

●●● معماری کامپیوتر (۱۳۸۵-۱۱-۱۳۳)

جلسه‌ی بیست و یکم



دانشگاه شهید بهشتی

دانشکده‌ی مهندسی برق و کامپیوتر

بهار ۱۳۹۱

احمد محمودی ازناوه

# فهرست مطالب

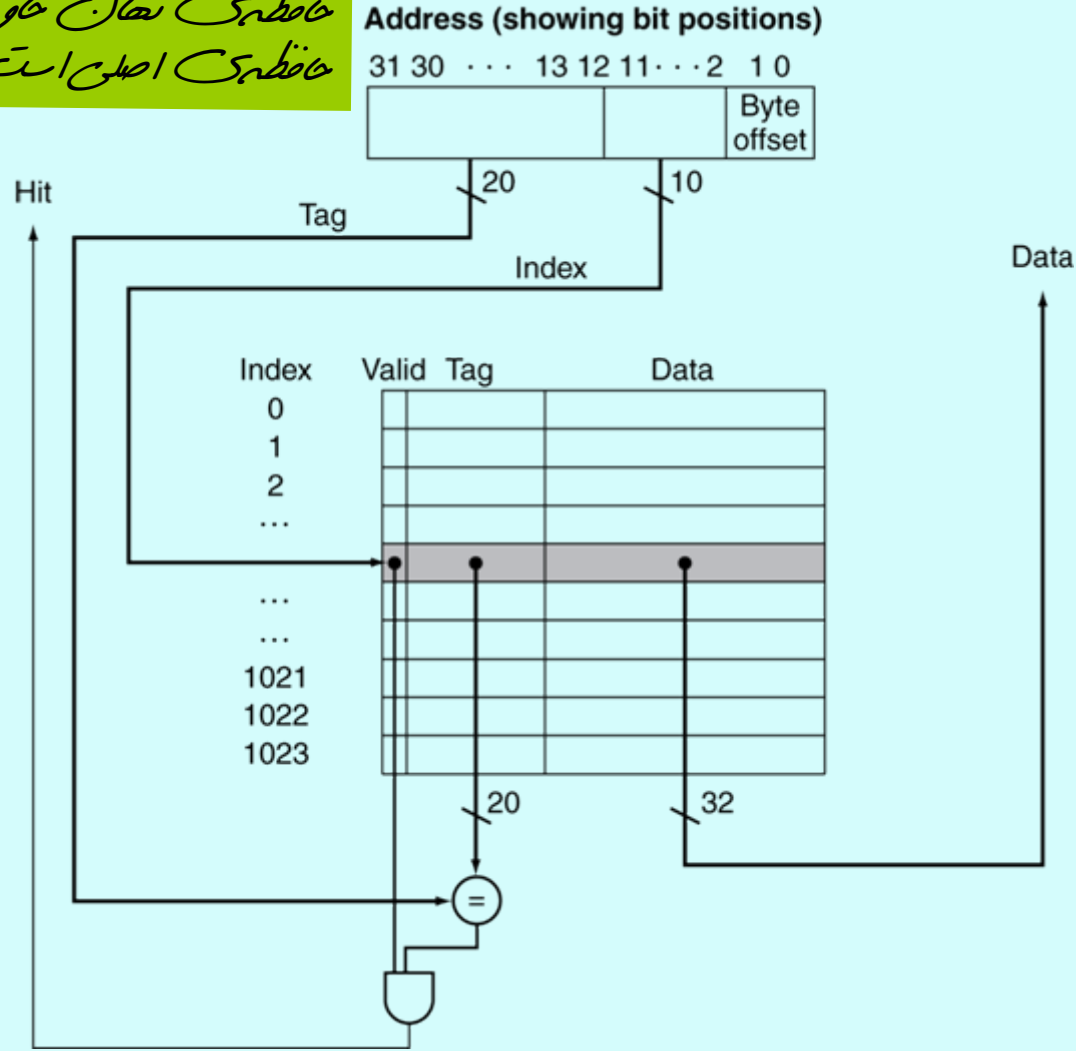
- حافظه‌ی نهان
  - نگاشت مستقیم
- نوشتن در حافظه‌ی نهان
- حافظه‌ی برگ‌برگ شده



