

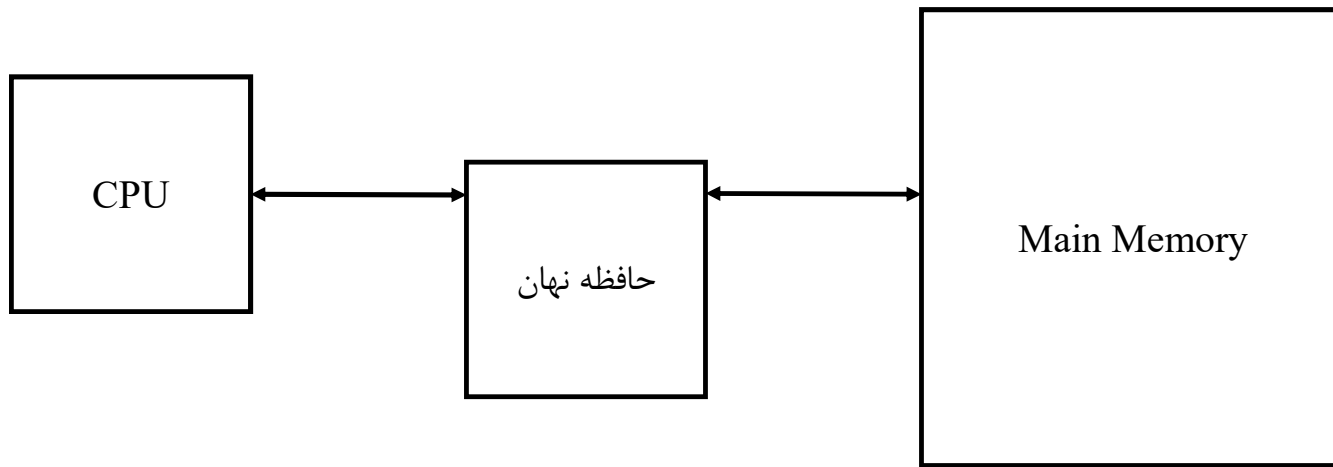
معماری کامپیوتر

جلسه هشتم: حافظه نهان-۲

حافظه نهان (Cache)



- در نظر گرفتن فضایی بین پردازنده و حافظه اصلی (مشابه جعبه کوچک کتابدار)
- باهدف ذخیره سازی بخشی از حافظه اصلی جهت دسترسی سریع تر



حافظه نهان (Cache)



- خاصیت هم‌جواری (Locality):
- هم‌جواری مکانی (Spatial Locality)
- هم‌جواری زمانی (Temporal Locality)
- لحاظ کردن این خاصیت در طراحی حافظه نهان

حافظه نهان (Cache)



- سوال های مهم:

- داده ها را چگونه از حافظه اصلی به حافظه نهان ببریم؟

- داده های جدید را چگونه در حافظه نهان جایگزین کنیم؟

- برای پاسخ به این دو سوال، دو بحث اساسی مطرح می شود:

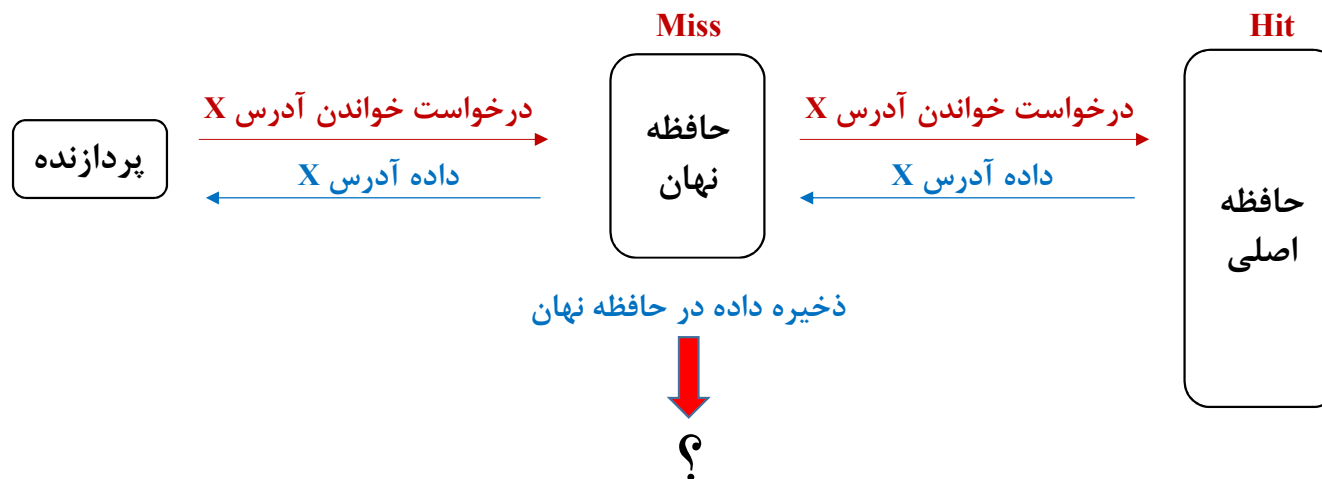
- سیاست جای دهی (Placement Policy)

- سیاست جای گزینی (Replacement Policy)

سیاست‌های جای‌دهی (Placement Policy)



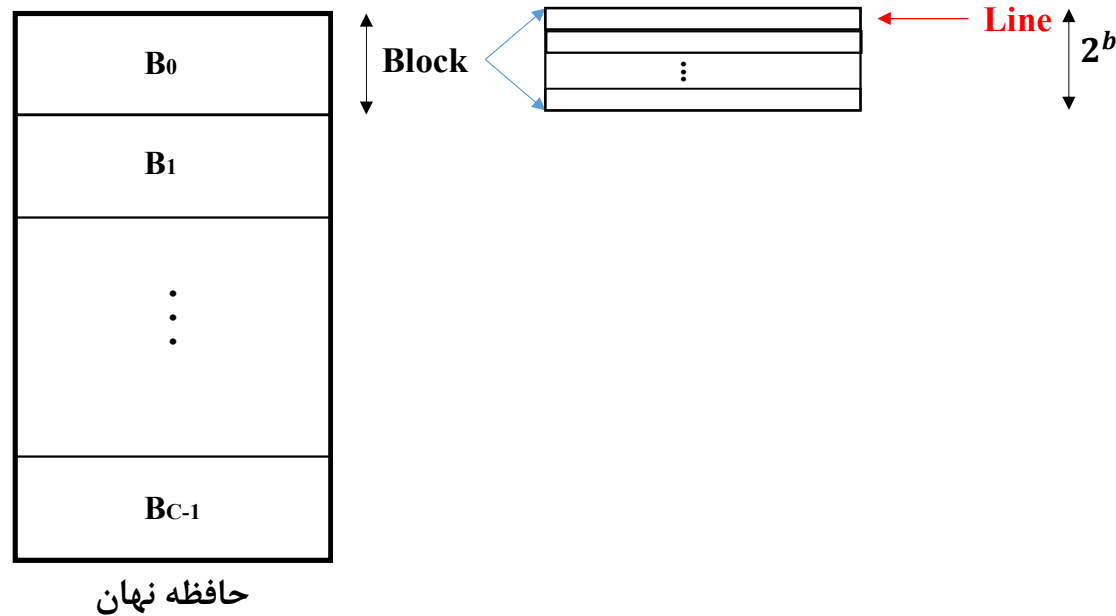
- تعیین آدرس داده در حافظه نهان براساس آدرس آن در حافظه اصلی (همجواری زمانی)



سیاست‌های جای‌دهی (Placement Policy)



