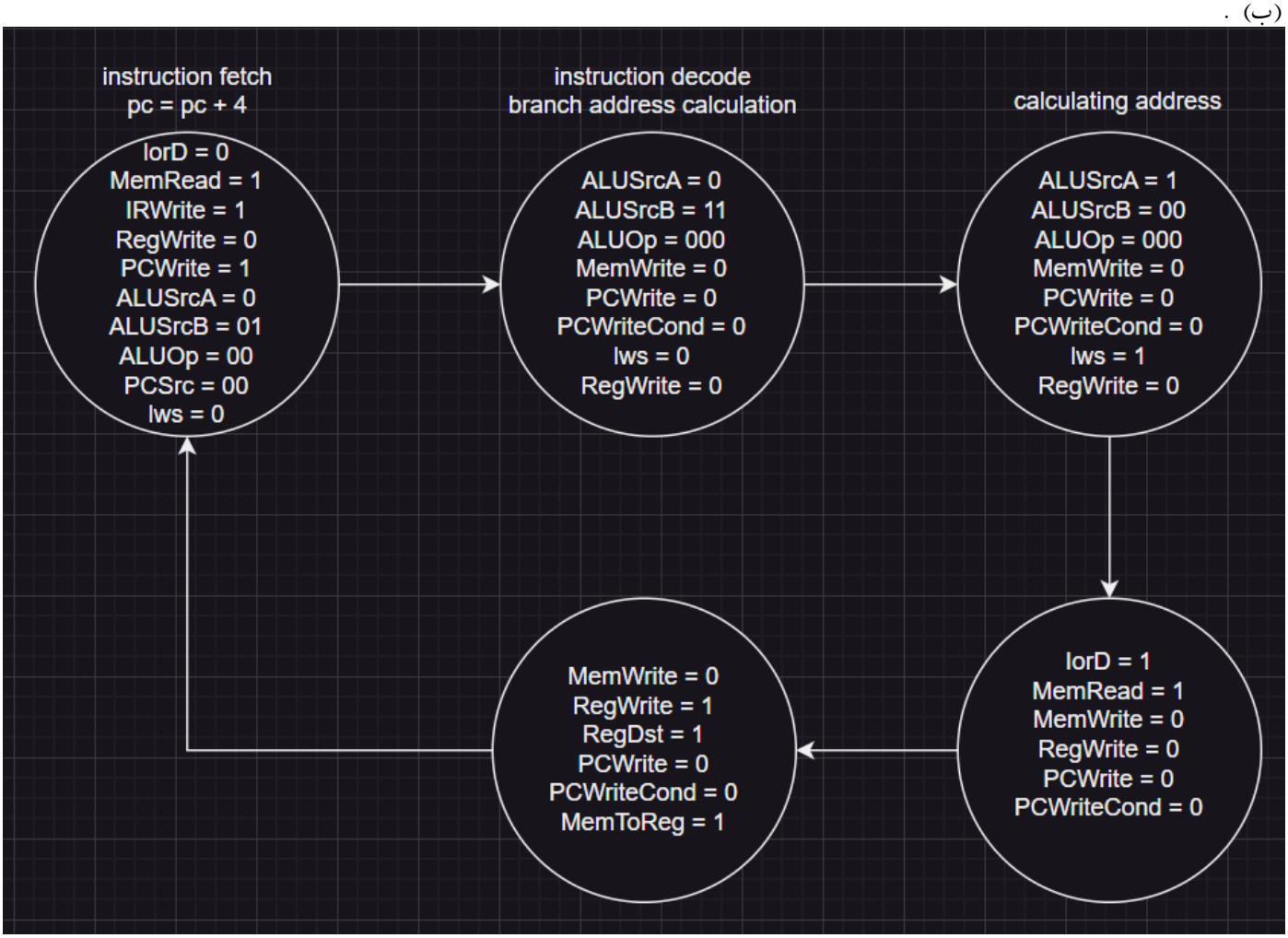
چرخه دوم:

ALUSrcA=0, ALUSrcB=11, ALUOp=000

(آ) .


