حافظه ۴

معماری کامپیوتر

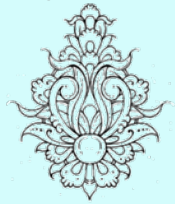
۱۳۹۵-۱۱-۱۳

جلسه‌ی بیست و دوم

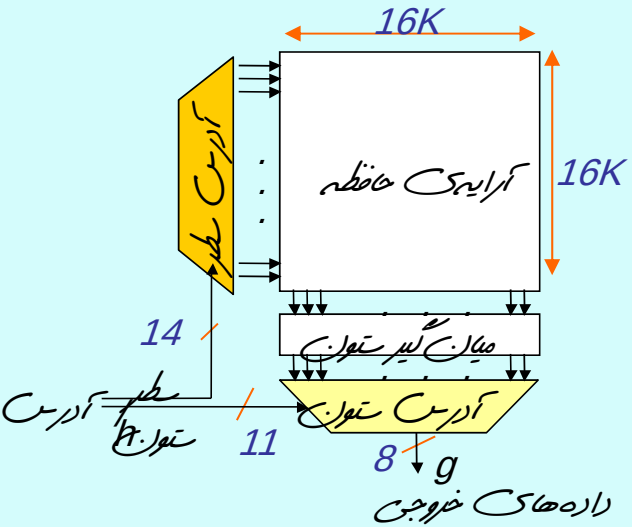
دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
بهار ۱۳۹۵
احمد محمودی ازناوه

فهرست مطالب

- سلسله‌مراتب در حافظه
- حافظه‌های انجمنی
- سیاست‌های جای‌دهی



ساختار حافظه‌های کنونی



- داده در آرایه مربعی ریخته می‌شود
- داده‌های یک سطر به صورت کامل خوانده می‌شوند.
- بدین‌ترتیب تاخیر (latency) برای خواندن کلمات پی‌درپی کاهش می‌یابد

Double data rate (DDR) DRAM

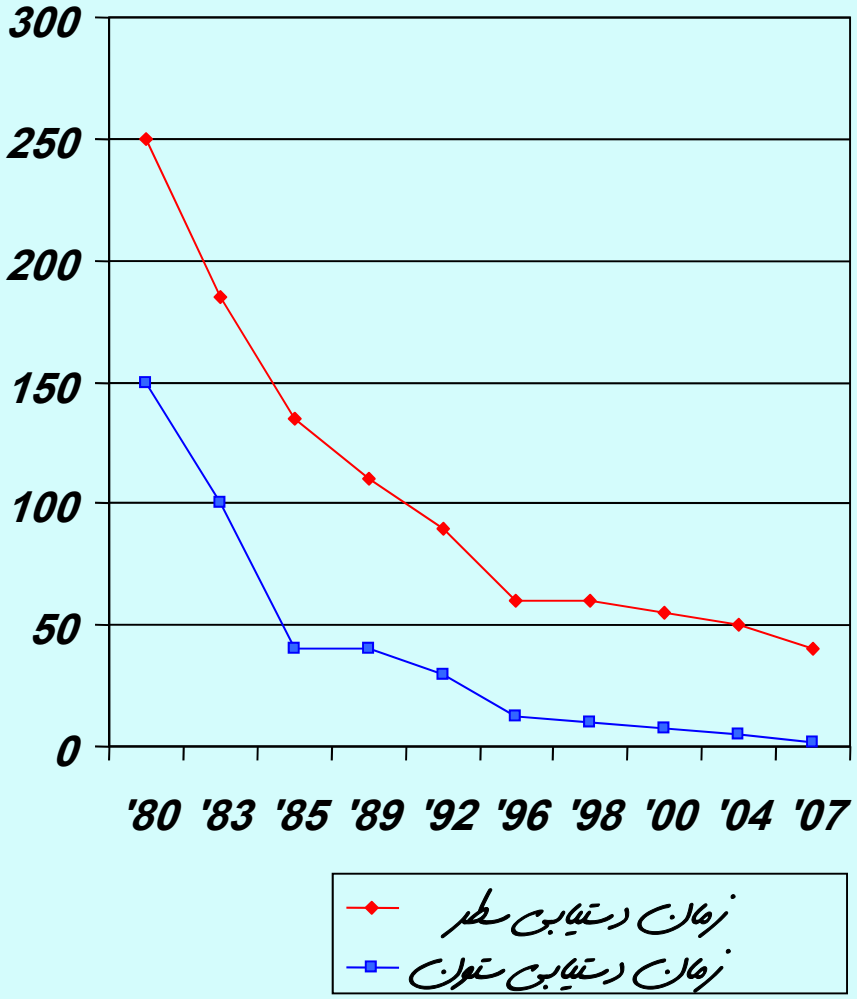
در هر دو لبه‌ی پالس ساعت داده خوانده می‌شود، در این شیوه داده به صورت interleaved سامان‌دهی شده است