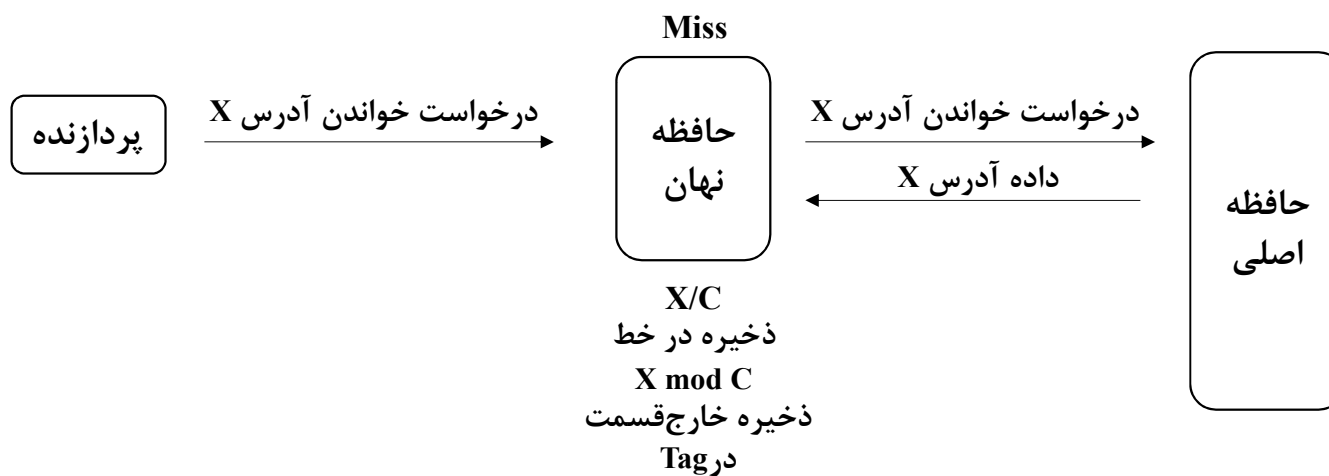
- پیاده‌سازی همجواری مکانی



حافظه نهان نگاشت مستقیم



- سیاست جایدهی



حافظه نهان نگاشت مستقیم



- قالب آدرس حافظه اصلی:



حافظه نهان نگاشت مستقیم (مثال)



