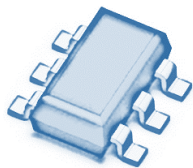


۱۳۹۷/۱۲/۱۴

## تکلیف ۲ – حافظه نهان

معماری کامپیوتر



۱) روش های نگاشت داده در حافظه نهان (cache) را نام ببرید و چگونگی به کار بردن این روش ها را توضیح دهید.

به سه روش Direct mapping, associative mapping, set associative mapping می توان داده ها را در حافظه نهان ذخیره کرد.

#### associative mapping:

این روش انعطاف پذیرترین روش در ذخیره سازی داده ها در حافظه نهان است. هم داده و هم آدرس داده ذخیره می شود، به همین دلیل می توان هر داده با هر آدرسی را در cache ذخیره کرد. برای پیدا کردن داده مورد نظر، آدرس آن را با آدرس های ذخیره شده مقایسه می شود. در این روش از نظر سخت افزاری مدارهای مقایسه گر لازم است به همین دلیل باعث افزایش هزینه ساخت می شود.

#### Direct mapping:

در این روش قسمتی از آدرس داده بر آدرس بلوکی که در آن ذخیره شده منطبق است (index) و قسمت باقی مانده با عنوان tag ذخیره می شود.

اگر  $2^n$  اندازه حافظه اصلی و  $2^k$  اندازه حافظه نهان باشد، آنگاه  $k$  کم ارزش آدرس با آدرس بر آدرس های حافظه نهان منطبق می شود و  $n-k$  بیت پر ارزش به عنوان tag ذخیره می شوند. هنگامی که یک داده را می خواهیم قسمت tag آدرس آن را با tag ذخیره شده در index متناظر با این آدرس مقایسه می کنیم.

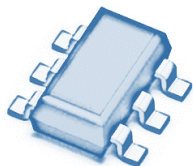
از معایب این روش این است که دو آدرس با index یکسان نمی توانند به طور همزمان در cache باشند و این موجب بالا رفتن نرخ miss می شود اگر آدرس هایی با index یکسان به طور مداوم درخواست شوند.

#### set associative mapping:

این روش ترکیبی از دو روش پیشین است .

در این روش بر خلاف direct mapping، می توان آدرس هایی با index یکسان را به طور همزمان در cache نگه داری کرد.

اگر اندازه مجموعه در این روش  $k$  باشد، حافظه به  $(k/\text{ظرفیت})$  قسمت تقسیم می شود که در هر قسمت  $k$  داده با index یکسان همراه با tag هایشان ذخیره می شوند.



## ۲) الف) علت استفاده از حافظه های سلسه مراتبی را توضیح دهید.

به دلیل اینکه سرعت پاسخگویی پردازنده مرکزی بالاست و اگر این پردازنده بخواهد از یک نوارخوان یا هر حافظه دیگری با قیمت پایین بخواند زمان اتلافی زیادی خواهیم داشت و همینطور اگر بخواهیم از حافظه هایی سریع مثل فلیپ فلاپ ها استفاده کنیم هزینه زیادی را باید پرداخت کنیم. پس از حافظه های سلسله مراتبی که ترکیبی از این دو روش اند استفاده میکنیم و حافظه های سریع تر و گران قیمت تر را به پردازنده نزدیک میکنیم.

## ب) چه معیارهایی را برای برتری دادن به یک حافظه در نظر میگیرید؟

۱) سرعت حافظه

۲) ظرفیت حافظه

۳) قیمت

۳) آیا در یک حافظه ی نهان مجموعه انجمنی با حجم ثابت، با افزایش  $k$  لزوماً درصد موفقیت افزایش می یابد؟ (اگر پاسخ مثبت است، دلیل بیاورید و اگر خیر، مثال نقض بزنید)

بله با افزایش  $k$  همواره مقدار hit ratio زیاد میشود. این موضوع را میتوان اینگونه توصیف کرد که با افزایش  $k$  رعایت همجواری های زمانی بیشتر میشود که این موضوع در کنار رعایت بیشتر همجواری مکانی در blocking باعث زیادتر شدن hit ratio میشود.

