



درس معماری کامپیوتر
نیم سال دوم ۰۴-۰۳
استاد: دکتر اسدی

تمرین سری چهارم

- پرسش‌های خود را در سامانه CW و تالار مربوط به تمرین مطرح نمایید.
- پاسخ سوالات را تایپ نمایید.
- پاسخ تمرین را به صورت یک فایل زیپ با فرمت HW1_401234567.zip آپلود کنید. فایل زیپ باید به صورتی باشد که پس از باز کردن آن بدون هیچ پوشه‌ای فایل‌های زیر با ساختار زیر قرار گرفته باشند:

```
1 .  
2 |-- HW4T_401234567.pdf  
3 |-- practical  
4 |   |-- HW4P_401234567_401234568.pdf  
5 |   |-- schematic.circ  
6 |   |-- verilog  
7 |       |-- circuit  
8 |           |-- main.v  
9 |           |-- ...  
10 |   |-- gates  
11 |   |-- ...  
12 |   |-- toplevel  
13 |   |-- ...
```

- در صورت عدم تطابق فایل آپلود شده با فرمت بالا، تمرین شما تصحیح نخواهد شد.
- پاسخ سوالات تئوری و گزارش تمرین‌های عملی باید به فرمت pdf باشد.
- هر دانشجو می‌تواند حداکثر سه تمرین را با دو روز تأخیر بدون کاهش نمره ارسال نماید.
- تمرینات عملی به صورت گروه‌های دو نفر تحویل داده شود.
- هر دو عضو گروه موظف هستند تمرینات خود را بارگذاری کنند.
- عواقب عدم تطابق بین پاسخ دو عضو گروه برعهده خودشان است.
- تحویل تمرین به صورت انگلیسی مجاز نیست. در صورت تحویل تمرین به صورت انگلیسی (حتی بخشی از تمرین) نمره تمرین موردنظر صفر در نظر گرفته می‌شود.
- در صورت مشاهده تقلب برای بار اول نمره هر دو طرف صفر می‌شود. در صورت تکرار نمره کل تمرینات صفر خواهد شد.
- استفاده از ابزارهایی مانند ChatGPT به منظور ابزار کمک آموزشی مجاز است به شرط آن که به خروجی آن اکتفا نشود.
- توجه شود که پروژه نهایی درس در گروه‌های چهار نفر تحویل گرفته می‌شود.
- سوالات با عنوان اختیاری نمره‌ای ندارند اما جواب دادن به آن‌ها کمک به‌سزایی در یادگیری درس می‌کند.

- (ب) اگر برنامه P1 با ۴۵۳۹ دستور روی ماشین M1 اجرا شود، زمان اجرای برنامه روی پردازنده چقدر خواهد بود؟
- (ج) با توجه به نرخ توزیع دستورات در برنامه P1، در چند درصد از زمان اجرای برنامه روی پردازنده، پردازنده بیکار^۲ است؟ (فرض کنید زمان چرخه ساعت پردازنده به اندازه حداقل مقدار مشخص شده در بخش الف است).
- جدول تأخیرهای اجزای مختلف پردازنده MIPS را به این شکل فرض کنید:

Control	D-Mem	Regs	ALU	Mux	Add	I-Mem	جزء
۲۰	۱۵۰	۵۰	۵۰	۱۰	۴۰	۱۰۰	تأخیر (ps)

با توجه به این جدول به سوالات زیر پاسخ دهید:

۳. (آ) برای هر یک از نوع دستورات عمل‌های زیر، تمامی اجزای سخت‌افزاری (به ترتیب استفاده) که مسیر بحرانی^۳ را تشکیل می‌دهند، فهرست کنید:

۱. R-type

۲. lw

۳. sw

۴. beq

(ب) با استفاده از جدول تأخیرها، تأخیر کل عبور برای هر یک از این چهار نوع دستورات عمل را محاسبه کنید.

(ج) کدام دستور مسیر بحرانی را مشخص می‌کند؟

(د) با توجه به مسیر بحرانی، فرض کنید که زمان چرخه پردازنده برابر با تأخیر مسیر بحرانی است. نرخ کلاک پردازنده را محاسبه کنید.