# زیر بخش مربوط به آدرس

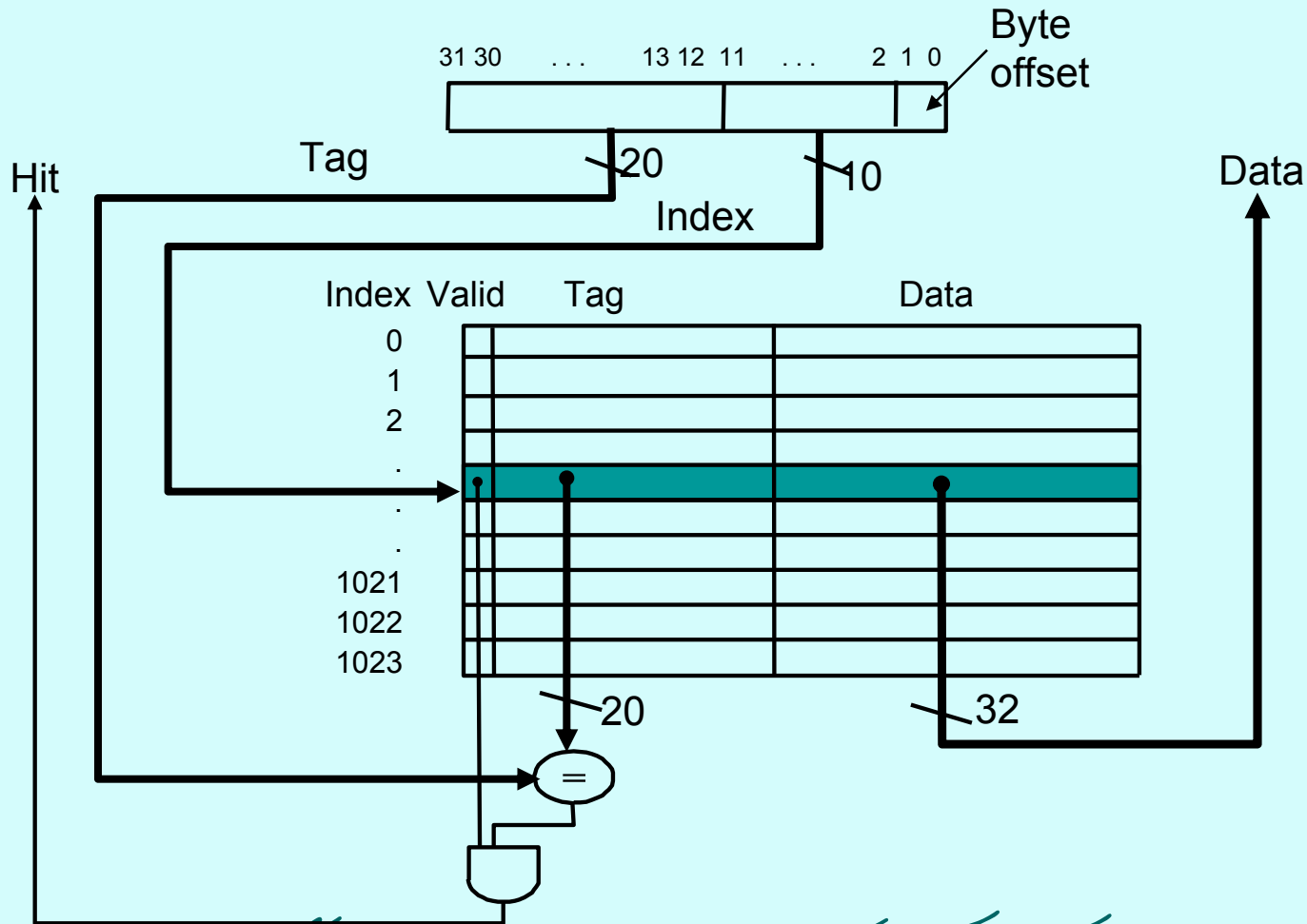
شاخصی (index) : بخش کم ارزش آدرس بلوک که بین حافظه‌های اصلی و حافظه‌های نهان مشترک است.

برچسب (tag) : مشخص می‌کند هر بلوک حافظه‌های نهان حاوی کدامین بلوک از حافظه‌های اصلی است