۴) فرض کنید که یک حافظه نهان با ۶۴ بلوک و اندازه بلوک ۱۶ بایت موجود است. چه شماره بلوکی به آدرس ۱۲۰۰ نگاشت می شود؟

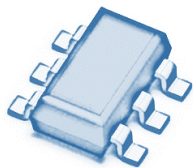
شماره بلوک ها از رابطه زیر محاسبه میشود:

شماره بلوک = (تعداد بلوک های حافظه نهان) mod (آدرس بلوک)

که در آن آدرس بلوک عبارت است از:

آدرس بایت

تعداد بایت در هر بلوک



و توجه کنید که این آدرس بلوک شامل تمام آدرس های بین

$$\frac{\text{آدرس بایت}}{\text{تعداد بایت در هر بلوک}} \times \text{تعداد بایت در هر بلوک}$$

9

$$1 - \frac{\text{آدرس بایت}}{\text{تعداد بایت در هر بلوک}} \times \text{تعداد بایت در هر بلوک}$$

لذا با ۱۶ بایت در هر بلوک ، آدرس بایت ۱۲۰۰ در آدرس بلوک، برابر خواهد بود با:

$$\left\lceil \frac{1200}{16} \right\rceil = 75$$

که به شماره بلوک  $75 \bmod 64 = 11$  در حافظه نهان نگاشت میشود.

در حقیقت چنین بلوکی تمامی آدرس های بین ۱۲۰۰ و ۱۲۱۵ را نگاشت میکند.

(۵) یک حافظه نهان با ۶۴ بایت ظرفیت داریم که برای استفاده از آن روش direct Mapping به کار گرفته شده است. در هر بلوک حافظه دو کلمه داده ۳۲ بیتی ذخیره می شود. اگر بخواهیم یک کلمه جدید را وارد cache کنیم تمام کلمات آن بلوک همراه آن به cache آورده می شود.

a. برای درخواست های زیر miss یا hit شدن را تعیین کنید.

b. هر آدرس را به باینری تبدیل کنید و قسمت های offset, index, tag را برای هر یک تعیین کنید.

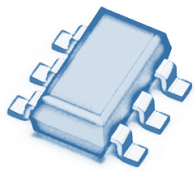
c. hit rate را محاسبه کنید.

۱، ۲۴، ۵، ۲۵، ۱۰، ۲۰، ۳، ۴، ۱۱، ۰، ۲، ۷، ۱۷، ۱۶، ۱۰

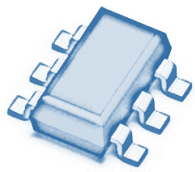
هر بلوک cache دو کلمه داده ۳۲ بیتی (۸ بایت) ظرفیت دارد، پس cache، ۶۴/۸ بلوک (۸ بلوک) دارد. در ابتدای کار حافظه خالی است و تقسیم بندی های آن به صورت زیر است. یک بیت offset، سه بیت index و چهار بیت باقی مانده tag

نحوه پاسخ گویی به درخواست ها :

۱۰ => ۰۰۰۰۱۰ (miss)

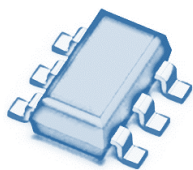


Offset => ۰  
Index => ۱۰۱  
Tag => ۰۰۰۰  
۱۶ => ۰۰۰۱۰۰۰۰ (miss)  
Offset => ۰  
Index => ۰۰۰  
Tag => ۰۰۰۱  
۱۷ => ۰۰۰۱۰۰۰۱ (hit)  
Offset => ۱  
Index => ۰۰۰  
Tag => ۰۰۰۱  
۷ => ۰۰۰۰۰۱۱۱ (miss)  
Offset => ۱  
Index => ۰۱۱  
Tag => ۰۰۰۰  
۲ => ۰۰۰۱۰۰۰۰ (miss)  
Offset => ۰  
Index => ۰۰۱  
Tag => ۰۰۰۰  
۰ => ۰۰۰۰۰۰۰۰ (miss)  
Offset => ۰  
Index => ۰۰۰  
Tag => ۰۰۰۰  
۱۱ => ۰۰۰۰۱۰۱۱ (hit)  
Offset => ۱  
Index => ۱۰۱  
Tag => ۰۰۰۰  
۴ => ۰۰۰۰۰۱۰۰ (miss)  
Offset => ۰  
Index => ۰۱۰



