

حافظه‌ی نهان

این حافظه، با گنجایش ۲۵۶ بایت، با ساختار نگاشت مستقیم در فایل "cache.v" پیاده‌سازی شده‌است. دارای ۶۴ خط می‌باشد که هر خط گنجایش ۴ بایت داده را دارد: بنابراین برای دسترسی به هر خط، ۶ بیت داده $\log_2 64 = 6$ کافی خواهد بود. با فرض اینکه طول آدرس حافظه‌ی اصلی برابر با ۱۲ بیت داده باشد، و این که اندازه‌ی هر خط برابر با ۴ بایت داده باشد و برای offset، بیتی در نظر بگیریم (اندازه‌ی هر word را ۴ بایت در نظر گرفته ایم)، اندازه‌ی tag برابر با ۶ بیت خواهد شد. در بلوک initial فایل داده شده، تمامی بیت‌های valid و tag را در ابتدا، برای تمامی خانه‌ها با ۰ مقدار دهی شده است.

حافظه‌ی RAM

این حافظه، با اندازه‌ی تقریبی ۱۶ کیلوبایت، در فایل "ram.v" پیاده‌سازی شده‌است. دارای 2^{12} خط که هر کدام، گنجایش ۴ بایت داده را دارند، می‌باشد، بنابراین طول آدرس برای آن ۱۲ بیت خواهد بود و هر کلمه دارای ۴ بایت اندازه می‌باشد.

ارتباط حافظه‌ی Cache و RAM

در فایل "cache_and_ram.v" ارتباط دو ماژول بالا، به صورت سنکرون با کلاک، توصیف شده‌است بدین صورت که مشخصات درخواست‌های به Cache، در رجیسترهایی ذخیره می‌شوند، اگر در لبه‌ی بالارونده‌ی کلاک، مقدار این رجیسترها تغییر کند، Cache با بررسی مقدار tag و index جدید، اقدام مناسب را انجام خواهد داد، توجه شود که در این حافظه‌ی نهان از مکانیزم write through برای نوشتن hit ها در cache بهره می‌برد، بنابراین برای آپدیت کردن داده‌هایی که در حافظه‌ی نهان یافت می‌شوند، علاوه بر نوسازی آن‌ها در cache باید به نوسازی آن‌ها در RAM نیز پردازیم. بنابراین برای زمان‌هایی که نسبت خواندن از حافظه بسیار بیشتر باشد، cache ما کارا تر خواهد بود.

یک سناریو که در آن Cache با دو راه بهتر از کش نگاشت مستقیم عمل می‌کند

یک مزیت که کش با دو راه نسبت به کش نگاشت مستقیم می‌تواند داشته‌باشد، دیرتر قربانی کردن داده در حافظه است: از طرفی داشتن set کوچک‌تر، به نوعی موجب قربانی دادن بیشتر خواهد شد. حالتی را در نظر بگیرید که در آن، کاربر بخواهد از دو خانه با tag های متفاوت ولی set های یکسان، به صورت متوالی استفاده کند، بنابراین این دو داده، محل یکسانی در حافظه اشغال خواهند کرد، حال اگر از کش با ساختار دوم استفاده کنیم تنها ۲ miss در ابتدای کار خواهیم داشت و سپس می‌توانیم به ۲ داده دسترسی داشته‌باشیم، این در حالی است که در کش با نگاشت مستقیم، به اندازه تعداد دفعات استفاده از دو داده، miss خواهیم داشت.