دارای درگاه‌های ورودی و خروجی مجزاست

Quad data rate (QDR) DRAM

نسل‌های مختلف DRAM

Year	Capacity	$/GB
1980	64Kbit	$1500000
1983	256Kbit	$500000
1985	1Mbit	$200000
1989	4Mbit	$50000
1992	16Mbit	$15000
1996	64Mbit	$10000
1998	128Mbit	$4000
2000	256Mbit	$1000
2004	512Mbit	$250
2007	1Gbit	$50



اندازه‌گیری کارایی حافظه‌ی نهان (نهان‌گاه)

زمان صرف شده توسط پردازنده

زمان اجرای دستورالعمل‌ها

زمان‌هایی که دچار تعلیق می‌شود

بیشتر به علت cache miss

شامل زمان دسترسی به حافظه‌ی نهان در صورت وجود داده (hit)

$$\text{CPU Time} = (\text{CPU execution clock cycles} + \text{Memory-stall clock cycles}) \times \text{Clock cycle time}$$

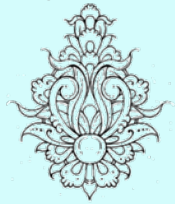
$$\text{Memory-stall clock cycles} = \text{Read-stall cycle} + \text{Write stall cycle}$$

$$\text{Read-stall cycle} = \text{Reads} \times \text{Read rate miss rate} \times \text{Read miss penalty}$$

per program

$$\text{Write-stall cycle} = \text{Writes} \times \text{Write rate miss rate} \times \text{Write miss penalty} + \text{Write buffer stall}$$

per program



اندازه‌گیری کارایی حافظه‌ی نهان (ادامه...)

با ساده‌سازی شرایط

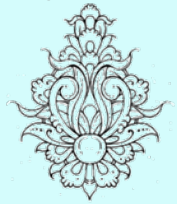
Memory stall cycles

$$= \text{Memory accesses} \times \text{Miss rate} \times \text{Miss penalty}$$

per program

$$= \text{Instructions} \times \text{Misses} \times \text{Miss penalty}$$

per program per Instruction



مثال

- سیستمی با شرایط زیر مفروض است:
  - I-cache miss rate = 2%
  - D-cache miss rate = 4%
  - Miss penalty = 100 cycles
  - Base CPI (ideal cache) = 2
  - Load & stores are 36% of instructions
- در صورتی که یک حافظه‌ی نهان ایده‌آل جایگزین کینم، افزایش سرعت چه مقدار خواهد بود؟

جریمه‌ی فقدان دستوالعمل

$$0.02 \times I \times 100 = 2 \times I$$

جریمه‌ی فقدان داده

$$0.36 \times I \times 0.04 \times 100 = 1.44 \times I$$

زمان اجرا — زمان تعلیق

$$2 \times I + 2 \times I + 1.44 \times I = 5.44 \times I$$

Actual CPI

$$5.44/2 = 2.72$$

بهبود کارایی

ادامه مثال

در صورتی که سرعت پردازنده را افزایش دهیم، بدون این که در سیستم حافظه تغییری ایجاد نشود، چه اتفاقی خواهد افتاد؟

قانون **Amdhal**

$$T_{improved} = \frac{T_{affected}}{improvement\ factor} + T_{unaffected}$$

- در صورتی که سرعت پردازنده‌ی مثال را دو برابر کنیم، در شرایطی که فرکانس پالس ساعت تغییر نکند، CPI چه تغییری خواهد کرد؟

