به نام خدا

دانشکده مهندسی کامپیوتر

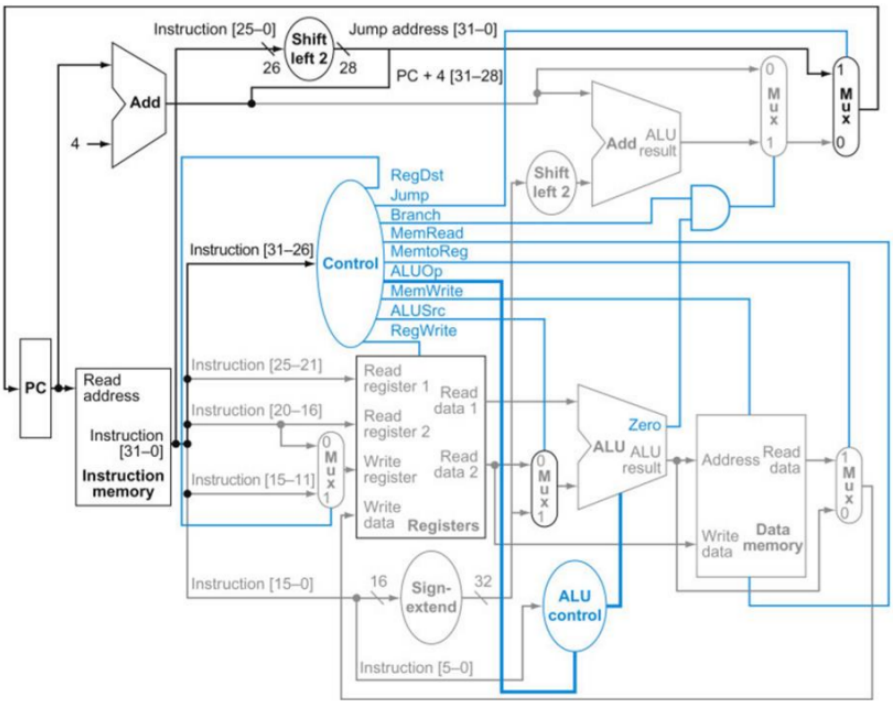
درس معماری کامپیوتر
نیم‌سال دوم ۰۲-۰۳
استاد: دکتر حسین اسدی

تمرین سری پنجم

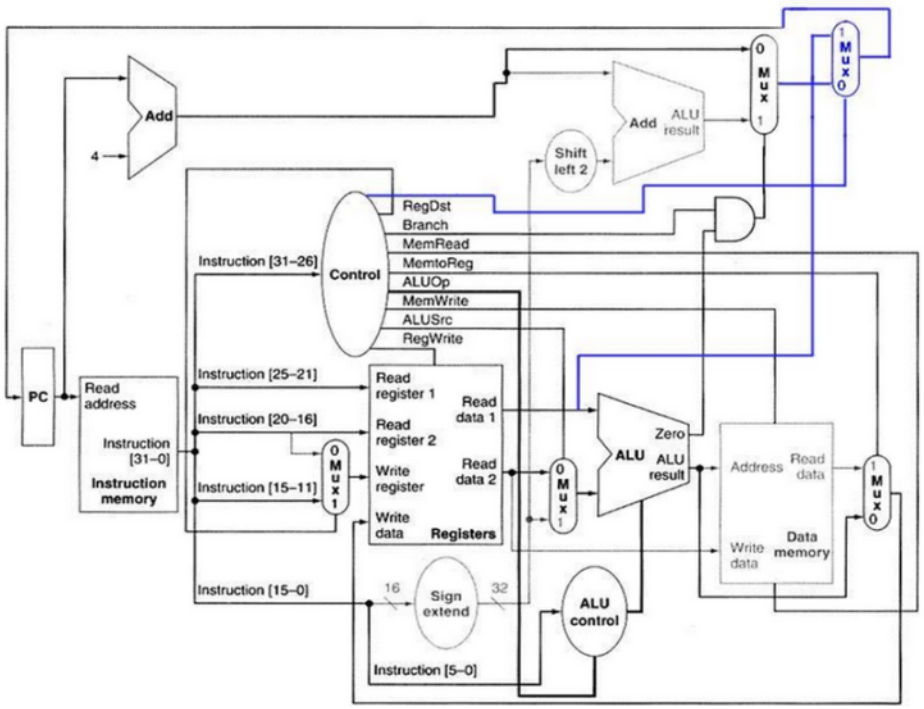
- پرسش‌های خود را در صفحه quera مربوط به تمرین مطرح نمایید.
- سوالات نظری را حتماً به صورت انفرادی و سوالات عملی را می‌توانید در گروه‌های دو نفر تحویل دهید.
- پاسخ‌ها را به صورت تایپی بنویسید.
- اسکرین‌شات‌ها، عکس‌ها و فایل‌های مربوط به سوال عملی را در فایل فشرده مربوطه در cw و quera قرار دهید. هر گونه عدم تطابق بین دو تمرین آپلود شده در دو سایت منجر به از دست رفتن نمره تمرین مربوطه می‌شود.
- پی دی اف قسمت تئوری را در سامانه cw و quera بارگذاری کنید.
- هر دانشجو می‌تواند حداکثر سه تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.

