Cpi x1 (+arta)

Address fields length x2 ✓

K-way associative x1✓

AMAT x1 = Average memory access time

Amdahl law + Zaman Ejra x3

Memory x2 (Hefzi)

Hefzi x3

multilevel cache x1

# JAVAD

## نمونه سوالات کنکور جلسه اول

سوال1) (متوسط رو به سخت، دقتی، Address fields length ) در یک سیستم، یک حافظه اصلی 16GB و یک حافظه نهان 128KB داریم. بلاک های حافظه نهان حاوی 4 Word هستند و هر Word معادل 2B است. اگر این حافظه نهان از مکانیزم Two way set associative بهره ببرد آنگاه اندازه فیلد های offset —-- بیت و index —-- بیت است؟

1. Offset = 2, index = 13
2. Offset = 4, index = 13
3. Offset = 2, index = 26
4. Offset = 4, index = 26

Offset : log 4 = 2 bit

#block = (128 \* 2^10 ) / (2 \* 4) = 2 ^ 14

Index = log(2 ^14 / 2) = log 2^13 => 13 bit

سوال 2) (آسون، دو بار فربه جمله رو تکرار کرده بیاریمش گناه دارن، Address fields length ) اگر با ثابت نگه داشتن اندازه بلاک مقدار k را در مکانیم k-way set associative بالا ببریم آنگاه تعداد بیت های مورد نیاز آدرسی دهی برای فیلد tag —-- میابد

1. افزایش
2. کاهش
3. بدون تغییر

تعداد بیت های index کاهش پیدا میکند ولی برای مشخص کردن بلاک مورد در هر ایندکس به تعداد بیت tag بیشتری نیاز داریم

~~سوال 3)~~ (سخت دقتی پتانسیل سوتی داره، Address fields length) در یک سیستم، یک حافظه اصلی 16MB و یک حافظه نهان 256KB داریم. بلاک های حافظه نهان حاوی 2 Word هستند و هر Word معادل 4B است. اگر این حافظه نهان از مکانیزم Four way set associative بهره ببرد آنگاه سربار موجود برای ذخیره tag چند کیلوبیت است؟

Offset : log2 = 1 bit

#block = (256 \* 2^10 ) / (2 \* 4) = 2 ^15

Index = log(2 ^15 / 4) = log 2^13 => 13 bit

Address length = log ( 16 \* 2 ^20) = 24 bit

Tag = 24 - ( 13 +1 ) = 10 bit

Overhead = 2 ^ 15 \* 10 bit = 320 Kb

سوال4) (سخت دقتی و زمان گیر اگر گیج بازی در میارن، ،multilevel cache AMAT) یک پردازنده دارای دو سطح حافظه نهان L1, L2 می باشد که hit rate آن ها به ترتیب 90% و 80% می باشد. زمان دسترسی به L1 برابر با 10 نانو ثانیه، زمان دسترسی به L2 برابر با 150 نانو ثانیه و زمان دسترسی به حافظه اصلی برابر 300 نانو ثانیه است. متوسط زمان دسترسی به حافظه توسط کدام فرمول به درستی نمایش داده می شود؟

1. 10 + (0.1 \* [150 + (0.2 \* 300)])
2. (0.8 \* 150) + (0.2 \* [150 + (0.2 \* 300)])
3. (0.9 \* 10) + (0.1 \* [10 + (0.8 \* 150) + (0.2 \* (150 +300)])
4. موارد 1 و 3

AMAT = hit time + miss rate \* miss penalty

## سوالات طرح شده

سوال 1) (متوسط، Amdahl law Zaman Ejra) فرض کنید شما تنها میتوانید 40% از کد یک برنامه را بهبود ببخشید. آیا رسیدن به speedup=1.8 امکان پذیر است؟

1. بله
2. خیر

جواب: امکان پذیر نیست زیرا این مقدار از بیشنه مقدار speedup با fraction = 0.4 بیشتر است.

طبق قانون ambdahl داریم:

Maximum speedup = 1/((1 - 0.4) + (0.4/infinity)) = 1/0.6 = 10/6 = 1.66

~~سوال 2)~~ ( آسون، اینو بدیم حال کنن،Cpi Zaman Ejra ) کدام یک از موارد از اجزای فرمول CPU Time است؟

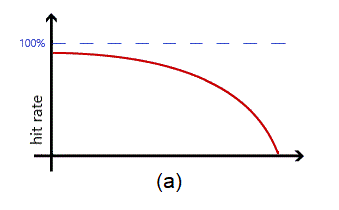
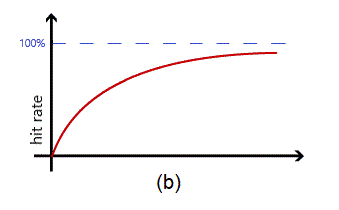
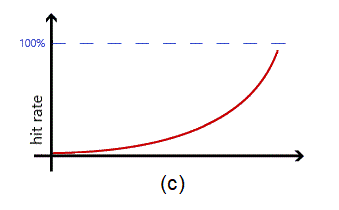
1. تعداد دستورات ( instruction count یا همان IC)
2. CPI
3. CPU frequency
4. همه موارد