$1\times I + 3.44\times I = 4.44\times I$

زمان صرف شده برای تعلیق حافظه

حالت اول: $\frac{3.44}{5.44} = 63\%$

حالت دوم: $\frac{3.44}{4.44} = 77\%$

Average Memory Access Time

متوسط زمان دست‌یابی

AMAT

- از سوی دیگر، افزایش کلاک پردازنده نیز منجر به عدم بهره‌وری (در اثر نبود اطلاعات) در حافظه‌ی نهان خواهد شد.
- زمان hit time، نیز می‌تواند زمان کل دستیابی به حافظه را افزایش دهد. این مسأله هنگامی رخ می‌دهد که گنجایش حافظه‌ی نهان افزایش یابد.
- با توجه دشواری‌های مطرح شده، گاهی معیار زیر تعریف می‌شود:

$$AMAT = Hit\ time + Miss\ rate \times Miss\ penalty$$

مثال: یک پردازنده با کلاک 1ns مفروض است، زمان مشخص شدن hit یک سیکل، جریمه‌ی فقدان بیست سیکل است. نرخ فقدان پنج درصد است، متوسط زمان دست‌یابی را حساب کنید.

$$AMAT = 1 + 0.05 \times 20 = 2ns$$

- دستیابی به بلوک‌های زیر در حافظه‌ی اصلی را در نظر بگیرید:

0 4 0 4 0 4 0 4

0 miss

00	Mem(0)

4 miss

01	
00	Mem(0) 4

0 miss

00	
01	Mem(4) 0

4 miss

01	
00	Mem(0) 4

0 miss

00	
01	Mem(4) 0

4 miss

01	
00	Mem(0) 4

0 miss

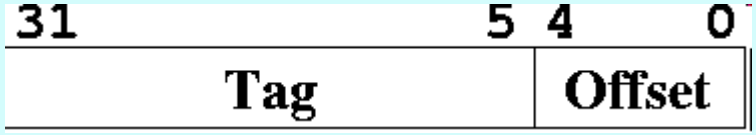
00	
01	Mem(4) 0

4 miss

01	
00	Mem(0) 4

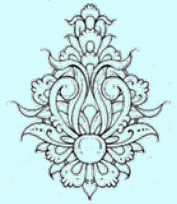
حافظه‌ی نهان (اشتراکی، انجمنی)

- تا کنون از شیوه‌ی نگاشت مستقیم برای جای‌گذاری بلوک‌ها استفاده کردیم.
  - شیوه‌ی دیگر این است که هر بلوک بتواند در هر جای حافظه‌ی نهان قرار بگیرد. این شیوه به نام انجمنی (associative) شهرت دارد.
  - در این صورت بر طول برچسب افزوده می‌شود و تمامی برچسب‌ها می‌باید مورد بررسی قرار گیرند.
  - برای چنین کاری از مقایسه‌کننده‌های موازی استفاده می‌شود که هزینه‌ی سخت‌افزاری بالایی دارد.



حافظه‌ی نهان با مجموعه‌های اشتراکی

- یک راه دیگر برای کاهش هزینه‌ها، استفاده از راهی بین نکاشت مستقیم و حافظه‌ی نهان اشتراکی است:
  - حافظه‌ی نهان با مجموعه‌های اشتراکی(شبه انجمنی)
- در این شیوه هر بلوک از حافظه‌ی اصلی در مکان‌های خاصی از حافظه‌ی اصلی می‌توانند قرار گیرند.
- در صورتی که هر بلوک در n محل از حافظه‌ی نهان قابل جای‌گذاری باشد آن را *n*-way set associative می‌نامند.
- در مقابل شیوه‌ی پیشین به اشتراکی کامل(تمام انجمنی) (Fully associative) مشهور است.



