

به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخ نامه بنویسید.
- ۲- کل پاسخ تمرینات را در قالب یک فایل pdf با شماره دانشجویی خود نام گذاری کرده در سامانه CW بارگذاری کنید.
- ۳- این تمرین ۶۰ نمره دارد که معادل ۰,۶ نمره از نمره کلی درس است.
- ۴- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر **کل نمره** این تمرین را از دست خواهند داد.

۱- (۲۰ نمره) فرض کنید می‌خواهیم دستورات مقابل را روی معماری شکل زیر اجرا کنیم.

```
a: 0xAC620014 = 101011 00011 00010 0000000000010100
b: 0x0082082A = 000000 00100 00010 00001 00000 101010
```

الف- کد اسمبلی، معادل هر دستور را مشخص کنید.

```
a: 0xAC620014    sw $2,20($3)
b: 0x0082082A    slt $1,$4,$2
```

ب- خروجی، واحد عملیات sign-extend دستور a و خروجی، واحد عملیات Shift left 2 دستور b را مشخص کنید.

[illegible]

ج- مقادیر ورودی واحد کنترل ALU را برای هر دستور مشخص کنید.

```
a: ALUop=00      Function bits=010100
b: ALUop=10      Function bits=101010
```

د- مقدار جدید PC پس از اجرای هر دستور چند خواهد بود؟ مسیری را که این مقادیر از آن به دست می‌آید مشخص کنید.

برای هر دو دستور PC+4 است.

فرض کنید مقادیر تمام خانه‌های حافظه صفر است. همچنین در زمان ابتدای اجرای هر دستور، مقادیر داخل ثبات‌ها به صورت زیر است. برای دستورات a و b به طور جداگانه به سوالات زیر پاسخ دهید.

	R0	R1	R2	R3	R4	R5	R6	R8	R12	R31
a	0	-1	2	-3	-4	10	6	8	2	-16
b	0	256	-128	19	-32	13	-6	-1	16	-2

ه- مقادیر خروجی، هر کدام از ماتریکس‌های این معماری در زمان اجرای این دستورات را مشخص کنید.

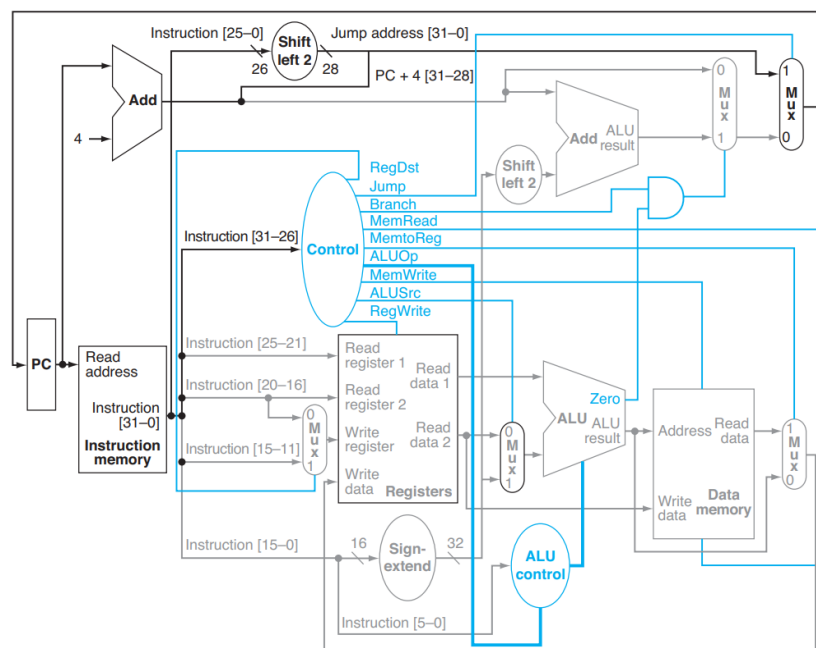
	RegDst	ALUSrc	MemtoReg	PCSrc	Jump
a	2 or 0 (RegDst is \times)	20	\times	PC+4	PC+4
b	1	-128	0	PC+4	PC+4

و- ورودی‌های ALU و دو واحد Add داخل معماری را مشخص کنید.

	ALU	Add (PC + 4)	Add (Branch)
a	-3 and 20	PC and 4	PC+4 and 20×4
b	-32 and -128	PC and 4	PC+4 and 2090×4

ز- مقادیر ورودی‌های مختلف Register File (شامل Read Register 1, Read Register 2, Write Register, Write Data, و RegWrite) را مشخص کنید.

	Read Register 1	Read Register 2	Write Register	Write Data	Regwrite
a	3	2	×	×	0
b	4	2	1	0	1



۲- (۲۰ نمره) فرض کنید بخواهیم دستور for را به میپس اضافه کنیم، این دستور به این شکل کار می‌کند که محتوای Rt را از Rs کم کرده و نتیجه را در Rs می‌نویسد و اگر حاصل صفر نبود به صورت نسبی به مقدار ثابتی که در دستور است پرش می‌کند.

