

بسمه تعالی



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

فاز سوم Pipe Line

تیم اول

امیرحسین عزیزی - ۴۰۰۱۰۵۱۲۲

سپهر میزانیان - ۴۰۰۱۰۹۶۸۴

امید دلیران - ۴۰۰۱۰۴۹۳۱

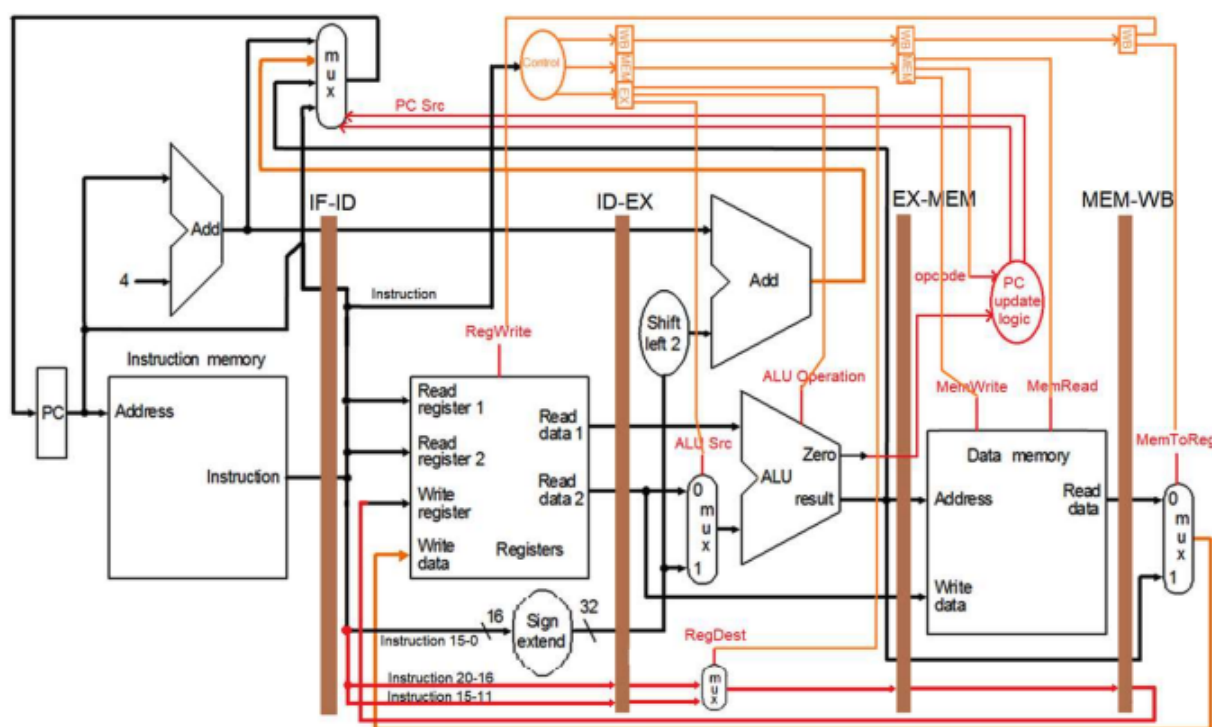
تیر ۱۴۰۲

توضیحات کلی

در ابتدا ما ساختار کلی CPU را به گونه‌ای تغییر دادیم تا به صورت خط لوله‌ای عمل کند. خط لوله شامل ۵ مرحله دریافت دستورات، ترجمه opcode، اجرای عملیات‌ها، دسترسی به حافظه و Write-Back می‌باشد. سپس به سراغ بررسی مخاطراتی که پیش می‌آیند رفتیم و آن‌ها را بررسی کردیم و همچنین راه‌های موجود برای برطرف کردن آن‌ها را ذکر کردیم.