۴. قصد داریم دستورات زیر را به دستورات سیستم MIPS تک چرخه‌ای خود اضافه کنیم. برای هر کدام از دستورات، تغییرات لازم را در شکل مسیر داده MIPS تک چرخه‌ای لحاظ کرده و رسم کنید. شما تنها اجازه افزودن MUX و گیت‌های منطقی را دارید. همچنین در هر مورد سیگنال‌های کنترلی را نیز تعیین کنید.

$$1. \text{lw } Rt, Rd(Rs) \# \text{Reg}[Rt] = \text{Mem}[\text{Reg}[Rd] + \text{Reg}[Rs]]$$

$$2. \text{addm } Rd, Rt, \text{Offset}(Rs) \# \text{Reg}[Rd] = \text{Reg}[Rt] + \text{Mem}[\text{sign extended offset} + \text{Reg}[Rs]]$$

۵. جدول نرخ توزیع دستورات پردازنده MIPS به صورت زیر است:

نرخ توزیع در برنامه P1	نوع دستور
۷%	Jump
۱۵%	Branch
۲۰%	LOAD
۱۹%	STORE
۳۹%	R-Type

فرض کنید تأخیر بخش Control برابر 30ps، D-Mem برابر 300ps، Regs برابر 80ps، ALU برابر 100ps، MUX برابر 20ps، Add برابر 90ps و I-Mem برابر 250ps است. در صورتی که با تغییراتی بتوان تأخیر ALU را ۱۰ درصد و تأخیر D-Mem را ۱۵ درصد کاهش داد، تأخیر Control، ۵۰ درصد افزایش می‌یابد. در این حالت، میزان Speedup را محاسبه کنید.

۶. بر اساس MIPS تک چرخه ای به سوالات زیر پاسخ دهید.

- (آ) می خواهیم یک دستور جدید که I-type می باشد را اضافه کنیم که به صورت `getpc $rt` می باشد و به این صورت عمل می کند که مقدار `pc` را در ثابت `rt` قرار می دهد. تغییرات لازم را روی مسیر داده ^۴ اعمال کرده و شکل را بکشید و سیگنال ها را هم مشخص کنید.
- (ب) **اختیاری** - حال فرض کنید که تاخیرهای این معماری به صورت خواندن از حافظه: ۶ نانوثانیه و نوشتن در حافظه: ۱۰ نانوثانیه و فایل ثابت ها: ۳ نانوثانیه و ALU: ۴ نانوثانیه و (سایر اجزا دارای تأخیر ناچیز ۱ نانوثانیه هستند). اگر زمان چرخه کلاک تا حد امکان کاهش یابد، فرکانس کلاک چقدر خواهد بود؟
- (ج) **اختیاری** - ما باید بین دو بهینه سازی عملکرد یکی را انتخاب کنیم: ۱. یک واحد ALU جدید (با تأخیر ۳ نانوثانیه) ۲. یک فایل ثابت جدید (با تأخیر ۲ نانوثانیه) کدام گزینه انتخاب بهتری است؟ توضیح دهید.

۷. **اختیاری** - با مجموعه دستورات پردازنده MIPS در اسلایدهای درس آشنا شدید. حال فرض کنید که می خواهیم مجموعه دستورات زیر را به این پردازنده اضافه کنیم. همچنین فرض کنید که پردازنده ما یک پردازنده single-cycle است. همچنین می توانیم یک واحد Adder دیگر برای پیاده سازی دستورات جدید به پردازنده اضافه کنیم. همینطور فرض کنید خانه های حافظه در کلاک پایین رونده نوشته می شوند و قابلیت خواندن و نوشتن همزمان نیز وجود دارد.

جدول ۱:

Instruction	Operation
jal address	Stores PC in \$31 and then go to address
cmpjz rs, lable	if \$t0 == 0 then jump to PC+4 + lable«2
addm rs,rt,imm	rs=(rs)+Mem[rt+imm]
las rs,imm(rt)	Mem[(rt)+imm]=(rs)+Mem[(rt)+imm]

حال مشخص کنید که برای اضافه کردن این دستورات به معماری مجموعه دستورات، باید چه تغییراتی اعمال کنیم و سپس تاخیر هر دستور را مطابق جدول داده شده به دست بیاورید. فرض کنید که هر دستور بر روی پردازنده اصلی بدون اضافه شدن سایر دستورات داده شده پیاده می شود.
توجه: در دستورات پرش فرم دستور به شکل زیر است:

op	address
31-26	25-0

در دستور `cmpjz` فرم دستور به شکل زیر است:

op	rs	rt	imm
31-26	25-21	20-16	15-0

همچنین در دستوراتی که مقدار immediate داریم، فرم دستورات به شکل زیر است:

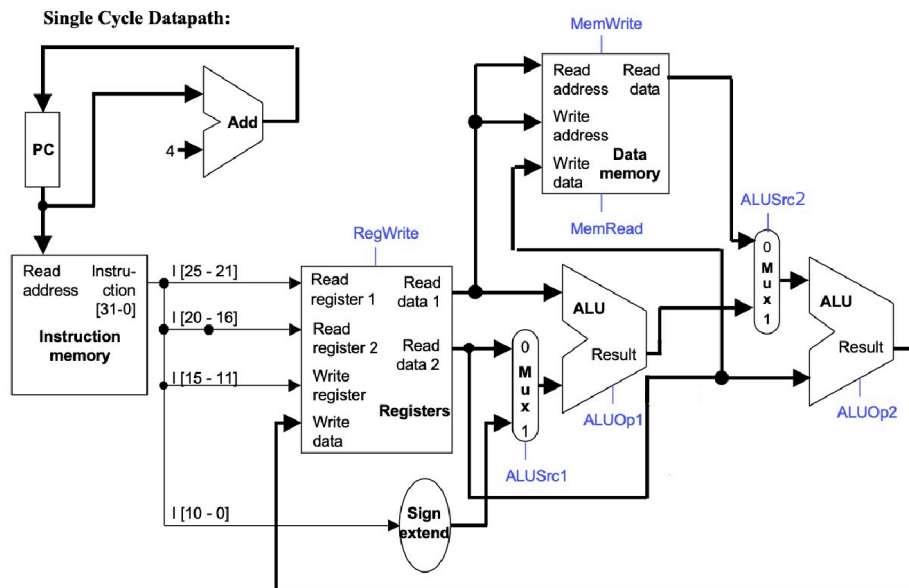
op	rs	rt	imm
31-26	25-21	20-16	15-0

تاخیر گیت AND و OR با هر تعداد ورودی را در صورت نیاز برابر با 5ps بگیرید.

جدول ۲: تاخیر ماژول‌های مختلف

I-Mem	Add	Mux	ALU	Registers	D-Mem	Control	ALU-Control
۲۱۰ps	۷۰ps	۲۰ps	۹۵ps	۸۰ps	۲۲۰ps	۵۰ps	۳۰ps

۸. اختیاری - با توجه به معماری زیر که با معماری تدریس شده در کلاس تفاوت‌هایی دارد، به سوالات پاسخ دهید.



الف) فرض کنید می‌خواهیم یک دستور جدید با نام `addi_st` به این پردازنده اضافه کنیم. این دستور به این صورت عمل می‌کند:

`addi_st (rs), rs, imm # Memory[R[rs]] = R[rs] + imm;`

چه تغییراتی باید در این پردازنده ایجاد شود تا از این دستور پشتیبانی کند؟ همچنین سیگنال‌های کنترلی برای این دستور را نیز مشخص کنید.

ب) حال سه دستور زیر را که توسط این پردازنده قابل اجرا هستند را در نظر بگیرید.

`lw_add rd, (rs), rt # rd = Memory[R[rs]] + R[rt];`
`addi_st (rs), rs, imm # Memory[R[rs]] = R[rs] + imm;`
`sll_add rd, rs, rt, imm # rd = (R[rs] << imm) + R[rt];`

اگر تاخیر هر واحد را طبق جدول زیر در نظر بگیریم، حداقل زمان لازم برای اجرای هر دستور را به دست آورید.

Func. Unit	Latency
Memory	۳ ns
ALU	۴ ns
Register File	۲ ns

تمارین عملی

۱. در این تمرین قصد داریم پردازنده MIPS را به صورت Single Cycle بسازیم. در ساخت این پردازنده لازم است که از ALU ساخته شده در تمرین ۳ استفاده کنید. توجه داشته باشید که در این پیاده‌سازی از ALU به صورت یک واحد ترکیبی (Combinational) استفاده می‌کنیم. بنابراین ALU شما باید به گونه‌ای تغییر کند که بدون کلاک عمل کند (نیازی به پشتیبانی از دستورات ضرب و تقسیم در ALU تغییر یافته نیست)

پردازنده شما باید دارای ۵ مرحله زیر را پیاده‌سازی کند (جدا سازی درست این ۵ مرحله در تمرینات بعدی و پیاده‌سازی pipeline اهمیت دارد):

- Instruction Fetch
- Instruction Decode
- Execution
- Memory Access
- Write Back

فایل قالب schematic.circ دارای قطعات کمکی زیر برای ساخت پردازنده است. با این حال لزومی به استفاده از این قطعات در طراحی خود وجود ندارد.