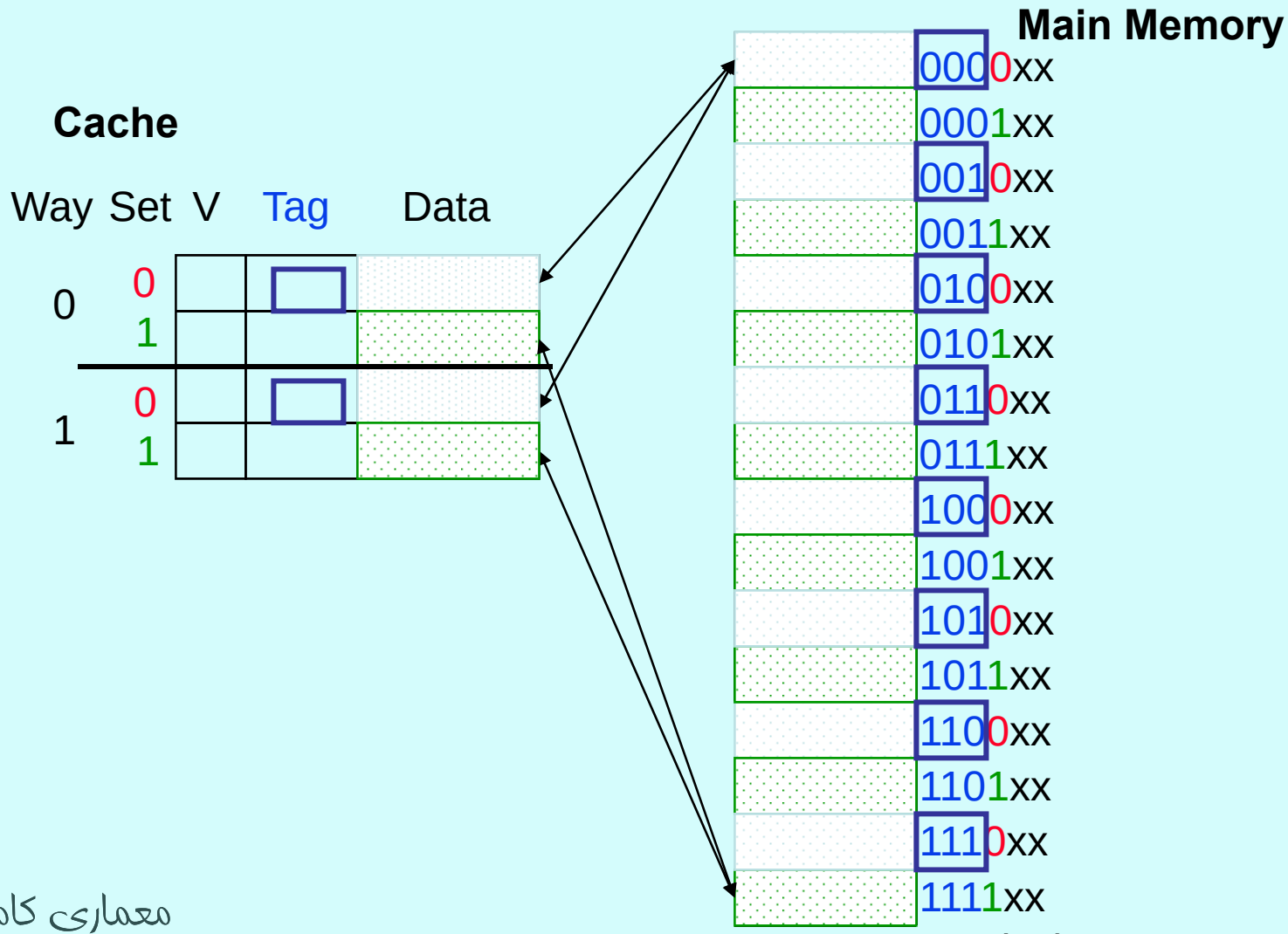
حافظه‌های نهان اشتراکی

(Block address) modulo (#Blocks in cache)