• فرضیات: آدرس حافظه ۳۲ بیتی، حافظه نهان با ۱۶ بلوک ۴ کلمه‌ای

• سوال: داده آدرس ۱۸۱ حافظه اصلی (XYZ) در کدام خانه حافظه نهان ذخیره می‌شود؟

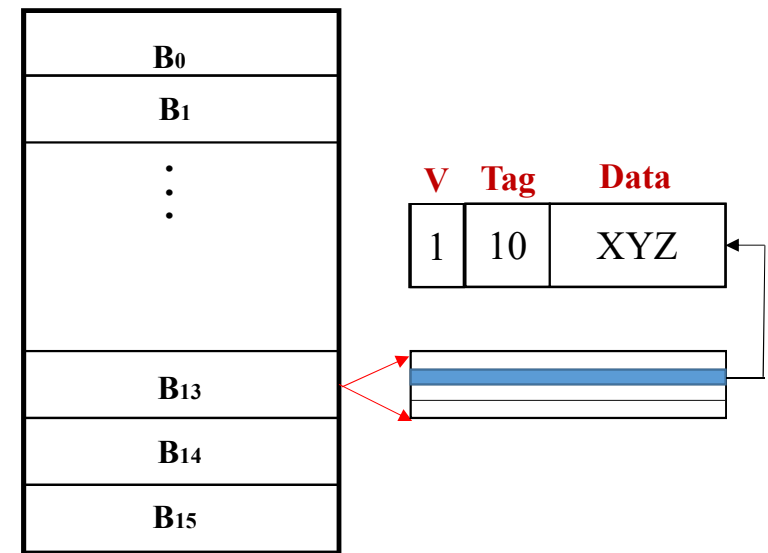
$$181 = (10110101)_b$$

$$B = 4 = 2^2 \longrightarrow \text{Offset} = 2 \text{ bit} : 01 = 2$$

$$C = 16 = 2^4 \longrightarrow \text{Index} = 4 \text{ bit} : 1101 = 13$$

Tag: 10

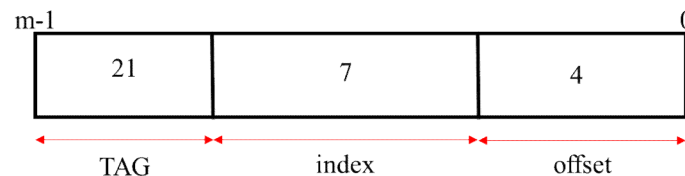
حافظه نهان



مثال



- درخواست خواندن از حافظه به آدرس 000B3A4F آمده است و مشخصات روبرو را داریم (نوع



طراحی: نگاشت مستقیم):

- **حجم حافظه اصلی، حافظه نهان و محل استقرار داده در حافظه نهان؟**

حجم حافظه نهان: 2^{11} بایت

حجم حافظه اصلی: 2^{32} بایت

آدرس: 00000000000010110011101001001111

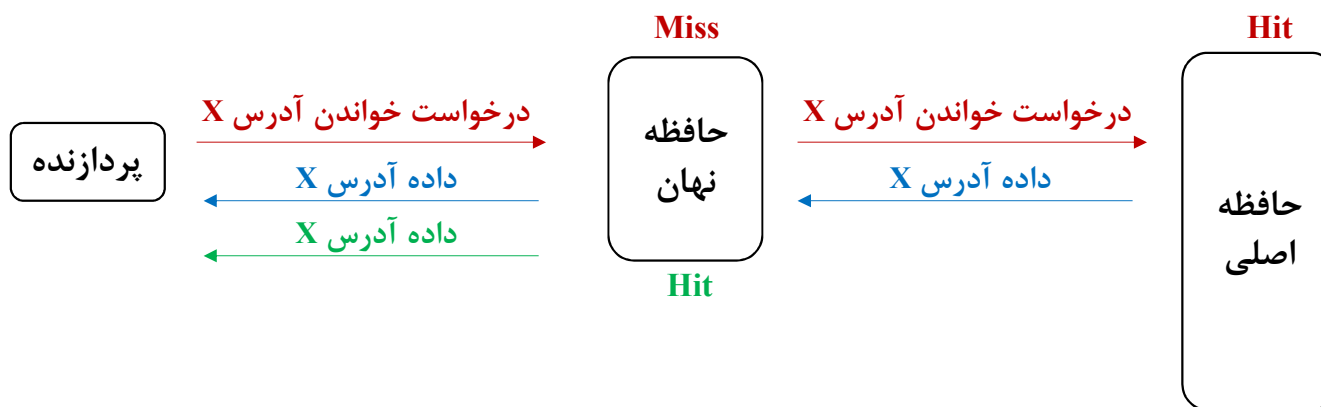
محل استقرار داده: کلمه ۱۵ از بلوک ۳۶ ام حافظه نهان

حافظه نهان نگاشت مستقیم



• مدیریت درخواست خواندن از حافظه:

- ارسال درخواست به حافظه نهان
- Miss: ارسال درخواست به حافظه اصلی و بروزرسانی حافظه نهان
- Hit: ارسال پاسخ درخواست



حافظه نهان نگاشت مستقیم



- مدیریت درخواست نوشتن در حافظه:

- ارسال درخواست به حافظه نهان

- Miss: ارسال درخواست به حافظه اصلی و بروزرسانی حافظه نهان

- Hit: بروزرسانی داده در حافظه نهان و انتقال تاثیر به حافظه اصلی

- Write Back

- Write Through

حافظه نهان نگاشت مستقیم



- نگاشت هر بلوک از حافظه اصلی به نقطه مشخص در حافظه نهان
- امکان نگاشت دو بلوک به یک فضا از حافظه نهان
- جایگزین شدن متوالی این دو بلوک بر اثر درخواست‌های متوالی
- مراجعات متعدد به حافظه اصلی، افزایش زمان دسترسی و پدیده thrashing
- مثال: درخواست‌های متوالی به خانه‌های ۰ و ۸ حافظه اصلی با فرض داشتن حافظه نهان هشت خطی
- هفت خانه حافظه نهان خالی و یک خانه دائما جایگزین شده و miss رخ می‌دهد