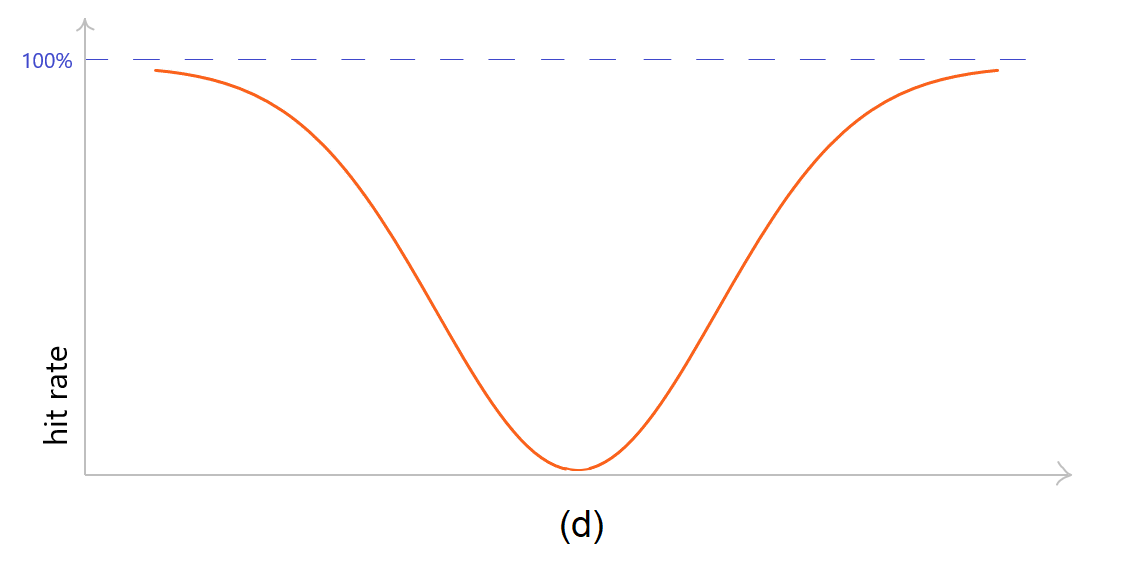
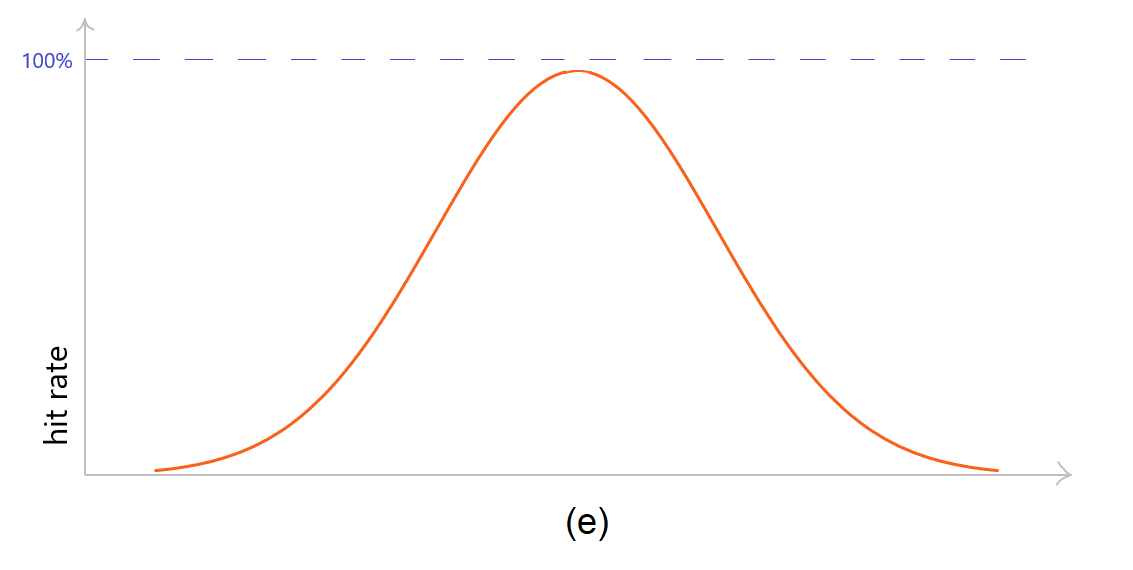
CPU time = IC \* CPI \* T = IC \* CPI / F

**Negar**

**سوال اول (تمرین 2) - متوسط - دقتی -** Hefzi **مفاهیم cache**

در مورد یک حافظۀ نهان نگاشت مستقیم با توجه به اصول همجواری زمانی و مکانی به سوالات زیر پاسخ دهید (از راست به چپ).