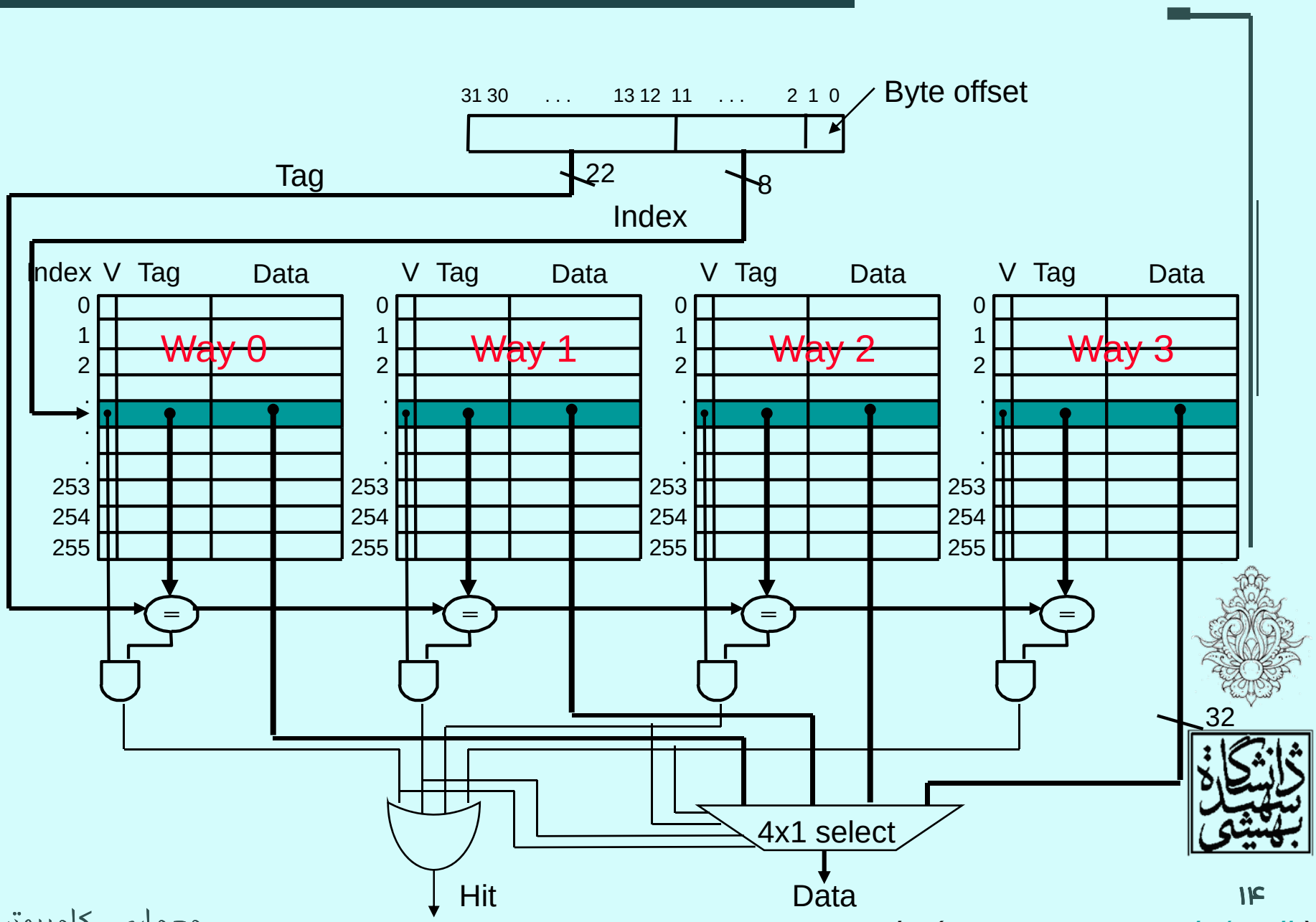
Direct Map

(Block address) modulo (#Sets in cache)

