

●●● معماری کامپیوتر (۱۳۹۱-۱۱-۱۳۳)

جلسه‌ی بیست و دوم



دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
بهار ۱۳۹۱
احمد محمودی ازناوه

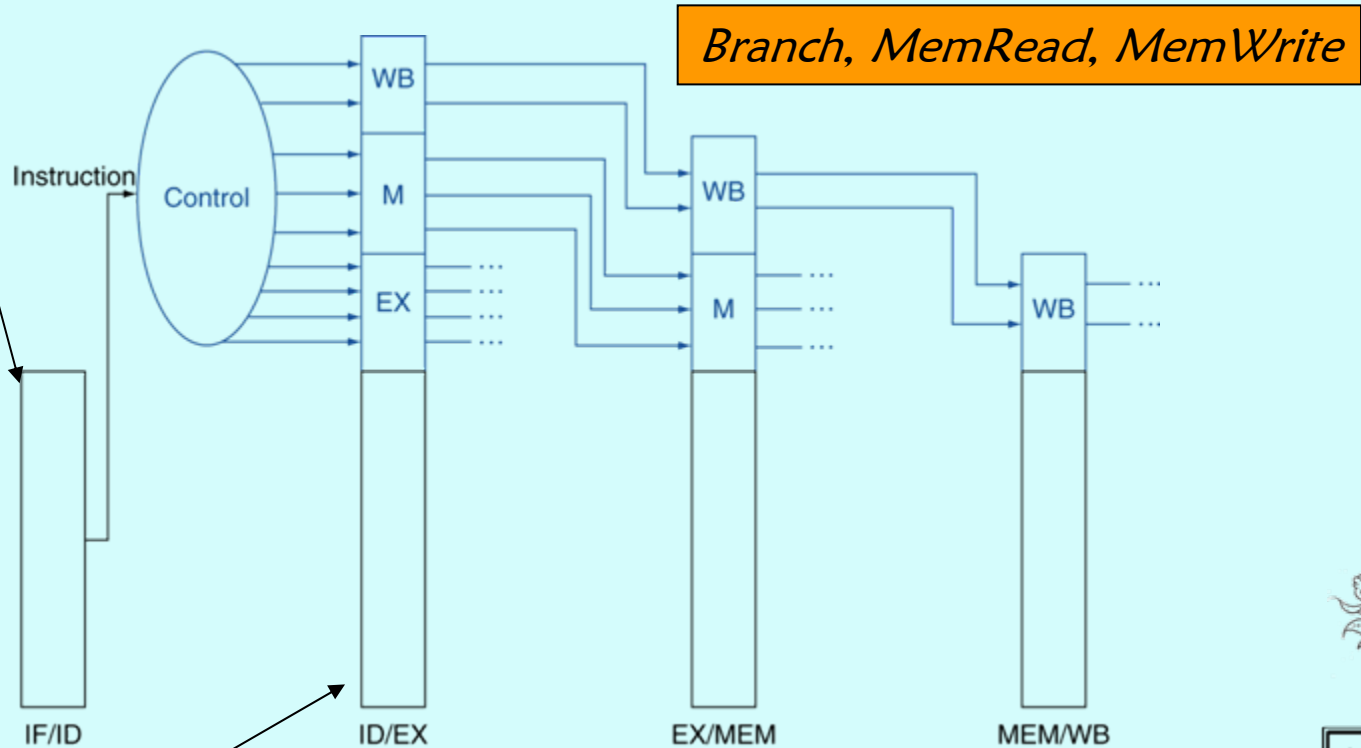
فهرست مطالب

- مروری بر جلسات پیش
- ساختار کنونی حافظه
- حافظه‌ی نهان اشتراکی
- مجموعه‌های اشتراکی



واحد کنترل خط لوله (ادامه...)

در این بخش می‌باید دستور العمل بعدی واکنشی شود، کاری که برای هم‌بندی دستورها به یک شیوه خواهد بود



همانند لام پیشین، نیاز به سیگنال کنترل خاصی نیست

RegDst, ALUOp, ALUSrc

MemtoReg, RegWrite



تشخیص نیاز به پیش فرستادن

- بنابراین، می‌توان با مقایسه‌ی محتوای ثبات‌ها، مداری برای کنترل پیش‌فرستادن داده طراحی کرد.

```
sub $2, $1, $3 EX/MEM.RegisterRd = ID/EX.RegisterRs = $2
and $12, $2, $5
```

در همه‌ی دستورالعمل‌های مقدار خروجی ALU، در ثبات نوشته نمی‌شود بدین ترتیب این راهکار در همه‌ی موارد درست نخواهد بود.



