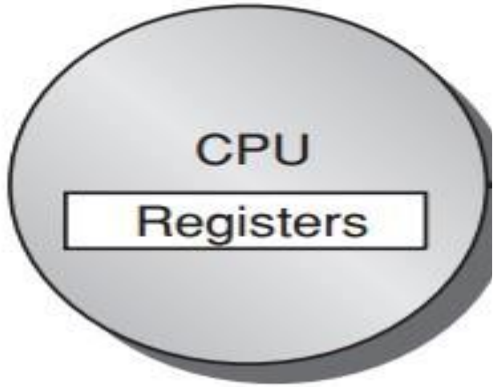
Memory Hierarchy هرمية الذاكرة
Cache Performance أداء الذاكرة المخبأية

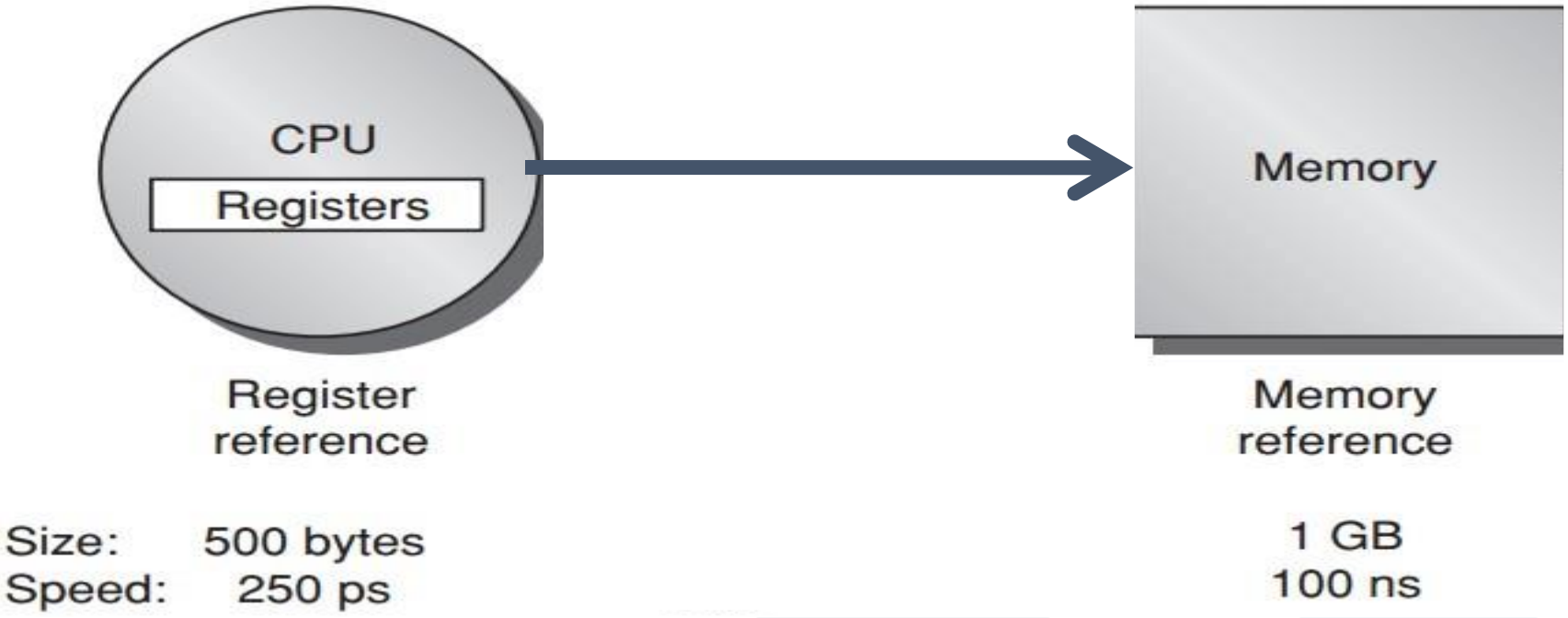
معالجة البيانات data processing
والتخزين المؤقت & temporary storage



