

گزارش فاز دوم پروژه معماری کامپیوتر - گروه Mascarpone

محمد مهدی قیدی - ۹۸۱۰۵۹۷۶

حسین آقایی - ۹۸۱۰۵۶۱۹

میلاد سعادت - ۹۸۱۰۰۴۴۲

پرهام چاوشیان - ۹۸۱۰۰۱۱۸

در این فاز به اضافه کردن یک حافظه نهان از نوع direct mapped به mips_core که در فاز پیش طراحی اش را کامل کرده بودیم پرداختیم. این ماژول با نام cache.sv به src اضافه شده است.

در این ماژول سیگنال هایی از جمله cache_word_in (برای ورودی گرفتن داده روی ماژول)، data_addr (برای دسترسی به آدرسی از مموری که دیتای مورد نظر در آن جا نوشته شده)، cache_we (برای write_enable شدن و اینکه امکان نوشتن روی حافظه نهان به وجود بیاید)، wait_signal (با توجه به اینکه خواندن و نوشتن روی مموری عملیاتی است که چند کلاک زمان میبرد مجبور شدیم این سیگنال را به حافظه نهان وصل کنیم تا در صورتی که این سیگنال روشن بود متوجه شویم که کش نباید کاری انجام دهد). همچنین همانطور که انتظار می رود سیگنال های clk و rst_b هم به ورودی کش وصل می باشند.

همچنین در خروجی کش، داده ها را روی cache_word_out خروجی می دهیم. برای شرایط خاص، مثل حالتی که دو بلاک به یک بلاک کش map می شوند و باید قبل از بازنویسی روی آن بلاک، بلاک قبلی را در مموری بنویسیم، سیگنال های mem_fetch و mem_write و write_mem_addr و fetch_mem_addr را تعریف کرده ایم که در شرایط مختلف از آنها استفاده می کنیم. (بسته به valid بودن، dirty بودن و میچ شدن یا نشدن tag بیت ها استفاده می کنیم. این موارد در کد قابل مشاهده و درک هستند و سعی شده کد تا حد امکان قابل فهم زده شده باشد.)

برای مثال اگر valid بیت مربوط به یک بلاک صفر باشد در آن بدون مکث و پردازش بیشتر مینویسیم. اگر اینطور نباشد، اگر dirty بیت ۱ باشد بلاک فعلی داخل کش را در مموری مینویسیم و داده جدید را به کش میاوریم و در حالتی که dirty صفر باشد مستقیم مینویسیم.

همچنین با توجه به اینکه ماژول کش به cpu اضافه شده است و تاخیر خواندن/نوشتن از مموری هم در این فاز شبیه سازی شده است، مجبور شدیم تاخیر کلاک را با یک متغیر counter در منطق mips_core شبیه سازی کنیم. روش کار هم بدین

صورت است که برای هر کدام از write و fetch به تنهایی ۶ کلاک و برای ترکیبشان با هم ۱۱ کلاک وقفه میاندازیم که این مورد را با استفاده از counter و مقدار دهی آن در زمانی که سیگنال های mem_write و mem_fetch فعال شده است انجام میدهیم و در هر کلاک یک واحد از آن کم میکنیم. سیگنال های ورودی حافظه نهان مانند cache_we و data_addr و cache_word_in و mem_write با استفاده از وضعیت state عوض می شوند و این state با توجه به همان counter و سیگنال های mem_fetch و mem_write تعیین می شود تا همواره مقدار درستی را بدهند.