PCWrite
PC
IorD
MemRead
Address
Memory
Write data
Mem Data
MemWrite
IRWrite
[31-26]
[25-21]
[20-16]
[15-11]
[15-0]
Instr register
Memory data register
RegDst
RegWrite
Read reg 1
Read reg 2
Write register
Write data
Read data 1
Read data 2
Register file
A
B
4
ALUSrcA
ALUSrcB
ALU
Zero
Result
ALUOp
ALU Out
PCSrc
Sign extend
Shift left 2
MemToReg
lws
SLL
SLL x2

(ب) .


instruction fetch
pc = pc + 4
IorD = 0
MemRead = 1
IRWrite = 1
RegWrite = 0
PCWrite = 1
ALUSrcA = 0
ALUSrcB = 01
ALUOp = 00
PCSrc = 00
lws = 0
instruction decode
branch address calculation
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 000
MemWrite = 0
PCWrite = 0
PCWriteCond = 0
lws = 0
RegWrite = 0
calculating address
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 000
MemWrite = 0
PCWrite = 0
PCWriteCond = 0
lws = 1
RegWrite = 0
IorD = 1
MemRead = 1
MemWrite = 0
RegWrite = 0
PCWrite = 0
PCWriteCond = 0
MemWrite = 0
RegWrite = 1
RegDst = 1
PCWrite = 0
PCWriteCond = 0
MemToReg = 1

سوال ۴

۴. فرض کنید بخواهیم به جای ۳۲ ثبات ۳۲ بیتی MIPS از ۶۴ ثبات ۶۴ بیتی استفاده کنیم،

آ) برای این کار چه تغییراتی باید در ساختار و معماری پردازنده اعمال کنیم؟

ب) در صورت اعمال این تغییرات، اندازه کدها چه مقدار افزایش خواهد یافت؟

ج) این تغییر چطور می‌تواند باعث کاهش سرعت نسبت به حالت قبل شود؟

د) در صورت استفاده از خطلوله، چه تغییرات دیگری نسبت به حالت قبل وجود دارد؟

(آ) تمام رجیستر ها، خانه های حافظه و گذرگاه های داده باید ۶۴ بیتی شوند، و تمام سلکت های رجیستر ها باید ۶ بیتی بشوند. سایز ISA نیز باید افزایش یابد تا بتواند با رجیستر های جدید کار کند.

(ب) اندازه ی کد ها متناسب با تغییر سایز ISA زیاد می شود.

(ج) هرچه مقدار داده ی نیاز به جابجایی بیشتر شود، از سرعت آن در خواندن و نوشتن می کاهد. برای مثال، حافظه های بیشتر، زمان بیشتری برای دسترسی به داده نیاز دارند، همچنین تاخیر محاسبات، mux ها و گیت ها بخاطر افزایش سایز ورودی، بیشتر می شود.

(د) نیاز است حواسمان باشد که طول هر مرحله نسبت به حالت قبل، افزایش داشته و باید فرکانس کلاک را کمتر کرد به اندازه ای که تاخیر به وجود آمده توسط افزایش اندازه ی محاسبات را پوشش دهد.

سوال ۵

۵. فرض کنید یک مسیر داده single-cycle با زمان چرخه T داریم. مسیر داده را به n مرحله به صورت خطلوله تقسیم می‌کنیم (فرض کنید مرحل مختلف دارای تأخیر تقریبا یکسانی هست). فرض کنید زمان چرخه در این حالت برابر با ((T/n)+ an) باشد. اگر T=10ns و a=0.1ns و n>=2 باشد آنگاه دوره چرخه را وقتی که گذردهی[1] خطلوله برای جریان بزرگی از دستورالعمل‌ها ماکزیمم باشد، محاسبه کنید.

به دنبال n ای می گردیم که thorughput را ماکزیمم کند، می دانیم برای گذردهی و زمان اجرا داریم:

$$throughput = \frac{1}{\text{execution time}}$$
$$\text{execution time} = IC \times CPI \times CT$$

CPI و IC که اعداد ثابتی هستند، در تعداد دستور زیاد، CPI = ۱ پس برای اینکه گذردهی را ماکزیمم کنیم کافیست زمان اجرا را مینیمم کنیم و برای اینکار تنها نیاز است زمان هر کلاک را کمینه کنیم.

$$IC = \frac{T}{n} + an \xrightarrow{\frac{d}{dn}} -\frac{T}{n^2} + a = 0 \rightarrow n = \pm\sqrt{\frac{T}{a}} \xrightarrow{T=10, a=0.1} n = 10$$

و در این حالت گذردهی برابر است با

$$throughput = \frac{1}{2 \times 10^{-9} \times IC} = \frac{5 \times 10^8}{IC}$$