~~الف) فرض کنید در این حافظۀ نهان اندازۀ بلوک های حافظه ثابت است. با این فرض نمودار hit rate بر حسب اندازۀ حافظۀ نهان کدام یک از شکل های (a) تا (e) می باشد؟~~

ب) حال فرض کنید اندازۀ حافظۀ نهان ثابت است. با این فرض نمودار hit rate بر حسب اندازۀ بلوک های حافظۀ نهان کدام یک از شکل های (a) تا (e) می باشد؟

الف) e - b

ب) a - e

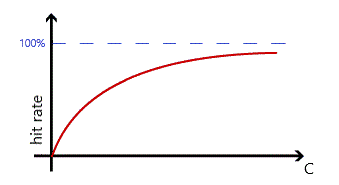
ج) d - c

د) b - b

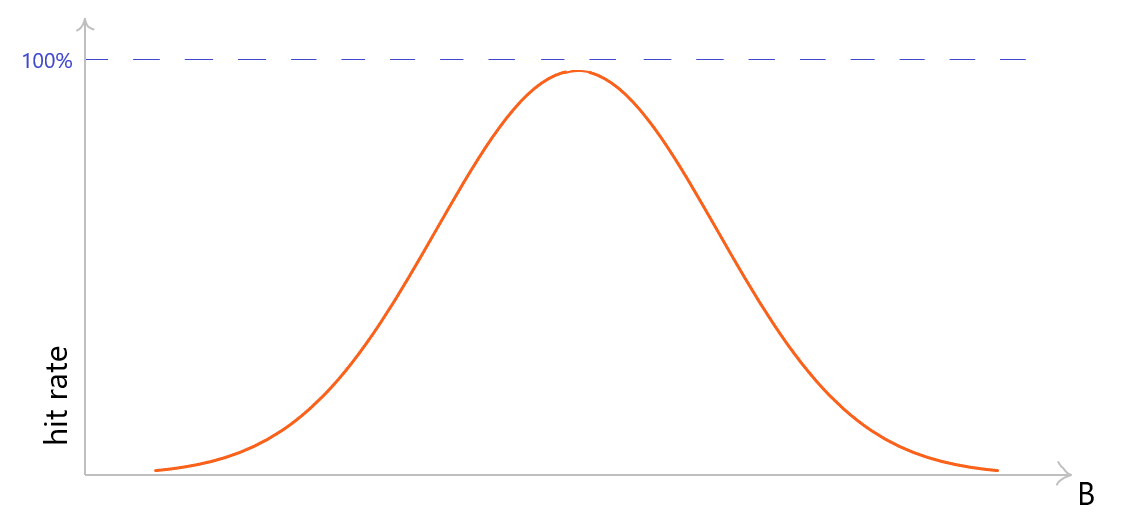
**پاسخ**

**گزینۀ الف** - اصل همجواری مکانی بیان می دارد اگر پردازنده دادهای را درخواست کرده به احتمال بالا دادهای نزدیک (از نظر محل قرارگیری در حافظۀ اصلی) به آن را نیز در آینده خواهد خواست. اساس بلوک بندی حافظۀ نهان نیز بر همین اصل است. به جای این که با هر رجوع به حافظه تنها دادهای که میخواهیم را در حافظۀ نهان ذخیره کنیم، داده های مجاور آن (که در بلوک یکسانی قرار دارند) را نیز به حافظۀ نهان می آوریم. با این کار تعداد مراجعه ها به حافظۀ اصلی کاهش یافته و عملکرد سیستم بهبود مییابد.

قسمت اول) با افزایش سایز حافظۀ نهان میزان hit rate نیز افزایش می یابد، زیرا حافظۀ نهان می تواند دادۀ بیشتری را در خود ذخیره کند تا در صورت نیاز به CPU بدهد. مقدار hit rate زمانی که اندازۀ حافظۀ نهان با حافظۀ اصلی برابر باشد میتواند به 100% برسد، هر چند که در عمل انجام چنین کاری توجیه ندارد و سعی میشود با کوچک نگهداشتن سایز حافظۀ نهان و استفاده از تکنیک هایی برای پیشبینی درخواستهای CPU این مقدار را به 100% نزدیک کنیم.



قسمت دوم) در نیمۀ اول نمودار میزان hit rate افزایش مییابد زیرا به اصل همجواری مکانی توجه بیشتری میشود (طبق توضیحات داده شده در ابتدای پاسخ). اما در نیمۀ دوم نمودار به همجواری زمانی توجه کمتری شده زیرا میزان overwrite زیاد شده و حافظۀ نهان نمیتواند دادههای مفید را برای مدت زیادی در خود نگه دارد و لذا میزان hit rate کاهش مییابد (میزان cache pollution زیادتر می شود و درون حافظۀ نهان دادۀ مفید نداریم). مقدار 100% برای این نمودار نیز یک مقدار مجانبی میباشد.



