به نام خداوند بخشنده مهربان

معماری کامپیوتر

دانشکده مهندسی کامپیوتر

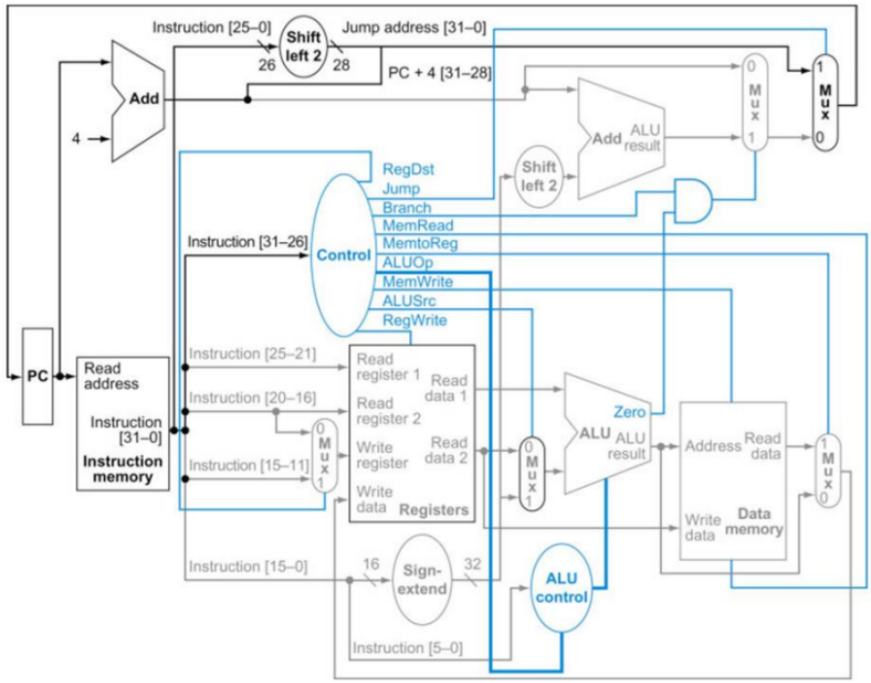
دکتر اسدی
بهار ۱۴۰۳

مهدی علی نژاد، ۴۰۱۱۰۶۲۶۶

تمرین ۵، تئوری

سوال ۱

۱. فرض کنید گونه‌ای از پردازنده میپس را در اختیار دارید که از دستورات addi و ori و slti و lw و sw و R-format و beq پشتیبانی می‌کند و مسیر داده آن به شکل زیر است. تابع منطقی تولیدکننده سیگنال‌های کنترلی خارج شده از واحد کنترل این پردازنده را بدست آورید.



instruction	opcode[5..0]	RegDst	Jump	Br	MR	MtR	ALUOp	MW	ALUSrc	RW
addi	001000	0	0	0	0	0	00	0	1	1
ori	001101	0	0	0	0	0	11	0	1	1
slti	001010	0	0	0	0	0	01	0	1	1
lw	100011	0	0	0	1	1	00	0	1	1
sw	101011	x	0	0	0	x	00	1	1	0
R-format	000000	1	0	0	0	0	10	0	0	1
beq	000100	x	0	1	0	x	01	0	0	0

$$\text{RegDst} = \overline{\Sigma_{i=0}^{5}\text{opcode[i]}}$$

$$\text{Jump} = 0$$

$$\text{Br} = \text{opcode}[2] \wedge \neg \text{opcode}[0]$$

$$\text{MR} = \text{opcode}[5] \wedge \neg \text{opcode}[3]$$

$$\text{MtR} = \text{MR}$$

$$\text{ALUOp}[0] = \text{opcode}[2] \vee (\text{opcode}[3] \wedge \neg \text{opcode}[0])$$
$$\text{ALUOp}[1] = \text{RegDst} \vee (\text{opcode}[2] \wedge \text{opcode}[0])$$