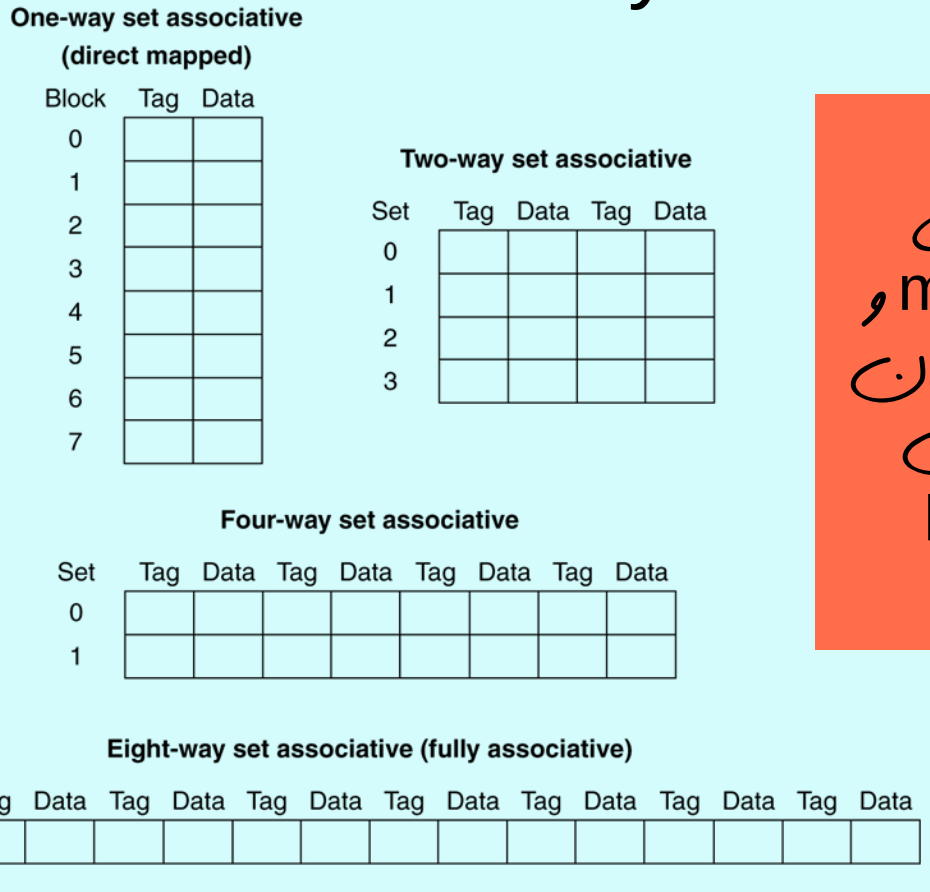
Set Associative


Main Memory
Cache
Way Set V Tag Data
0
1
0
1
0
1
0000xx
0001xx
0010xx
0011xx
0100xx
0101xx
0110xx
0111xx
1000xx
1001xx
1010xx
1011xx
1100xx
1101xx
1110xx
1111xx

Four-Way Set Associative Cache

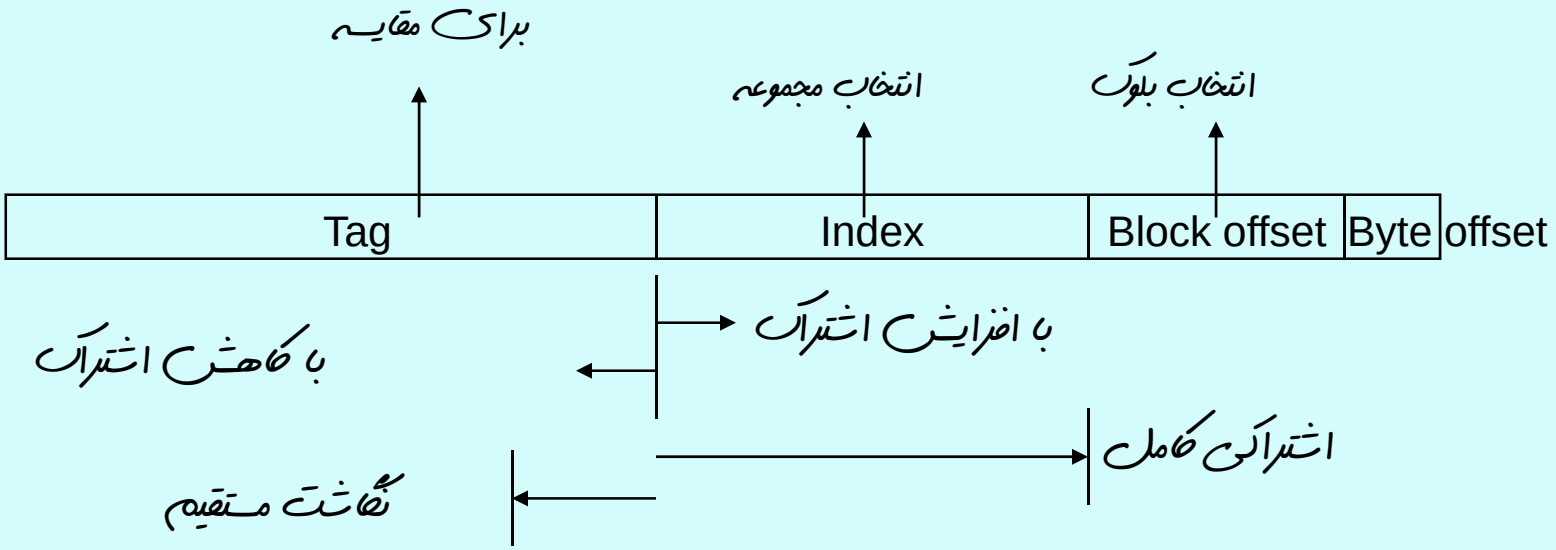


- تمام شیوه‌های جای‌دهی را به نوعی می‌توان n-way set associative دانست.