**سوال دوم (تمرین 2) - ساده - سریع - CPI**

فرض کنید در یک پردازنده از سیاست write-back استفاده می کنیم و مقدار base CPI برابر با 3 می باشد. به علاوه در حدود 15% از دستورات پردازنده مربوط به نوشتن در حافظه بوده و هر بار نوشتن در حافظه به طور متوسط 120 کلاک طول می کشد. وجود دستورات نوشتن بر حافظۀ اصلی پردازنده را چند برابر کندتر کرده است؟

الف) 18

ب) 6

ج) 54

د) 7

**پاسخ**

**گزینۀ د** - برای مقایسه عملکرد سیستم در دو حالت، CPI واقعی را با base CPI مقایسه میکنیم.

Effective CPI = 3 + 0.15 × 120 = 3 + 18 = 21

بنابراین:

Effective CPI / Base CPI = 21 / 3 = 7

در نتیجه میتوان گفت پردازنده 7 برابر کندتر شده است.

**~~سوال سوم (تمرین 2) - سریع - دقتی - تقریبا" حفظی Hefzi- مفاهیم cache~~**

چه شرطی کافیست تا بلوکی از حافظۀ نهان dirty تلقی شود؟ فرض کنید در یک حافظۀ نهان مجموعه انجمنی 4 راهه[[1]](#footnote-0) 512 set داریم. هر بلوک از این حافظه نیز حاوی هشت word چهار بایتی می باشد. در نهایت نیاز به چند بیت dirty در این حافظۀ نهان داریم؟

الف) تمام داده های آن تغییر کرده باشد - 211 بیت

ب) تنها یکی از داده های آن تغییر کرده باشد - 214 بیت

ج) تمام داده های آن تغییر کرده باشد - 214 بیت

د) تنها یکی از داده های آن تغییر کرده باشد - 211 بیت

**پاسخ**

**گزینۀ د** - برای اینکه یک بلوک از حافظۀ نهان dirty تلقی شود، تنها کافیست یکی از دادههای آن تغییر کرده باشد. زمانی که پردازنده دادهای را از حافظۀ اصلی درخواست میکند این داده داخل حافظۀ نهان ذخیره میشود. حال اگر پردازنده پس از خواندن داده آن را تغییر دهد و دادۀ تغییر یافته را بر روی حافظۀ نهان بنویسد کل بلوک مربوط به آن داده dirty محسوب می شود زیرا دادهای دارد که مقدار آن با وضعیت فعلی حافظۀ اصلی مطابقت ندارد.

از آنجایی که هر بیت dirty به یک بلوک از حافظه متناظر میشود کافیست تعداد بلوکهای حافظۀ نهان را بدست آوریم. تعداد بلوکها در حافظۀ نهان توصیف شده برابر است با:

4 way × 512 blocks/way = 2048 blocks

بنابراین این حافظۀ نهان در مجموع نیاز به 2048 بیت dirty دارد.

**سوال چهارم (تمرین 2 - lecture 9) - متوسط - دقتی - حفظی از همه مباحث Hefzi**

صحیح یا غلط بودن عبارات زیر را مشخص کنید.

الف) کلاکی که CPU از آن استفاده می کند معمولا با کلاک گذرگاه حافظۀ اصلی یکسان نیست.

ب) حافظۀ اصلی خود مدیریت می کند که داده درون Instruction Cache ذخیره شود یا Data Cache.

~~ج) به فرایندی که در آن CPU درون Instruction Cache دستوری را می نویسد store گفته می شود.~~

د) در حافظه های نهان سطوح بالاتر (L2 Cache به بعد) کاهش hit time اولویت اول را دارد.

ه) به دلیل این که در حافظۀ نهان سطح 3 (L3 Cache) فضای زیادی در اختیار داریم، ترجیح می دهیم در آن قسمت مربوط به ذخیره سازی داده ها و دستورات را جدا کنیم.

~~و) با افزایش associativity در یک حافظۀ نهان، قسمت Index آدرسی که توسط CPU درخواست می شود کوچک تر شده و قسمت Tag آن بزرگ تر می شود.~~

**پاسخ**

الف) **صحیح** – کلاکی که پردازنده از آن استفاده می کند نسبت به کلاک گذرگاه حافظه سریعتر است.

ب) **غلط** – حافظه از این که چیزی که تحویل میدهد داده است یا دستور خبر ندارد، بلکه CPU است که میداند نیاز به داده داشته یا دستور و به D-Cache رجوع کند یا I-Cache. پس از آن هر یک از این حافظه های نهان بسته به نیاز به حافظۀ اصلی رجوع می کنند، اما همچنان حافظه خبر ندارد که کدام یک به او درخواست داده اند.