برای پیش‌گیری از این مسأله، می‌توان از سیگنال‌های کنترلی مربوط به WB ذخیره شده در ثبات‌های خط لوله استفاده نمود.

```
EX/MEM.RegWrite, MEM/WB.RegWrite
```

همچنین در صورتی که ثبات شماره‌ی صفر به عنوان مقصد یک دستور استفاده شده باشد، باید از پیش فرستادن جلوگیری کرد.

```
EX/MEM.RegisterRd ≠ 0,
MEM/WB.RegisterRd ≠ 0
```



شرایط پیش فرستادن



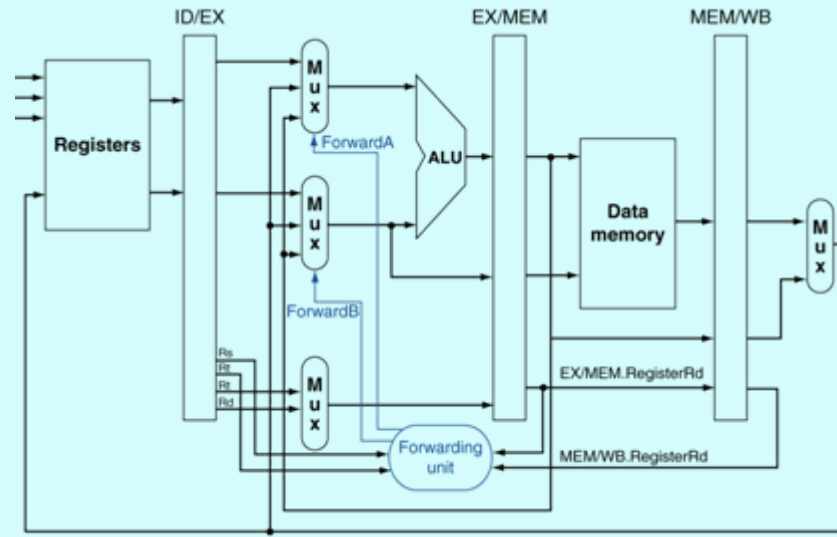
- EX hazard
 - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd \neq 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
ForwardA = 10
 - if (EX/MEM.RegWrite and (EX/MEM.RegisterRd \neq 0) and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
ForwardB = 10
- MEM hazard
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd \neq 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRs))
ForwardA = 01
 - if (MEM/WB.RegWrite and (MEM/WB.RegisterRd \neq 0) and (MEM/WB.RegisterRd = ID/EX.RegisterRt))
ForwardB = 01

اگر هر دوی این شرایط با هم رخ داد، چه اشکالی ایجاد می شود؟

شرایط پیش فرستادن (ادامه...)

- در این قطعه برنامه هر دو نوع مخاطره رخ می‌دهد.

```
add $I,$I,$2
add $I,$I,$3
add $I,$I,$4
```



در این حالت آخرین نتیجه باید فرستاده شود، در نتیجه داده‌ی موجود در مرحله‌ی MEM فرستاده می‌شود.

- بنابراین باید تخییراتی در مخاطره‌ی MEM بدهیم



شرایط پیش فرستادن (ادامه...)



- MEM hazard

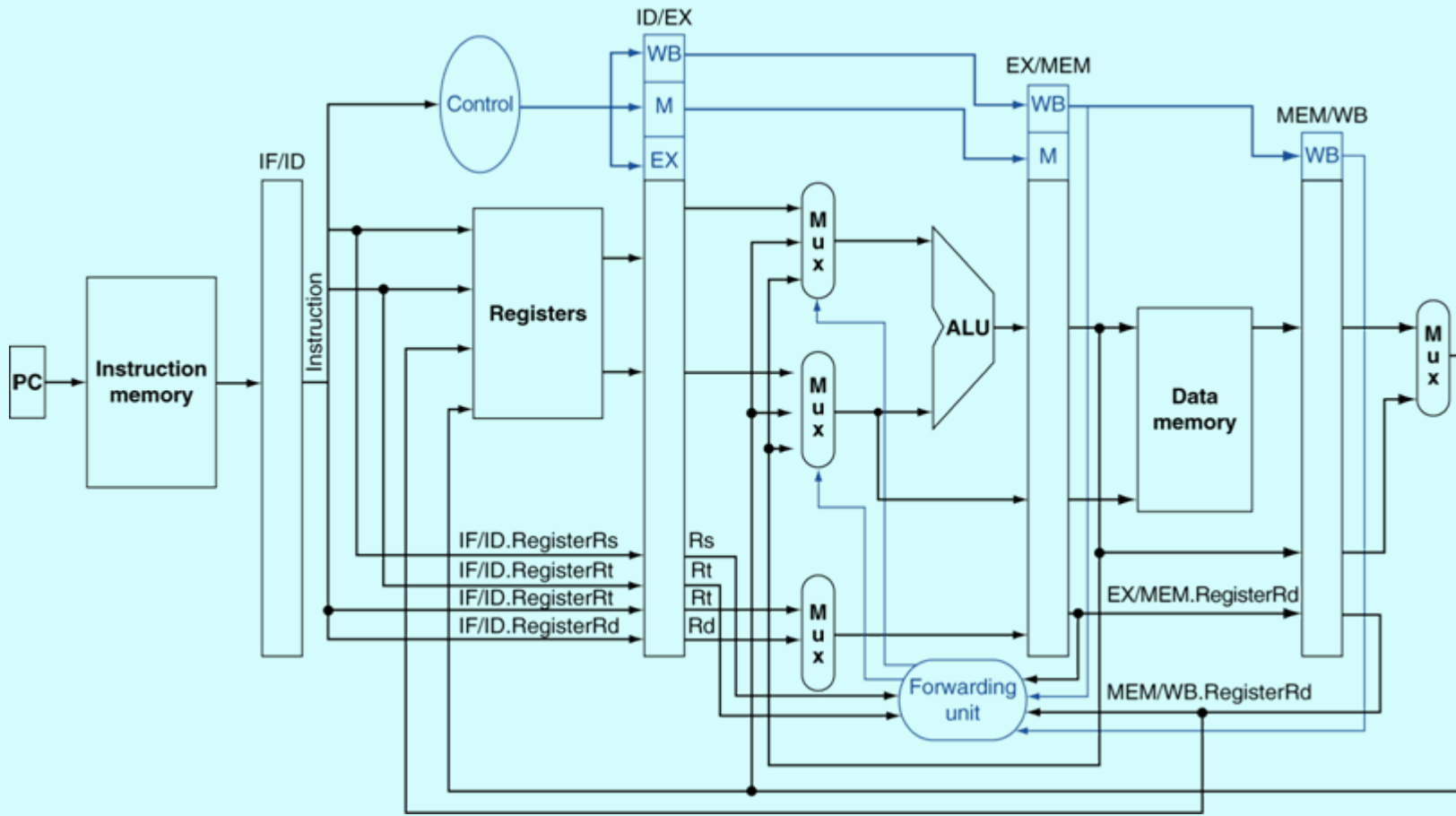
- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd \neq 0)
 and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd \neq 0)
 and (EX/MEM.RegisterRd = ID/EX.RegisterRs))
 and (MEM/WB.RegisterRd = ID/EX.RegisterRs))