تمارین تئوری

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.



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.

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.

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.

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.

تمارین عملی

در این تمرین قصد داریم مسیرداده و واحد کنترلی پردازنده MIPS را به صورت کامل طراحی کنیم. همچنین از قطعاتی که در دو تمرین قبل طراحی کرده‌اید یعنی ALU و بانک ثبات و حافظه در ساخت پردازنده خود استفاده کنید. پس از پایان این تمرین شما با دانش معماری خود یک واحد پردازشی تولید کرده‌اید که توانایی اجرای برنامه‌های ساده را دارد. دقت کنید که این تمرین را به صورت کامل انجام دهید و حتما تست‌های کافی را روی آن انجام دهید چرا که در غیر این صورت در فازهای بعدی با مشکل مواجه خواهید شد. پس از ساخت کامل این پردازنده باید بتوانید برنامه‌های ساده را روی آن اجرا کنید که نحوه تست کردن را نیز در ادامه ذکر می‌کنیم.

برای یادآوری دستورات تمرین سری قبل آورده شده‌اند. مانند پردازنده‌ی MIPS این پردازنده نیز سه نوع کدگذاری دستور دارد: Register, Immediate, Jump. تعریف هر کدام از این دستورات در زیر آمده است:

- **Register Instructions**: این دستورات همان طور که از اسم آن‌ها پیدا است برای زمانی استفاده می‌شوند که قرار است به کمک محتوای دو ثبات، یک ثبات دیگر را مقداردهی کنیم. این دستورات دارای فرمت زیر هستند:

opcode	rs	rt	rd	funct
4 bits	3 bits	3 bits	3 bits	3 bits

دقیقا مثل پردازنده‌ی MIPS در تمامی دستورات این نوع، ثبات opcode آن‌ها برابر صفر است و بر اساس مقدار سیگنال funct می‌توان نوع عملیات را تعیین کرد. جدول عملیات در زیر آمده است:

Mnemonic	Operation	funct
ADD	rd ← rs + rt	000
SUB	rd ← rs - rt	001
AND	rd ← rs & rt	010
OR	rd ← rs | rt	011
MULT	rd ← rs * rt	100
XOR	rd ← rs ^ rt	101
JR	PC ← rs	111

- **Immediate Instructions**: این دستورات خود سه نوع هستند:(۱) یا مسئول یک پرش شرطی[1] هستند، (۲) یا برای load و store استفاده می‌شوند (۳) یا اینکه برای انجام دادن یک عملیات با یک مقدار ثابت و ثبات هستند. فرمت این دستورات مانند شکل زیر است:

opcode	rs	rt	immediate
4 bits	3 bits	3 bits	6 bits

لیست این دستورات و opcodeهای آن در زیر آمده است:

Mnemonic	Operation	opcode
ADDi	rt ← rs + SIGN_EXTEND(immediate)	0010
SUBi	rt ← rs - SIGN_EXTEND(immediate)	0011
ANDi	rt ← rs & immediate	0100
ORi	rt ← rs | immediate	0101
SB	MEM[rs + SIGN_EXTEND(immediate)] ← rt	0110
LB	rt ← MEM[rs + SIGN_EXTEND(immediate)]	0111
BEQ	`if` (rt == rs): PC ← PC + SIGN_EXTEND(immediate << 1)	1000
BNQ	`if` (rt != rs): PC ← PC + SIGN_EXTEND(immediate << 1)	1000

در رابطه با این نوع دستورات به نکات زیر توجه کنید:

[1] conditional branch

- دقت کنید که مقدار immediate در دستورات `ANDi` و `ORi` به هیچ وجه sign extend نمی‌شوند.
- اگر به یاد داشته باشید زمانی که در پردازنده میپس پرش نسبی [۲] داشتیم به اندازه‌ی ۲ بیت مقدار immediate را شیفت می‌دادیم چرا که دستورات 4 byte aligned بودند. اما در اینجا از آنجا که دستورات 2 byte aligned هستند باید یک واحد مقدار immediate را شیفت دهید. در دستورات نیز علامت >> به معنای شیفت دادن است.

- **Jump Instructions**: این دستورات برای پرش استفاده می‌شوند. شکل این دستورات به صورت زیر است:

opcode	NOT USED	address
4 bits	5 bits	7 bits

همان طور که مشاهده می‌کنید در اینجا طول آدرس ۷ بیت است چرا که با توجه به طراحی ما و اندازه‌ی حافظه این پردازنده می‌تواند حداکثر ۱۲۸ دستور را در خود ذخیره و اجرا کند. همچنین همان طور که از اسم آن پیدا است بیت‌های NOT USED صرفا بدون استفاده هستند و به عبارتی dont care در نظر گرفته می‌شوند. این سری از دستورات شامل دو دستور زیر هستند:

Mnemonic	Operation	opcode
J	PC $\leftarrow$ address << 1	1110
JAL	R[7] $\leftarrow$ PC + 2, PC $\leftarrow$ address << 1	1111

در نهایت نیز به این موضوع توجه کنید که تمامی مقادیر opcode و funct که نوشته شده‌اند پیشنهادی هستند و در صورت نیاز می‌توانید آن‌ها را تغییر دهید. اما در صورتی که آن‌ها را تغییر دهید حتما در گزارش خود ذکر کنید که دستورات شما چه opcode یا functهایی دارند.

تست پردازنده ۱

برای تست این پردازنده باید تابع فیبوناچی با ورودی ۵ را به پردازنده دهید و نشان دهید خروجی صحیح است.

تست پردازنده ۲ و ۳

دو تست دیگر به انتخاب خود بنویسید. در هر دو تست باید از تعداد زیادی **پرش و عملیات محاسباتی** استفاده کنید تا صحت برنامه شما به صورت کامل مشخص شود. همچنین خروجی تست‌ها باید منطقاً قابل ارزیابی و تحلیل باشند به صورتی که هدف تست را باید در گزارش خود بنویسید. همچنین ورودی و خروجی و روند تست‌های شما باید به صورت کامل و واضح مشخص باشند به صورتی که بتوانید با یک شبه‌کد [۳] تست خود را در گزارش توضیح دهید.

گزارش

برای گزارش این تمرین روند ساخت پردازنده را شرح دهید. همچنین نحوه انجام تست‌ها، شبه‌کدها، کد دستورات ماشین، نحوه انتقال دستورات ماشین به ROM و همچنین خروجی نهایی را باید به صورت کامل در گزارش خود شرح دهید.

[2] relative jump
[3] psudo code