ج) **غلط** – پردازنده هیچ گاه دستوری را روی حافظه نمی نویسد و دستورات برای CPU حالت Read-Only دارند. به فرآیند نوشتن داده بر روی Data Cache اما store گفته می شود.

د) **غلط** – در این سطوح تمرکز ما بر افزایش hit rate است و در سطح L1 الویت با کاهش hit time می باشد.

ه) **غلط** – به طور معمولا در حافظه های نهان سطوح پایین تر قسمت داده و دستورات جدا می شوند، اما در سطوح بالاتر هر دو در یک مکان ذخیره می شوند تا بتوانیم به طور متوازن از فضای زیادی که در اختیار داریم استفاده کنیم.

و) **صحیح** – در حقیقت Index نشان دهندۀ شماره set ای از حافظۀ نهان است که دادۀ مورد نظر در آن قرار دارد. از طرفی با فرض یکسان بودن سایز حافظۀ نهان هر چه associativity آن افزایش یابد (که معادل با افزایش تعداد way هاست)، تعداد set های حافظۀ نهان کاهش می یابد. بنابراین تعداد بیت های مورد نیاز برای آدرس دهی آن کاهش می یابند.

**سوال پنجم (lecture 9) - ساده - سریع - حفظی از multilevel cache**

در حافظه­ های نهان چند لایه با دور شدن از پردازنده به طور معمول هر یک از پارامترهای زیر به ترتیب چه تغییری می ­کنند؟

حجم حافظه – سرعت حافظه – associativity حافظه

الف) افزایش – افزایش – کاهش

ب) افزایش – کاهش – افزایش

ج) کاهش – افزایش – کاهش

د) کاهش – کاهش – افزایش

**پاسخ**

**گزینۀ ب** - با دور شدن حافظه ­های نهان از پردازنده (رفتن به cache های با level بالاتر) اندازۀ این حافظه­ ها افزایش یافته و در عوض سرعت آن­ها کاهش می ­یابد. همچنین به طور معمول associativity حافظه­ های سطوح بالاتر بیش­تر است.

**سوال ششم (lecture 9) - متوسط - حفظی از cache considerations** Hefzi

افزایش حجم حافظه، اندازۀ بلوک ها و associativity در یک حافظۀ نهان به ترتیب ممکن است چه پیامدهایی داشته باشد؟

الف) کاهش compulsory miss - افزایش miss penalty - افزایش access time

ب) افزایش miss penalty - کاهش hit rate - افزایش access time

ج) افزایش access time - کاهش compulsory miss - کاهش conflict miss

د) کاهش capacity miss - کاهش conflict miss - کاهش hit rate

**پاسخ**

**گزینۀ ج** - طبق جدول زیر تمامی موارد ذکر شده در گزینۀ ج درست هستند.

| **Negative performance effect** | **Effect on miss rate** | **Design change** |
| --- | --- | --- |
| May increase access time | Decrease capacity misses | Increase cache size |
| May increase access time | Decrease conflict misses | Increase associativity |
| Increases miss penalty. For very large block size, may increase miss rate due to pollution. | Decrease compulsory misses | Increase block size |

# 

# 

# ARTA

### **سوال اول (کلاس کنکور) - ارزش 0.7 - سختی 0.5 (فقط کافیه بدونه، زمان زیاد نمیبره) - مشابه سوال 2** Zaman Ejraیک برنامه چهار نوع دستور D,C،B،A را دارا است. این دستور ها به ترتیب 20، 20، 50 و 10 درصد از کل زمان اجرای برنامه را شامل می شوند. کدام گزینه می تواند زمان اجرای برنامه را بیشتر از بقیه کاهش دهد ؟

**الف)** تمام دستور ها دو برابر سریع تر شوند.

**ب)** دستور های A و B چهار برابر سریع تر شوند.

**ج)** دستور های C و D چهار برابر سریع تر شوند.

**د)** دستور C پنج برابر سریع تر شود.

**پاسخ - گزینه الف**

| د | ج | ب | الف | Initial |  |
| --- | --- | --- | --- | --- | --- |
| 20 | 20 | 5 | 10 | 20 | A |
| 20 | 20 | 5 | 10 | 20 | B |
| 10 | 12.5 | 50 | 25 | 50 | C |
| 10 | 2.5 | 10 | 5 | 10 | D |
| 60 | 55 | 70 | 50 | 100 | Total |

در نتیجه گزینه **الف** بهترین راه حل است.

### 

### **~~سوال دوم (کلاس کنکور)~~ - ارزش 0.9 - سختی 0.7 (اینم زمانبر نیست کافیه بدونه باید چیکار کنه) - مشابه 1** Zaman Ejraدر یک کامپیوتر در مجموعه دستوراتش، 30 درصد آنها 2 برابر سریع تر شدند، 20 درصد 3 برابر کاهش سرعت داشتند و 15 درصد آنها حذف شدند. سرعت این برنامه با چه نسبتی تغییر می کند.