مهمترین برتری شیوه‌های اشتراکی کاهش miss-rate و بزرگ‌ترین مشکل آن افزایش مشکل آن افزایش hit-time است

طیف اشتراک (ادامه...)



مثال

- یک حافظه‌ی نهان چهار بلوکی در اختیار داریم،
- هدف مقایسه‌ی miss-rate در حالات زیر است:
  - نگاشت مستقیم
  - اشتراکی دو بلوکه
  - اشتراکی کامل
- ترتیب بلوک‌های به صورت زیر است:
  - 0, 8, 0, 6, 8

Direct mapped

Block address	Cache index	Hit/miss	Cache content after access			
			0	1	2	3
0	0	miss	**Mem[0]**			
8	0	miss	**Mem[8]**			
0	0	miss	**Mem[0]**			
6	2	miss	Mem[0]		**Mem[6]**	
8	0	miss	**Mem[8]**		Mem[6]	

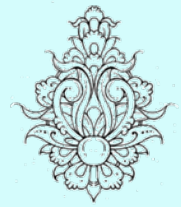
مثال

2-way set associative

Block address	Cache index	Hit/miss	Cache content after access			
			Set 0		Set 1	
0	0	miss	**Mem[0]**			
8	0	miss	Mem[0]	**Mem[8]**		
0	0	hit	**Mem[0]**	Mem[8]		
6	0	miss	Mem[0]	**Mem[6]**		
8	0	miss	**Mem[8]**	Mem[6]		

Block address		Hit/miss	Cache content after access			
0		miss	**Mem[0]**			
8		miss	Mem[0]	**Mem[8]**		
0		hit	**Mem[0]**	Mem[8]		
6		miss	Mem[0]	Mem[8]	**Mem[6]**	
8		hit	Mem[0]	**Mem[8]**	Mem[6]	

