حافظه ۳

معماری کامپیوتر

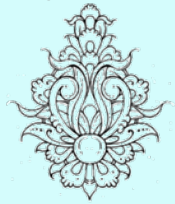
۱۳۵۰۱۱۰۱۳

جلسه‌ی بیست و یکم

دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
بهار ۱۳۹۲
احمد محمودی ازناوه

فهرست مطالب

- سلسله‌مراتب در حافظه



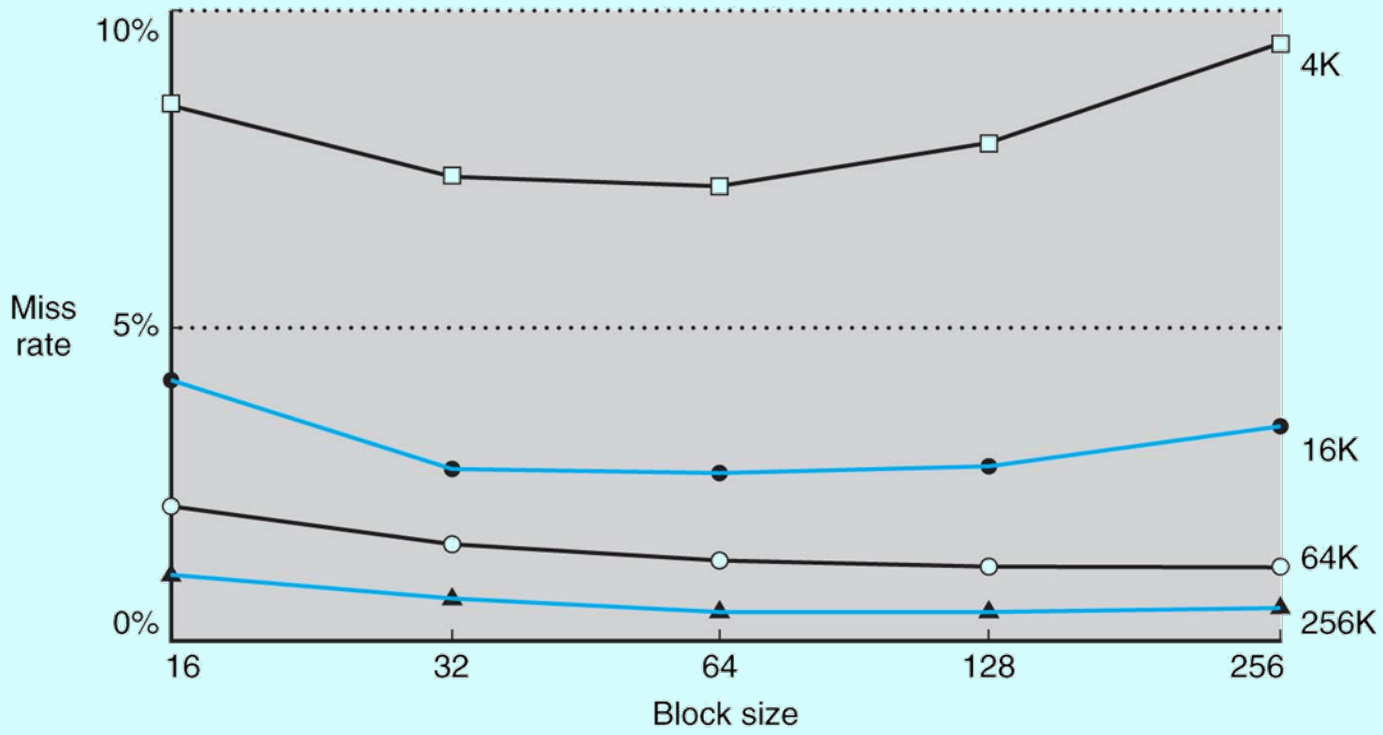
بزرگی بلوک‌ها

- 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.

اما

بزرگی بلوک‌ها (ادامه...)

- بدین‌ترتیب مزیت کاهش miss rate تحت‌الشعاع 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.