# حافظه‌ی نهان با نگاشت مستقیم

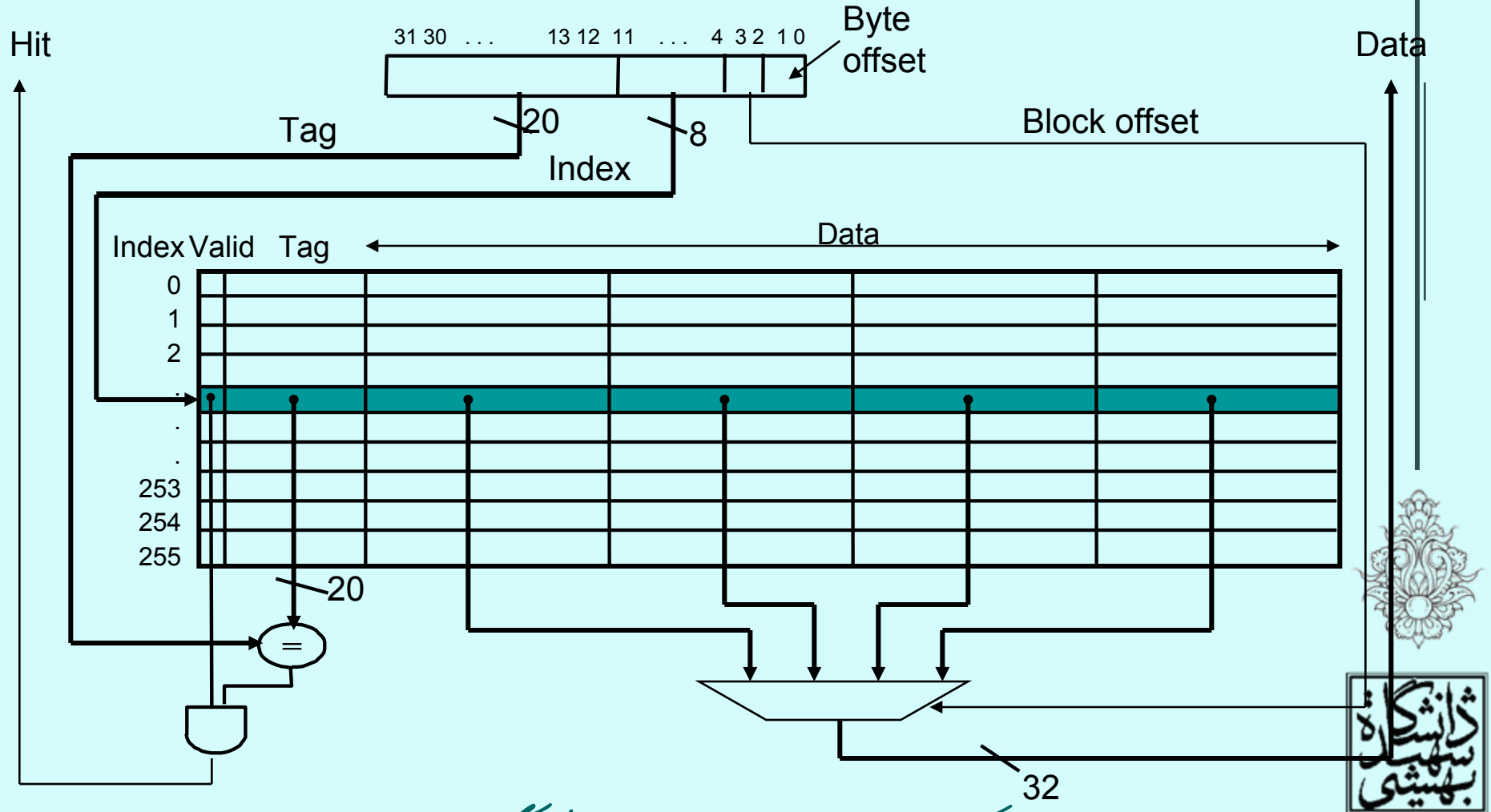
- گنجایش حافظه‌ی نهان 1KW با بلوک‌های یک کلمه‌ای



