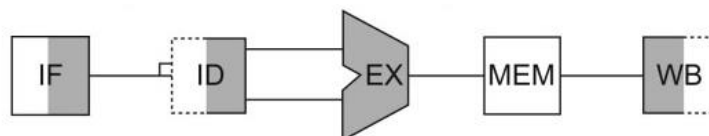




به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخنامه 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 ۱۰۰ نمره دارد که معادل ۰.۵ نمره از نمره کلی درس است.
- ۴- تحویل این تمرین در تاریخ «تحویل امتیازی» ۰.۱ نمره اضافه دارد.
- ۵- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر کل نمره این تمرین را از دست خواهند داد.

دو پردازنده MIPS را در نظر بگیرید، پردازنده A و پردازنده B؛ هر دو پردازنده یک خط لوله ۵ مرحله‌ای مطابق شکل ۱ دارند.



شکل ۱- بلوک دیاگرام مسیر داده ۵ مرحله‌ای مبتنی بر خط لوله در MIPS

ویژگی‌های خط لوله این دو پردازنده در جدول زیر آمده است.

| | پردازنده A | پردازنده B |
|-----------------------------------|--|---|
| Data Forwarding | این پردازنده data forwarding ندارد، بلکه کامپایلر ترتیب دستورها را مشخص می‌کند یا دستور nop اضافه می‌کند تا دستورهای وابسته به درستی اجرا شوند. | تشخیص وابستگی داده و اعمال data forwarding در سخت‌افزار انجام می‌شود. با تشخیص وابستگی دستورات یک عملوند را از مرحله memory (برای دستورات محاسباتی) یا از مرحله write-back (برای دستور load) به مرحله execute ارسال می‌کند. |
| Internal register file forwarding | این امکان وجود دارد. (برای مثال یک دستور در لبه بالارونده clock در یک ثبات می‌نویسد و دستور دیگر در لبه پایین‌رونده همان clock به محتوای آن ثبات دسترسی دارد.) | مشابه پردازنده A |
| Branch Prediction | پیش‌فرض همه پرش‌های شرطی این است که پرش انجام می‌شود. مقصد پرش در مرحله decode محاسبه شده و در مرحله بعد از آن در دسترس است. | مشابه پردازنده A |

این قطعه کد را در نظر بگیرید:

Loop:

```
lw $1, 0($4)
lw $2, 400($4)
add $3, $1, $2
sw $3, 0($4)
sub $4, $4, #4
bne $4, $0, Loop
```

Initially, \$1=0, \$2=0, \$3=0, and \$4=400

۱- این قطعه کد را با کمترین میزان تغییر (اضافه کردن دستور nop یا تغییر ترتیب دستورات) بازنویسی کنید طوری که با حداقل تاخیر در پردازنده A به درستی اجرا شود. (۲۰ نمره)

دستور sub را به قبل از دستور add منتقل می‌کنیم و برای جبران در دستور sw هم یک تغییر جزئی می‌دهیم.

Loop:

```
lw $1, 0($4)
lw $2, 400($4)
sub $4, $4, #4
nop
add $3, $1, $2
nop
nop
sw $3, -4($4)
bne $4, $0, Loop
```

۲- جدول زیر را به ازای اولین بار تکرار حلقه در پردازنده A پر کنید. (۱۰ نمره)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|----|----|----|-----|-----|-----|----|-----|----|----|-----|-----|----|-----|----|
| lw | IF | ID | EX | MEM | WB | | | | | | | | | | |
| lw | | IF | ID | EX | MEM | WB | | | | | | | | | |
| sub | | | IF | ID | EX | MEM | WB | | | | | | | | |
| nop | | | | - | - | - | - | | | | | | | | |
| add | | | | | IF | ID | EX | MEM | WB | | | | | | |
| nop | | | | | | - | - | - | - | | | | | | |
| nop | | | | | | | - | - | - | - | | | | | |
| sw | | | | | | | | IF | ID | EX | MEM | EX | | | |
| bne | | | | | | | | | IF | ID | EX | MEM | EX | | |
| lw | | | | | | | | | | - | IF | ID | EX | MEM | WB |

۳- تعداد چرخه‌های موردنیاز برای اجرای کد خود را در پردازنده A محاسبه کنید و محاسبات خودتان را این‌جا بنویسید. (۲۰ نمره)

پاسخ این سوال را دو جور می‌توانیم حساب کنیم:

اجرای هر بار حلقه ۱۰ چرخه دارد که با دستورات چرخه قبل و بعد هم‌پوشانی ندارد. کل حلقه هم ۱۰۰ بار تکرار می‌شود و در بار آخر سه چرخه اضافه هم نیاز داریم تا دستورات چرخه به طور کامل اجرا شوند. بنابراین کل این قطعه ۱۰۰۳ چرخه طول می‌کشد. می‌توانیم بگوییم در هر چرخه ۶ دستور اصلی داریم و ۳ تا nop و یک چرخه معطلی بعد از bne که مشابه یک nop است. پس هر چرخه ۱۰ دستور دارد و ۱۰۰ بار تکرار می‌شود، پس در کل باید ۱۰۰۰ دستور را اجرا کنیم. البته آخرین nop را نباید اجرا کنیم، بنابراین ۹۹۹ دستور داریم. اجرای ۹۹۹ دستور در یک خط لوله ۵ مرحله‌ای $999 + 5 = 1004$ چرخه طول می‌کشد (۵ چرخه برای دستور اول و ۹۹۸ چرخه برای هر کدام از دستورات دیگر)

۴- این قطعه کد را دوباره با کمترین میزان تغییر (اضافه کردن دستور nop یا تغییر ترتیب دستورات) بازنویسی کنید طوری که این بار با حداقل تاخیر در پردازنده B به درستی اجرا شود. (۲۰ نمره)

به این سوال می‌توانیم دو پاسخ بدهیم، یا ترتیب دستورات را مثل سوال ۱ عوض کنیم یا دستورات را به همان ترتیب اصلی خودشان اجرا کنیم. اما در حالت اول باید یک nop قبل از دستور add وارد کنیم. علت نیاز به این nop را در پاسخ به سوال ۵ می‌بینیم.

Sol.4.1

Loop:

```
lw $1,0($4)
lw $2,400($4)
sub $4,$4,#4
nop
add $3,$1,$2
sw $3,-4($4)
bne $4,$0,Loop
```

Sol.4.2

Loop:

```
lw $1,0($4)
lw $2,400($4)
add $3,$1,$2
sub $4,$4,#4
sw $3,-4($4)
bne $4,$0,Loop
```

۵- جدولی مشابه جدول بند ۲ به ازای اولین بار تکرار حلقه در پردازنده B پر کنید. (۱۰ نمره)

راه حل ۴-۱: در پردازنده B عمل forwarding هر دستور به دستور بعدی انجام می‌شود، بنابراین دستور add نمی‌تواند از forwarding دستور lw استفاده کند و باید برای اجرا تا تمام شدن lw دوم صبر کند و به همین دلیل به یک nop بعد از sub نیاز داریم. دستور sw می‌تواند از forwarding دستور add استفاده کند که در جدول به رنگ سبز نمایش داده شده است.

| Sol.4.1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---------|----|----|----|-----|-----|-----|----|-----|-----|-----|----|-----|----|
| lw | IF | ID | EX | MEM | WB | | | | | | | | |
| lw | | IF | ID | EX | MEM | WB | | | | | | | |
| sub | | | IF | ID | EX | MEM | WB | | | | | | |
| nop | | | | - | - | - | - | - | | | | | |
| add | | | | | IF | ID | EX | MEM | WB | | | | |
| sw | | | | | | IF | ID | EX | MEM | WB | | | |
| bne | | | | | | | IF | ID | EX | MEM | WB | | |
| lw | | | | | | | | - | IF | ID | EX | MEM | WB |

راه حل ۴-۲: اینجا از همان ترتیب اولیه برنامه استفاده کردیم و موارد forwarding به رنگ سبز نشان داده شده.

| Sol.4.2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---------|----|----|----|-----|-----|----|-----|-----|-----|-----|----|-----|----|
| lw | IF | ID | EX | MEM | WB | | | | | | | | |
| lw | | IF | ID | EX | MEM | WB | | | | | | | |
| add | | | IF | ID | - | EX | MEM | WB | | | | | |
| sw | | | | IF | - | ID | EX | MEM | WB | | | | |
| sub | | | | | | IF | ID | EX | MEM | WB | | | |
| bne | | | | | | | IF | ID | EX | MEM | WB | | |
| lw | | | | | | | | - | IF | ID | EX | MEM | WB |

می‌بینیم که تعداد چرخه‌های موردنیاز در هر دو راه حل یکی است.

۶- تعداد چرخه‌های موردنیاز برای اجرای کد را در پردازنده B محاسبه کنید و محاسبات خودتان را این‌جا بنویسید. (۲۰ نمره)

طبق استدلالی که برای سوال ۳ انجام دادیم، کامل شدن هر دور حلقه به ۸ چرخه نیاز دارد، بنابراین کامل شدن ۱۰۰ دور حلقه ۸۰۰ چرخه نیاز دارد و علاوه بر آن حلقه آخر به سه چرخه دیگر هم نیاز دارد، پس روی هم ۸۰۳ چرخه نیاز است.

با استدلال دیگری، در هر دور اجرای حلقه ۸ دستور اجرا می‌شود، بنابراین در مجموع ۸۰۰-۱ دستور اجرا می‌شود. اجرای ۷۹۹ دستور در یک خط لوله ۵ مرحله‌ای $۷۹۸+۵=۸۰۳$ چرخه طول می‌کشد.