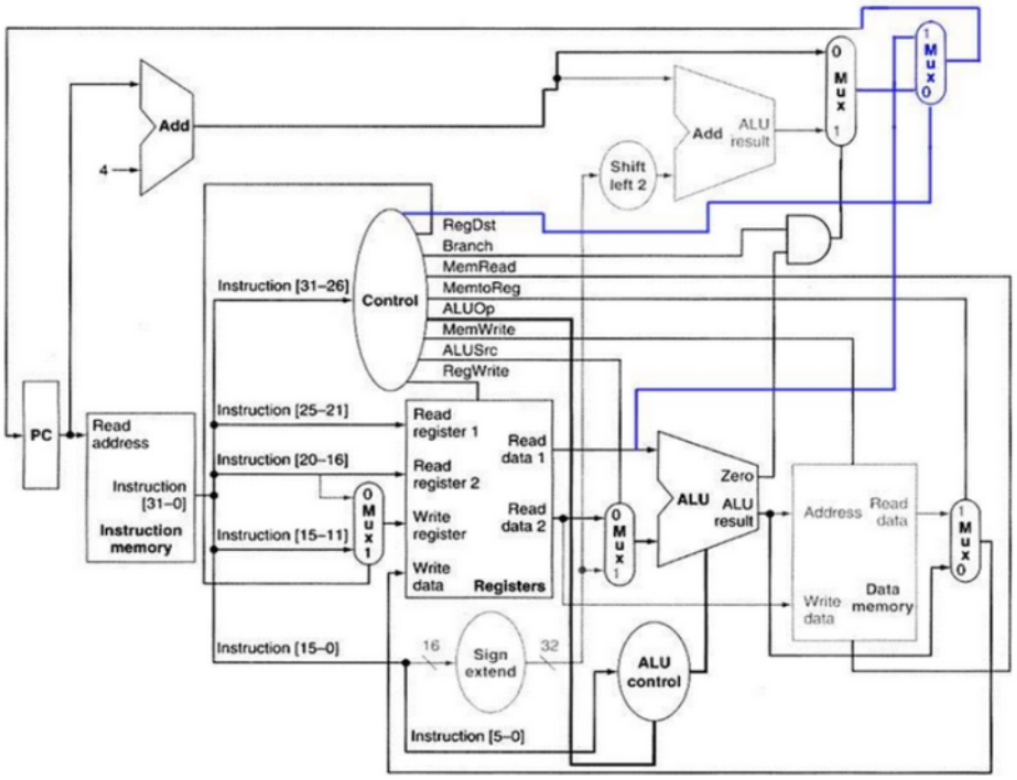
$$\text{MW} = \text{opcode}[5] \wedge \text{opcode}[3]$$

$$\text{ALUSrc} = \text{oopcode}[3] \vee \text{opcode}[0]$$

$$\text{RW} = \overline{\text{Br} \vee \text{MW}}$$

سوال ۲

۲. در مسیرداده پردازنده MIPS تغییرات زیر که با رنگ آبی در مدار مشخص شده‌اند را اضافه کرده‌ایم:



آ) مشخص کنید که سیگنال کنترلی جدیدی که اضافه کرده‌ایم چیست و به چه منظوری اضافه شده است؟

ب) در صورتی که این سیگنال ۱ شود، مقادیر سیگنال‌های کنترلی RegDst، Branch، MemWrite، ALUSrc را مشخص کنید.

(آ) تغییرات آبی رنگ می تواند به هدف پشتیبانی کردن دستور jump register باشد، که عملا آدرس یکی از ثبات ها را درون PC می ریزد.

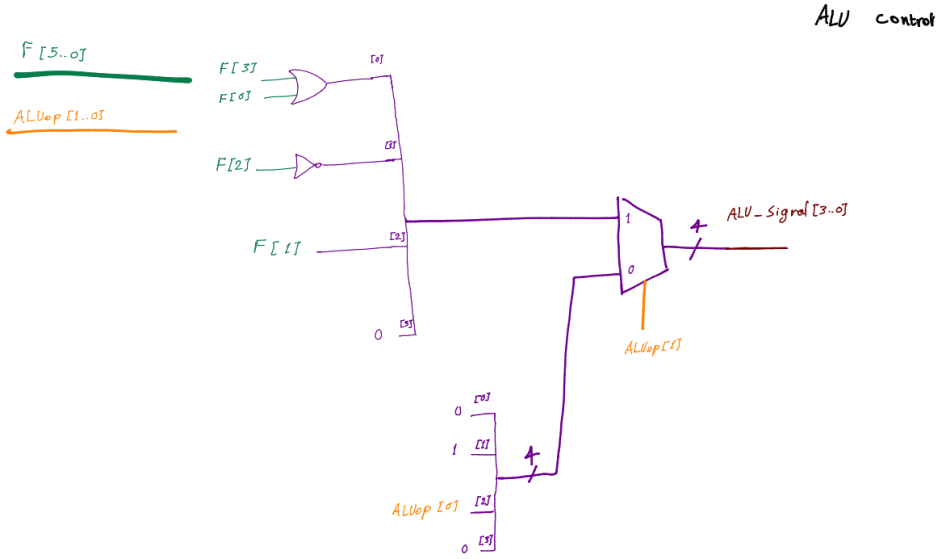
(ب) در این صورت سیگنال MemWrite صفر و باقی سیگنال ها don't care است.