با استفاده از یک بلوک، کدام نوع همجواری در نظر گرفته شده است؟

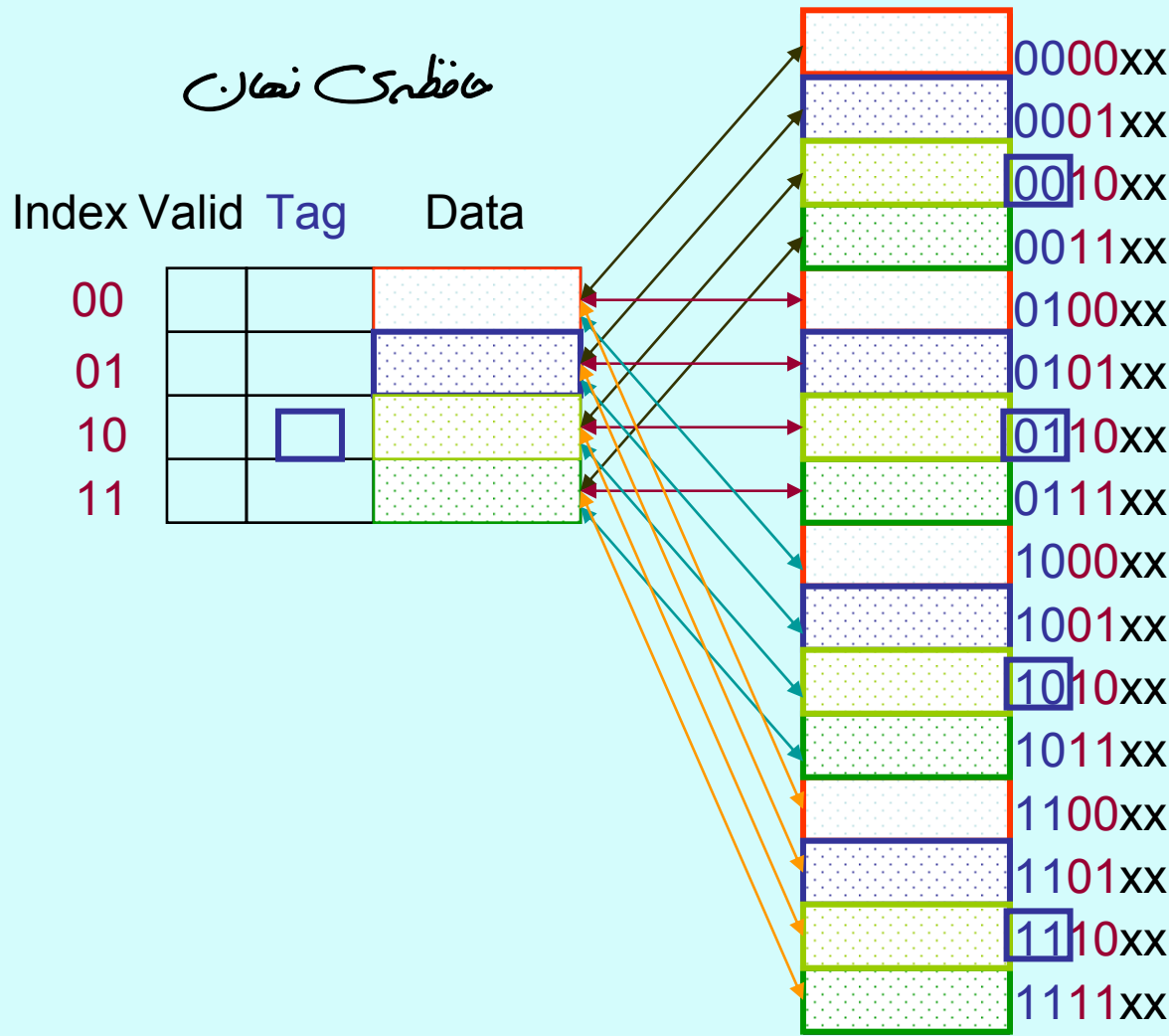


حافظه‌ی نهان با نگاشت مستقیم (ادامه...)  
 • گنجایش حافظه‌ی نهان 1KW با بلوک‌های چهار کلمه‌ای



در این حالت، کدام نوع همجواری در نظر گرفته شده است؟





# مثال

- یک سیستم با مشخصات زیر را در نظر بگیرید:
  - سی‌ودو بیت آدرس
  - حافظه‌ی نهان با نگاشت مستقیم
  - حافظه‌ی نهان با  $2^n$  بلوک؛ شاخص با  $n$  بیت
  - اندازه‌ی هر بلوک  $2^m$  کلمه

اندازه‌ی بخش برجسته چند بیت خواهد بود؟ **پیش**

پاسخ  $32-(n+m+2)$

تعداد کل بیت‌های حافظه‌ی مورد نیاز برای حافظه‌ی نهان را حساب کنید؟ **پیش**

پاسخ  $2^n \times (\text{block size} + \text{tag size} + \text{valid field size})$   
 $= 2^n \times (2^m \times 32 + (32 - n - m - 2) + 1) = 2^n \times (2^m \times 32 + 31 - n - m)$

البته باید توجه داشت، معمولا حجم داده قابل ذخیره‌سازی به عنوان گنجایش ذکر می‌شود، کامپیوتر



## مثالی دیگر

- برای یک حافظه‌ی نهان با شیوه‌ی نگاشت مستقیم با گنجایش 16KB و بلوک‌های چهار کلمه‌ای چند بیت حافظه نیاز داریم؟ (۳۲ بیت خط آدرس وجود دارد)

$$16KB \rightarrow 2^{12} \text{ word}$$

$$16KB \rightarrow 2^{10} \text{ Block}$$

$$1 \text{ Block} \rightarrow 4 \text{ word } (2^2)$$

$$32 - (10 + 2 + 2) = 18 \text{ bit for tag} +$$

$$1 \text{ bit for valid bit}$$

$$2^{10} \times [4 \times 32 + 18 + 1] = 147Kbits$$

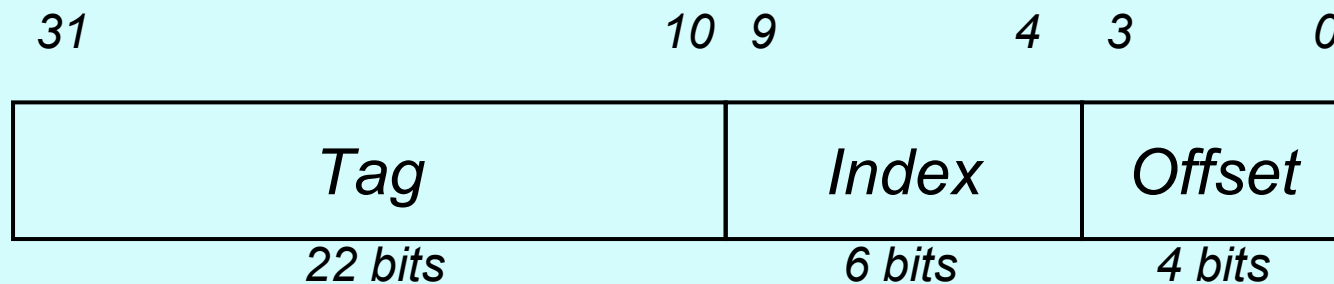
پاسخ





باز هم مثال

- یک حافظه‌ی نهان 64 بایتی، هر یک به اندازه‌ی ۱۶ بایت مفروض است. آدرس 1200 حافظه‌ی اصلی در کدامین بایت حافظه‌ی نهان قرار می‌گیرد؟