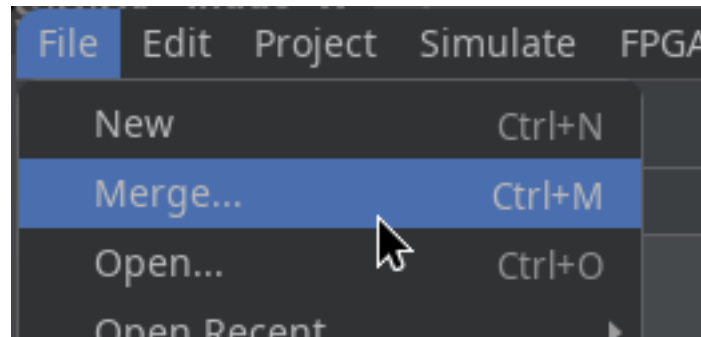
- Instruction Memory
- Data Memory
- Register File
- Clock Generator

پردازنده شما باید از دستورات زیر پشتیبانی کند. برای بررسی opcode مد نظر برای هر دستور، فایل داوری را بررسی کنید:

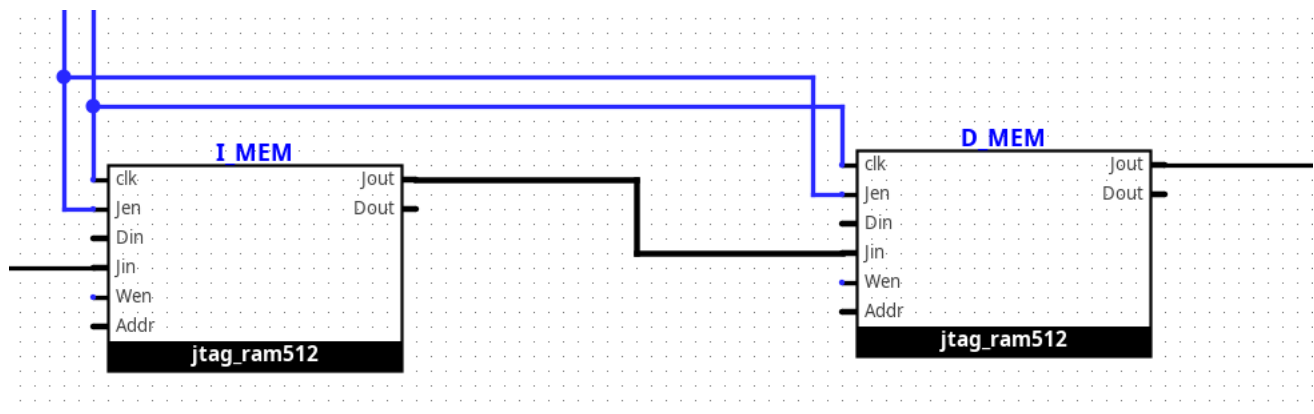
```
1 add $1, $2, $3
2 addi $1, $2, $3
3 sub $1, $2, $3
4 and $1, $2, $3
5 or $1, $2, $3
6 xor $1, $2, $3
7 sll $1, $2, $3
8 srl $1, $2, $3
9 sra $1, $2, $3
```

برای اینکه بتوانیم وضعیت اولیه حافظه برنامه و داده را آماده کنیم، از مکانیزمی به نام JTAG استفاده می‌کنیم. به طور خلاصه JTAG حافظه مدار را به صورت یک شیفت رجیستر بزرگ به هم متصل می‌کند و به همین شکل، دسترسی آسان برای فرایند های اینچنینی مانند خروجی گرفتن و ورودی دادن وضعیت حافظه می‌دهند.

ماژول های حافظه برایتان آماده شده اند، آنها را در مدارتان باید به شکل زیر بارگیری و متصل کنید :



شکل ۱: نحوه استفاده از ماژول حافظه داده شده



شکل ۲: نحوه اتصالات JTAG حافظه

Inputs:

- clk
- rst
- Jin (32 bit)
- Jen

Outputs:

- R1...R32 (32 bit)
- Jout (32 bit)