حافظه نهان نگاشت مستقیم



• مزیت:

- سادگی پیاده‌سازی
- کارایی بالا به دلیل سرعت پیدا کردن داده

• عیب:

- مکانیزم همجواری زمانی ضعیف
- نگاشت مستقیم هر بلوک از حافظه اصلی به نقطه مشخص در حافظه نهان (پدیده thrashing)
- راه‌حل: تغییر شیوه طراحی

حافظه نهان انجمنی



• برای حل مشکل Thrashing

- نگاشت مستقیم و محدودیت آدرس را برمی داریم
- مجاز بودن نگاشت هر بلوک حافظه به هر آدرسی در حافظه نهان
- برای جستجوی داده خاص، مقایسه همزمان تمامی خانه‌های حافظه نهان با داده موردنظر

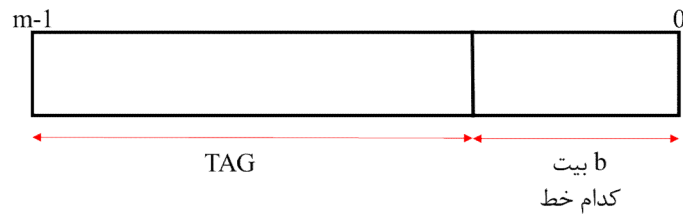
• این طراحی از حافظه نهان: Fully Associative cache

حافظه نهان انجمنی



Fully Associative cache •

- هر خانه حافظه اصلی می تواند با هر خانه ای از حافظه نهان همبسته شود
- کاربردی بودن برای حافظه های نهان کوچک
- نیاز به حافظه CAM برای مقایسه محتوا



- بسیار پرهزینه
- توان مصرفی بالا
- مساحت زیاد

حافظه نهان انجمنی



• راه حل میانی DM و FA: Set Associative Cache

• مشابه نگاشت مستقیم ولی

0	<div>T</div> 0	<div>T</div> 8
1	<div>T</div>	<div>T</div>
2	<div>T</div>	<div>T</div>
3	<div>T</div>	<div>T</div>

• طول حافظه نهان را $1/k$ کرده و عرض آن را k برابر می کنیم

• در شکل روبرو، به جای ۸ خط یک بایتی، ۴ خط دو بایتی خواهیم داشت

• این طراحی: 2 way set associative

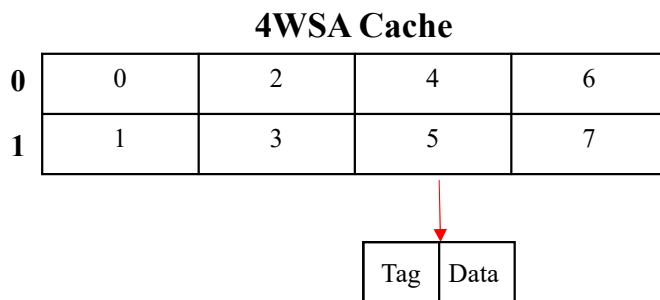
• بدین ترتیب توانایی ذخیره حافظه نهان در هر خط k برابر می شود

