

درس معماری کامپیوتر

تمرین سری چهارم

۱۴۰۴/۰۲/۰۱

متن باقری

۴۰۲۱۰۵۷۲۷

۱) (آ) ۱. MSB خروجی ALU (alu برای این دستور مانند دستور beq باید sub انجام دهد) را به یک گیت نات متصل کرده (حاصل آن را با x نشان می‌دهیم) و یک MUX بعد از MUX PCSrc اضافه می‌کنیم که بیت سلکتور آن x است، به ورودی صفر آن خروجی mux قبلی و به ورودی یک آن خروجی adder بعد از shift left 2 را متصل می‌کنیم.

۲. برای این دستور نیز ALU باید sub انجام دهد. MSB خروجی ALU را با zero extend به ۳۲ بیت می‌رسانیم. این مقدار (x) باید داخل rd قرار بگیرد، برای همین قبل از write data registers یک mux قرار داده که مقدار ورودی صفر آن همان مقدار قبلی و مقدار ورودی یک آن x است. بیت سلکتور این mux در صورتی که opcode دستور slt باشد باید یک شود.

۳. بعد از این دستور، ۱۶ بیت پایین‌تر تبدیل به صفر می‌شوند. Writer register رجیستر فایل rd است. Imm16 را از instruction، zero extend کرده و به ALU می‌فرستیم، برای این کار بعد از mux ALUSrc یک mux دیگر قرار می‌دهیم که ورودی صفر آن به خروجی mux قبلی متصل است و ورودی یک آن imm16 است که از instruction آمده. بیت سلکتور آن نیز در صورتی که opcode دستور lui باشد باید یک شود. برای اجرای این دستور ALU باید مقدار imm16 ورودی را ۱۶ بیت به چپ شیفت بدهد.

active high (ب)

Signal	bge	slt	lui
RegWrite	0	1	1
ALUSrc	0	0	1 (or don't care)
ALU operation	sub	sub	left shift 16
MemWrite	0	0	0
MemRead	0	0	0
MemtoRe	1 (or don't care)	1 (or don't care)	1
PCSrc	0 (or don't care)	0	0
Mux PCSrc 2	1	-----	-----
Mux before Write data in RF	-----	1	-----
Mux ALUSrc 2	-----	-----	1

(۲) آ 8ns چون بیشترین زمان مورد نیاز برای اجرای یک دستور 8ns است و باید همه دستورات را بتوان در یک کلاک اجرا کرد.

(ب)

$$8 \times 4539 = 36312 \text{ ns} = 36.312 \mu\text{s}$$

(ج)

$$44\% \times (8 - 6) + 24\% \times (8 - 8) + 12\% \times (8 - 7) + 18\% \times (8 - 5) + 2\% \times (8 - 2) = 1.66 \quad 1.66 \div 8 = 20.75\%$$

$$\frac{8 - 6.34}{8} = 20.75\%$$

۳) آ و ب) پس از حل تمرین، TAها گفتند برای دستوراتی که در write back RF دارند، تاخیر RF را مجدداً در انتها

اضافه کنید. این تغییرات را با رنگ قرمز اضافه کردم.

R-type:

pc: add → mux

$$40 + 10 = 50 \text{ ps}$$

Imem → RF → mux → ALU → mux

$$100 + 50 + 10 + 50 + 10 = 220 \text{ ps}$$

$$220 + 50 = 270 \text{ ps}$$

lw:

pc: add → mux

$$40 + 10 = 50 \text{ ps}$$

Imem → RF → ALU → Dmem → mux

$$100 + 50 + 50 + 150 + 10 = 360 \text{ ps}$$

$$360 + 50 = 410 \text{ ps}$$

sw:

pc: add → mux

$$40 + 10 = 50 \text{ ps}$$

Imem → RF → ALU → Dmem

$$100 + 50 + 50 + 150 = 350 \text{ ps}$$

at this point the sign extended value is right at the mux and the mux selector is ready too (cause control takes only 20 ps)

beq:

Imem \rightarrow RF \rightarrow mux \rightarrow ALU (now the zero flag is ready)

$$100 + 50 + 10 + 50 = 210 \text{ ps}$$

pc \rightarrow Imem \rightarrow sign-extend \rightarrow shesfit left 2 \rightarrow add \rightarrow (the zero flag is needed here) mux

$$100 + 0 + 0 + 40 = 140 \text{ ps}$$

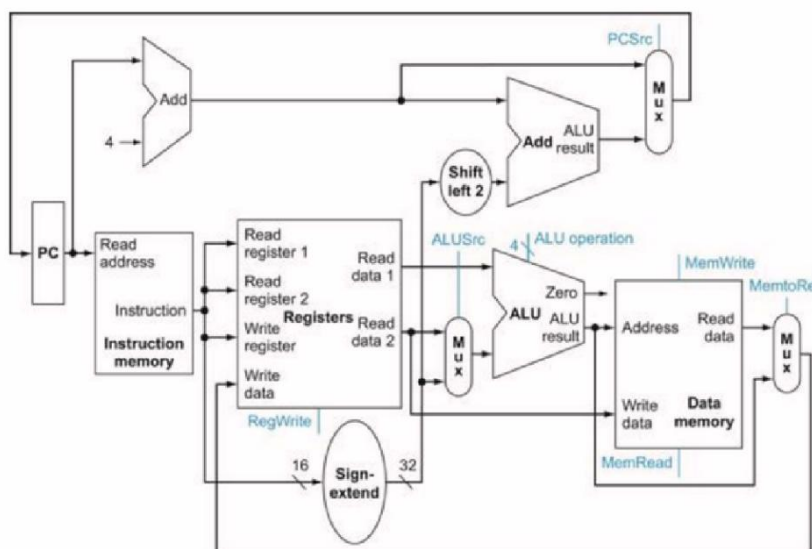
So we need to wait 210 ps for zero flag to be ready, then the mux takes 10 ps more. So in total it takes **220 ps**

ج) دستور lw چون بیشترین زمان را نیاز دارد. $360 \text{ ps} < 410 \text{ ps}$

(د)

$$\frac{1}{360 \times 10^{-12}} = 2.78 \text{ GHz}$$

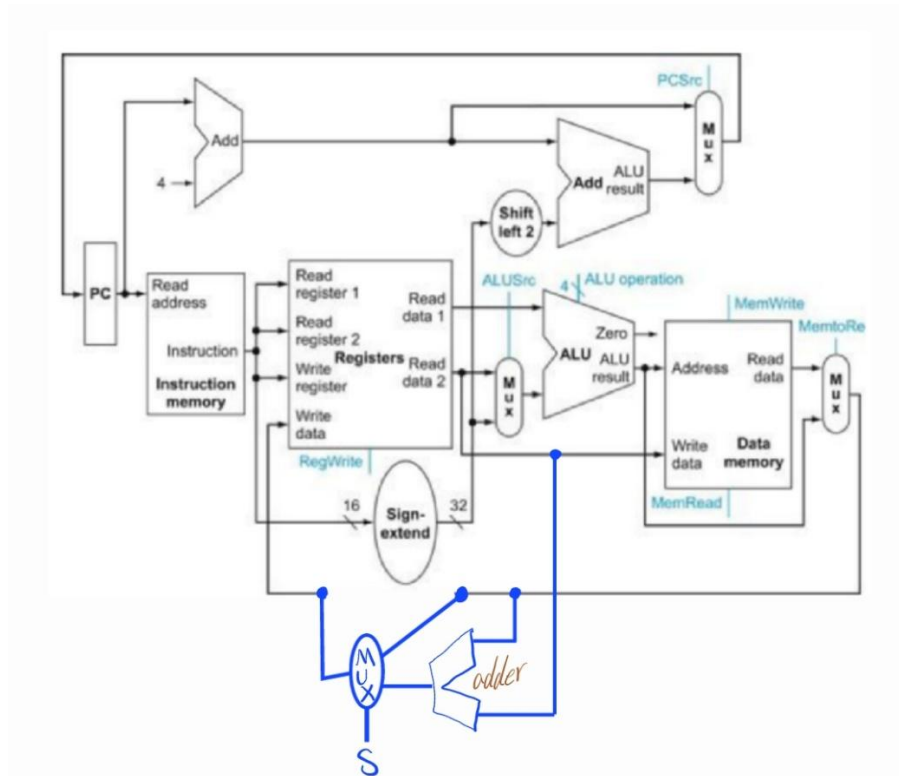
$$\frac{1}{410 \times 10^{-12}} = 2.44 \text{ GHz}$$



از مدار شکل بالا برای این دستور استفاده می‌کنیم. دو رجیستر rd و rs وارد ALU شده و با هم جمع می‌شوند. حاصل به data memory address می‌رود. از مموری خوانده شده و در رجیستر rt در RF نوشته می‌شود. سیگنال‌ها:

فرض کنید اگر سیگنال MUX select برابر با صفر باشد، مقدار بالایی انتخاب می‌شود.

Signals	
RegWrite	1
ALUSrc	0
ALU operation	add
MemWrite	0
MemRead	1
MemtoRe	0
PCSrc	0



با مداری که قبلا داشتیم مقدار $Mem[\text{sign extended offset} + \text{Reg}[Rs]]$ را به دست آورده و سپس به کمک

adderی که اضافه کردیم، آن را با $\text{Reg}[Rt]$ جمع کرده و در $\text{Reg}[Rd]$ می‌نویسیم.

فرض کنید اگر سیگنال MUX select برابر با صفر باشد، مقدار بالایی انتخاب می‌شود.

Signals	
S	1
RegWrite	1
ALUSrc	1
ALU operation	add
MemWrite	0
MemRead	1
MemtoRe	0
PCSrc	0

(۵) آ از آنجا که معماری که در حال بررسی آن هستیم single-cycle است و می‌دانیم دستور lw بیشترین زمان را نیاز دارد و تعیین کننده clock rate است، کافی است lw را بررسی کنیم. در سوال ۳ مسیر آن را یافتیم:

$\text{Imem} \rightarrow \text{RF} \rightarrow \text{ALU} \rightarrow \text{Dmem} \rightarrow \text{mux}$

$$250 + 80 + 100 + 300 + 20 = 750 \text{ ps}$$

$$750 + 80 = 830 \text{ ps}$$

After optimization:

$$250 + 80 + 90 + 255 + 20 = 695 \text{ ps}$$

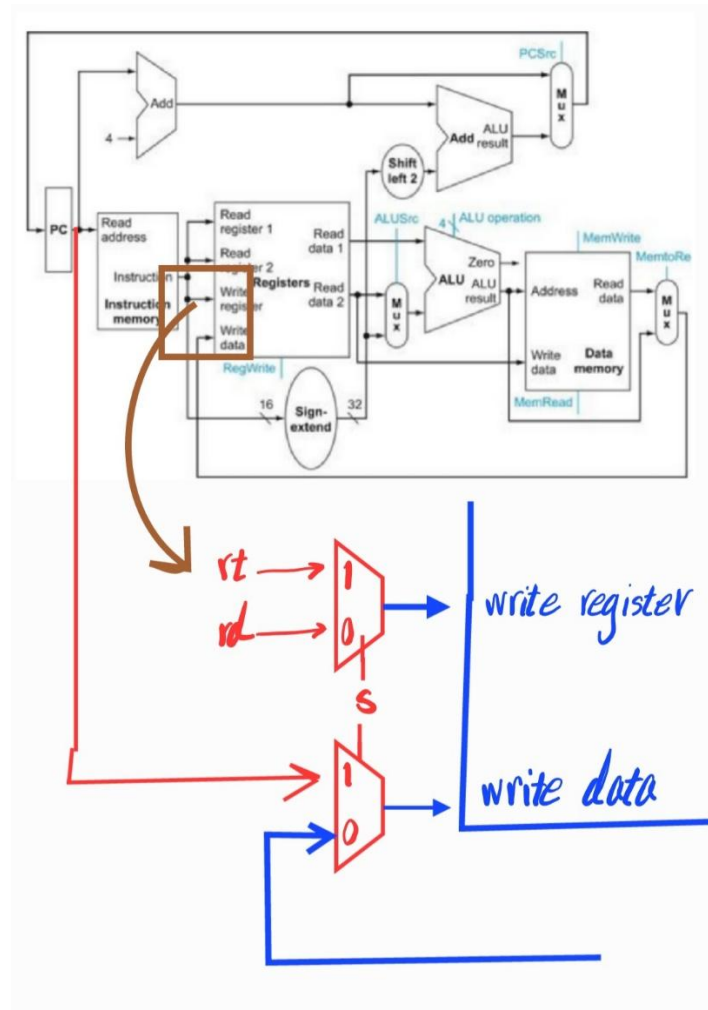
$$695 + 80 = 775 \text{ ps}$$

$$\text{speedup} = 750 / 695 = 1.07913669$$

$$830 / 775 \approx 1.07$$

بنابر این حدود ۸ درصد افزایش سرعت داریم. توجه داریم که تاخیر control با اینکه به ۴۵ps می‌رسد، باعث افزایش تاخیر کلی دستور نمی‌شود.

(۶) آ بخشی از مدار که در آن تغییر ایجاد می‌شود را بزرگ نمایی کردیم و اجزایی که جدید اضافه می‌شوند را با رنگ قرمز نمایش دادیم. سیگنال کنترل دو mux اضافه شده (s) در صورتی که opcode دستور با getps برابر باشد یک می‌شود و در غیر این صورت صفر است:



Signals	
S	1
RegWrite	1
ALUSrc	don't care
ALU operation	don't care
MemWrite	0
MemRead	0
MemtoRe	don't care
PCSrc	0