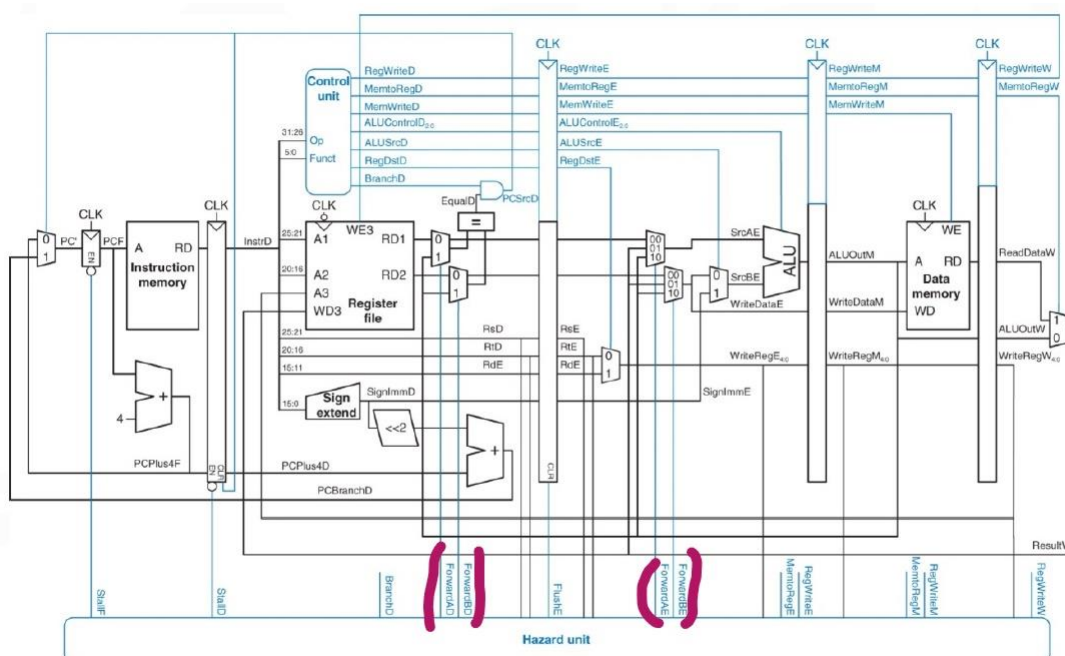


# گزارش پروژه نهایی درس معماری کامپیوتر

نگار هنرور صدیقیان  
99243076

در این پروژه به دنبال پیاده‌سازی یک پردازنده پایپ لاین میپس هستیم که در صورتی که دستور SW بلافاصله پس از دستور LW قرار گرفت نیازی به stall شدن نداشته باشد. برای اینکار لازم است تا به forwarding unit سیگنال جدیدی به نام c forward اضافه کنیم. در این راستا ابتدا لازم است تا به بررسی ساختار forwarding unit پردازیم که در کنار hazard unit وظیفه سامان دهی دستورات را به عهده دارند. ( در تصویر پردازنده پایپ لاین که در ادامه قرار گرفته و در صورت پروژه درج شده است forwarding unit به عنوان بخشی از hazard unit در نظر گرفته شده است.)



سیگنال های خروجی مشخص شده در تصویر مربوط به forwarding هستند. این دو سیگنال مقدار مورد نیاز را از رجیستر ها به mux های موردنظر برده و :

1) یکی از آن ها دیتا را از EX/MEM Stage به ALU منتقل میکند.

2) یکی از آن ها نیز دیتا را از MEM/WB Stage به ALU منتقل میکند.

و در بعضی منابع به این دو، forward های ALU-ALU نیز گفته میشود.

این سیگنال ها read and write hazard های مربوط به این دو استیج را برطرف میکنند.

حال ما به دنبال پیاده سازی سومین سیگنال مربوط به forwarding unit هستیم. این سیگنال برخلاف دو سیگنال قبلی داده را به ALU منتقل نمیکند.

برای درک بهتر نحوه عملکرد آن:

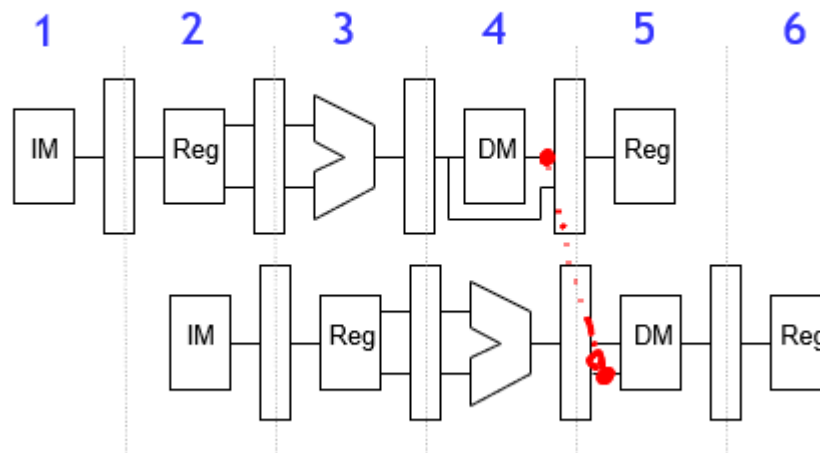
فرض کنید دستور فوق را داریم:

Lw \$1, 0(\$2)

Sw \$1, 0(\$2)

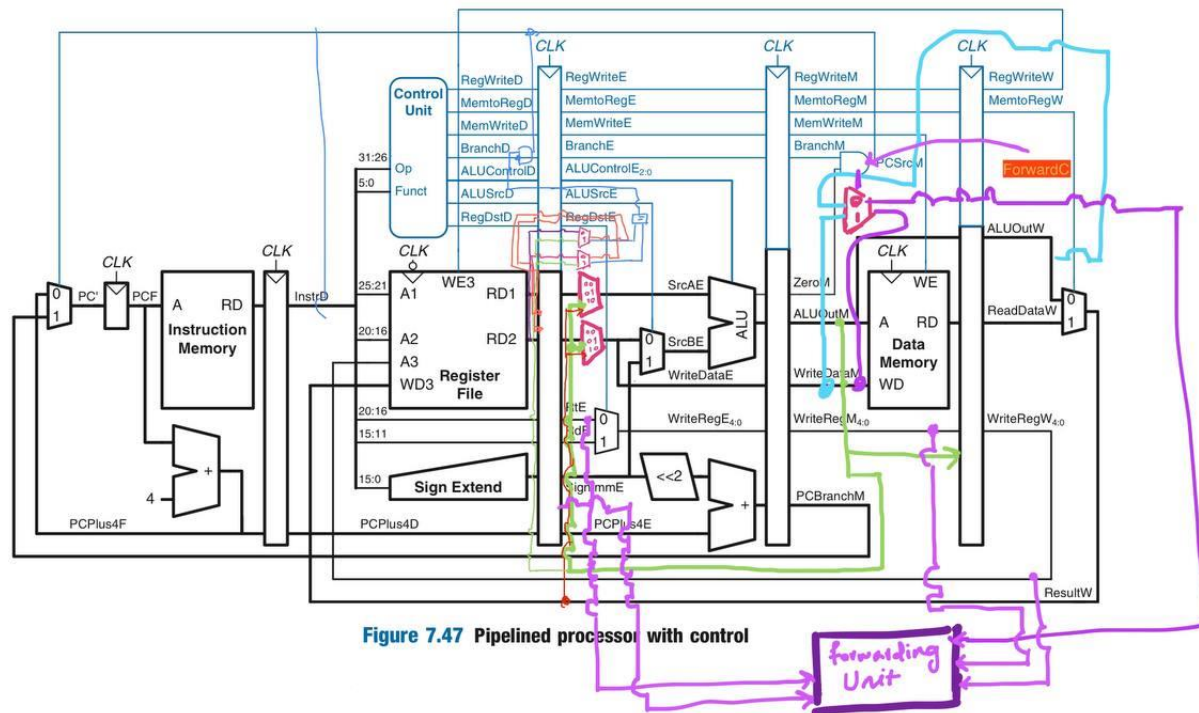
به طور کلی نتیجه حاصل از دستور lw تا انتهای سیکل پنجم (انتهای کلاک پنجم) آماده ارسال به sw که به آن در سیکل چهارم خود (ابتدای کلاک پنجم) نیاز دارد نمیشود، اما:

- In what cycle is:
  - The load value available? The end of 4<sup>th</sup> clock cycle
  - The store value needed? The beginning of 5<sup>th</sup> clock cycle



در نتیجه به سیگنالی نیاز داریم تا داده را از MEM/WB Stage به ابتدای آن منتقل کند.

به این ترتیب و با جدا در نظر گرفتن forwarding unit از hazard detection unit خواهیم داشت:



تصویر بالا forwarding unit را با تمامی اجزای آن از جمله forward c که وظیفه انتقال داده را از MEM/WB Stage به ابتدای آن برعهده دارد. به این ترتیب بخشی از hazard unit نیز طراحی شده است. در ادامه برای تکمیل طراحی hazard unit با توجه به آنچه در کتاب مرجع پروژه ذکر شده خواهیم داشت:

ما در اینجا با انواع hazard رو به رو هستیم.

1-data hazard

2-control hazard

در ابتدا به بررسی data hazard ها میپردازیم:

- Data hazard های ایجاد شده توسط دستورات محاسباتی (R-type) به کمک forwarding unit و سیگنال های forward A & forward B برطرف میشوند. همچنین برطرف کردن data hazard های مربوط به دستور sw بعد از lw نیز همانطور که گفت شد بر عهده forward C است.
- در زمانی که برای دستور lw دچار hazard میشویم hazard detection unit کمک سیگنال های stall IF and stall ID باعث میشود تا رجیسترهای استیج های ID and IF مقادیر قدیمی خود را حفظ کنند و سیگنال stall E فعال میشود تا رجیستر مربوط به مرحله EX/ از مقدار فعلی خود پاک شود. در اینجا یک bubble خواهیم داشت.

حالت دوم hazard به واسطه (I-type instructions) branch رخ میدهد.

در این حالت لازم است تا برای هر branch پایپ لاین را به مدت 3 کلاک سایکل stall کنیم تا دستور بعد از branch در استیج مموری گرفته شود. راهکار دیگر branch prediction است.