$$\text{Block address} = \lfloor 1200/16 \rfloor = 75$$

$$\text{Block number} = 75 \text{ modulo } 64 = 11$$



یا سبز

## بزرگی بلوک‌ها

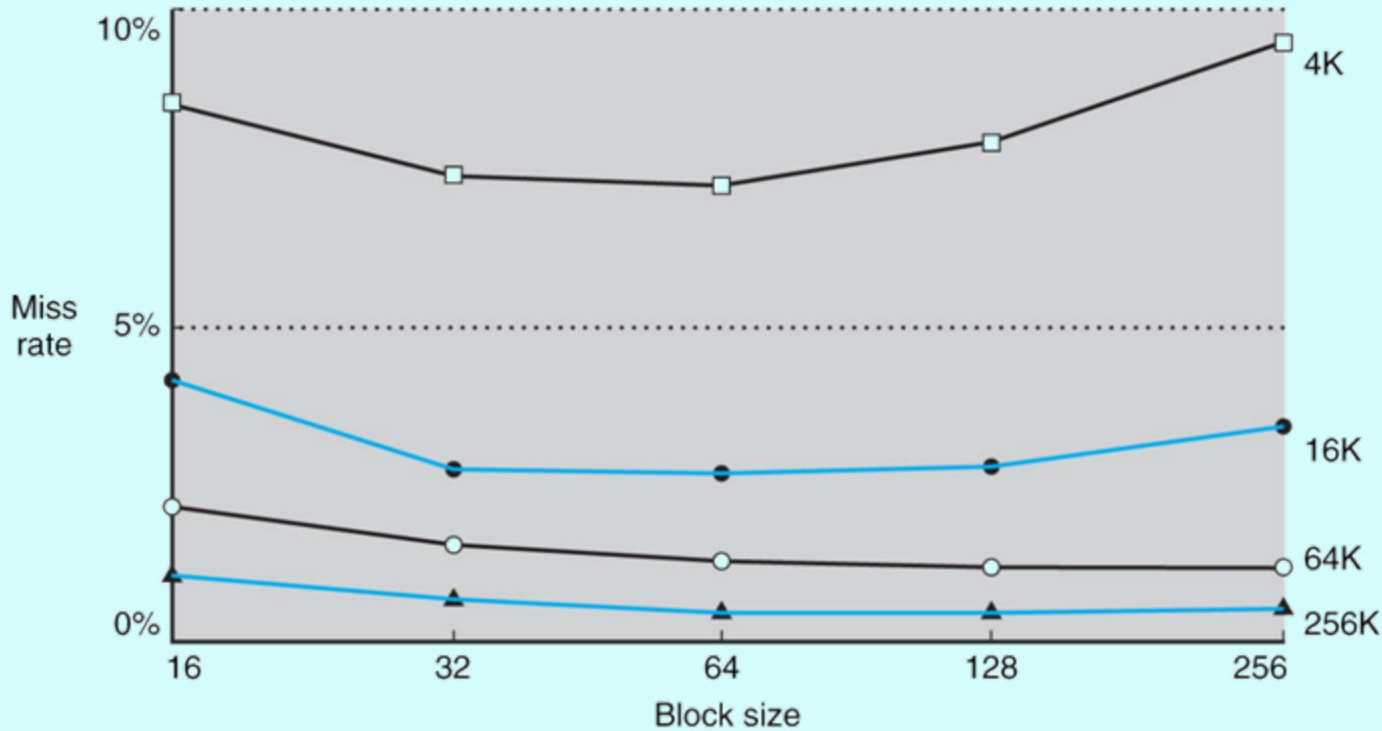
۱۵

- بلوک‌های بزرگ‌تر ← همجواری مکانی بهتر
  - در نتیجه miss rate کاهش می‌یابد.
- با حافظه‌ی نهان با حجم ثابت:
  - افزایش حجم بلوک ← کاهش تعداد بلوک‌ها
  - افزایش رقابت بین بلوک‌ها ← افزایش miss rate
- بلوک‌های بزرگ جریمه‌ی فقدان بالاتری دارند.
  - می‌باید در صورت عدم وجود بلوک در حافظه‌ی نهان، بلوک بزرگ‌تری به حافظه‌ی نهان منتقل شود.
  - با طراحی بهتر حافظه، می‌توان تا حدی بر این مشکل غلبه کرد.



# بزرگی بلوک‌ها (ادامه...)

بدین ترتیب مزیت کاهش miss rate تحت الشعاع قرار می‌گیرد.



نبود/وجود بلوک در حافظه‌ی نهان

• در صورتی که بلوک مربوط به آدرس مورد نظر در حافظه‌ی نهان وجود داشته باشد:

– پردازنده به روند عادی خود ادامه می‌دهد.

• **در غیر این صورت** *freezing the content of temporary register*

– خط لوله دچار تعلیق می‌شود یا وقفه‌ای رخ می‌دهد

– داده از سطوح پایین‌تر به حافظه‌ی نهان منتقل می‌شود.