• هریک از این دوبخش tag خاص خود را دارد

حافظه نهان انجمنی



• راه حل میانی DM و FA: Set Associative Cache



• مشابه نگاشت تمام انجمنی

• ذخیره داده در هریک از خانه‌های یک خط

• جستجوی خانه‌های هر خط به صورت موازی

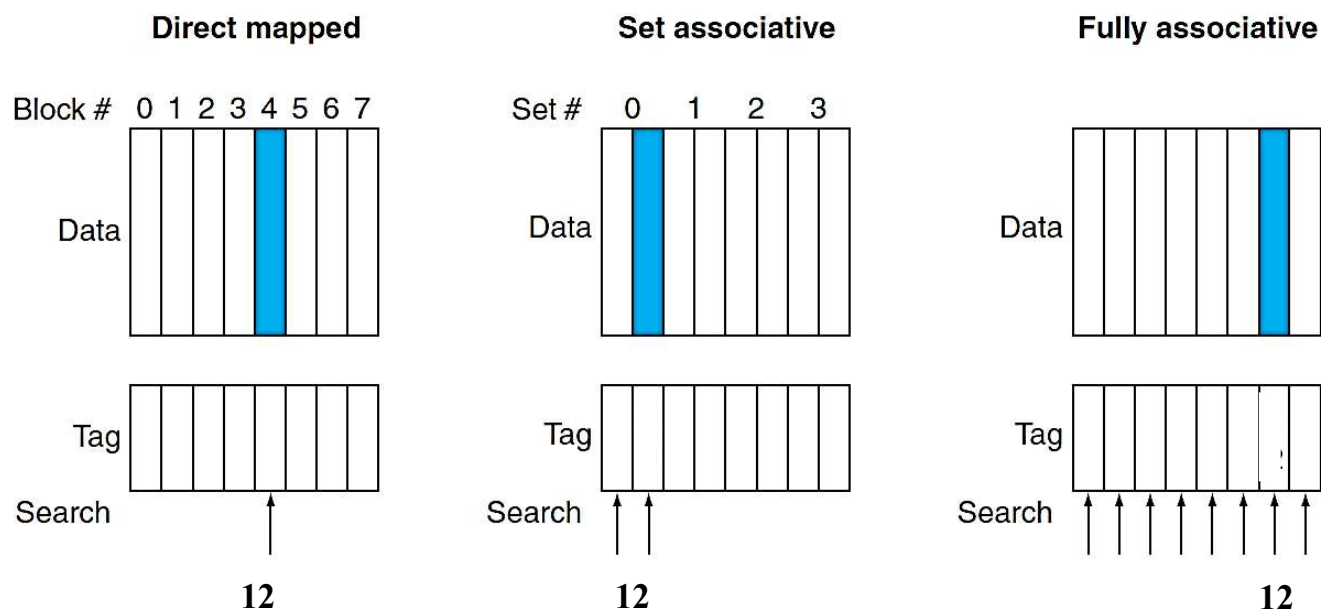
• افزایش کارایی در اثر افزایش k

• بیشینه k برابر با سایز حافظه نهان که به طراحی تمام انجمنی می‌رسیم

$$\# \text{ Set in Cache} = (\text{Memory Address}) \bmod (\text{Number of sets}(C/k))$$

$$\text{Tag in Cache} = (\text{Memory Address}) \div (\text{Number of sets}(C/k))$$

سیاست‌های جایدهی حافظه نهان



سیاست‌های جایدهی حافظه نهان



One-way set associative (direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

جایدهی در حافظه نهان (مثال)



- جلسه استاد با دانشجویان در کلاسی با ۳۰ صندلی
- نوبت هر دانشجو که باشد، نامش صدا زده می شود
- اگر نباشد جستجو شده و به کلاس می آید که زمان بر است
- ممکن است هر دانشجو چندین بار صدا زده شود یا اصلا صدا زده نشود



جایدهی در حافظه نهان (مثال)

- سناریو ۱) صندلی‌ها از ۰ تا ۲۹ ردیف شماره‌گذاری شوند با ورود هر دانشجو باقی‌مانده سه رقم آخر شماره دانشجویی به ۳۰ محاسبه شده و روی آن صندلی می‌نشینند
 - در صورت پر بودن، دانشجوی قبلی به بیرون می‌رود
- مشابه حافظه نهان دسترسی مستقیم



