

معماری کامپیوتر
گزارش فاز دوم
پروژه استاد سربازی



امیرمهدی کوششی
محمد صادق مجیدی
احسان موفق
پویا اسمعیلی آخوندی

در این فاز، کش طراحی شده از نوع Direct Mapping است. همان طور که می دانیم، کش حافظه ای شامل بلاک ها است که میان پردازنده و حافظه ی اصلی قرار دارد.

در اینجا، هر بلاک کش شامل یک ورد است. در نتیجه تعداد ورد هایی که در حافظه ی کش نگه داری می شوند، با تعداد بلاک های آن یکسان است. از آنجایی که به صورت پیشفرض، حافظه ی کش شامل 2 به توان 13 بایت است و هر ورد، 4 بایت دارد، لذا حافظه ی کش به طور پیشفرض حافظه ی کش 2 به توان 11 بلاک دارد.

در صورتی که حافظه ی مورد نظر برای دسترسی به یک بایت از حافظه را 32 بیتی بگیریم، می دانیم که 2 بیت کم ارزش نشان دهنده ی بایت مورد نظر از ورد است. 11 بیت بعدی آن (به طور پیشفرض) نشان دهنده ی ورد مورد نظر (بلاک مورد نظر) از بین تمامی بلاک های حافظه ی کش است. 19 بیت دیگر آن (به طور پیشفرض) بیت های تگ ما خواهند بود. مشخصا برای پیاده سازی باید برای هر بلاک، بیت valid و بیت dirty نیز در نظر بگیریم که این را با کمک آرایه پیاده سازی کردیم.

همچنین، برای پیاده سازی تاخیری که میان ارسال داده از حافظه اصلی به کش وجود دارد، دو متغیر تعریف کردیم. یکی از این متغیر ها برای تشخیص hit شدن داده ی داخل کش به هنگام خواندن داده است. در این حالت، در صورت miss شدن، این متغیر شروع به شمارش می کند (set می شود به مقدار ثابتی) و تا 5 کلاک بعدی که داده برای کش آماده می شود، منتظر می ماند و وقتی داده آماده شد، این متغیر ریست می شود و داده ی مورد نیاز ما در کش وجود خواهد داشت. به طریق مشابه، وقتی داده ای در کش که dirty شده است، لازم است که به حافظه ی اصلی بازگردد، متغیر دوم نقش این عمل را دارد که 5 کلاک منتظر می ماند تا داده به طرز صحیحی به حافظه ی اصلی ارسال شود و پس از آن، داده می تواند از

حافظه ی اصلی به کش انتقال یابد. (که در این حالت هم احتمالا متغیر اول
نیاز دارد دوباره 5 کلاک بشمارد تا داده به طرز صحیحی در کش قرار
گیرد.)