الف- این دستور از چه نوعی است؟

ب- شکل سوال قبل و جداول زیر را طوری تغییر دهید که امکان اجرای این دستور را داشته باشیم.

پاسخ:

الف- دستور از نوع I-type است، چون به دو عملوند از نوع ثبات و یک عدد ثابت نیاز دارد.

ب- در این دستور طبق روال سایر دستورات شماره دو ثبات Rs و Rt به فایل ثبات داده می‌شود و محتوای آنها وارد ALU می‌شود. بنابراین باید $ALUSrc=0$ باشد. چون ALU باید عمل تفریق را انجام دهد، می‌توانیم $ALUOp$ را برابر با 01 قرار بدهیم. خروجی Zero باید با یک سیگنال کنترلی جدید (مثلا به نام BNE) AND شود و با سیگنال PCSrc فعلی OR شود و وارد مالتی‌پلکسر بالا سمت راست شود. بنابراین در صورتی که حاصل تفریق صفر نباشد، مقدار Offset با $PC+4$ جمع خواهد شد و بنابراین یک پرش انجام خواهد گرفت.

از طرفی، خروجی ALU باید در Rs نوشته شود، بنابراین باید یک مالتی‌پلکسر به ورودی Write register در ثبات فایل اضافه کنیم که امکان انتخاب بیت‌های ۲۱ تا ۲۵ را هم بدهد. این جا هم به یک سیگنال کنترلی جدید نیاز داریم (مثلا به نام RegDst2). مقدار سایر سیگنال‌های کنترلی هم به صورت زیر است:

$RegDst=\times$, $Jump=Branch=MemRead=MemWrite=Mem2Reg=0$, $RegWrite=1$

Instruction opcode	ALUOp	Instruction operation	Func field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	and	0000
R-type	10	OR	100101	or	0001
R-type	10	set on less than	101010	set on less than	0111

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUOp0
R-format	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

۳- (۱۰ نمره) در مدارهای مجتمع، گاهی نقص‌های موجود در سیلیکون باعث گیر کردن مقدار یک سیگنال روی صفر یا یک می‌شود.

در هر مورد توضیح دهید با گیر کردن سیگنال، اجرای کدام دستورات دچار مشکل خواهند شد؟ توضیح دهید.

الف- سیگنال RegDst روی مقدار صفر گیر کند.

ب- سیگنال ALUOp0 روی مقدار یک گیر کند.

ج- سیگنال RegWrite متصل به فایل ثابت روی مقدار صفر گیر کند.

پاسخ:

الف- اگر RegDst=0 باشد، در همه دستورات بیت‌های ۱۶ تا ۲۰ (ثبات Rt) به عنوان شماره ثبات مقصد تعیین می‌شود که یعنی در همه دستوراتی که WriteBack داریم، پاسخ روی Rt نوشته می‌شود، یعنی همه دستورات R-type اشتباه اجرا می‌شوند و فقط دستور lw و سایر دستوراتی که WriteBack ندارند، درست اجرا می‌شود.

ب- اگر ALUOp0=1 باشد، دستور beq درست اجرا می‌شود اما اجرای سایر دستورات مختل می‌شود.

ج- اگر RegWrite=0 هیچکدام از دستوراتی که نیاز به WriteBack دارند درست اجرا نمی‌شوند که این شامل همه دستورات R-type و دستور lw می‌شود.

۴- (۱۰ نمره) تاخیرهای زیر را برای بخش‌های مختلف شکل سوال یک در نظر بگیرید:

I-Mem	Add	Mux	ALU	Regs	D-Mem	Control
220ps	75ps	20ps	95ps	70ps	240ps	45ps

فرض کنید ۸٪ دستورات از نوع پرش غیرشرطی، ۱۱٪ پرش شرطی، ۱۶٪ از نوع load، ۱۵٪ از نوع store و بقیه R-type هستند. در صورتی که با تغییراتی بتوان تاخیر ALU را به ۸۵ و تاخیر D-Mem را به ۱۸۰ پیکوثانیه کاهش داد، و تاخیر کنترل به ۵۵ پیکوثانیه برسد، میزان تسریع چقدر خواهد بود؟ (تاخیرها را برای نوشتن و خواندن برابر مقدارهای گفته شده در نظر بگیرید، تاخیر بخش‌های دیگر را ناچیز فرض کنید).

پاسخ:

جداول زیر مراحل اجرای دستورات را نشان می‌دهد و زمان اجرای هر گروه در دو ستون آخر محاسبه شده است:

							Before	After
jump	I-Mem	Control	Mux				285	295
branch	I-Mem	Regs	Mux	ALU	Mux	Mux	445	435
load	I-Mem	Regs		ALU	D-Mem	Mux	715	645
store	I-Mem	Regs		ALU	D-Mem		625	555
R-type	I-Mem	Regs	Mux	ALU	Mux	Regs	495	485

$$speedup = \frac{8 \times 285 + 11 \times 445 + 16 \times 715 + 15 \times 625 + 50 \times 495}{8 \times 295 + 11 \times 435 + 16 \times 645 + 15 \times 555 + 50 \times 485}$$