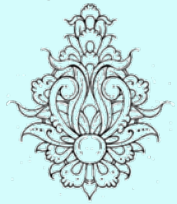
in order processr

out-of-order processor

در این شیوه به جای اجرای برنامه بر اساس توالی دستورالعمل‌ها، ترتیب اجرای دستورها براساس 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 مقداردهی 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، «نوشتن تمام‌عیار» است، بدین معنا که هر آن چه در حافظه‌ی نهان نوشته می‌شود در حافظه‌ی اصلی نیز نوشته شود.

write through

نوشتن در حافظه‌ی نهان (ادامه...)

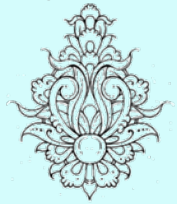
- در صورتی که write miss رخ دهد، چه فرآیندی طی می‌شود؟
  - داده از حافظه اصلی به حافظه‌ی نهانی منتقل می‌شود، مقدار مورد نظر نوشته شده و سپس حافظه‌ی اصلی به روز خواهد شد!
  - چنین شیوه‌ای موجب کندی خواهد شد:
    - مثال: در صورتی که CPI=1 (بدون cache miss) و ده درصد دستورات store باشد و برای نوشتن در حافظه‌ی اصلی صد سیکل لازم باشد، effective CPI برنامه‌ی مورد نظر چقدر خواهد بود؟

Effective CPI = 1 + 0.1×100 = 11

میان‌گیر نوشتن

- داده‌های به جای این که مستقیم 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 حافظه‌ی 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

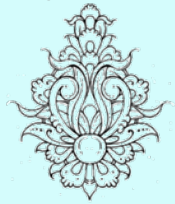
باعث کردن فایده‌ای نخواهد داشت!



write back (copy back)

الگوی بازنویسی

- 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 تغییر یافته (dirty block) را متمایز کنیم.
- در اینجا نیز می‌توان از بافر استفاده نمود.
- طبعاً طراحی سخت‌تر خواهد شد.