ForwardA = 01

- if (MEM/WB.RegWrite and (MEM/WB.RegisterRd \neq 0)
 and not (EX/MEM.RegWrite and (EX/MEM.RegisterRd \neq 0)
 and (EX/MEM.RegisterRd = ID/EX.RegisterRt))
 and (MEM/WB.RegisterRd = ID/EX.RegisterRt))

ForwardB = 01

شرایط پیش فرستادن (ادامه...)



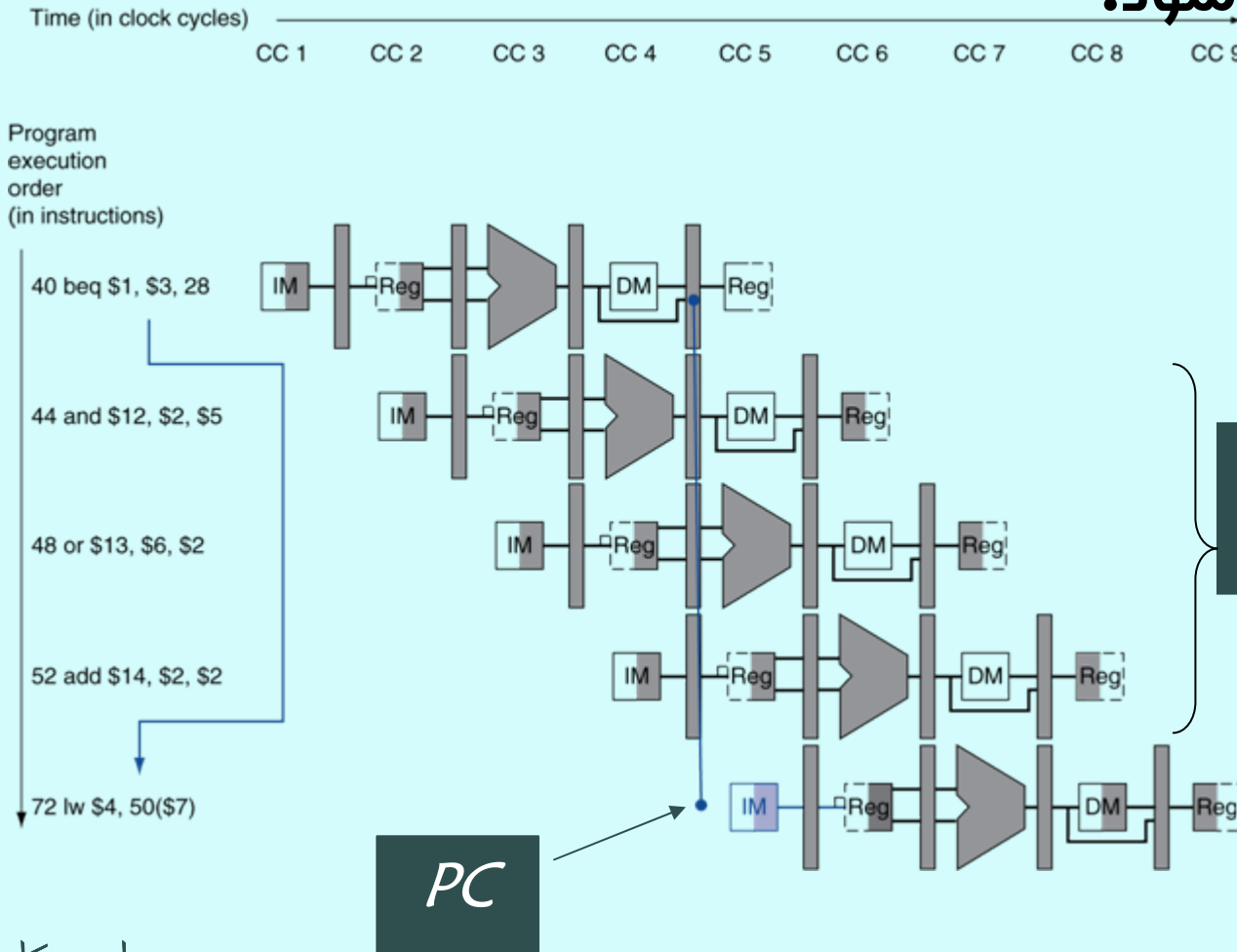
در این شکل بخشی که برای ارسال داده‌ی ثابت به ALU بود، حذف شده است