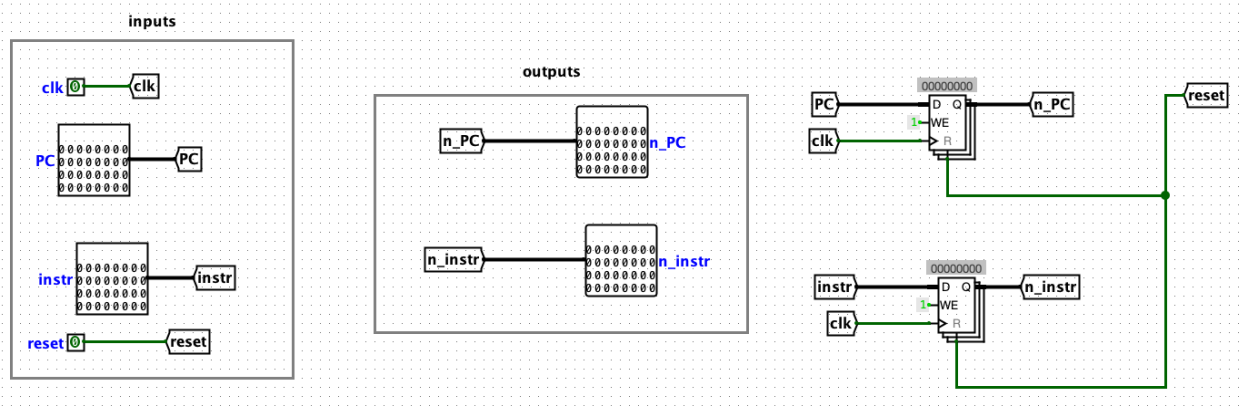
خط لوله

شمای کلی خط لوله‌ای که در نظر گرفته شده است به صورت زیر است:



در این ساختار از ۴ بافر استفاده شده است که در هر کدام سیگنال‌های مورد نیاز هر استیج از استیج قبلی در آن‌ها ذخیره می‌شود، همچنین سیگنال‌های کنترلی مورد نیاز نیز در بافرها قرار داده شده است.

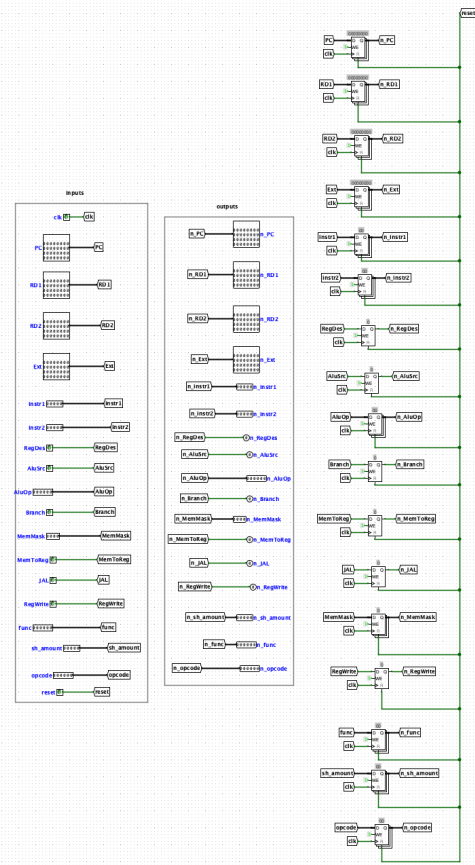
بافر IF-ID



در این مرحله بیت‌های instruction و PC+4 نگهداری می‌شوند.