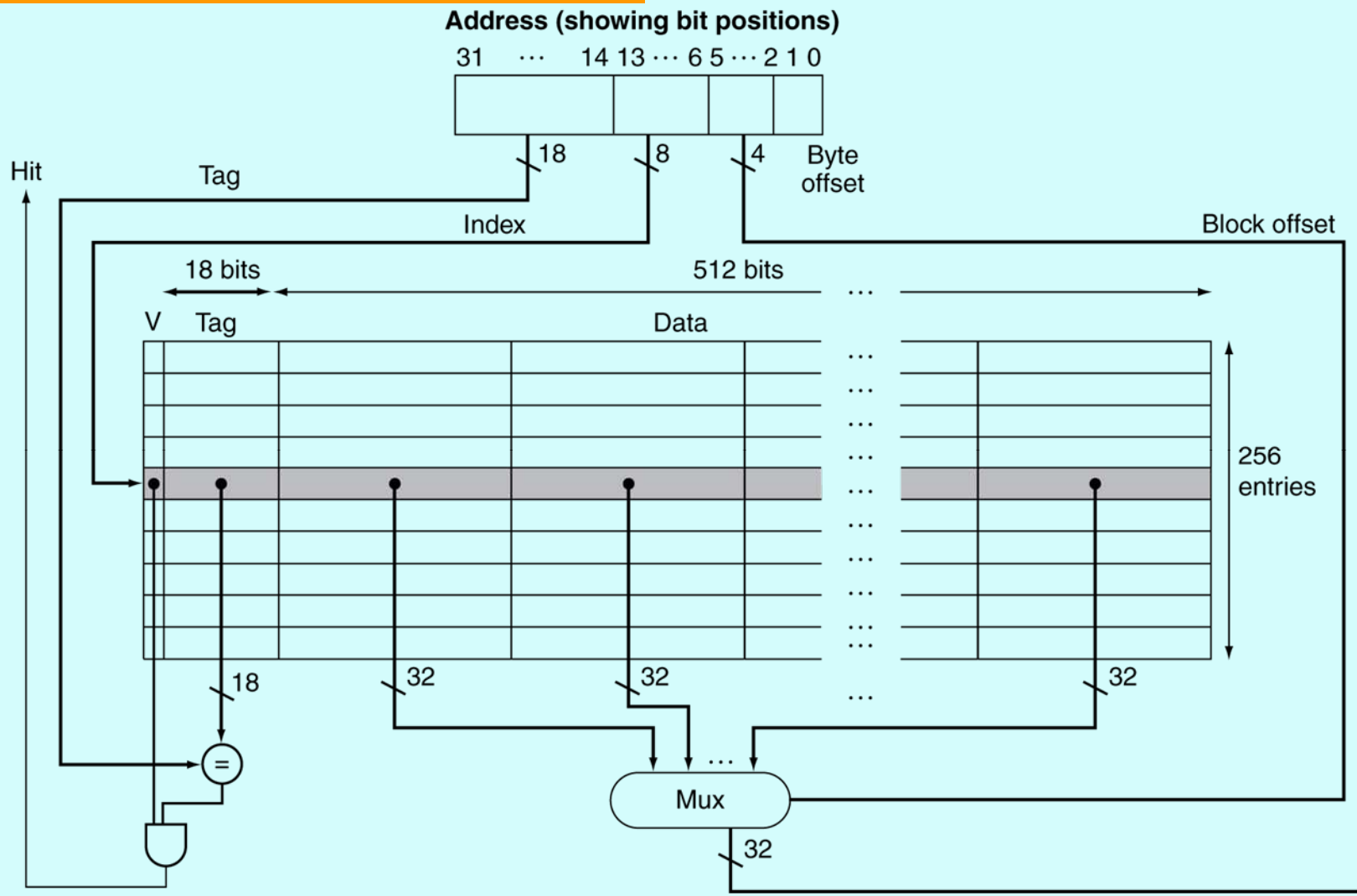


مثال: پردازنده‌ی Intrinsity FastMATH

- 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:
  - برای ارسال آدرس، یک سیکل گذرگاه
  - برای خواندن داده، پانزده سیکل گذرگاه
  - برای ارسال داده، یک سیکل گذرگاه

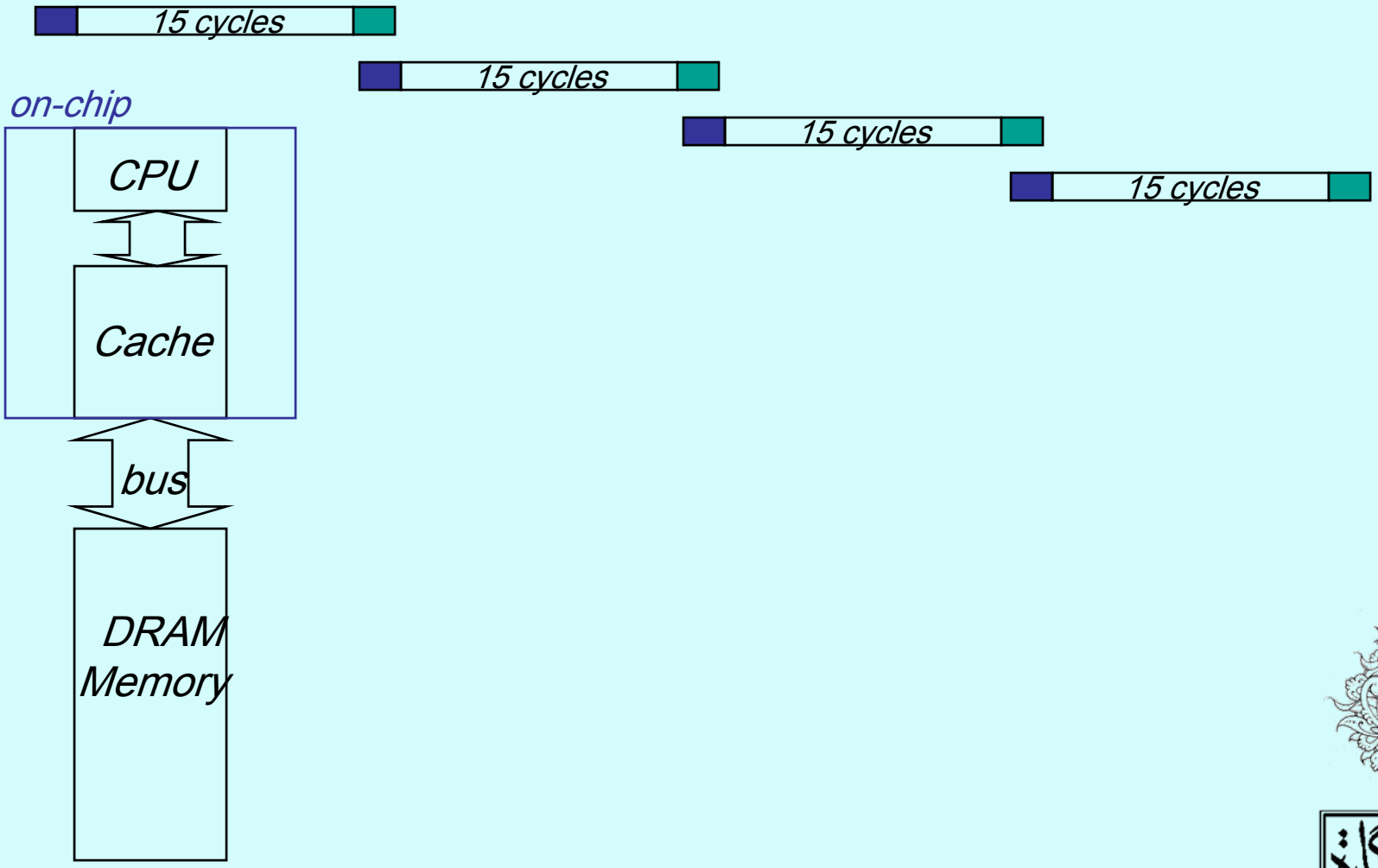
سیکل ساعت گذرگاه از ساعت پردازنده بیگر کندتر است

