

گزارش فاز سوم پروژه معماری کامپیوتر - گروه Mascarpone

محمد مهدی قیدی - ۹۸۱۰۵۹۷۶

حسین آقایی - ۹۸۱۰۵۶۱۹

میلاد سعادت - ۹۸۱۰۰۴۴۲

پرهام چاوشیان - ۹۸۱۰۰۱۱۸

در این فاز ما به پیاده‌سازی یک خط لوله یا همان پایپ لاین برای پردازنده Mips طراحی شده پرداختیم. ابتدای امر ذکر این نکته خالی از لطف نیست که با توجه به پیچیدگی های بسیار زیاد خط لوله پنج مرحله‌ای (five-stage pipeline) مانند وصل شدن استیج‌های نگه‌دارنده استیج‌های قبلی، وصل شدن استیج‌ها به استیج‌های پیشین و افزایش بسیار تعداد reg ها و wire ها در پیاده سازی module ها این تصمیم توسط تیم گرفته شد که پیاده‌سازی به صورت ۳-مرحله‌ای و شامل مراحل واکشی (fetch) انجام محاسبات منطقی (alu) و در نهایت مرحله نوشتن (write) است. پس از بررسی‌ای که انجام دادیم با این پیاده‌سازی بهبود مورد نظر به طور معقول و قابل انتظاری حاصل میشد. اما در حالت ۵-مرحله‌ای به دلیل افزایش بار پیاده‌سازی و سربار کلاک و انتقال داده بین استیج‌ها و ... این کار به صرفه نبود.

توضیحات کلی هر مرحله از پیاده سازی بدین صورت می‌باشد:

- در مرحله واکشی یا همان fetch دستور از آدرس مشخصش که در pc قرار دارد خوانده شده و در متغیر inst قرار داده می‌شود تا در مراحل بعدی اجرا شود. این عملیات توسط ماژول fetch_operands که در هسته cpu یا همان mips_core یک نمونه از آن را گرفتیم، اجرا می‌شود. پس از آن برای اینکه در مرحله بعد بتوانیم به صورت موازی با واکشی دستور بعدی این دستوری که در حال حاضر واکشی شده را اجرا کنیم به یک ماژول احتیاج داشتیم تا نتایج مرحله پیشین را برای اجرا در مرحله بعدی ذخیره کند. این اطلاعات با وصل کردن سیم های خروجی fetch_operands به یک نمونه از fetch_result انجام می‌شود.
- تغییر جدیدی که در این فاز اضافه شده است این است که، واحد محاسبه منطقی که در فاز های پیشین در ماژول های مختلف انجام می‌شده بود در ماژول ALU تجمیع شده اند تا هم کار محاسبه منطقی به صورت جامع به یک ماژول اختصاص یابد و هم کد خواناتر و دقیق‌تر شود. برای این استیج هم یک ماژول alu_result طراحی شده تا نتایج این استیج را در خودش نگه دارد تا برای اجرا در فاز بعدی استفاده شود.
- در آخرین مرحله، با توجه به اینکه در فاز های قبلی منطق کل پراسسور را در ماژول mips_core پیاده سازی کرده بودیم و تمام قسمت‌های مربوط به نوشتن نیز در این ماژول بود، کار نوشتن را به همین ماژول سپردیم و نتایج

آمده از استیج قبلی با ذخیره شدن در ماژول mips_core پس از کلاک های بعدی در مموری و/یا کش نوشته می شوند.

پیاده سازی تا حد خوبی خواناست و سعی شده است با توجه به تعداد بسیار زیاد سیم ها و کانکشن های بین ماژول ها، نحوه نامگذاری متغیرها و ... به صورتی انجام شود که خوانایی کد تا حد امکان بالا رود تا راحتی مطالعه، دیباگ و ... کد افزایش یابد.

یک دیاگرام و schema ی کلی از معماری سطح بالای این فاز می تواند به شکل زیر نمایش داده شود:

