

\* نحوه تولید آدرس حافظه:

برای تولید آدرس ابتدای حلقه `for` که 32 بار تکراری شود با احتمال 50%:

عدد 2 و با احتمال 50% عدد 1 را تولید می کنیم و با کنار هم قرار دادن این 32 رقم

یک رشته ی 32 بیتی می سازیم. این کار را 100 بار با یک حلقه ی `for` انجام

می دهیم تا 100 رشته 32 بیتی داشته باشیم و آن ها را در یک `ArrayList`

ذخیره می کنیم. حالا برای اینکه از روی این 100 عدد 500 عدد برای ذخیره در فایل

تولید کنیم در یک حلقه `for` ابتدا خود عدد را به یک `ArrayList` جدید اضافه می کنیم،

پس یک عدد تصادفی در بازه 0 تا 3 تولید می کنیم:

1. اگر عدد تصادفی 0 بود به عدد چیزی اضافه نمی کنیم و به `ArrayList` آن را `Add` می کنیم

2. اگر برابر 1 بود، بیت آخر عدد را `not` می کنیم و حالا این عدد را `Add` می کنیم

3. اگر برابر 2 بود، بیت یکی مانده به آخر را `not` می کنیم، `Add` می کنیم

4. اگر برابر 3 بود، بیت دوماه آخر را `not` می کنیم و `Add` می کنیم

5. اگر حالت 1 اتفاق بیفتد و تابع هم جواری خاصی تولید کرده ایم چون خود عدد تکراری شود

و اگر حالت 2 و 3 و 4 اتفاق بیفتد هم جواری خاصی تولید کرده ایم چون اعدادی تولید می کنیم که

1، 2 و 4 تا با اعدادمان اشتباه دارند.



پس برای هر عدد ابتدا عددش را به `ArrayList` جدید `Add` می کنیم سپس 4 تا عدد به با عددهای  
هم عواری مکانی داریم تولید می کنیم و بلافاصله بعد از عدد مان به `ArrayList` ، `Add` می کنیم  
در نهایت برای اینکه تولید اعدادمان را به `for` حلقه `for` به 100 بار تکرار  
می شود به حلقه حتماً از `ArrayList` با ما به حلقه حتماً می تغییر از آن ، به صورت  
و تمام حلقه ها می کنیم .

حافظه قربانی victim

"victim cache"

حافظه قربانی victim یک حافظه قربانی کوچک است که معمولاً به صورت تمام انجمنی

بپایه سازی می شود و تمام بلوک های اخراج شده از خط منظرش در cache را

ذخیره می کند. این حافظه قربانی به منظور کاهش تعداد miss ها به حافظه قربانی نگاشت

مستقیم اضافه می شود و هر خط آدرسی که از حافظه قربانی سطح اول خارج می شود در

victim نگه داشته می شود.

تعداد حافظه قربانی victim :

① cache miss / victim hit ← بلوک مورد نظر در حافظه قربانی victim

یافت می شود و جای آن با بلوک که در cache ، میس (miss) شده بود عوض می شود

و این بلوک جدید از این به بعد به عنوان MRU در victim شناخته می شود



⑤ cache miss / victim miss - به طور مورد نظر از حافظه اصلی آلوده می شود

دیتایی که از cache خارج می شود وارد victim cache می شود

آمارهای کامپیوتر نشان داده اند که اضافه کردن victim میزان miss rate

را بین ۱۰ تا ۱۰۰ درصد در موارد مختلف کاهش می دهد.



## "Trace cache"

حافظه یان Trace

حافظه یان trace یک حافظه یان ویژه است که برای ذخیره یان دستور العمل ها را

که می توان Trace را (اثر) معروفه ذخیره می کند.

حافظه یان Trace به این روش یانی باند دستور کار و هم چنین گام ها می توان معرفی

یک می کند و این کار را با استفاده از ریت سازی اثر (trace) دستور های که قبل

از حافظه آورده شده اند، انجام می دهد.

بردارنده های سوپر اسکلر برای کارایی بهتر در یک دوره جدید دستور را تقاضا می کند

دستور های که باید آورده شوند همیشه در مکان های متوالی در حافظه قرار دارند

(به دلیل دستور های jump و branch)، بنابراین بردارنده های یک منطبق و

سخت اقرار اضافی نیاز دارند تا دستور های که پشت سر هم می آیند را حاضر کند.

برای مثال ۴ دستور العمل زیر را که مربوط به یک حلقه `for` ساده اند در نظر بگیرید:



این بلوک ها در حافظه به صورت A, B, C و D، متوالی ذخیره می شوند. اثر شاخه ای D

به عنوان "Not-taken" پیش بینی شود، واحد `fetch` می تواند بلوک های A, B, C و

را که در حافظه متوالی اند بیاورد اما اثر شاخه ای D به عنوان "taken" پیش بینی



شماره واحد Fetch اندیشه‌های D و B و A به حافظه می‌تواند بسته شود

این آدرس این روش‌های غیرمستقیم در یک دوره بسیار مشکل است. بنابراین در

مواردی مشابه این حالت حافظه بان trace به کمک برداشته می‌آید

\* منطق hit و miss :

خطوط اثر (trace) بر اساس شماره برنامه‌ی اولین دستور و مجموعه‌ای از بین می‌جا

یافته (branch predictions) در حافظه بان (cache) ذخیره می‌شوند

این ذخیره سازی کمک می‌کند مسیرهای مختلف اثر (trace) که با آدرس یکسانی

شروع می‌شوند دستاویز branch برای آن‌ها متفاوت است، مرتب شوند.

در این دستور، شماره برنامه (program counter) کنونی به همراه مجموعه‌ای از

branch predictions جای می‌شود تا مقدار hit یا مشخص شود.

اگر hit وجود داشته خط اثری به Fetch Unit رسانده می‌شود که لازم نیست

برای بایش دستور را به cache یا حافظه اصلی برود. Cache به تأمین کردن

واحد Fetch ارائه می‌دهد تا زمانی که خط trace پایان یابد.

اگر miss وجود داشته یک trace جدید شروع به سامانه شدن می‌کند.

\* منابع :  
① این رشته شدن دستور را از اصلی بین instruction cache, Trace cache

و هم چنین در خود Trace cache

② عدد بیت توان و پیچیدگی سخت افزار



+ نتیجه گیری:

میانگین نرخ موفقیت برای ۵ فایل در درون و عبور specialized cache

در اعداد مختلف حدود ۰.۱۴۸ و با عبور specialized cache حدود

۰.۱۸۵ است. این تفاوت نشان می دهد که با اضافه شدن

victim cache مقدار نرخ موفقیت (hit Ratio) افزایش یافته،

چون victim cache به یک حافظه پتان Direct Mapped آمده

و بزرگ های خارج شده از آن را ذخیره می کند. این عمل از بروز مجدد

در حافظه miss penalty دارند جلوگیری می کند در نتیجه مقدار

miss ها کاهش می یابد.