



تمرین سری چهارم

- تمرین‌های خود را در قالب فایل تایپ‌شده و ذخیره شده به فرمت *PDF* به صورت یک فایل *Zip* با نام *1STDID_HW* که *STDID* شماره‌ی دانشجویی شما است، در صفحه‌ی درس در *CourseWare (CW)* بارگذاری کنید.
- دانشجویان مجاز هستند دو تمرین را حداکثر با دو روز تاخیر ارسال کنند.
- سؤالات خود را صرفاً در فروم مربوطه در *CW* بپرسید.



تمرین سری چهارم

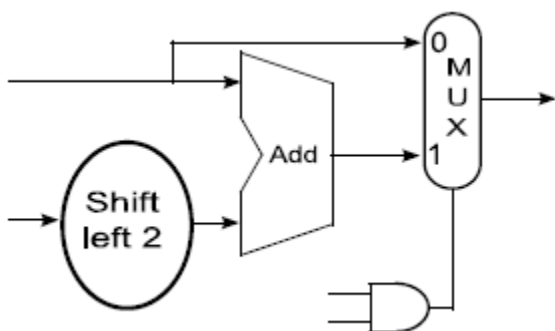
1- *Datapath* یک پردازنده *MIPS* که به صورت *Single-cycle* است را در نظر بگیرید. به سوالات مطرح شده پاسخ دهید.

الف – مقادیر گفته شده برای دستورات داده شده را مشخص کنید.

<i>Instruction</i>	<i>ALUSrc</i>	<i>MemRead</i>	<i>RegDst</i>	<i>RegWrite</i>	<i>ALUResult</i>
<i>addi \$8, &9, -4</i>	1	0	0	1	
<i>lw \$t2, 8(\$at)</i>	1	1	0	1	

ب – اگر بر اثر مشکلی در سخت افزار ما مقدار سیگنال *RegDst* همواره 1 باشد. چه تغییری در عملکرد دستور *addi* داده شده در قسمت قبل ایجاد می شود؟

2- قسمت نشان داده شده در *datapath* پردازنده *MIPS* را در نظر بگیرید. این بخش برای پیاده سازی دستور خاصی به آن افزوده شده است.



الف – این دستور چیست؟ چرا؟ (کار بخش نشان داده شده را شرح دهید)

ب – هدف از قرار دادن واحد شیفت 2 بیتی چیست؟ کاملاً توضیح دهید.

ج – برای اجرای این دستور دو سیگنال کنترلی *Zero* و *Branch* به طراحی افزوده شده است. توضیح دهید به هر کدام چه نیازی است و آیا یکی از آنها کافی است؟

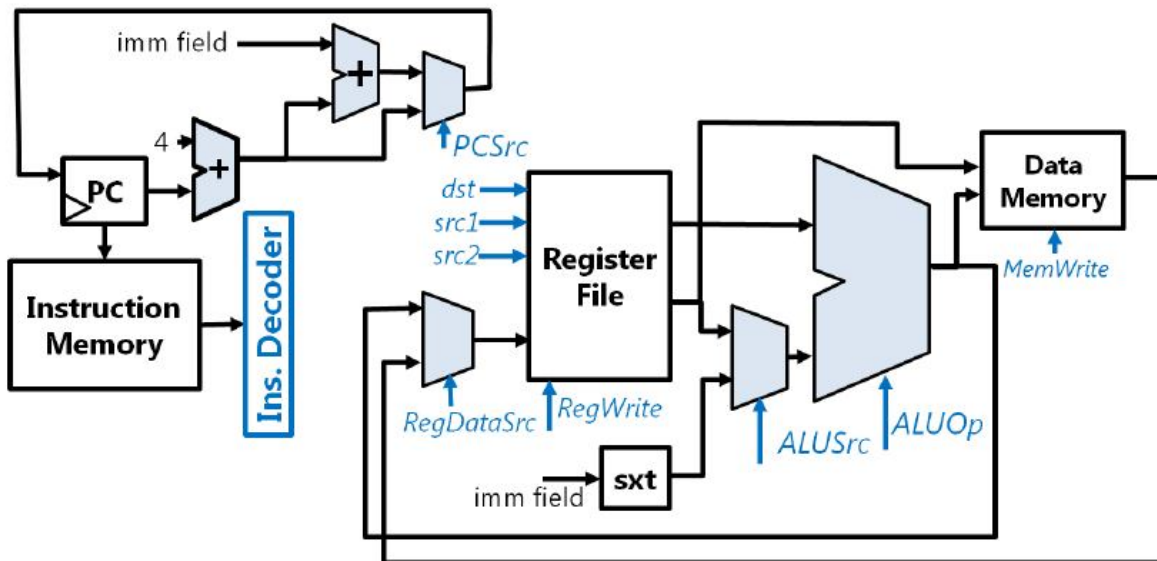
3- شکل ذیل را در نظر بگیرید تغییرات لازم برای اضافه کردن دستورات گفته شده را در مسیر داده و واحد کنترل توضیح دهید و در شکل نمایش دهید.

