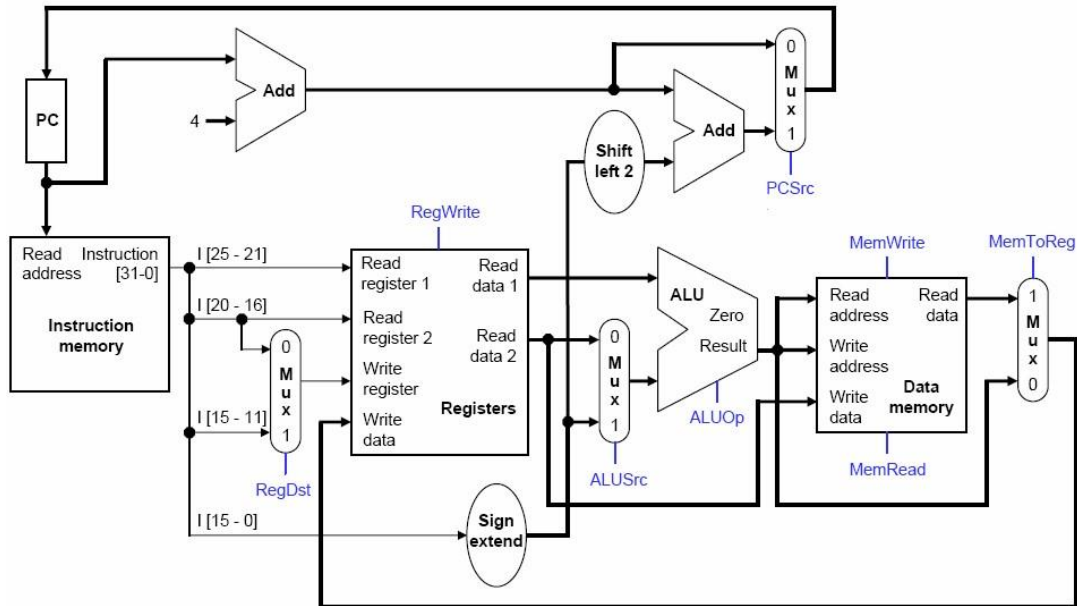


1. یک رجیستر فایل با اندازه  $16 \times 128$  دارای ۳ خط خواندن و یک خط نوشتن است. تعداد مالتی پلکسرها و دیکدرهای داخل این رجیستر فایل و اندازه هر کدام را مشخص نمایید.

2. برای دستور `Load $rd,$rs,$rt, const6bit` که کار زیر را انجام میدهد، یک مسیر داده ساده طراحی کنید.  
 $\$rd = \text{Memory} [\$rs + \$rt + \text{const6bit}]$

(مثال: `Load $2, $3, $4, 38 # $2=Memory[$3+$4+38]`)

3. با توجه به مسیر داده شکل زیر سیگنالهای کنترلی موجود در جدول زیر را طوری مقاردهی کنید که دستور مورد نظر اجرا شود.



	MemToReq	ALUSrc	ReqDst	PCSrc	RegWrite
Lw					
j (jump)					
Bne					

4. با توجه به شکل سؤال قبل، آیا دستور  $R0 = R1 + \text{Memory}[R2]$  که در آن  $R1$ ،  $R0$  و  $R2$  رجیستر هستند، میتواند در این مسیر داده (datapath) انجام شود.

5. با توجه به مسیر داده سوال سه، سیگنالهای کنترلی موجود در جدول زیر را طوری مقداردهی کنید که دستور مورد نظر اجرا شود.

	MemToReq	ALUSrc	ReqDst	PCSrc	RegWrite
lw					
add					
bne/beq					
sw					

6. در نظر داریم که دستورالعمل jr (پرش رجیستری) را به مسیر داده پردازنده single cycle سؤال ۳ اضافه کنیم. هر مسیر داده و سیگنال کنترلی را که لازم میدانید به این شکل اضافه کنید تا این دستورالعمل پشتیبانی شود. یادآوری: دستور jr از دستورات نوع R و یک دستور پرش بدون شرط است. آدرس مقصد در این دستور پرش، محتوای رجیستری است که در دستور آمده است. به طور مثال در دستور jr \$rs آدرس پرش محتوای داخل رجیستر rs خواهد بود. فرمت دستور jr به صورت زیر است.

000000	rs	00000	00000	00000	001000
6	5	5	5	5	6

7. با توجه به شکل سؤال ۳ توضیح دهید که آیا ما میتوانیم در پردازنده single cycle بدون اینکه تغییری در سخت افزار آن بدهیم دستوری نظیر swap داشته باشیم که محتوای دو رجیستر را در یک کلاک با هم عوض نماید؟

8. فرض کنید به جای اینکه دستور lw به صورت lw rt, offset(rs) (مثلا به صورت lw \$t0, 12(\$t1)) تعریف شده باشد، به صورت lw rd, rs, rt تعریف شود که کار آن به صورت  $rd = M[rs + rt]$  باشد. تغییرات لازم را بر روی datapath شکل سؤال ۳ پیشنهاد کنید تا بتوانیم شکل جدید دستور lw را پشتیبانی کنیم.

9. یک رجیستر فایل (Register File) با ۴ رجیستر ۳۲ بیتی طراحی کنید.

