

به نام خداوند جان و خرد



دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

گزارش کار

Cache فاز دوم: افزودن

پروژه درس معماری کامپیوتر

سپهر میزانیان - ۴۰۰۱۰۹۶۸۴

امیرحسین عزیزی - ۴۰۰۱۰۵۱۲۲

امید دلیران - ۴۰۰۱۰۴۹۳۱

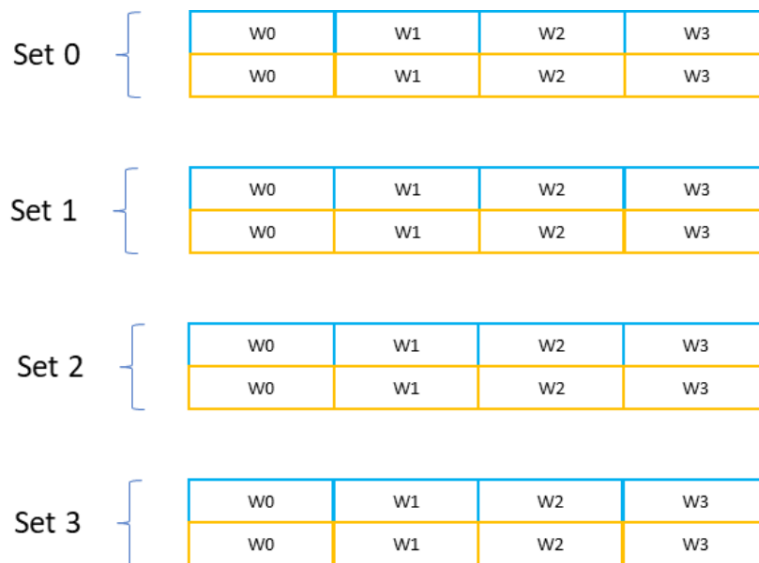
خردادماه ۱۴۰۲

توضیحات فاز

در فاز نخست، ما به طراحی یک Datapath مربوط به پردازنده MIPS پرداختیم. در این فاز و مطابق با توضیحات داده شده، باید حافظه بدون تاخیر با یک حافظه دارای تاخیر (با ۵ کلاک تاخیر) جایگزین شده و با طراحی یک 2-way set associative cache تلاش شود تا اثر این تاخیر بر روی اجرای دستورات در پردازنده کاهش یابد.

۱ ساختار کلی Cache

ساختار کلی Cache به صورت یک کش 2-way set associative است. در این کش، هر ست دارای ۲ بلاک، هر بلاک دارای ۴ کلمه و کل کش نیز شامل ۴ ست است. همچنین سیاست جایگزینی نیز به صورت True LRU خواهد بود. در داک توضیحات پروژه، شمای کلی کش آمده است:



۱ - ساختار کلی Cache

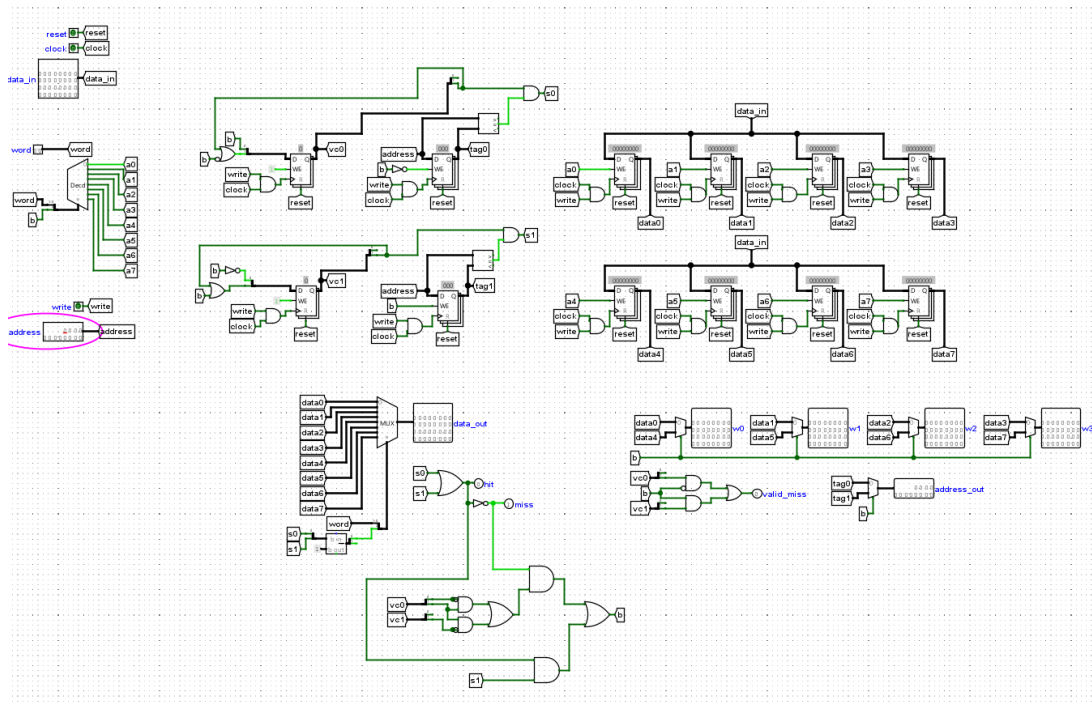
۲ ماژول‌های مربوط به Cache

در CPU، از ماژول Cache استفاده شده است. همچنین برای ساخت این ماژول، از ماژولی به نام set استفاده شده که توضیحات و شمای هردو در ادامه آورده خواهد شد.

۱-۲ ماژول set

این ماژول یک آدرس ۱۲ بیتی مربوط به همین ست را دریافت کرده و به ازای تگ ورودی، hit یا miss شدن درخواست CPU را بررسی و در خروجی تولید می‌کند. همچنین در خود این ماژول، یک سیستم جایگزینی True LRU پیاده‌سازی شده که هنگام نوشتن داده جدید در صورت پر بودن ست، داده روی داده‌ای که اخیراً استفاده نشده نوشته می‌شود. این موضوع، با کمک یک رجیستر به نام C هندل شده که مقدار آن مطابق با توضیحات داده شده سر کلاس درس برای این سیستم جایگزینی، پس از هر بار hit یا miss به‌روزرسانی می‌شود.

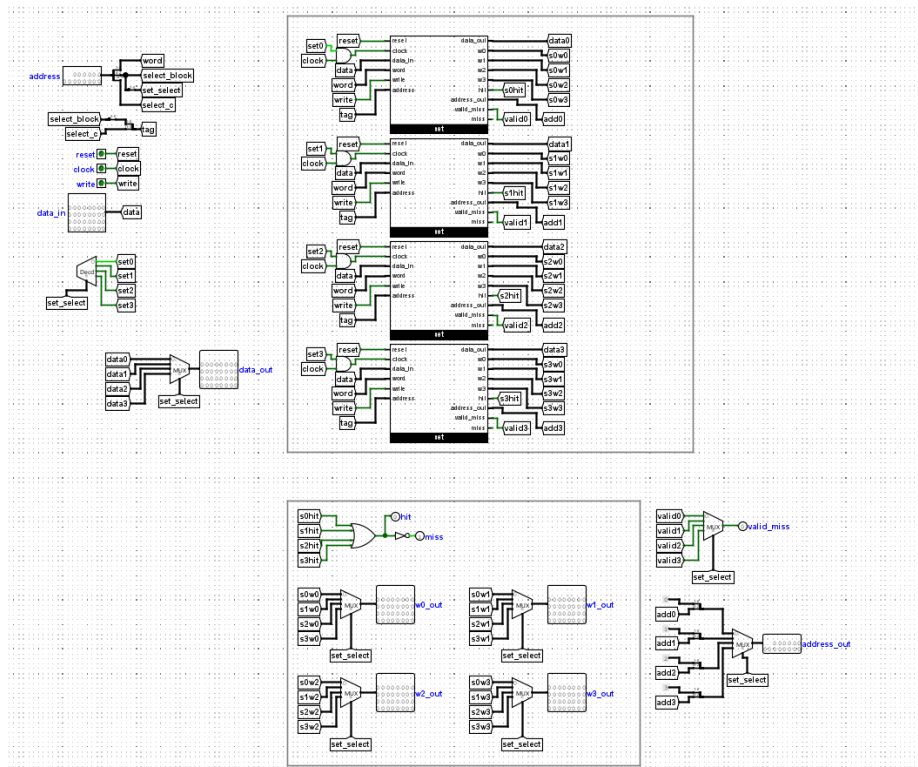
پروژه درس معماری کامپیوتر - فاز دوم



۲- ساختار کلی مازول set. ورودی و خروجی ها در سمت چپ تصویر مشخص شده اند.