*in order processor*

*out-of-order processor*

در این شیوه به جای اجرای برنامه بر اساس توابعی دستورالعمل‌ها، ترتیب اجرای دستورها بر اساس فراهم بودن داده‌ها صورت می‌پذیرد



نبود/وجود بلوک در حافظه‌ی نهان (ادامه...)

• در صورتی که دستورالعمل در حافظه‌ی نهان وجود نداشته باشد:

– آدرس مربوط به آن دستور (PC-4) به حافظه‌ی اصلی فرستاده می‌شود.

– حافظه‌ی اصلی داده را می‌خواند.

– داده در حافظه‌ی نهان نوشته می‌شود. بیت‌های برچسب و بیت اعتبار مقداره‌ی می‌شوند.

– اجرای دستورالعمل از سر گرفته می‌شود.



## نوشتن در حافظه‌ی نهان

- هنگامی نوشتن در حافظه‌ی نهان مطرح می‌شود، اوضاع کمی پیچیده‌تر خواهد شد.
- در چنین حالتی بین حافظه‌ی اصلی و حافظه‌ی نهان نوعی ناهماهنگی (**inconsistency**) به وجود می‌آید.
- ساده‌ترین راه، write through است، بدین معنا که هر آن چه در حافظه‌ی نهان نوشته می‌شود در حافظه‌ی اصلی نیز نوشته شود.



نوشتن در حافظه‌ی نهان (ادامه...)

• در صورتی که write miss رخ دهد، چه فرآیندی طی می‌شود؟

– داده از حافظه اصلی به حافظه‌ی نهانی منتقل می‌شود، مقدار مورد نظر نوشته شده و سپس حافظه‌ی اصلی به روز خواهد شد!

– چنین شیوه‌ای موجب کندی خواهد شد:

• مثال: در صورتی که  $CPI=1$  (بدون cache miss) و ده درصد دستورات store باشد و برای نوشتن در حافظه‌ی اصلی صد سیکل لازم باشد، effective CPI برنامه‌ی مورد نظر چقدر خواهد بود؟

$$Effective\ CPI = 1 + 0.1 \times 100 = 11$$



- داده‌های به جای این که مستقیماً در حافظه نوشته شوند، در یک بافر (میان‌گیر) نوشته خواهند شد. سپس پردازنده به فعالیت خود ادامه خواهد داد. در این هنگام محتوای بافر به حافظه اصلی منتقل خواهد شد.

– در صورتی که بافر پر شود، به ناچار پردازنده دچار تعلیق خواهد شد.

– در صورتی که نرخ تکمیل نوشتن داده در حافظه کندتر از درخواست‌های پردازنده برای نوشتن باشد

**بافر کردن فایده‌ای نخواهد داشت!**





# write back (copy back)

- داده‌ی مورد نظر تنها در حافظه‌ی نهان نوشته می‌شود، و در هنگام جابجایی به حافظه‌ی اصلی انتقال می‌یابد.
- در این صورت باید به نحوی بلوک‌های تخریب یافته (dirty block) را متمایز کنیم.
- در اینجا نیز می‌توان از بافر استفاده نمود.
- طبعاً طراحی سخت‌تر خواهد شد.