تراشگاه
تجهیزات
بهرشتی

مفاهیم کنترل

- نتیجه‌ی دستور پرش در مرحله‌ی **MEM** مشخص می‌شود.



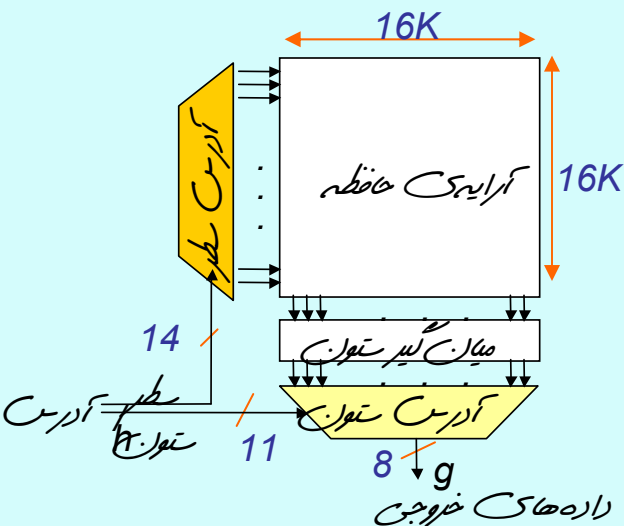
مفاظرات كنترلی (ادامه...)

- ایجاد تعلیق، موجب كندی می‌شود.
- یک راه حل، این است كه فرض کنیم هیچ پرشی انجام نمی‌شود.
- در صورت تحقق، اجرای دستورات واكشی شده، ملغی می‌گردد.
- برای این كار سیگنال‌های كنترلی **غیرفعال** می‌شوند.
- دستورات العمل‌ها از ثبات خط لوله پای می‌شوند.

flush



ساختار حافظه‌های کنونی



- داده در آرایه مربعی ریخته می‌شود
- داده‌های یک سطر به صورت کامل خوانده می‌شوند.
- بدین‌تریب 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 سازماندهی شده است

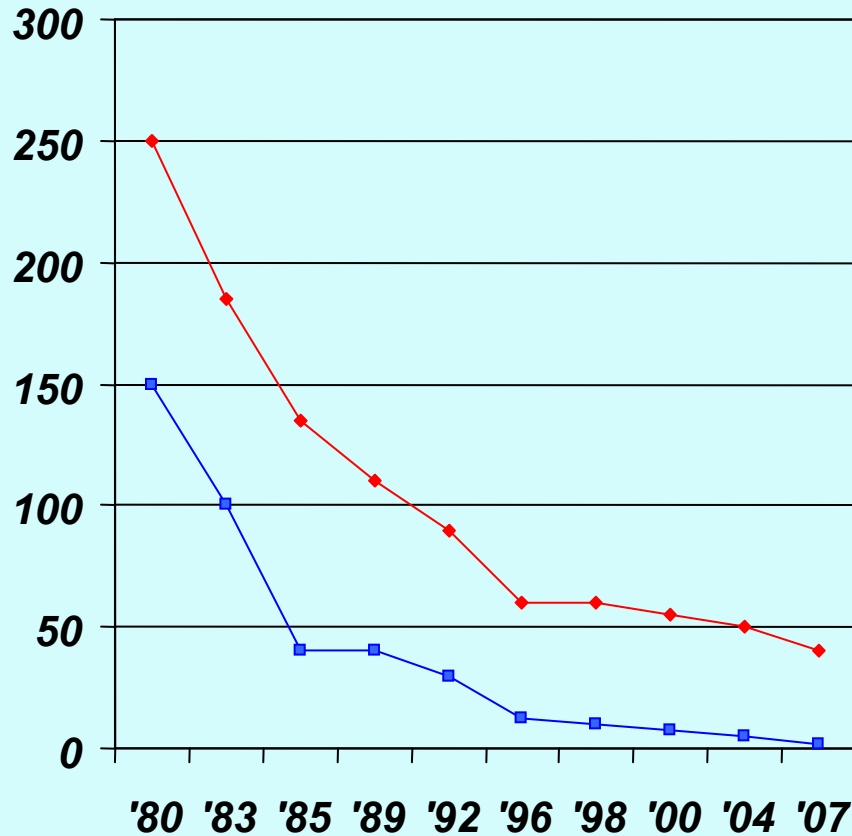
دارای درگاه‌های ورودی و خروجی مجزا است

Quad data rate (QDR) DRAM



DRAM نسل‌های مختلف

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



—♦— زمان دستیابی سطح
—■— زمان دستیابی ستون



اندازه‌گیری کارایی حافظه‌ی نهان (نهان‌گاه)

