به نام خدا

معماری کامپیوتر تمرین 5

جناب آقای دکتر اسدی

سارا آذرنوش

98170668

1)

با توجه به مطالب گفته شده در کلاس mux و سیگنال قابلیت حذف ندارد و برای جایگذاری با توجه به جدول میتوان از معکوس Regdest و memtoreg به جای یکدیگر استفاده کرد (هر یک تنها برای دستورات lw و r-format با مقادیر معکوس استفاده میشوند) و یا برای regdest از aluop1 و برای memtoreg از memread استفاده کرد (اگر تغییر ایجاد شود و sw و beq تداخل ایجاد نکنند).

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUp0
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

2)

در صورتی که سیگنال در یکی از مقادیر گیر کند و تنها آن مقدار باشد دستوراتی که با مقدار دیگر هستند کار نمیکنند و با توجه به جدول مقادیر دستورات زیر کار نمیکند.

Signal	Stuck at zero	Stuck at 1
Regwrite	کار نمیکنند lw و R_format	کار نمیکنندbeq و Sw
Aluop0	کار نمیکندbeq	کار نمیکنند lw و R_format و Sw
Aluop1	کار نمیکندR_format	کار نمیکنند lw و beq و Sw
Branch	کار نمیکندbeq	کار نمیکنند lw و R_format و Sw
Memread	کار نمیکندlw	کار نمیکنند beq و R_format و Sw
Memwrite	کار نمیکندsw	کار نمیکنند lw و R_format و beq

3)

الف) با استفاده از کارت متوجه میشویم000111 برای دستور bgtz است

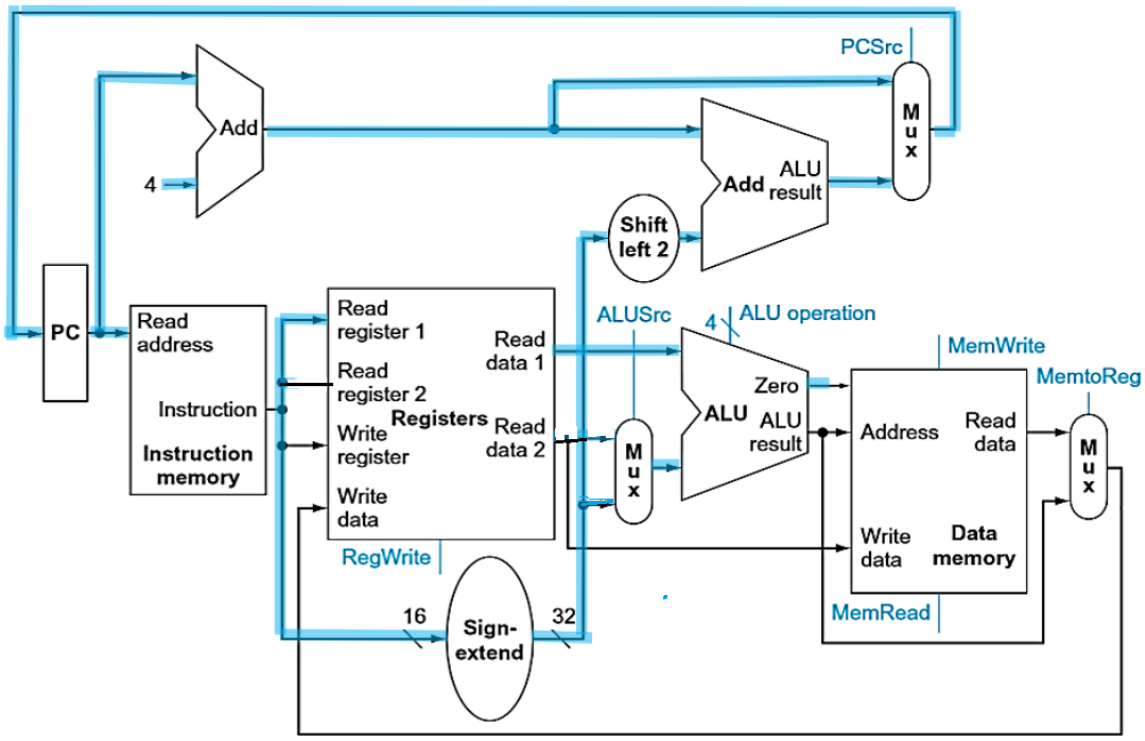
Branch to target if $rs >=0

Bgtz rs rt=0 label

000111 01001 00000 1111101011111010

R9 = 10 != 0

ب)