• *jar* (پرش به آدرس یک رجیستر)

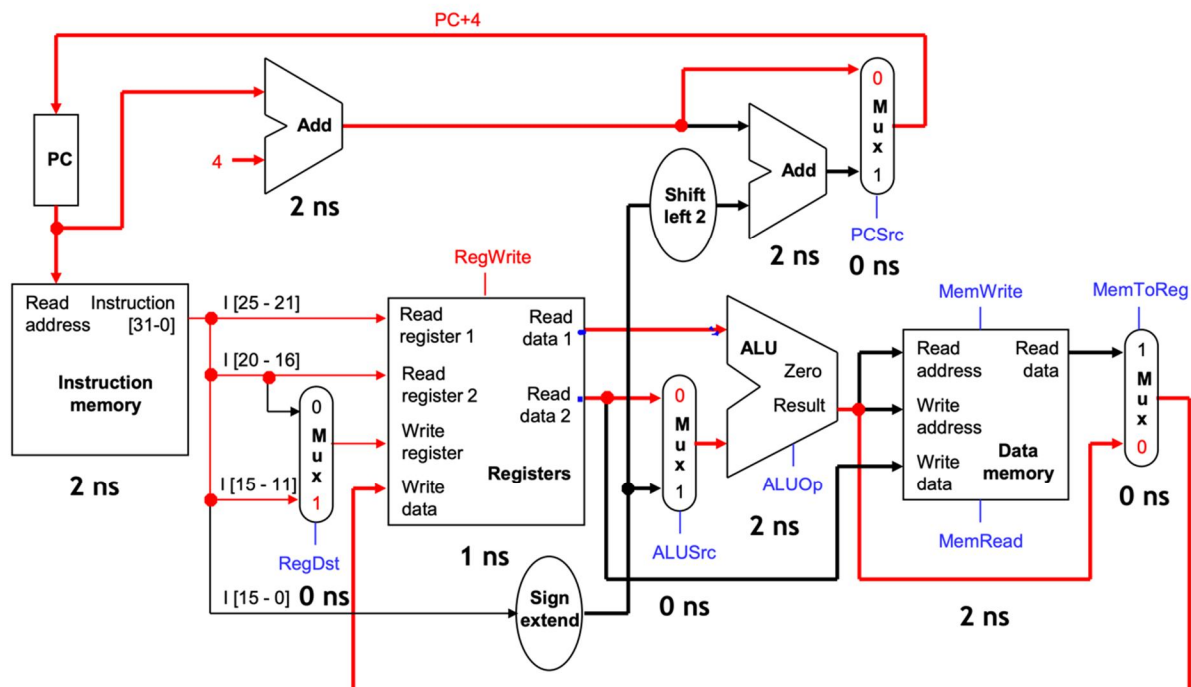


تمرین سری چهارم

- *jal* (کپی آدرس $PC + 4$ در یک رجیستر و پرش به آدرس جدید)



-4 با توجه به شکل داده شده به سوالات زیر پاسخ دهید:



الف) این *datapath* مسیر حرکت چه دستوری را نشان می دهد؟

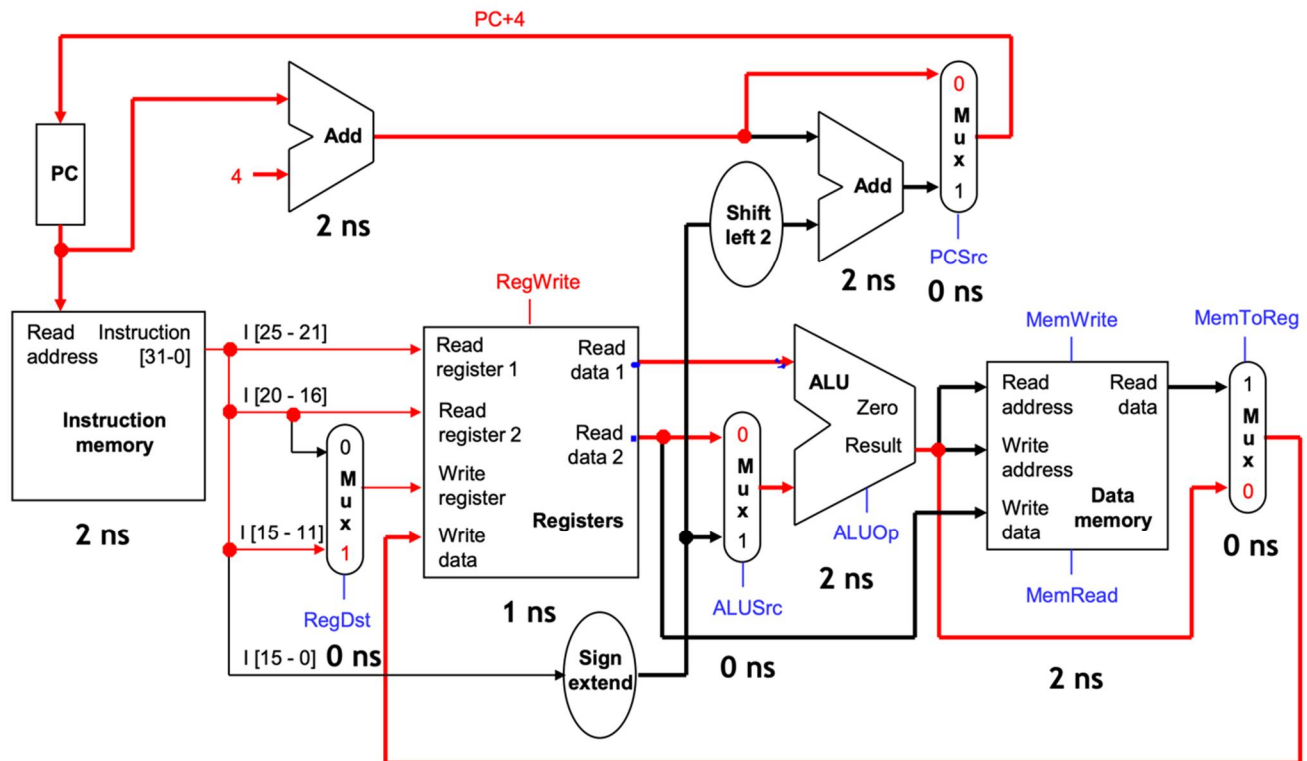
ب) تاخیر اجرای این دستور چقدر می باشد؟



تمرین سری چهارم

ج) با توجه به *datapath* داده شده و معادله های پایداری زمانی بهترین کران بالا را برای *hold-time* و هم چنین با فرض این که *set up-time* برابر $0.3ns$ باشد بهترین کران پایین برای کلاک را بیابید.

5- فرض کنید که *datapath* پردازنده MIPS ما دچار خطا شده است حال بسته به هر مورد به سوالات پاسخ دهید:



الف) سیگنال های کنترلی *RegDst* و *MemToReg* دچار ایراد شده اند و همیشه مقدار صفر به خود می گیرند و نمی توان به آن ها مقدار داد. با وضع موجود کدام دستورات هنوز درست عمل می کنند؟

ب) با توجه به ایراد ایجاد شده در بخش الف اگر تنها بتوانیم مشکل یکی از دو سیگنال را برطرف کنیم با توجه به جدول زیر تعمیر کدام یک مقرون به صرفه تر است؟