**الف)** سرعت 1.2 برابر می شود **ب)** سرعت 0.83 برابر می شود

**ج)** سرعت 1.1 برابر می شود **د)** سرعت 0.91 برابر می شود.

**پاسخ - گزینه د**

زمان اجرای اولیه را T در نظر گرفته و با استفاده از آن T جدید را حساب می کنیم. نسبت تغییر سرعت برابر می شود با تقسیم T قدیمی بر T جدید.

T(new) =

در نتیجه نسبت سرعت جدید به قدیم برابر می شود با 1 تقسیم بر 1.1 برابر با 0.91 .

### **سوال سوم (کلاس کنکور) - ارزش 0.9 - سختی 0.9 - سوال نابیه ولی وقت گیره، هم باید بلوکشو حساب کنه بعد ببینه توی اون setش هست نیست و miss یا hit رو تشخیص بده.**

### K-way associativeیک cache با ساختار 2way set associative و با گنجایش 4 بلوک 4 کلمه ای موجود است. اگر اعداد زیر آدرس کلمه هایی باشند که به ترتیب از چپ به راست توسط پردازنده از حافظه درخواست می شوند، کدام آدرس های منجر به Miss می شوند ؟ (حافظه cache در ابتدا خالی بوده و از سیاست FIFO استفاده می شود.)

1, 3, 4, 7, 8, 9, 18, 3, 6, 16

**الف)** 6، 18، 8، 3، 1

**ب)** 3، 18، 8، 4، 1

**ج)** 16، 6، 9، 8، 1

**د)** 3، 9، 7، 4، 1

**پاسخ - گزینه ب**

با محاسبه خارج قسمت تقسیم هر عدد بر 4 شماره بلوک آن را پیدا می کنیم. همین طور با توجه به اینکه 4 بلوک داریم و ساختار cach به صورت 2way set associative است پس 2 تا set داریم. (ردیف 4 و 5 نمایشگر بلوک های داخل آن set هستند)

| Add | 1 | 3 | 4 | 7 | 8 | 9 | 18 | 3 | 6 | 16 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block | 0 | 0 | 1 | 1 | 2 | 2 | 4 | 0 | 1 | 4 |
| Set | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
|  | 0 | 0 | 1 | 1 | 2 | 2 | 4 | 0 | 1 | 0 |
|  | - | - | - | - | 0 | 0 | 2 | 4 | - | 4 |
| h/m | m | h | m | h | m | h | m | m | h | h |

Misses : 1, 4, 8, 18, 3

### **سوال سوم v2**

### K-way associativeیک cache با ساختار 2way set associative و با گنجایش 4 بلوک 4 کلمه ای موجود است. اگر اعداد زیر آدرس کلمه هایی باشند که به ترتیب از چپ به راست توسط پردازنده از حافظه درخواست می شوند، کدام آدرس های منجر به Miss می شوند ؟ (حافظه cache در ابتدا خالی بوده و از سیاست FIFO استفاده می شود.)

1, 16, 4, 12, 3, 9, 8, 18, 6, 14

**الف)** 6، 18، 8، 3، 1

**ب)** 3، 18، 8، 4، 1

**ج)** 9، 12، 4، 16، 1

**د)** 3، 9، 7، 4، 1

**پاسخ - گزینه ج**

با محاسبه خارج قسمت تقسیم هر عدد بر 4 شماره بلوک آن را پیدا می کنیم. همین طور با توجه به اینکه 4 بلوک داریم و ساختار cach به صورت 2way set associative است پس 2 تا set داریم. (ردیف 4 و 5 نمایشگر بلوک های داخل آن set هستند)

| Add | 1 | 16 | 4 | 12 | 3 | 9 | 8 | 18 | 6 | 14 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block | 0 | 4 | 1 | 3 | 0 | 2 | 2 | 4 | 1 | 4 |
| Set | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
|  | 0 | 4 | 1 | 3 | 4 | 2 | 2 | 2 | 3 | 3 |
|  | - | 0 | - | 1 | 0 | 4 | 4 | 4 | 1 | 1 |
| h/m | m | m | m | m | h | m | h | h | h | h |

Misses : 1, 16, 4, 12, 9

### **سوال سوم v3**

### K-way associativeیک cache با ساختار 2way set associative و با گنجایش 4 بلوک 4 کلمه ای موجود است. اگر اعداد زیر آدرس کلمه هایی باشند که به ترتیب از چپ به راست توسط پردازنده از حافظه درخواست می شوند، کدام آدرس های منجر به Miss می شوند ؟ (حافظه cache در ابتدا خالی بوده و از سیاست FIFO استفاده می شود.)

1, 16, 3, 12, 4, 8, 9, 18, 6, 14

**الف)** 6، 18، 12، 16، 1

**ب)** 14، 18، 8، 4، 1

**ج)** 18، 9، 8، 4، 1

**د)** 8، 4، 12، 16، 1

**پاسخ - گزینه د**

با محاسبه خارج قسمت تقسیم هر عدد بر 4 شماره بلوک آن را پیدا می کنیم. همین طور با توجه به اینکه 4 بلوک داریم و ساختار cach به صورت 2way set associative است پس 2 تا set داریم. (ردیف 4 و 5 نمایشگر بلوک های داخل آن set هستند)

| Add | 1 | 16 | 3 | 12 | 4 | 8 | 9 | 18 | 6 | 14 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block | 0 | 4 | 0 | 3 | 1 | 2 | 2 | 4 | 1 | 3 |
| Set | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
|  | 0 | 4 | 4 | 3 | 1 | 2 | 2 | 2 | 1 | 1 |
|  | - | 0 | 0 | - | 3 | 4 | 4 | 4 | 3 | 3 |
| h/m | m | m | h | m | m | m | h | h | h | h |

Misses : 1, 16, 12, 4, 8

### 

### **سوال سوم v4**

### K-way associativeیک cache با ساختار 2way set associative و با گنجایش 4 بلوک 4 کلمه ای موجود است. اگر اعداد زیر آدرس کلمه هایی باشند که به ترتیب از چپ به راست توسط پردازنده از حافظه درخواست می شوند، کدام آدرس های منجر به Miss می شوند ؟ (حافظه cache در ابتدا خالی بوده و از سیاست FIFO استفاده می شود.)

1, 3, 9, 12, 10, 8, 9, 2, 6, 14

**الف)** 6، 10، 12، 9، 1

**ب)** 2، 9، 12، 3، 1

**ج)** 2، 9، 10، 3، 1

**د)** 14، 8، 12، 9، 1

**پاسخ - گزینه الف**

با محاسبه خارج قسمت تقسیم هر عدد بر 4 شماره بلوک آن را پیدا می کنیم. همین طور با توجه به اینکه 4 بلوک داریم و ساختار cach به صورت 2way set associative است پس 2 تا set داریم. (ردیف 4 و 5 نمایشگر بلوک های داخل آن set هستند)

| Add | 1 | 3 | 9 | 12 | 10 | 8 | 9 | 2 | 6 | 14 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block | 0 | 0 | 2 | 3 | 2 | 2 | 2 | 0 | 1 | 3 |
| Set | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
|  | 0 | 0 | 2 | 3 | 2 | 2 | 2 | 2 | 1 | 1 |
|  | - | - | 0 | - | 0 | 0 | 0 | 0 | 3 | 3 |
| h/m | m | h | m | m | m | h | h | h | m | h |

Misses : 1, 9, 12, 10, 6

### 

### 

### **سوال چهارم (کلاس کنکور) - ارزش 1.0 - سختی 0.5 - فقط نکته داره همین، کافیه سر کلاس دقت کرده باشه و نکته رو بدونه درجا جوابو میزنه.** Cpi

### در یک برنامه، دستور ها مطابق جدول زیر می باشند :

| دستور | تعداد کلاک | تعداد دستور |
| --- | --- | --- |
| Add/Sub | 2 | 5000 |
| Mul | 6 | 1500 |
| Load | 5 | 3500 |
| Store | 5 | 2000 |

در طول اجرای این برنامه چند بار به حافظه مراجعه می شود ؟

**الف)** 5500 **ب)** 3500 **ج)** 12000 **د)** 17500

**پاسخ - گزینه د**

برای اجرای هر کدام از دستورات نیاز داریم تا دستور مربوطه را ابتدا از حافظه دریافت کنیم (fetching) در نتیجه 12000 مراجعه به حافظه تنها به این منظور داریم. علاوه بر آن دستورات Load و Store نیز دستوراتی هستند که با حافظه در ارتباط خواهند بود در نتیجه به ازای هر کدام از آنها نیز یک مراجعه به حافظه دیگر هم داریم که می شود 5500. پس مجموعا 17500 مراجعه به حافظه داریم.

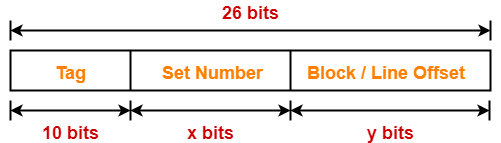
### **~~سوال پنجم~~ (طرح شده - از سری تمرینای دوم) - ارزش 0.6 چون مثل همین توی تمرینا داشتیم - سختی 0.65 - وقت گیریش 5 از 10** Address fields lengthفرض کنید در یک سیستم حجم حافظه اصلی 64MByte می باشد و در آدرس دهی 10 بیت برای بخش tag درنظر گرفته شده. اگر سیاست جای دهی حافظه نهان آن 4way set associative باشد، اندازه این حافظه نهان چند بایت است ؟