Fully associative



قابلیت اشتراک تا چه حد؟

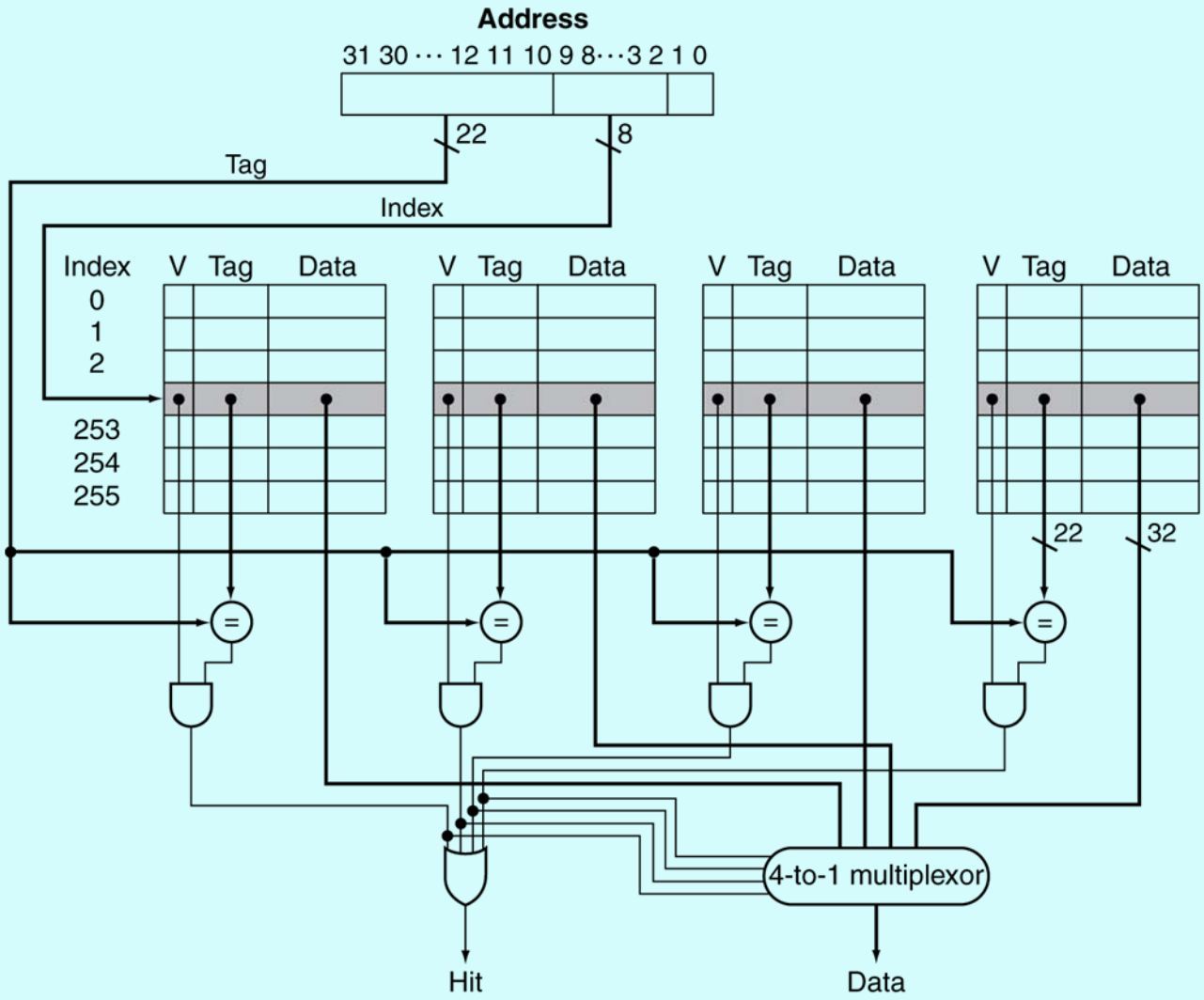
سوال؟؟

- هر چه قابلیت اشتراک بیشتر باشد، نرخ miss-rate کاهش می‌یابد.
- تا چه حد این قابلیت را افزایش دهیم؟
- نتایج شبیه‌سازی یک سیستم، با 64KB و بلوک‌های شانزده کلمه‌ای که با SPEC2000:

1-way: 10.3%
2-way: 8.6%
4-way: 8.3%
8-way: 8.1%

هر چند miss-rate کاهش می‌یابد، اما روند این کاهش، رفته‌رفته کم‌تر می‌شود

ساختار حافظه‌ی نهان با مجموعه‌های اشتراکی



در حافظه‌های نهان انجمنی، زمان جستجو اهمیت بسزایی در کارایی دارد

Content Addressable Memory (CAM)

Associative memory

حافظه‌ی انجمنی!!

حافظه‌ی تداعی‌گر

- در بسیاری کاربردها، لازم است یک آیتم در یک جدول (حافظه) جستجو شود، که فرآیندی زمان‌بر است.
- در صورتی که بتوان حافظه‌ای ساخت که با ارائه‌ی داده، آدرس را بیابد، کارایی فرآیند جستجو به صورت مؤثری بهبود خواهد یافت. چنین حافظه‌ای، حافظه‌ی تداعی‌گر خوانده می‌شود.
- این نوع حافظه‌ها، هزینه‌ی بالاتر نسبت به حافظه‌های معمولی دارند، و بدین‌سبب در کاربردهایی که زمان جستجو نقشی حیاتی دارد، به کار می‌روند.

در حافظه‌های نهان انجمنی، علاوه بر داده، (بخشی) از آدرس را نیز ذخیره می‌کند. چنین حافظه‌ای از یک حافظه‌ی معمولی و یک حافظه‌ی تداعی‌گر تشکیل شده است.

Replacement Policy

سیاست جایگزینی

- در نگاشت مستقیم، جایی که بلوک باید در آن قرار گیرد مشخص است.
- در روش‌های اشتراکی بلوک در چند محل متفاوت می‌تواند قرار گیرد.
  - در درجه‌ی اول مکانی انتخاب می‌شود که بیت‌اعتبار آن غیر فعال است.
  - در غیر این صورت از بلوکی که کم‌تر مورد استفاده قرار گرفته است(اخیرا کم‌استفاده‌ترین بلوک)، از حافظه‌ی نهان خارج می‌شود.

هر چه تعداد مجموعه‌های مشترک افزایش یابد، هزینه‌ی سخت‌افزاری LRU افزایش خ

Least-recently used (LRU)

راه دیگر، انتخاب تصادفی است

برای حالات اشتراک با مجموعه‌های بزرگ کارایی یک نی با LRU دارد