۲-۲ مازول cache

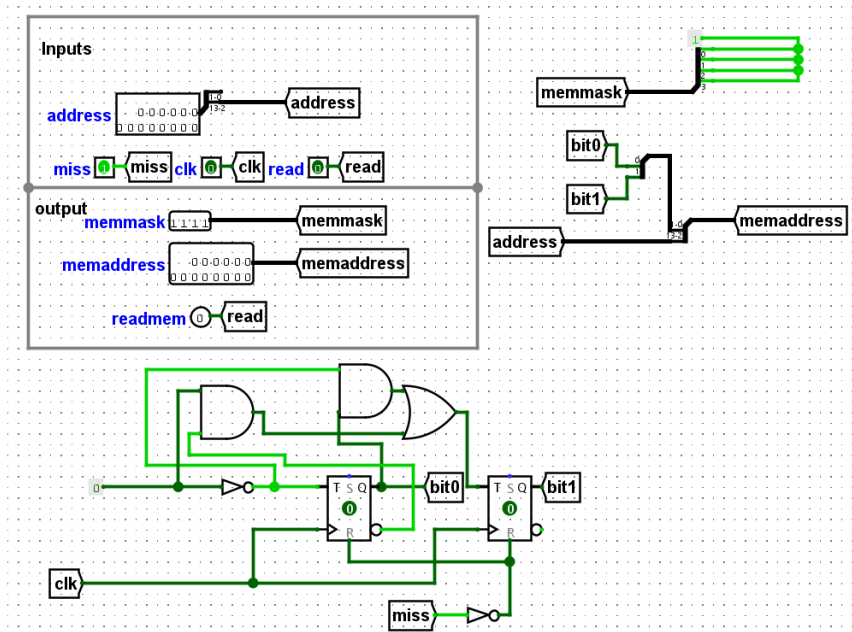
این مازول که در واقع همان کش قرار گرفته در پردازنده است، از ۴ ست تشکیل شده که با توجه به ۲ بیت کم ارزش آدرس ۱۴ بیتی دریافتی، ۱۲ بیت باقی مانده را به ستها می دهد و هیت یا میس را بررسی می کند.



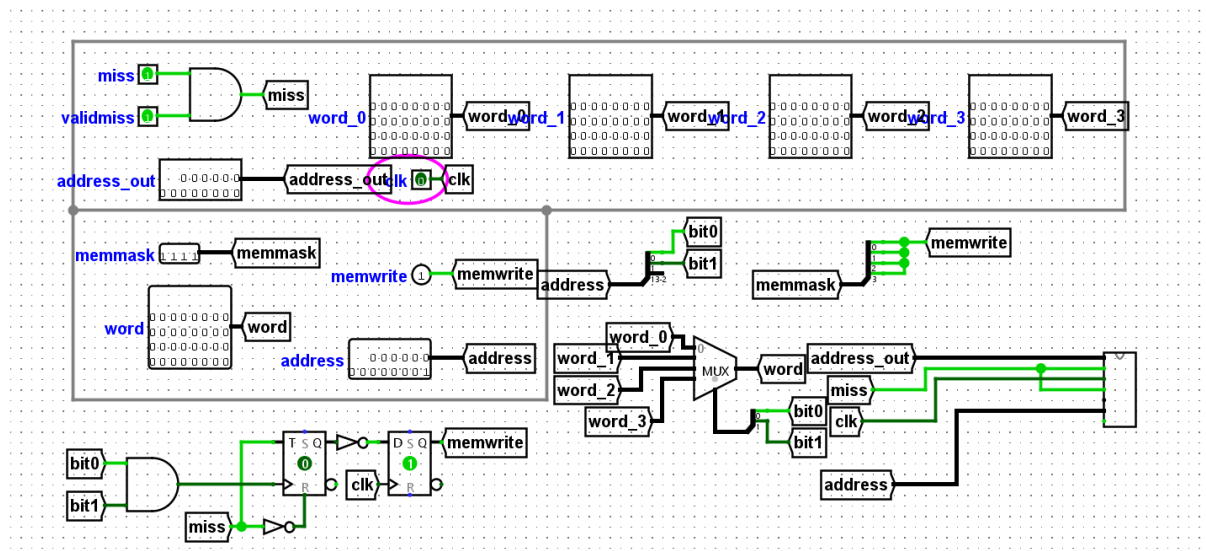
۳ - ساختار کلی مازول cache که از ۴ ست تشکیل شده است.