سوال ۳

۳. مدار کنترل‌کننده ALU در پردازنده MIPS را رسم کنید که ورودی‌های ALUOp و Funct را به‌عنوان سیگنال‌های کنترلی دریافت و خروجی مربوطه را تولید می‌کند. دقت کنید که در این سوال علاوه‌بر رسم مدار کنترل‌کننده ALU باید شرح کاملی نیز از آن ارائه گردد.

ALUOp		Funct field						
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	Operation
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

این جدولی است که در اسلاید ها آمده شده که سیگنال های ورودی و خروجی واحد کنترلی ALU را مشخص کرده.



این مداری است که طبق جدول حقیقت عمل خواهد کرد. طبق جدول در حالت هایی که ALUOp۱ مقدار صفر دارد، سیگنال های F[۵..۰] همگی دونت کر هستند و سیگنال خروجی تنها وابسته به ALUOp۰ خواهد بود که با وصل کردن مستقیم آن به بیت سوم خروجی و وصل کردن vcc به بیت دوم و gnd به بیت اول و چهارم به خروجی خواسته شده می رسیم. برای حالتی که F[۵..۰] خروجی را تعیین می کنند نیز با کمی دقت به جدول درستی متوجه می شویم که بیت های F[۵..۴] در تعیین خروجی نقشی ندارند. بیت چهارم خروجی نیز همواره صفر است. بیت سوم معادل F۱ است. بیت دوم معادل $\overline{F۲}$ و بیت اول نیز معادل F۰ | F۳ است.

سوال ۴

۴. یکی از مشکلات متداول در مدار های مجتمع بر پایه سیلیکون، ثابت شدن مقدار یک سیگنال روی ۰ یا ۱ است.

آ) فرض کنید سیگنال RegWrite متصل به رجیستر فایل روی مقدار ۰ ثابت شده باشد، در این صورت اجرای چه دستوراتی با مشکل مواجه می‌شوند؟ توضیح دهید.

ب) در صورتی که بیت صفرم ALUop روی مقدار ۰ ثابت شود، اجرای کدام دستورات با مشکل مواجه می‌شوند؟ توضیح دهید.

ج) در صورتی که RegDest روی ۱ ثابت شود، کدام دستورات با مشکل مواجه می‌شوند؟ توضیح دهید.

(آ) عملا مقدار رجیستر ها دیگر تغییر نمی کند، پس هر دستوری که مقصد رجیستر داشته باشد به مشکل می خورد. از جمله R-type، Load، addi، ori، ... .

(ب) بیت های ALUop برای دستورات store و load ، ۰۰ است، به معنی جمع.
برای دستورات branch ۰۱ است به معنی تفریق. و برای R-type ۱۰ است که در این حالت به funct. عملیات مشخص می شود. اگر بیت صفرم آن روی مقدار صفر گیر کند، دستورات branch با مشکل مواجه می شوند.

(ج) در اجرای دستوراتی که رجیستر دوم همان رجیستر مقصد است به مشکل می خوریم. مثل دستورات I-type، load، ... .

سوال ۵

۵. فرض کنید در مسیرداده یک پردازنده قابلیت اجرای دستور slt وجود دارد، به این شکل که ورودی اول a و ورودی دوم b در دستور داده می‌شوند و در مسیرداده مقدار $X = a - b$ توسط ALU محاسبه می‌شود و اگر overflow رخ دهد flag مربوطه f برابر با یک می‌شود. درنهایت با داشتن بیت‌های آخر a،b،x و f مقدار خروجی دستور slt بدست می‌آید. منطق بدست آوردن خروجی را ابتدا در یک جدول نشان دهید و سپس عبارت منطقی آن را بنویسید.

MSB_a	MSB_b	MSB_x	f	slt
1	0	x	x	1
0	1	x	x	0
0	0	1	x	1
0	0	0	x	0
1	1	1	x	1
1	1	0	x	0

$$\text{stl} = ((\text{MSB}_a \oplus \text{MSB}_b) \wedge \text{MSB}_a) \vee ((\text{MSB}_a \odot \text{MSB}_b) \wedge \text{MSB}_x)$$