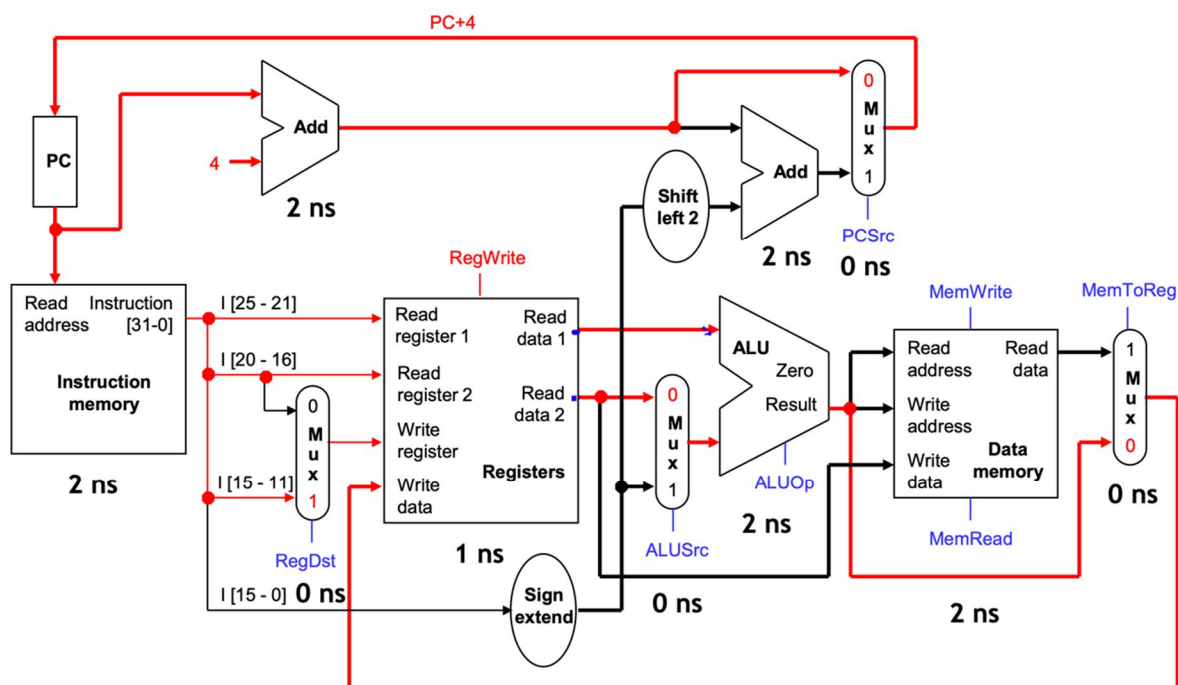
Instruction	Frequency
Arithmetic	48%
Loads	22%
Stores	11%
Branches	19%



تمرین سری چهارم

ج) فرض کنید که سیستم ما دچار خطا شده است و سیگنال *MemToReg* توسط *control unit* به درستی مقداره‌ی نمی شود چگونه می توان بدون ایجاد تغییر در *control unit* با کمترین دست کاری مشکل را برطرف کرد؟

6- تغییرات لازم برای آن که *datapath* زیر بتواند دستور *jal* را پشتیبانی نماید را بنویسید (نمره کامل به پاسخ کامل تعلق می گیرد). سپس سیگنال های کنترلی مربوط به این دستور را نیز مقدار دهی کنید:





تمرین سری چهارم

سوال عملی:

طراحی واحد محاسبه گر PC

در این تمرین هدف طراحی یک واحد محاسبه گر PC می باشد. این واحد که توسط یک شیفت رجیستر مقدار فعلی PC را در خود نگه می دارد، با گرفتن سیگنال های مشخص شده در شکل به عنوان ورودی، مقدار PC بعدی را محاسبه کرده و در لبه ی مثبت کلاک آن را به خروجی می فرستد. این واحد باید توانای بازنشانی سنکرون، $Increment$ یا افزایش PC به اندازه ی یک دستور به جلو، $jump$ و $Branch$ را دارا باشد. شیوه ی محاسبه ی آدرس در حالت های $branch$ و $jump$ مشابه پردازنده ی $MIPS$ است که در جدول زیر آمده است. دقت شود سیگنال $Reset$ در این واحد $Active Low$ است و با صفر شدن آن مقدار PC به صفر تغییر می کند.

$Reset$	CLK	$Jump$	$Branch$	PC
0	↑	X	X	$PC + 4$
1	↑	0	0	0
1	↑	0	1	$PC + 4 + (SE(BranchOffset) << 2)$
1	↑	1	0	$(PC + 4)[31..28]:(JumpAddress) << 2$
1	↑	1	1	X

منظور از SE عملیات $Sign Extend$ است. در نهایت ماژول طراحی شده ی شما باید به صورت زیر باشد. برای تست ماژول طراحی و ساخت یک $Waveform$ نیز الزامیست.