ج) در دستور bgtz مقدار داده شده در دو رجیستر را خوانده و اگر برابر بود به آدرس میرود اما در این دستور مقدار یکی از رجیسترهای داده شده با 0 برابر باشد به آدرس میرود بنابراین مانند beq است با این تفاوت که مقدار رجیستر دوم را نمیخواند و همان 0 را مقایسه میکند. اگر aluSrc = 1 و دیگر سیگنال ها مانند beq باشد میتوان دستور bgtz را اجرا کرد.

Instruction	RegDst	ALUSrc	Memto-Reg	Reg Write	Mem Read	Mem Write	Branch	ALUOp1	ALUp0
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

د)

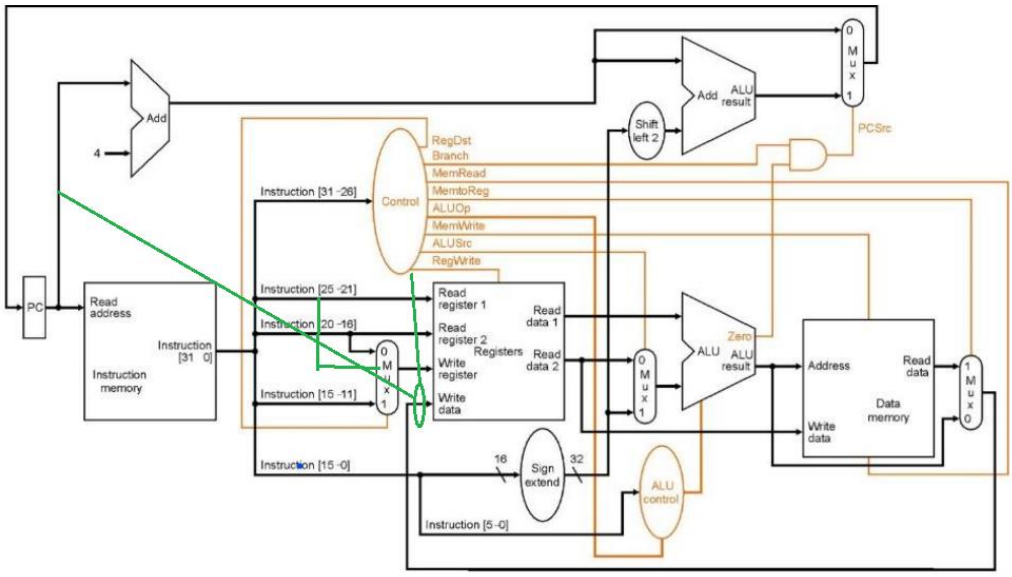
```
if [Rs] < 0 then PC ← [PC] + 4 + ([I15]14 || [I15..0] || 02)
else PC ← [PC] + 4
```

```
0 + 100 + 11111111111111111110101111101000 =
11111111111111111110101111101100
```

