



## آزمایش دوم: پیاده سازی پردازنده MIPS

گرد آورنده: علیرضا یزدان پناه

### قسمت چهارم آزمایش پیاده سازی MIPS (جلسه چهارم):

در جلسات گذشته، مراحل واکشی، کدگشایی، اجرا و نوشتن در رجیسترفایل (WB) پردازنده پیاده سازی گردید. در این جلسه، ابتدا مرحله‌ی حافظه و مازول تشخیصی مخاطره را پیاده سازی کنید، سپس کل پردازنده را تست کنید.

برای پیاده سازی حافظه نکات زیر را در نظر بگیرید:

- ۱- خواندن به صورت ترکیبی و نوشتن در حافظه به صورت ترتیبی انجام می‌شود. نوشتن در لبه بالارونده رونده‌ی کلاک صورت می‌گیرد.
- ۲- در این مرحله به کمک سیگنال MEM\_W\_EN داده‌ها در حافظه نوشته می‌شود. هرگاه این سیگنال مقدارش در سر بالارونده کلاک یک باشد، مقدار در در آدرس مشخص شده از حافظه، تغییر می‌کند.
- ۳- داده‌های درون حافظه به کمک سیگنال MEM\_R\_EN خوانده می‌شود.
- ۴- این حافظه تنها یک خط آدرس دارد که هم برای نوشتن و هم برای خواندن استفاده می‌شود. آدرس در حقیقت داده‌ی محاسبه شده تو سط ALU در مرحله‌ی قبل می‌باشد. این مقدار در رجیسترهای پایپ ذخیره شده است.
- ۵- برای پیاده‌سازی فضای حافظه می‌توانید از تعریف آرایه‌ای از رجیسترها استفاده کنید. مانند زیر:  
Reg [31:0] memory [0:63];
- ۶- خواندن و نوشتن فقط از آدرس‌های مضرب ۴ (به دلیل ۳۲ بیتی بودن معماری) انجام می‌شود. به طور مثال: در ازای خواندن از آدرس‌های ۱۰۲۴، ۱۰۲۵، ۱۰۲۶ و ۱۰۲۷ نتایج یکسانی خوانده می‌شود یعنی ۴ بایت از آدرس ۱۰۲۴ خوانده می‌شود.
- ۷- فضای آدرس‌دهی حافظه داده از ۱۰۲۴ شروع می‌شود. در این صورت برای خواندن یا نوشتن آدرس ۱۰۲۴ مقادیر ۴ بایت اول آرایه memory یعنی memory[0] خوانده یا در آن نوشته می‌شود (همه آدرس‌های ورودی به این مرحله از ۱۰۲۴ کسر شود).
- ۸- پس از اضافه کردن واحد حافظه به پردازنده، یک کد برای تست پردازنده بنویسید که شامل هازارد نباشد (چون هنوز واحد تشخیص هازارد به پردازنده اضافه نشده است).
- ۹- مازول Write Back را پیاده سازی نمایید، این مازول بسیار ساده خواهد بود، این مازول داده خوانده شده از حافظه و داده محاسبه شده توسط ALU را به همراه شماره رجیستر مقصد از طریق رجیسترهای پایپ‌لاین به عنوان ورودی دریافت می‌کند و با توجه به سیگنال‌های کنترلی محتوای رجیسترفایل را تغییر می‌دهد. در رجیسترهای پایپ‌لاین میان دو مرحله‌ی memory و write back، سیگنال‌های کنترلی WB\_EN و MEM\_R\_EN ذخیره شده‌اند. هرگاه WB\_EN برابر یک باشد، با توجه به MEM\_R\_EN داده‌ی خوانده شده از حافظه یا داده‌ی محاسبه شده توسط ALU در رجیستر فایل نوشته می‌شود.



## آزمایش دوم: پیاده سازی پردازنده MIPS

گرد آورنده: علیرضا یزدان پناه

حال باید ماژول تشخیصی مخاطره افزوده شود اما در این بخش به طور مختصر انواع مخاطره ها را ذکر می کنیم:  
به طور کلی در پردازنده ها سه نوع مخاطره وجود دارد:

- الف) مخاطره ساختاری:** این مخاطره در بطن ساختار خط لوله وجود دارد به همین دلیل ساختاری نا گرفته است. این مخاطره بین مرحله WB و ID به دلیل همزمانی خواندن و نوشتن از ثبات های عمومی ناشی می شود. برای رفع این مخاطره نوشتن در ثبات های عمومی را به لبه پایین رونده منتقل کردیم. بنابراین این مخاطره در پردازنده رفع شده است.
- ب) مخاطره کنترلی:** این مخاطره ناشی از دستورات پرش است. در صورتی که دستور پرش وارد خط لوله شود به دلیل تأخیر در تشخیص و محاسبه آدرس پرش دو دستور به اشتباه وارد خط لوله می شود. برای رفع این مشکل سیگنال های Flush به پردازنده افزوده شد. بنابراین این مخاطره نیز رفع شده است.
- ج) مخاطره داده ای:** مخاطره داده ای به صورت زیر دسته بندی می گردد:

۱- **خواندن پس از نوشتن (RAW):** مخاطره خواندن بعد از نوشتن در حالتی رخ می دهد که یک دستور، رجیستری که هنوز محاسبه یا ذخیره نشده است را فراخوانی می نماید. در این حالت می بایست دستورات جدید تا محاسبه یا ذخیره شدن آن رجیستر متوقف گردند. در مثال زیر دستور ۲ نیاز به رجیستر R2 دارد که دستور ۱ آنرا محاسبه می نماید، بنابراین تا ذخیره یا محاسبه شدن مقدار R2 دستور ۲ باید متوقف گردد.