زمان صرف شده توسط پردازنده

بیشتر به علت cache miss

زمان‌هایی که دچار تعلیق می‌شود

زمان اجرای دستورالعمل‌ها

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

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

per program

Write buffer stall

+



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 کنیم، افزایش سرعت چه مقدار خواهد بود؟

جریمه‌ی فقدان دستورالعمل

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

جریمه‌ی فقدان داده

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

زمان تعلیق

زمان اجرا

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

$$5.44 / 2 = 2.72$$

بهبود کارایی

Actual CPI



ادامه مثال

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

قانون Amdahl

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

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

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

$$\frac{3.44}{5.44} = 63\%$$

زمان صرف شده برای تعلیق حافظه

$$\frac{3.44}{4.44} = 77\%$$

حالت دوم



- 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:

$$AMAT = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

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 حافظه اصلی را در نظر بگیرید:

0 4 0 4 0 4 0 4

0 miss

00	Mem(0)

4 miss

00	Mem(0)

0 miss

01	Mem(4)

4 miss

00	Mem(0)

0 miss

01	Mem(4)

4 miss

00	Mem(0)

0 miss

01	Mem(4)

4 miss

00	Mem(0)



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.

31	5 4	0
Tag	Offset	



حافظه‌ی نهان با مجموعه‌های اشتراکی

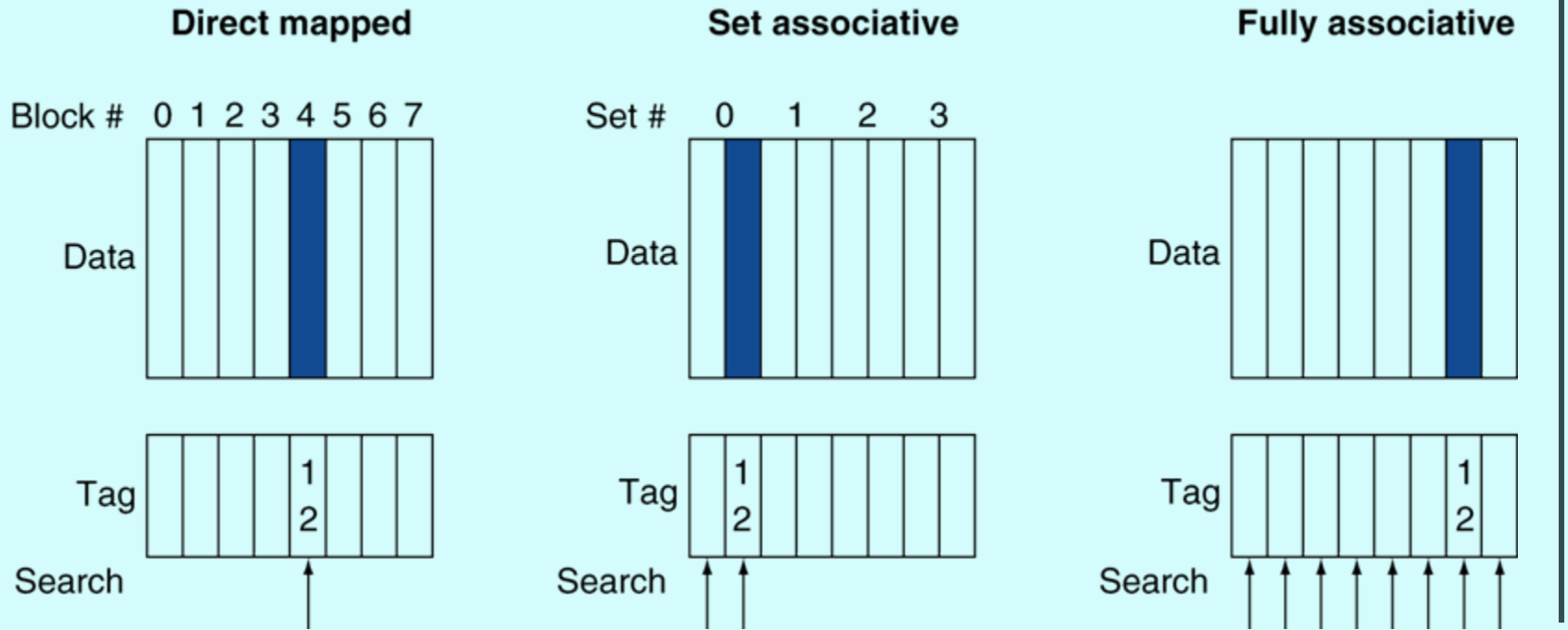
- یک راه دیگر برای کاهش هزینه‌ها، استفاده از راهی بین نکاشت مستقیم و حافظه‌ی نهان اشتراکی است:

– حافظه‌ی نهان با مجموعه‌های اشتراکی

- 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 (**Fully associative**) مشهور است.