۳ مازول‌های MemToCache و MemAddressGen

در ادامه، دو مازول به نامهای MemToCache و MemAddressGen طراحی شده‌اند که هدف آنها تسهیل انتقال داده میان کش و مموری است. مازول MemAddressGen، آدرس یک ورود در مموری را دریافت کرده و طی ۴ کلاک، آدرس‌های مربوط به بلاکی که این ورود در آن قرار دارد را تولید می‌کند. از این مازول، در مازول MemToCache استفاده شده است. این مازول تنها هنگام جایگزینی کردن داده صدا زده شده و با کمک مازول MemAddressGen، یک بلاک از کش را گرفته و کلمه به کلمه به حافظه می‌دهد، سپس کلمه به کلمه از حافظه داده گرفته و به کش بازمیگرداند. درون کش، خود مازول set تعیین می‌کند که داده ورودی باید جای کدام داده قدیمی نوشته می‌شد.



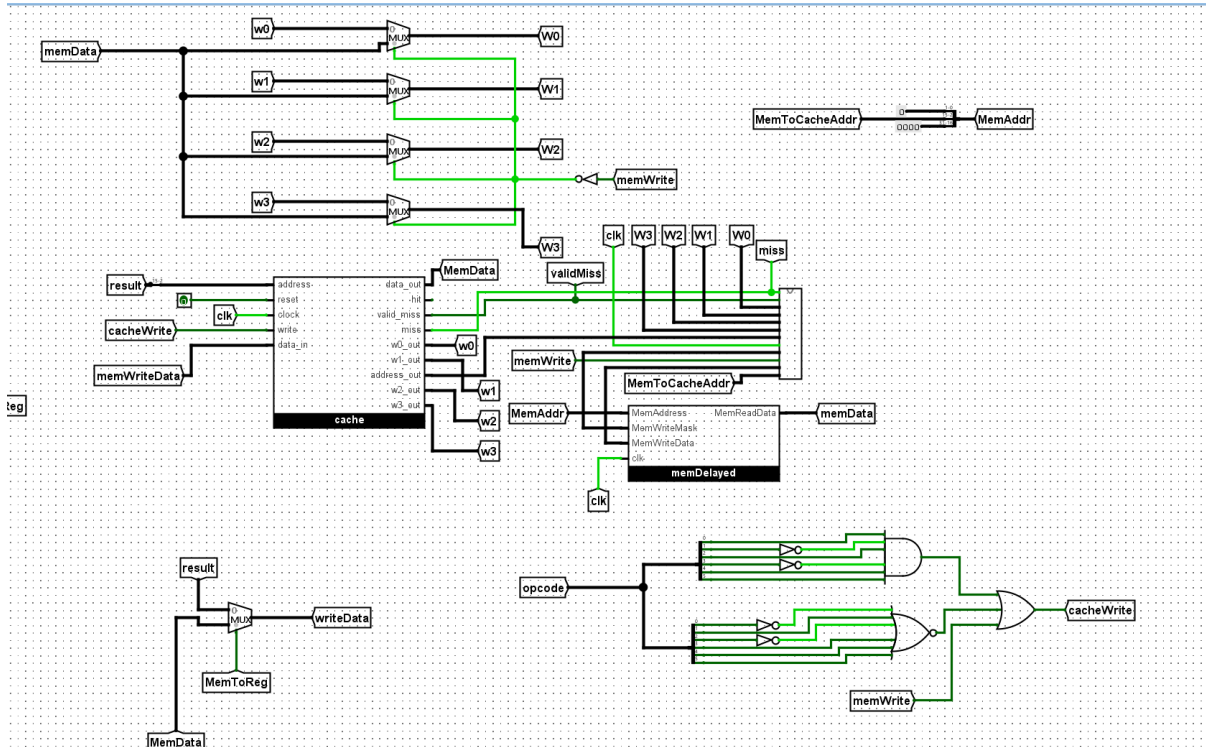
۴- مازول MemAddressGen تولید کننده آدرس یک بلاک



۵- مازول MemToCache که برای جایگزینی داده به کار می‌رود.