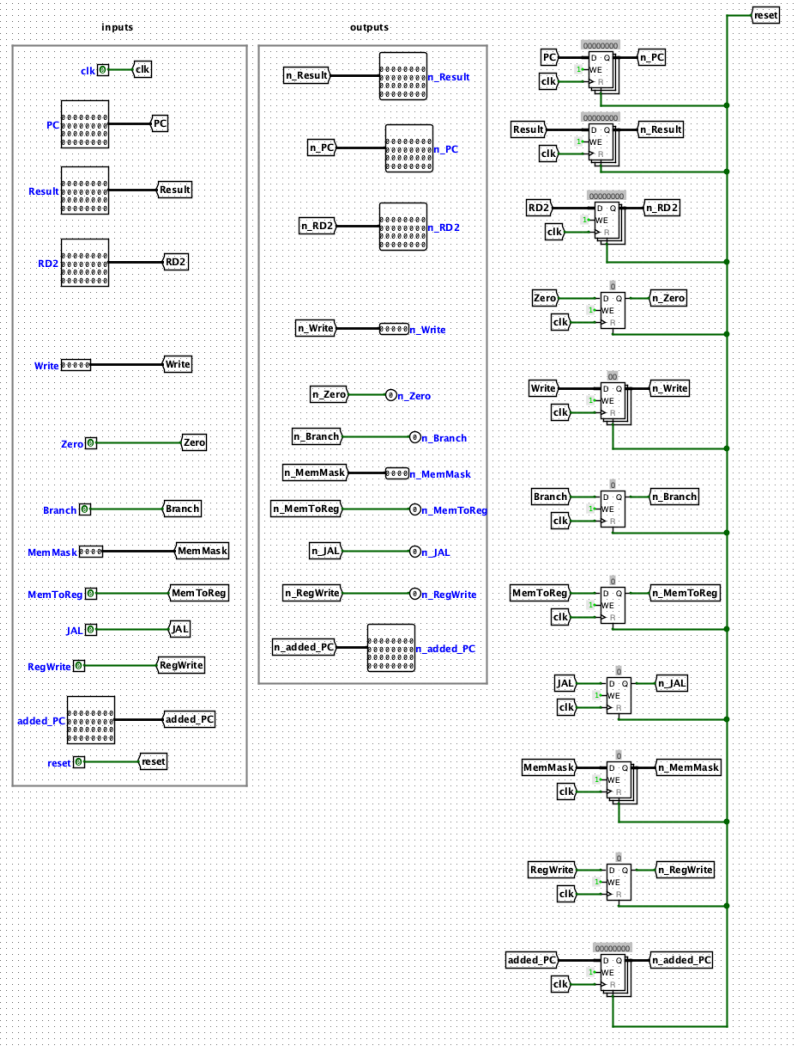
بافر ID-EX

در این مرحله ترجمه opcode به سیگنال‌های کنترلی و نتایج خواندن رجیسترها قرار می‌گیرد.



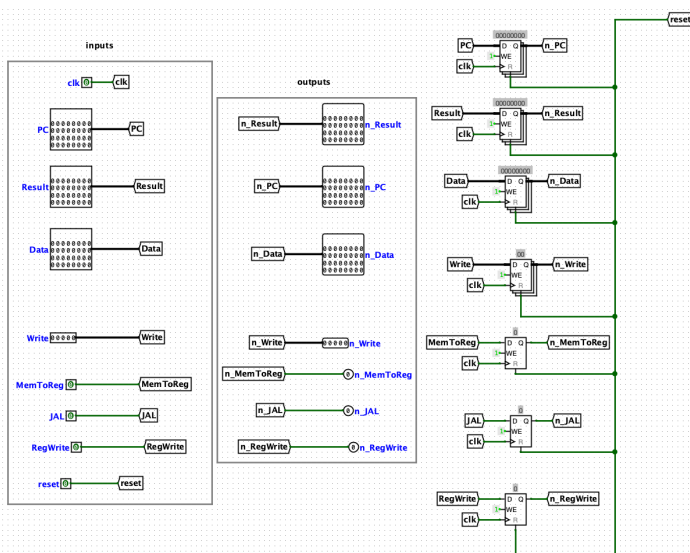
بافر EX-MEM

در این مرحله نیز مقادیری همچون مقادیر PC و خروجی ALU و سیگنال‌هایی همچون سیگنال JAL قرار داده می‌شوند.



بافر MEM-WB

در این مرحله هم مقادیر PC و خروجی مموری قرار داده می‌شود.



در اجرای دستورات به صورت خط لوله‌ای مخاطراتی پیش می‌آید که می‌توان آن‌ها را به ۳ دسته تقسیم کرد:

- مخاطرات ساختاری
- مخاطرات داده‌ای
- مخاطرات کنترلی

مخاطرات ساختاری

هنگامی که نیاز به استفاده همزمان از یک قطعه سخت‌افزاری شود مخاطره ساختاری رخ داده است، برای مثال در یک سیکل ممکن است نیاز به نوشتن و خواندن از مموری داشته باشیم که ممکن نیست. چنین مخاطره‌ای را می‌توانیم با دو روش از بین ببریم

- روش اول: انداختن حباب در خط لوله تا اجرای دستورات به تعویق بیافتد.
- روش دوم: جدا کردن حافظه دستورات و داده‌ها.

لازم به ذکر در این پروژه روش دوم پیاده شده است.