Tag => 0000  
۳ => 00000011 (hit)  
Offset => ۱  
Index => 001  
Tag => 0000  
۲۰ => 00010100 (miss)  
Offset => ۰  
Index => 101  
Tag => 0001  
1۰ => 00001010 (hit)  
Offset => ۰  
Index => 101  
Tag => 0000  
۲۵ => 00011001 (miss)  
Offset => ۱  
Index => 100  
Tag => 0001  
۵ => 00000101 (miss)  
Offset => ۱  
Index => 010  
Tag => 0000  
۲۴ => 00011000 (hit)  
Offset => ۱  
Index => 100  
Tag => 0001  
1 => 00000001 (hit)  
Offset => ۱  
Index => 000  
Tag => 0000

$$\text{Hit rate} = \frac{\# \text{ hit}}{\# \text{ all request}} = \frac{6}{15} = 0.40$$



۶) در مورد ناهنجاری بلیدی (Belady's Anomaly) تحقیق کنید و آن را توضیح دهید.

پدیده‌ای است که در آن با افزایش تعداد قاب‌ها، تعداد نقص صفحه‌ها هم افزایش می‌یابد. تا قبل از اثبات ناهنجاری بلیدی، اعتقاد بر این بود که با افزایش تعداد قاب‌ها، تعداد نقصه صفحه‌ها یا کاهش می‌یابد یا تغییر نمی‌کند. و...

۷) فرض کنید حافظه اصلی از پنج بلوک تشکیل شده است. پردازنده بلوک‌های حافظه را به ترتیب زیر درخواست میکند.

۳، ۲، ۱، ۰، ۳، ۲، ۴، ۳، ۲، ۱، ۰، ۴

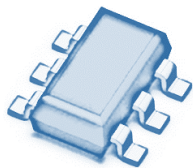
اگر در ابتدا حافظه نهان خالی باشد:

a. وضعیت حافظه نهان را در صورتی که اندازه آن سه بلوک و سیاست جایگزینی FIFO باشد را رسم کنید.

3	3	3	0	0	0	4	4	4	4	4	4
null	2	2	2	3	3	3	3	3	1	1	1
null	null	1	1	1	2	2	2	2	2	0	0
miss	miss	miss	miss	miss	miss	miss	hit	hit	miss	miss	hit

b. وضعیت حافظه نهان را در صورتی که اندازه آن چهار بلوک و سیاست جایگزینی FIFO باشد را رسم کنید.

3	3	3	3	3	3	4	4	4	4	0	0
null	2	2	2	2	2	2	3	3	3	3	4
null	null	1	1	1	1	1	1	2	2	2	2
null	null	null	0	0	0	0	0	0	1	1	1
miss	miss	miss	miss	hit	hit	miss	miss	miss	miss	miss	miss



c. برای حالت های بالا مقدار Miss Rate را محاسبه کنید.

$$\text{miss rate} = 9/12 = 0.75$$

$$\text{miss rate} = 10 / 12 = 0.833333$$

۸) یک حافظه نهان (cache) با ظرفیت ۶۴ kByte برای ذخیره داده داریم. اندازه هر بلوک این حافظه ۵۱۲ بیت است. هر کلمه (word)، ۳۲ بیت و اندازه آدرس حافظه اصلی هم ۳۲ بیت است.

a. تعداد بیت index, tag, offset را در ۳۲ بیت آدرس مشخص کنید.

در هر بلوک  $\frac{512}{32}$  (۱۶ کلمه) ذخیره می شود. بنابراین ۴ بیت برای offset لازم است. حافظه دارای  $\frac{64 * 2^{10}}{64}$  ( $2^{10}$  بلوک) است. بنابراین ۱۰ بیت برای index لازم است. ۳۲-(۱۰+۴) بیت برای tag لازم است.

b. اگر در هر بلوک تنها یک کلمه (word) ذخیره شود و اندازه حافظه نهان همین مقدار باشد، در این حالت تعداد بیت offset, tag, index را در ۳۲ بیت آدرس مشخص کنید.

با وجود یک کلمه در هر بلوک بیتی برای offset نیاز نیست. حافظه دارای 16384 بلوک است پس ۱۴ بیت برای index نیاز است. ۳۲-۱۴ بیت برای tag اختصاص می یابد.

c. در هر یک از دو حالت قبل اندازه کلی حافظه نهان را با احتساب مقادیر tag محاسبه کنید.

$$a : 534 \quad b : 800$$