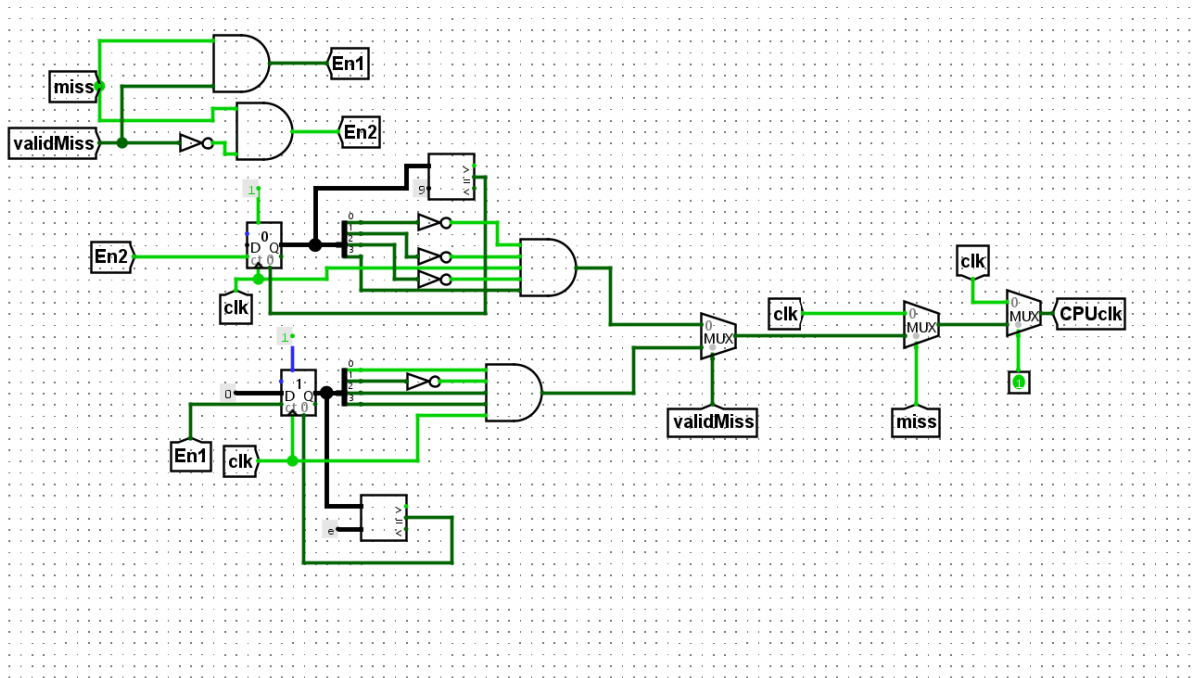
۴ جایگذاری Cache در مدار اصلی

در ادامه، مطابق با توضیحات داده شده در گروه و داک پروژه، ماژول Cache جایگزین حافظه در datapath طراحی شده در فاز ۱ شد. به این صورت که تمام ورودی‌ها (آدرس، سیگنال write و دیتا ورودی) به جای مموری به کش داده شده‌اند. همچنین برای ارتباط کش با memDelay ۴ ورود از طریق ماژول با مموری در ارتباط خواهند بود:



ع-ارتباط بین ماژول cache، MemToCache و memDelayed

همچنین با توجه به سیگنال‌های miss و valid miss تولید شده در کش، کلاک cpu تولید خواهد شد، به این صورت که cpu در صورت ۱ بودن miss و ۱ بودن valid miss به مدت ۱۳ کلاک، ۱ بودن miss و ۰ بودن valid miss ۸ کلاک و در صورت ۰ بودن miss ۰ کلاک منتظر می‌ماند.



۷- به دست آوردن کلاک cpu با سیگنال های cache

۵ نتیجه گیری

در این فاز، آموختیم تا یک حافظه نهان به صورت 2-way set associative طراحی کرده و با چالش‌های مربوط به اختلاف سرعت پردازش در حافظه و پردازنده و همچنین چگونگی انتقال داده میان حافظه، کش و پردازنده آشنا شدیم. در نهایت نیز با چگونگی کارکرد سیاست جایگزینی True LRU آشنا شده و آنرا پیاده‌سازی کردیم.