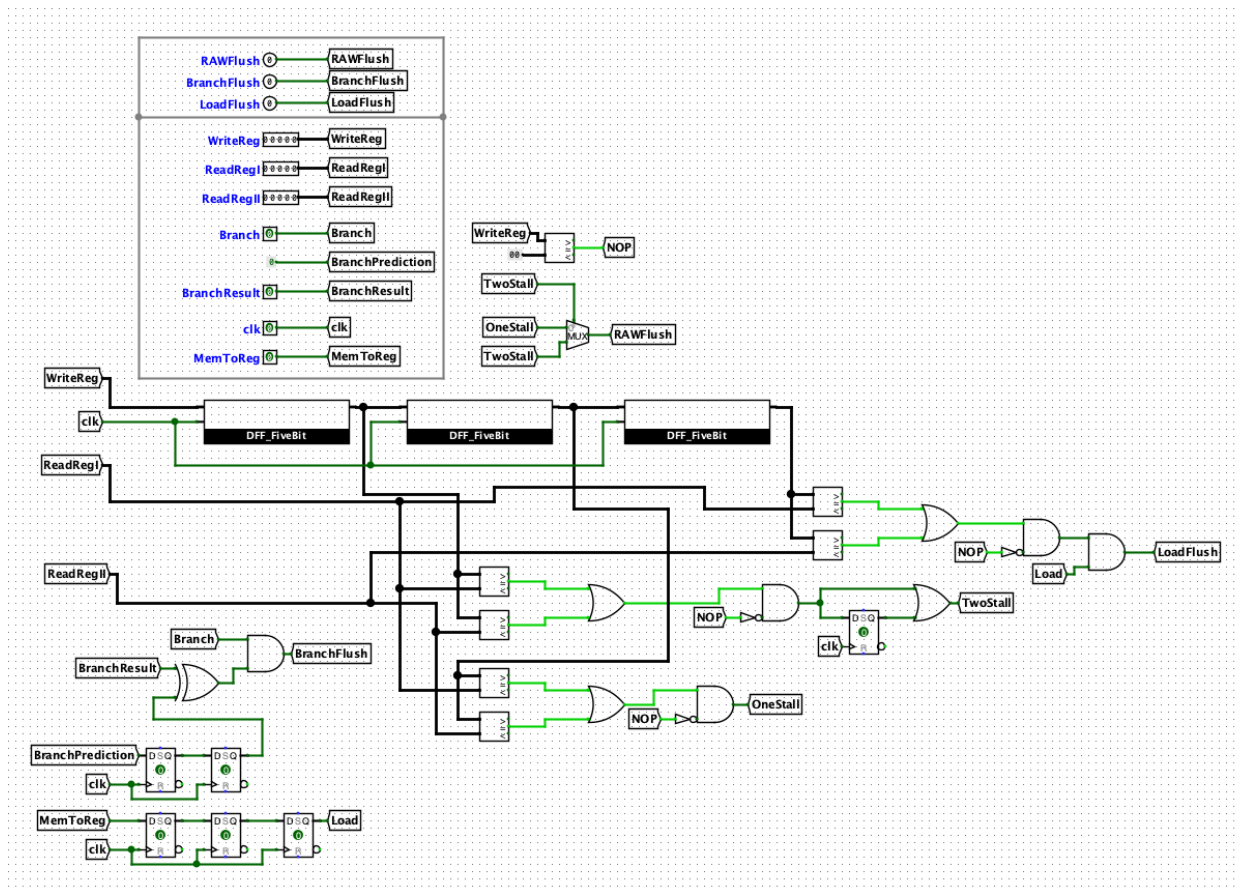
مخاطرات داده‌ای

این مخاطره زمانی رخ می‌دهد که چند دستور به یک داده نیاز داشته باشند، برای مثال RAW، در این نوع از مخاطره، یک دستور تغییراتی در یک داده می‌دهد اما قبل از آنکه آن داده در رجیستر ذخیره شود دستور دیگری آن داده را می‌خواند در حالی که هنوز مقدار قدیمی در رجیستر قرار دارد. البته WAR و WAW مخاطره به حساب نمی‌آیند و مشکلی در اجرای برنامه ایجاد نمی‌کنند. راه حلی که برای از بین بردن این مخاطره در نظر گرفته می‌شود، فروارد کردن نتایج دستورات از یک مرحله از خط لوله به مرحله دیگر است. البته در این روش نیز نیاز به حباب انداختن در اجرای دستورات وجود دارد.

مخاطرات کنترلی

هنگامی این نوع مخاطره رخ می‌دهد که دستوری که لود شده است، اجرا شدنش وابسته به نتایج دستور و یا دستورهای قبلی باشد. برای مثال دستورات branch می‌توانند باعث این نوع مخاطره شوند. یکی از راه‌حلهایی که می‌توان برای این نوع مخاطره در نظر گرفت، مجدداً انداختن حباب در اجرای دستورات است، به این نحو که اگر به دستورات برنج رسیدیم تا مشخص شدن نتیجه‌ش کار بقیه مدار را متوقف کنیم. راهکار دیگر محاسبه زودتر نتیجه branch است. همچنین می‌توان در مدار یک Branch Predictor قرار داد تا نتیجه برنچ‌ها را پیش‌بینی کند. که این روش را در ادامه پیاده سازی می‌کنیم.

ماژول Pipeline Flush



در این ماژول بررسی می‌شود که رجیسترهای مورد استفاده در دستور، با دستورات موجود در خط لوله دچار مخاطره RAW می‌شوند یا خیر، و در صورتی که مخاطره رخ بدهد مشخص می‌کند که به چه تعداد باید حباب در خط لوله ایجاد کنیم. همچنین بررسی می‌شود که اگر نتیجه برنج درست بوده است در خط لوله حباب ایجاد شود.

نتیجه‌گیری

در این فاز به طراحی و پیاده‌سازی خط لوله پرداختیم و همچنین مخاطرات ناشی از خط لوله را بررسی کردیم و در ادامه راهکارهایی را برای برطرف کردن آن‌ها ذکر کردیم. در فاز بعد به پیاده‌سازی Branch Predictor که ارتباط تنگاتنگی با برطرف کردن مخاطره‌های خط لوله دارد می‌پردازیم.