جایدهی در حافظه نهان (مثال)

• سناریو ۲) صندلی‌ها را به صورت زوج در هر ردیف از ۰ تا ۱۴ شماره گذاری شوند و با ورود هر

دانشجو باقی مانده سه رقم آخر شماره دانشجویی به ۱۵ محاسبه شده و روی آن صندلی

می‌نشیند

• در صورت پر بودن هر دو صندلی این ردیف، دانشجوی قبلی به بیرون می‌رود

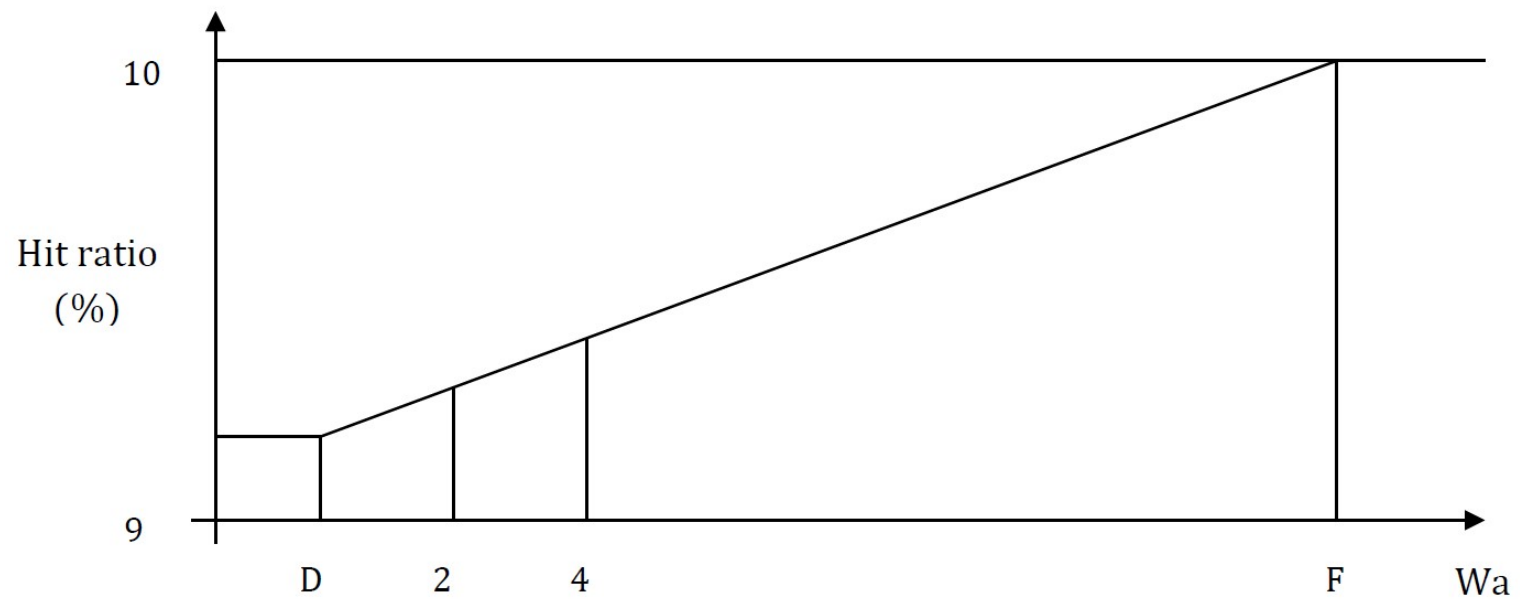
• مشابه حافظه نهان two way set associative

جایدهی در حافظه نهان (مثال)



- سناریو ۳) هر دانشجویی که وارد شد روی هر صندلی که خالی بود می‌نشیند اگر خالی نبود با یکی از دانشجویانی که قبلاً آمده جایگزین می‌شود
- مشابه حافظه نهان تمام انجمنی

جایدهی در حافظه نهان



قالب آدرس در حافظه نهان