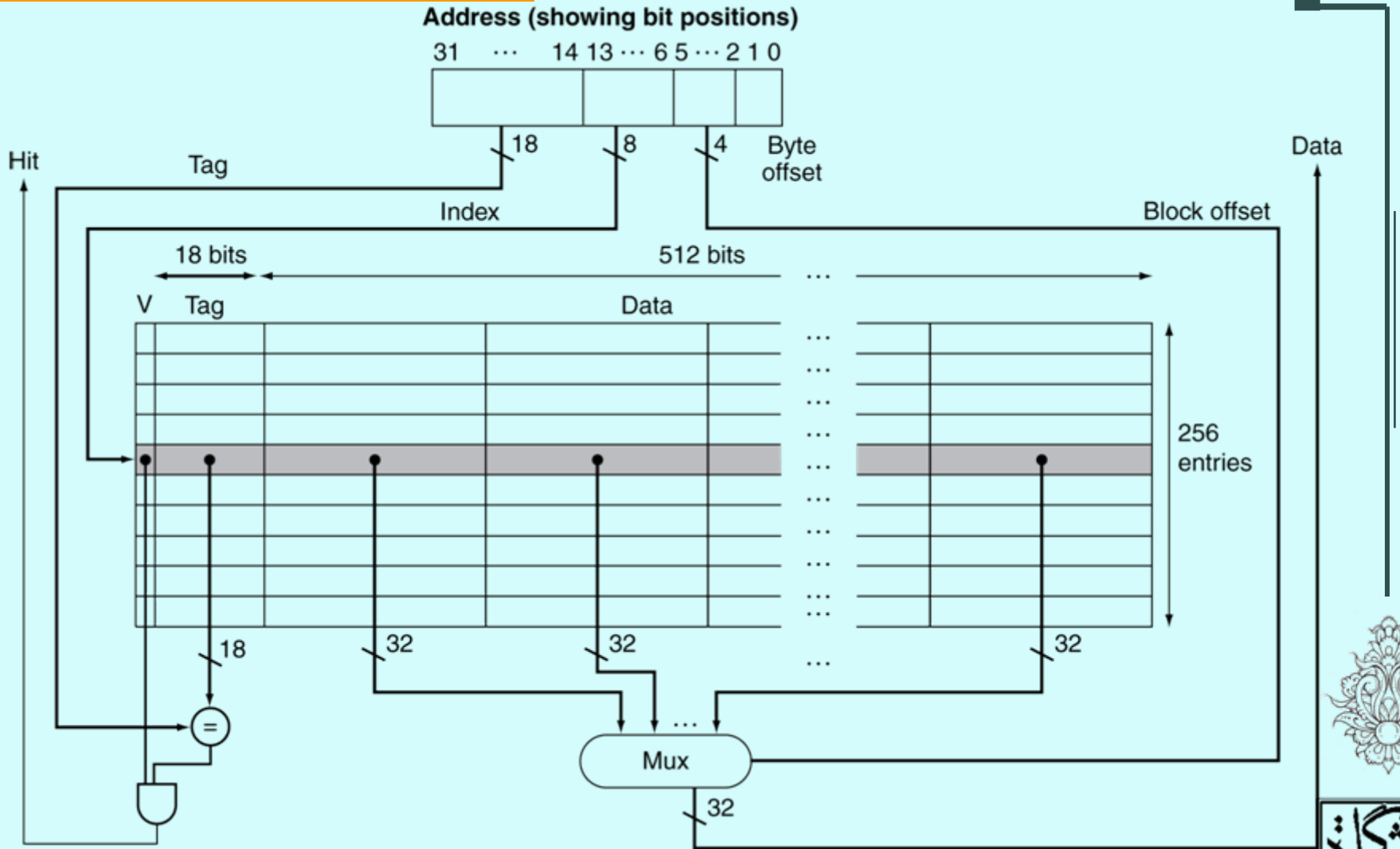
## مثال: پردازنده‌ی Intrinsity FastMATH

- Intrinsity FastMATH یک پردازنده‌ی سریع درون‌کار است که از معماری MIPS بهره می‌گیرد.
- این پردازنده یک خط لوله‌ی دوازده مرحله‌ای دارد.
- دارای دو حافظه‌ی داده و دستورالعمل می‌باشد.
- برای هر حافظه یک حافظه‌ی نهان با گنجایش 16KB، با بلوک شانزده کلمه‌ای وجود دارد.
- در این حال برای هر حافظه‌ی نهان سیگنال‌های کنترلی مجزا نیاز خواهیم داشت.



*Split cache: separate I-cache and D-cache*

# Intrinsity FastMATH



*SPEC2000 miss rates*  
*I-cache: 0.4%*  
*D-cache: 11.4%*  
*Weighted average: 3.2%*

## نقش حافظه‌ی اصلی

- در صورتی که از DRAM با پنهای یک کلمه برای حافظه‌ی اصلی استفاده کنیم.
- برای دستیابی به محتوای حافظه، زمان‌های دستیابی به قرار زیر می‌باشد:

یک سیکل گذرگاه	– برای ارسال آدرس،
پانزده سیکل گذرگاه	– برای خواندن داده،
یک سیکل گذرگاه	– برای ارسال داده،