1. SUB R2,R0,R1
2. AND R3,R2,R1

در این آزمایش سعی در برطرف نمودن این نوع هازارد را داریم.

۲- **نوشتن پس از خواندن (WAR):** در مثال زیر رجیستر R1 در حالتی ممکن است قبل از خوانده شدن توسط دستور ۱ مقدار آن به وسیله دستور ۲ تغییر کند. به این رخداد هازارد داده ای از نوع WAR گفته می شود.

1. SUB R2,R0,R1
2. AND R1,R3,R4

این نوع هازارد در پردازنده های In-Order مانند پردازنده مورد بحث در این دستور کار رخ هرگز رخ نخواهد داد.

۳- **نوشتن پس از نوشتن (WAW):** هازارد داده ای نوشتن پس از نوشتن نیز همچون هازارد WAR در پردازنده های In-Order رخ نمی دهد. این نوع مخاطره ممکن است در پردازنده های Out-of-order رخ دهد و ترتیب نوشتن در رجیستر مقصد تغییر کند.

برای رفع هازارد RAW ماژول Hazard Detection Unit همانند شکل زیر به پردازنده ARM اضافه نمایید. در این واحد، منابع Src1 و Src2 در مرحله ID با مقصدهای مراحل EXE و MEM به صورت مجزا مقایسه می شود و در صورت برابر بودن یکی از منابع با مقصدها، سیگنال کنترلی Hazard\_Detected\_Signal را برابر ۱ قرار می دهد. این سیگنال باید دستورات درون IF و رجیستری های پس از آن را متوقف نماید و حبابی را به خط لوله تزریق نماید. برای ایجاد حباب کافی است تمامی سیگنال های حیاتی پردازنده صفر گردد. برای پیاده سازی این مرحله به ترتیب زیر عمل کنید:

۱- واحد تشخیصی هازارد با حداقل ورودی های زیر ایجاد نمایید.



## آزمایش دوم: پیاده سازی پردازنده MIPS

گرد آورنده: علیرضا یزدان پناه

```
1 module hazard_Detection_Unit(  
2     input  [4:0]src1,  
3     input  [4:0]src2,  
4     input  [4:0] Exe_Dest,  
5     input  Exe_WB_EN,  
6     input  [4:0]Mem_Dest,  
7     input  Mem_WB_EN,  
8     output hazard_Detected  
9 );
```

۲- تمامی حالاتی که هازارد RAW رخ می دهد را در نظر بگیرید و خروجی واحد تشخیص هازارد را یک کنید. این حالت ها به شرح زیر است:

- برابری src1 با مقصد EXE در صورت یک بودن WB\_EN در مرحله اجرا
  - برابری src1 با مقصد MEM در صورت یک بودن WB\_EN در مرحله حافظه
  - برابری src2 با مقصد EXE در صورت یک بودن WB\_EN در مرحله اجرا و دو منبعی بودن دستور
  - برابری src2 با مقصد MEM در صورت یک بودن WB\_EN در مرحله حافظه و دو منبعی بودن دستور
- ۳- به دستورات Immediate که دارای تنها یک منبع هستند توجه نمایید. سیگنال دارای منبع دوم بودن که براساس سیگنال فوری بودن (immediate) و دستور STR تولید شده است را نیز به هازارد اضافه کنید.
- ۴- ورودی های مورد نیاز واحد هازارد را به خروجی دیگر مراحل متصل نمایید.
- ۵- قابلیت Freeze را به رجیستر PC در مرحله IF و رجیستری های بعد از IF اضافه نمایید و خروجی واحد تشخیص هازارد را به آن متصل نمایید.
- ۶- برای ایجاد حباب در خط لوله سیگنال های کنترلی خروجی از واحد کنترل را صفر نمایید. برای اینکار خروجی ماژول مخاطره را با خروجی بررسی شرط دستورات or کنید.
- ۷- دستورات برنامه محک را بدون تغییر اجرا نمایید.
- ۱۰- پردازنده خود را اشکال یابی نمایید و درستی نتایج را به کمک دستیار آموزشی بررسی نمایید.
- ۱۱- گزارش این بخش را به همراه خروجی SignalTap تکمیل نمایید.
- ۱۲- در بعضی از دستورات src1 وجود ندارد اما کد پیاده سازی شده آنها را تشخیص نمی دهد. در صورت رفع این مشکل ۵ درصد نمره اضافه منظور می شود. در صورت پیاده سازی تغییرات خود را در گزارش قید کنید.

موفق باشید

نصیحت کن