حافظه‌های نهان اشتراکی



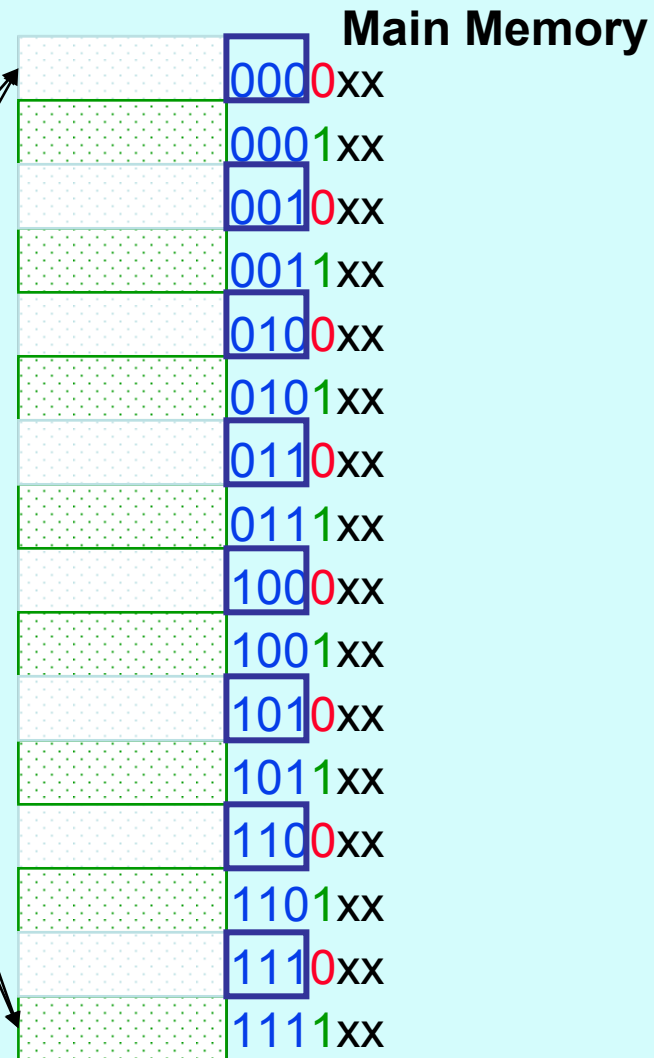
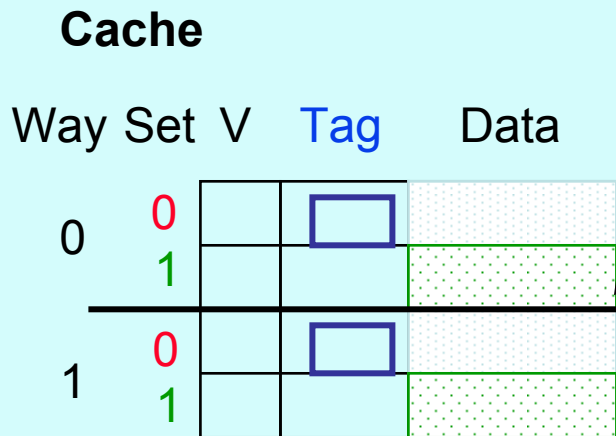
(Block address) modulo (#Blocks in cache)