سیکل ساعت گذرگاه از ساعت پردازنده بهر کندتر است

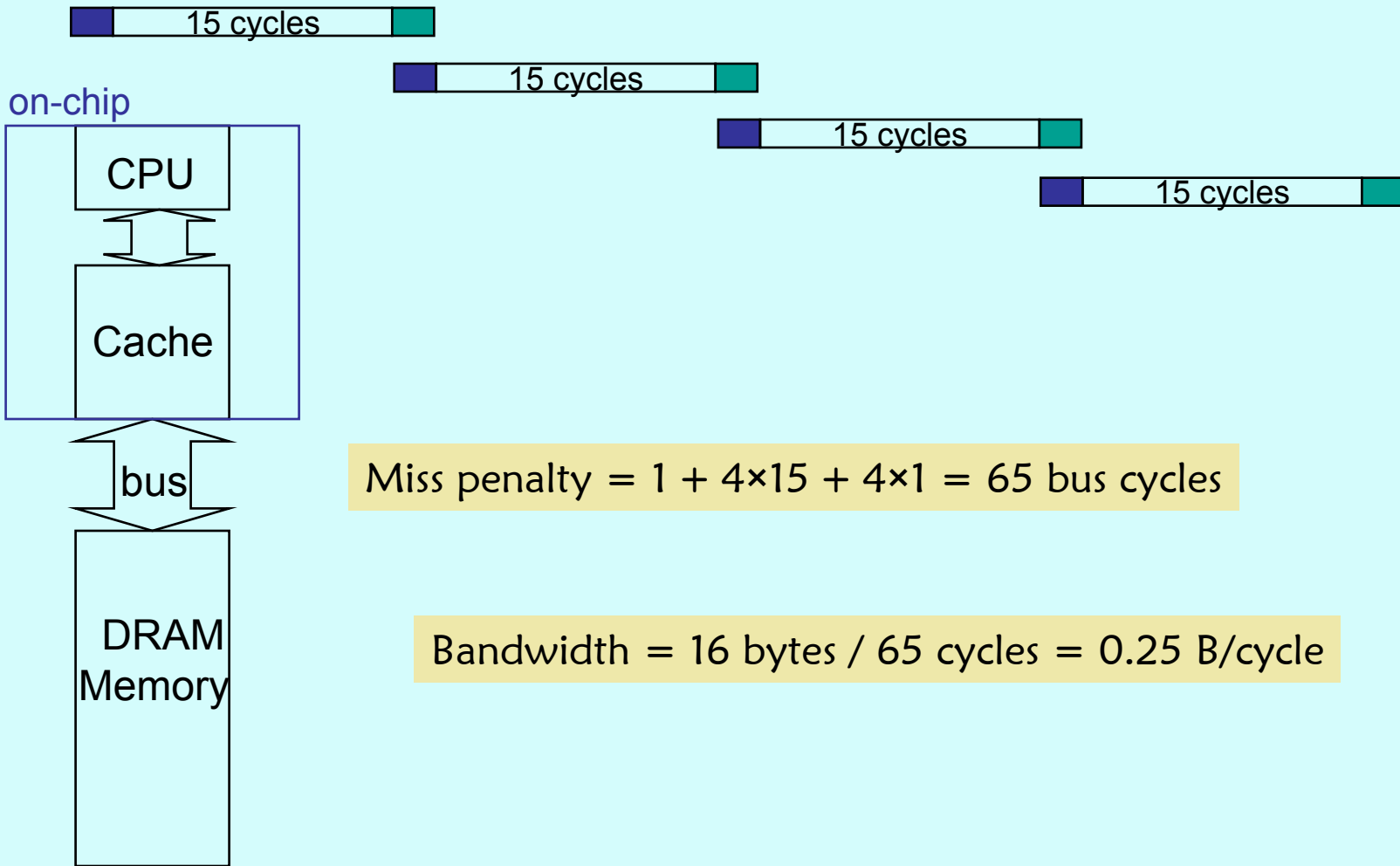
- **جریمه‌ی فقدان** را برای خواندن چهار کلمه به دست آورید.

$$\text{Miss penalty} = 1 + 4 \times 15 + 4 \times 1 = 65 \text{ bus cycles}$$

$$\text{Bandwidth} = 16 \text{ bytes} / 65 \text{ cycles} = 0.25 \text{ B/cycle}$$

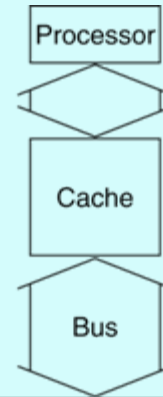
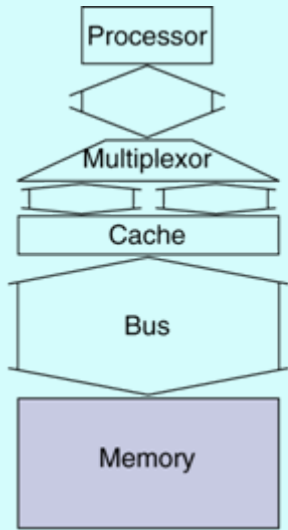
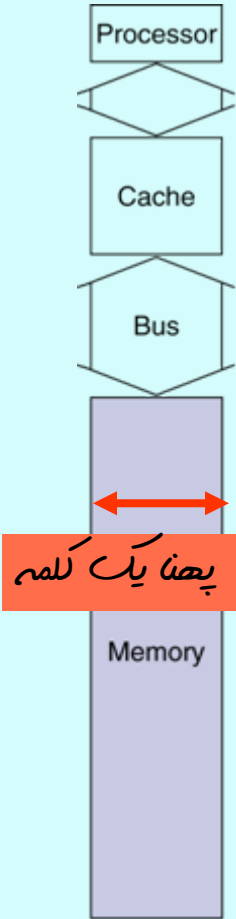


# نقش حافظه‌ی اصلی (ادامه...)



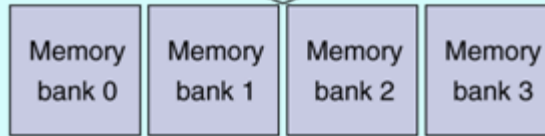
# افزایش پهنای باند حافظه اصلی

Miss penalty = 1 + 15 + 1 = 17 bus cycles  
 Bandwidth = 16 bytes / 17 cycles = 0.94 B/cycle



interleaved Memory

حافظه بزرگ بزرگ شده



Miss penalty = 1 + 15 + 4×1 = 20 bus cycles  
 Bandwidth = 16 bytes / 20 cycles = 0.8 B/cycle