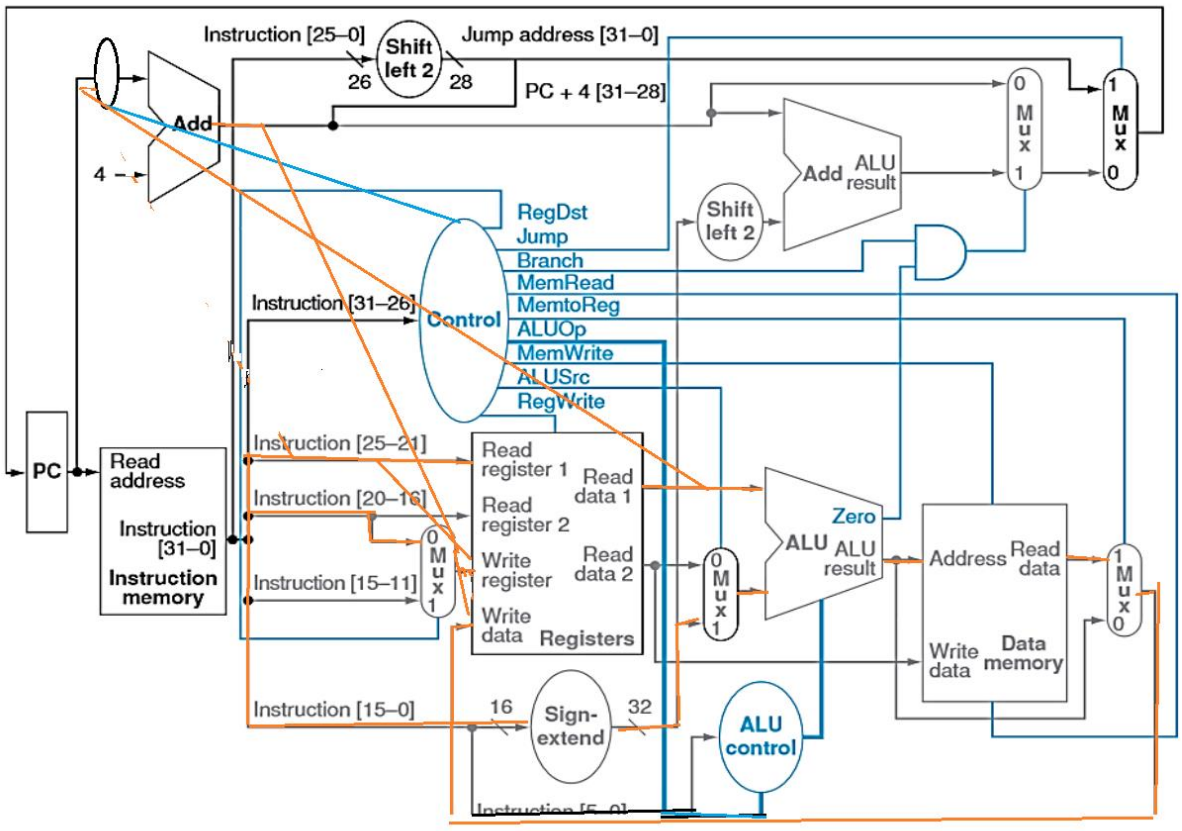
4)

6 بیت op 5 بیت rs و بقیه 0

ادرس rs را میگیرد و در بیت 21 تا 26 است و مقدار pc را در آن مینویسد بنابراین ماکس ورودی Wr نیاز به ماکس 3 ورودی دارد و سیگنال ورودی 2 بیتی میشود و برای مثال عدد 2 را برای این دستور درنظر میگیریم و مقدار pc را در wd مینویسد و برای تداخل یک ماکس قرار میدهیم و یک سیگنال کنترلی تا مقدار خروجی از dm و pc را تمایز دهد.



5)



در ابتدا دستور lw را همانند دیتاپس گفته شده و نمایش داده شده انجام داده سپس دستورaddi از دستورات i-format را نمایش میدهد در ابتدا برای دستور lw مقدار ثبات rs را خوانده سپس با مقدار اکستند شده 32 بیتی imm جمع میکنیم و آن را به عنوان آدرس به dm داده و مقدار خوانده شده را در ادرس داده شده در rf میریزیم. برای addi نیز مقدار خوانده شده در ابتدا از رجیستر با ادرس rs را با 4 جمع میکنیم برای اینکار از ادر pc استفاده میکنیم و یک ماکس قرار میدهیم تا تعیین کند مقدار pc یا مقدار خوانده شده جمع گردد. و سپس مقدار جمع شده را در همان رجیستر rs مینویسیم.

ب)

خیر rf باید این قابلیت را داشته باشد که م در دو جا در rt و در rs بنویسد و هم اکنون تنها قابلیت نوشتن در یک مکان را دارد و باید تغییر پیدا کند برای مثال برای اینکار میتوان از دو ورودی برای نوشتن استفاده کرد چون هم زمان است از ماکس نمیتوان استفاده کرد و باید ورودی دیگری در نظر گرفت برای ادرس و مقدار مورد نظر برای نوشتن.

عملی:

یک ماکس 8 تایی با 3 ورودی انتخاب میکنیم تا تناسب با جدول خروجی دهد. متناسب با جدول ماژول های مورد نظر را ساخته و به ماکس متصل میکنیم. در جمع اگر و تفریق پر ارزش ترین را اور کرده و به عنوان SIGN خروجی میدهیم و چک میکنیم اگر هر یک از جواب جمع و تفریق 0 بود ZEERO یک میشود.