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

Miss penalty = 1 + 4×15 + 4×1 = 65 bus cycles

Bandwidth = 16 bytes / 65 cycles = 0.25 B/cycle

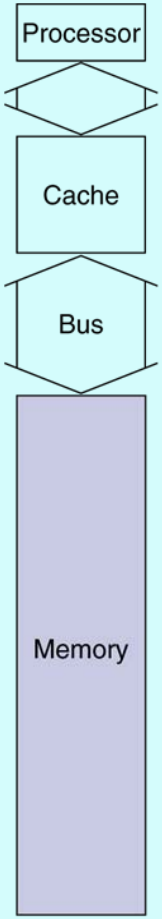
نقش حافظه‌ی اصلی (ادامه...)



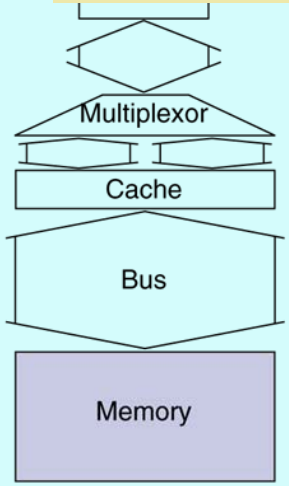
Miss penalty = 1 + 4×15 + 4×1 = 65 bus cycles

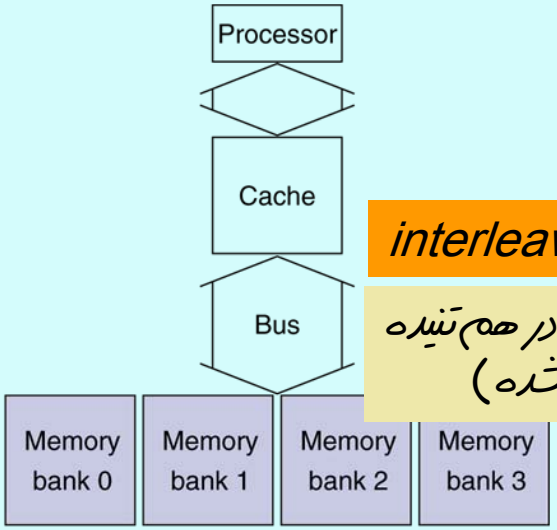
Bandwidth = 16 bytes / 65 cycles = 0.25 B/cycle

افزایش پهنای باند حافظه‌ی اصلی



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





interleaved Memory

حافظه با آلگوی در هم تنیده
(بُرش‌بُرش شده)

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