Direct Map

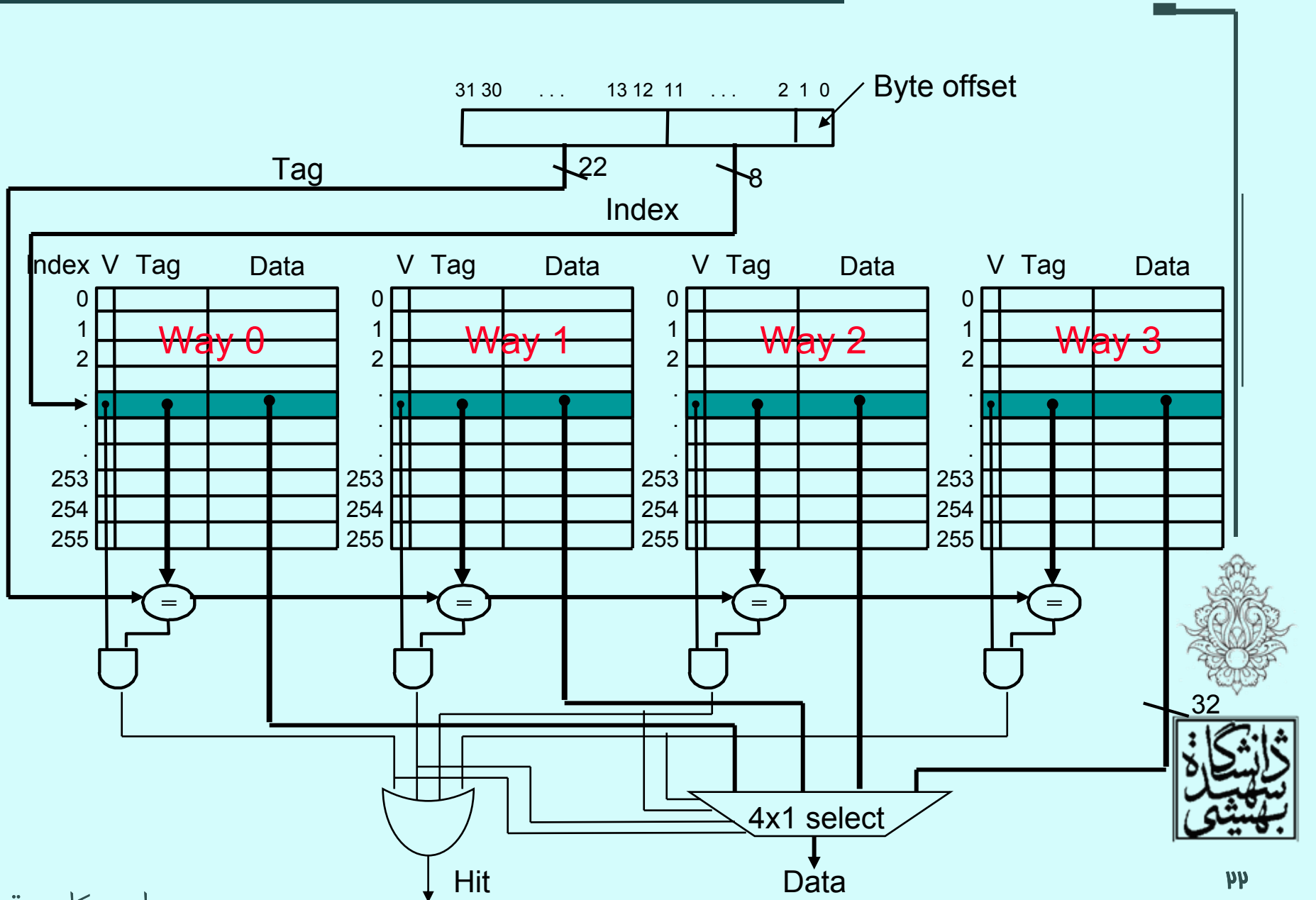
(Block address) modulo (# Sets in cache)

Set Associative





Four-Way Set Associative Cache



32
 دانشگاه
 شهید
 بهشتی

طیف اشتراک

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

One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

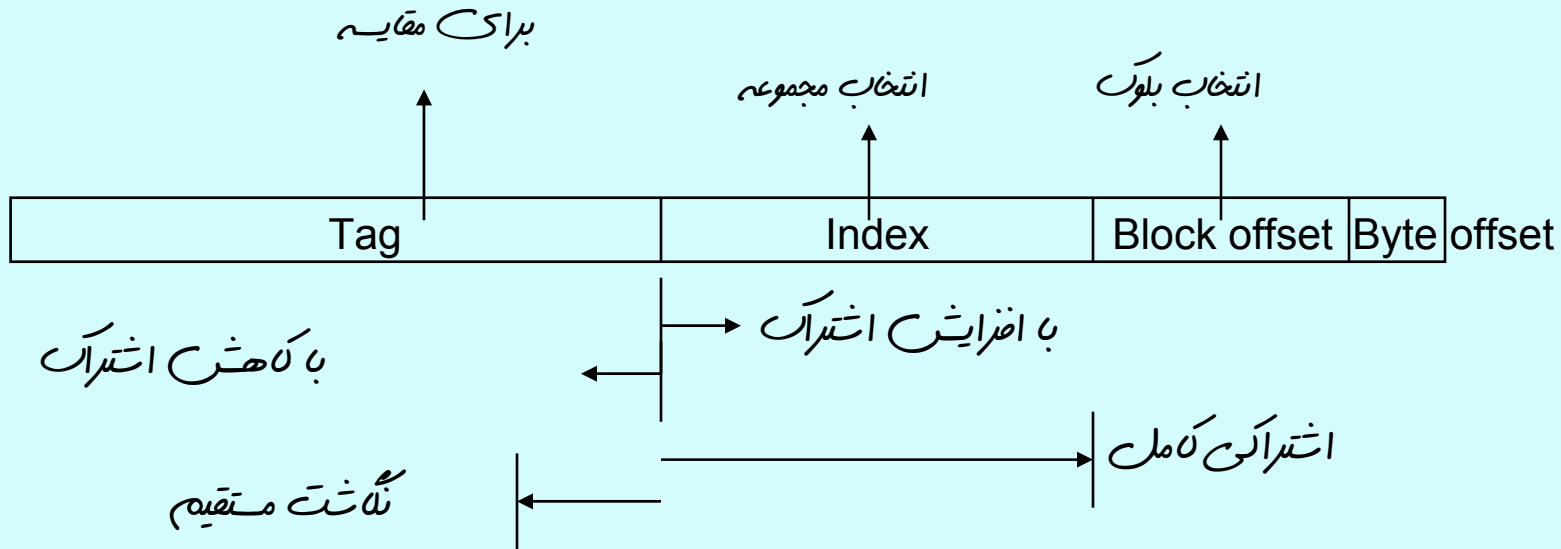
Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

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



طیف اشتراک (ادامه...)



