به نام خدا

دانشکده مهندسی کامپیوتر

معماری کامپیوتر

نیم‌سال دوم ۹۹-۰۰

استاد: دکتر اسدی

تمرین سری ششم

- پاسخ تمرین‌های تئوری را به صورت فایل تایپ شده در فرمت PDF در قسمت مربوطه در سامانه CW بارگذاری نمایید.
- پرسش‌های خود را می‌توانید در فروم ایجاد شده در سایت درس مطرح کنید.
- هر دانشجو می‌تواند حداکثر دو تمرین را با دو روز تاخیر بدون کاهش نمره ارسال نماید.

سوال ۱.

سه تفاوت سخت‌افزاری که میان مسیرداده‌ی[۱] پردازنده‌های تک-سیکلی[۲] و چند-سیکلی[۳] وجود دارد را ذکر کنید و علت وجود این تفاوت‌ها را توضیح دهید.

سوال ۲.

در تمرین قبل stuck-at-0 و stuck-at-1 را برای پردازنده تک-سیکلی میپس بررسی کردید. حال در این تمرین بررسی کنید که برای هر یک از سیگنال های زیر stuck-at-0 و stuck-at-1 رخ دهد، چه مشکلاتی پیش خواهد آمد؟

RegWrite, MemRead, MemWrite, IRWrite, PCWrite, PCWriteCond

سوال ۳.

با توجه به اینکه تاخیر هر یک از بخش‌های یک پردازنده مطابق جدول زیر هستند، به سوالات زیر پاسخ دهید.

(*setup time = hold time = 0.15 ns*)

	Instruction-cache	Register-Read	ALU	PC update	Data-cache	Register-Write
Delay	*3 ns*	*2 ns*	1.5 *ns*	0.5 *ns*	3 *ns*	2.5 *ns*

الف) اگر این پردازنده تک-سیکلی باشد، مقدار Clock Cycle Time این پردازنده را بدست آورید.

ب) اگر این پردازنده چند-سیکلی باشد، قسمت‌های مختلف را به طریقی دسته‌بندی کنید تا مقدار Clock Cycle Time حداقل باشد، سپس زمان اجرایی هر یک از دستورات beq، lw، sw و R-type در این پردازنده را بدست آورید.

پ) آیا می‌توان اظهار داشت که همواره پردازنده‌ی Multi-Cycle از پردازنده‌ی Single-Cycle (با فرض یکسان بودن معماری‌ها) عملکرد بهتری در اجرای برنامه‌ها دارد؟ با ذکر مثال دلیل خود را توضیح دهید.

[۱] Datapath
[۲] Single-Cycle
[۳] Multi-Cycle

سوال ٤.

سیگنال‌های کنترلی دستورهای زیر را در یک پردازنده چند مرحله‌ای مشخص کنید.

الف) Jump 1023

ب) Addi $r4, $r1, 123

سوال ٥.

فرض کنید بخواهیم یک دستور فرضی jump memory را به پردازنده چند-سیکلی میپس اضافه کنیم.

Jmem rt, offset(rs)

این دستور جزء دستورات I-type است و فرمت آن مطابق زیر می‌باشد:

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

اگر این دستور به صورت زیر عمل کند:

Memory[R[rs]+offset] = PC+4;
PC = Memory[R[rt]];

با افزودن کمترین سخت‌افزار ممکن (سیم‌ها و مالتی پلکسرها) تغییرات لازم را در مسیر داده ایجاد کنید تا این دستور پشتیبانی شود. FSM را نیز تکمیل کنید.

سوال ٦.

می خواهیم دستور jal را به پردازنده ی چند-سیکلی میپس اضافه کنیم. با کمترین سخت افزار ممکن تغییرات لازم را در این پردازنده ایجاد کنید و FSM را نیز تکمیل کنید.

سوال ٧.

قصد داریم پشتیبانی از دستور ldi را به پردازنده Multicycle اضافه کنیم. شکل زیر را به عنوان مرجع در نظر بگیرید. روند اجرای دستور و سیگنال‌های کنترلی مربوطه برای اجرای درست این دستور را مشخص نمایید. همچنین مجاز هستید در صورت نیاز تغییراتی در مسیرداده‌ی داده شده انجام بدهید.

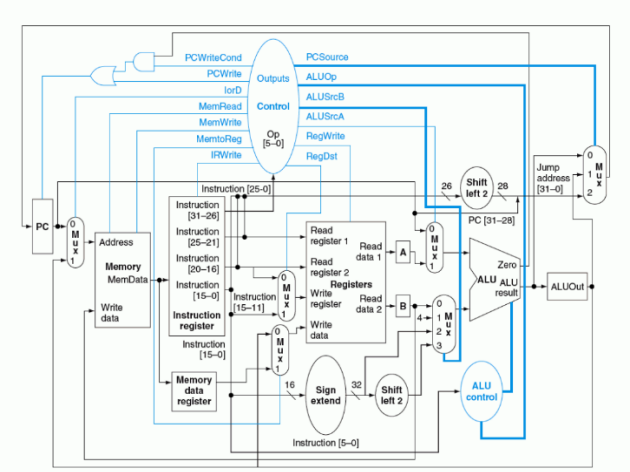
فرمت دستور ldi مشابه دستور زیر است

ldi $r0 IMM

و اجرای آن را می‌توان معادل دستور \lr{RTL}\ زیر دانست:

$r0 <- MEM[MEM[(Base,IMM)]]

در اصل این دستور مانند است که براساس Immediate داده شده یک مقدار را از روی حافظه بار کرده و سپس از آن مقدار به عنوان آدرس برای بار کردن یک مقدار جدید استفاده کنیم.



سوال ۸

فرض کنید که در پردازنده Multicycle شکل بالا، از Register Fileای استفاده کرده‌ایم که فقط قابلیت خواندن همزمان از روی یک ثبات را دارد. چه تغییراتی باید در این دیتاپس و FSM آن اعمال کنیم تا همچنان به درستی کار کند.

سوال عملی:

در این سوال می خواهیم یک محاسبه‌گر PC را طراحی کنیم. از یک شیفت ثبات برای نگه‌داری مقدار فعلی PC استفاده می‌کنیم.

این واحد محاسبه‌گر موارد زیر را به عنوان ورودی می‌گیرد:

- jumpAddress[25:0]
- BranchOffset[15:0]
- سیگنال های تک بیتی reset, clock, jump, branch

و به عنوان خروجی، مقدار بعدی PC که ۳۲ بیتی است را در لبه ی مثبت کلاک به خروجی می‌فرستد.

جدول زیر نحوه محاسبه PC را با توجه به سیگنال های ورودی نشان می‌دهد:

Reset	Jump	Branch	PC
0	X	X	0
1	0	0	PC + 4
1	0	1	PC + 4 + (SignExtend(BranchOffset) << 2)
1	1	0	(PC + 4)[31:28]:(JumpAddress << 2)[27:0]
1	1	1	X

قابل توجه است که:

- این واحد حساس به لبه‌ی مثبت کلاک است.
- سیگنال Reset نیز Active low است.

برای آزمون این واحد، طراحی و ساخت یک waveform و ارزیابی تمامی سیگنال های کنترلی الزامی است.