**الف)** 128Bytes **ب)** 256Bytes **ج)** 512Bytes **د)** 64Bytes

**پاسخ - گزینه ب**

برای محاسبه حجم حافظه نهان ابتدا باید ببینیم تعداد کل بیت های آدرس دهی چند تاست:

بعد از تعداد بیت های tag کم می کنیم. که میشه جمع بیت های block و set :



با توجه به اینکه نمیدونیم داخل هر بلوک چند word وجود داره نمیتونیم جدا جدا x و y رو حساب کنیم اما دونستن جمعشون برای ما کفایت می کنه. حجم حافظه نهان برابره با (تعداد set ها داخل حافظه نهان \* تعداد بلوک های هر set \* تعداد word هر بلوک). تعداد بلوک های هر set که با توجه به 4way set associative بودن حافظه میدونیم 4 هست، جمع x و y هم میدونیم برابره با 16. در نتیجه :

cache memory size = =

# 

# Shayan

1 (طرح شده ) سوال حفظی ، زمان بری = 0.3 ، ارزش = 0.6 ، سختی = 0.5- کدام گزینه به ترتیب از چپ به راست پاسخ درست به سوالات زیر میباشد؟ Memory

الف) حافظه ی دائمی ای که داده ها و دستورالعملهای راه اندازی کامپیوتر را در خود جای داده , داده ها پس از خاموش

شدن پاک نمیشوند،چیست؟

ب) کدام حافظه باید بارها در ثانیه تازه سازی شود؟

ج) کدام حافظه پس از قطع شدن منبع انرژی اطلاعاتش پاک میشود؟

1) ROM - SRAM - ROM 2) RAM - ROM - DRAM 3) SRAM - DRAM \_ROM 4) ROM-DRAM-RAM

پاسخ : گزینه 4

2 (طرح شده ) سوال حفظی ، زمان بری = 0.4 ، ارزش = 0.6 ، سختی = 0.6

- کدام گزینه از راست به چپ به ترتیب مزیت EEPROM به EPROM و تفاوت ROM و PROM را شرح میدهد؟ Memory

1. در EEPROM عملیات پاک کردن با استفاده از نور فرابنفش انجام می شود و کار پاک کردن راحت تر میشود - PROM به شکل از پیش برنامه ریزی شده تولید میشود در صورتی که که ROM به شکل یک حافظه خالی تولید میشود
2. در EEPROM عملیات پاک کردن با استفاده از نور فرابنفش انجام می شود و کار پاک کردن راحت تر میشود - ROM به شکل از پیش برنامه ریزی شده تولید میشود در صورتی که که PROM به شکل یک حافظه خالی تولید میشود
3. در EEPROM عملیات پاک کردن به شکل الکتریکی انجام می شود و کار پاک کردن راحت تر میشود - ROM به شکل از پیش برنامه ریزی شده تولید میشود در صورتی که که PROM به شکل یک حافظه خالی تولید میشود
4. در EEPROM عملیات پاک کردن به شکل الکتریکی انجام می شود و کار پاک کردن راحت تر میشود - PROM به شکل از پیش برنامه ریزی شده تولید میشود در صورتی که که ROM به شکل یک حافظه خالی تولید میشود

پاسخ : گزینه 4

الف) از آنجایی که عملیات erase کردن در EEPROM به شکل الکتریکی صورت میگیرد در نتیجه نیازی نیست این حافظه را جهت بردن به حالت خام از مدار خارج کنیم تا زیر نور فرابنفش قرار دهیم که پاک شود و با عبور جریان الکتریکی می توان آن را erase کرد.

ب) PROM به عنوان یک حافظه خالی تولید میشود ، به گونه ای که تمام بیت های آن به وسیله اتصال دیود ، یک میشود و میتوان آن را بعدا برای یک بار برنامه ریزی کرد اما ROM در طول فرایند تولیدش از پیش برنامه ریزی میشود و دیگر قابلیت برنامه ریزی ندارد

3- (کنکور ) سوال محاسباتی ، زمان بری = 0.7 ، ارزش = 0.8 ، سختی = 0.8 نیاز به دقت دارد Amdahl law Zaman Ejraدر یک کامپیوتر مجموعه دستورالعمل‌ها به گونه‌ای تغییر کرده است که 20 درصد زمان یک برنامه 4 برابر تسریع شده، 30 درصد آن 2 برابر کاهش سرعت یافته و 10 درصد از دستورات آن حذف شده است . سرعت این برنامه به نسبت حالت قبل چه تغییر کرده است؟

1. بدون تغییر است 2) افزایش یافته است. 3) سرعت کاهش یافته است 4) مفروضات کافی نیست

پاسخ گزینه 3

T new = 0.2 T old/4 + 0.3 T old \*2 + 0.4 T old = 1.05 T old

=> speed up = T old / T new = 1 / 1.05 => سرعت کاهش یافته است

1. 4-Way Set associative [↑](#footnote-ref-0)