مثال

- یک حافظه‌ی نهان چهار بلوکی داریم،
- هدف مقایسه‌ی miss-rate در حالات زیر است:
 - نگاشت مستقیم
 - اشتراکی دو بلوکی
 - 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]	



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؟

- 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:

سوال؟؟

1-way: 10.3%

2-way: 8.6%

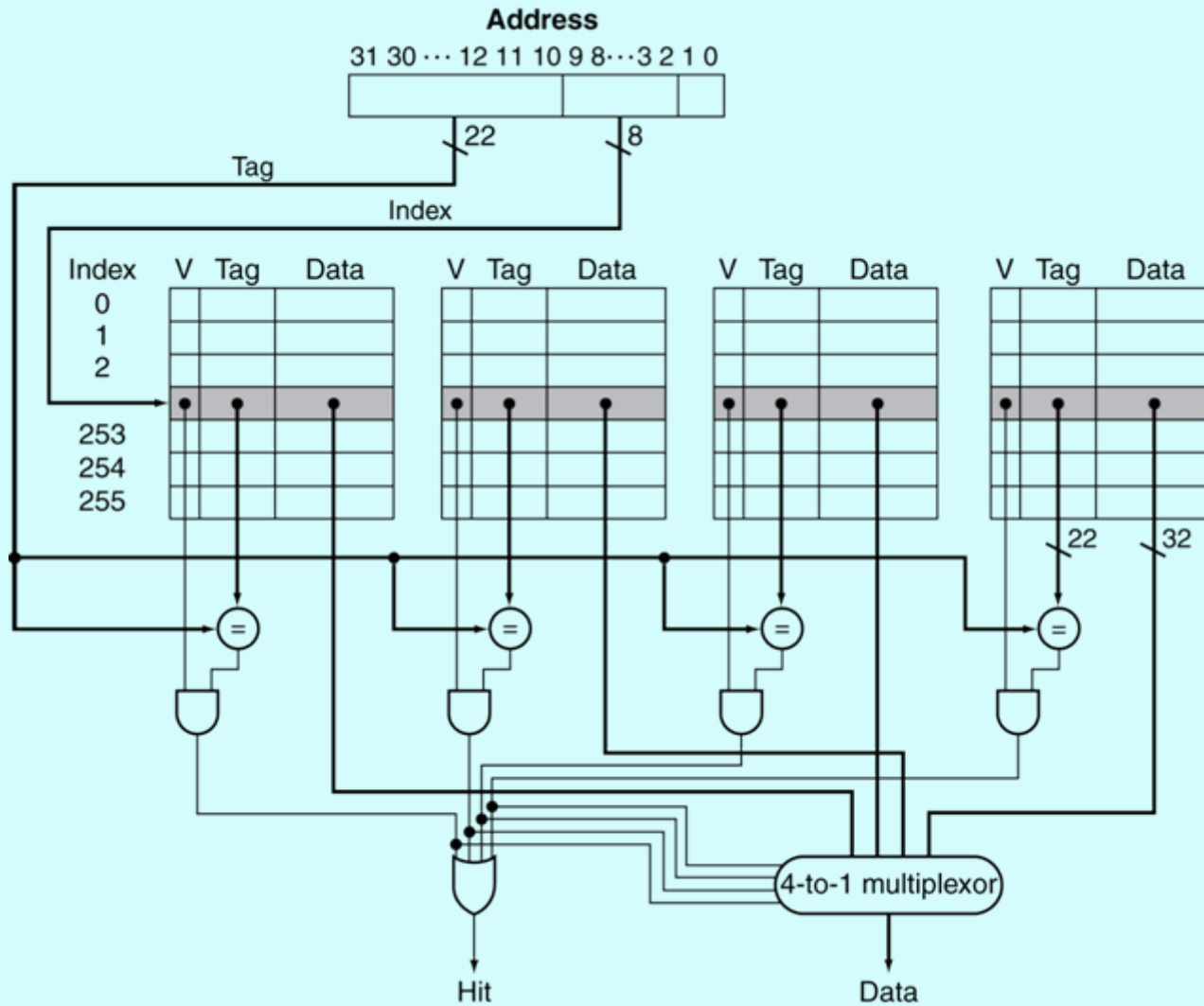
4-way: 8.3%

8-way: 8.1%

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



ساختار حافظه‌ی نهان با مجموعه‌های اشتراکی



در حافظه‌های نهان انجمنی، زمان جستجو اهمیت بسیاری دارد