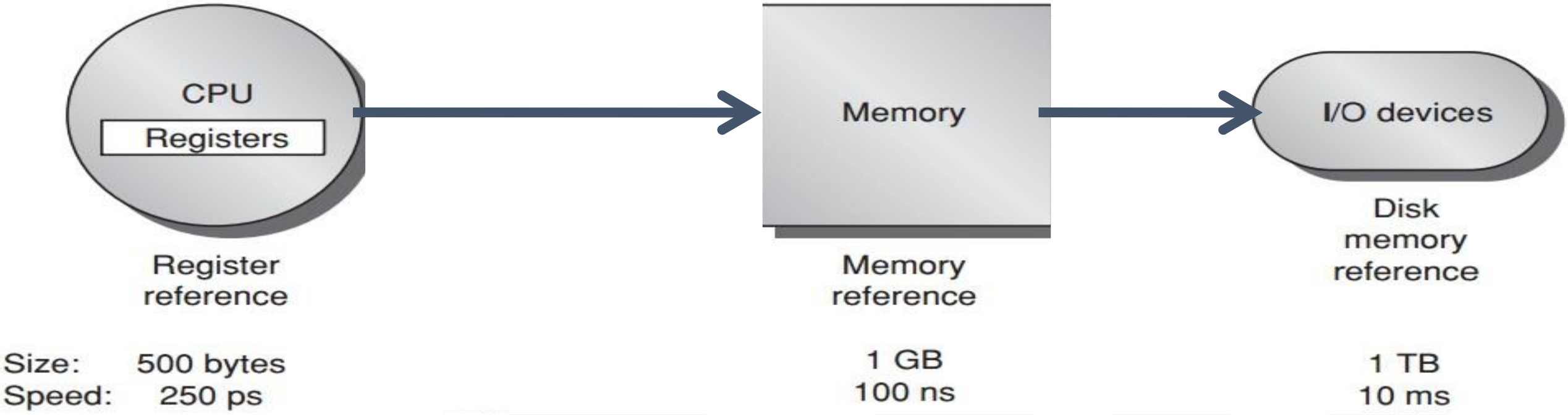
Register reference

Size: 500 bytes
Speed: 250 ps

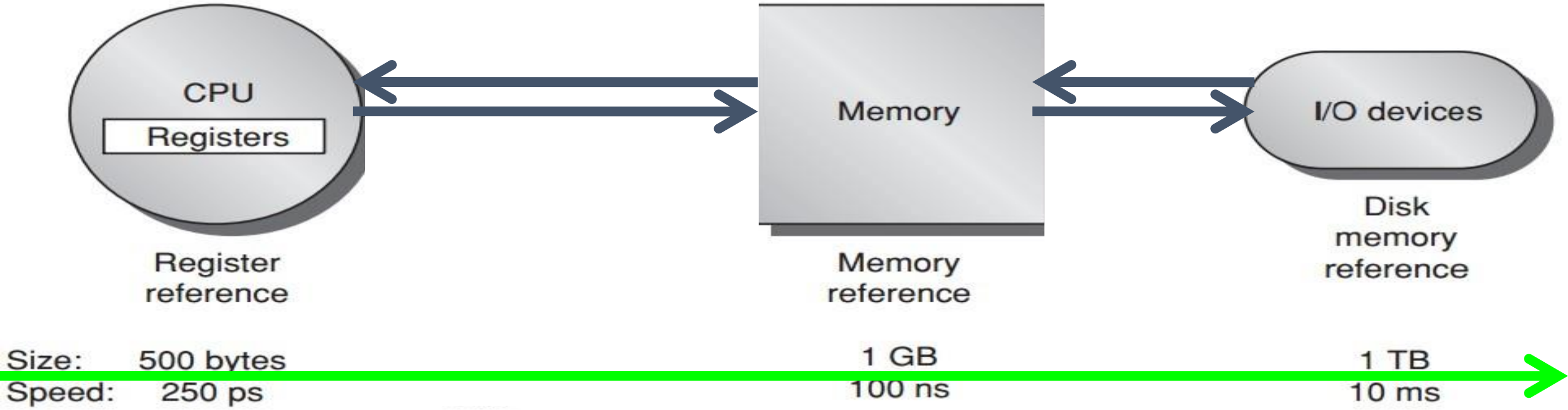
temporary storage التخزين المؤقت



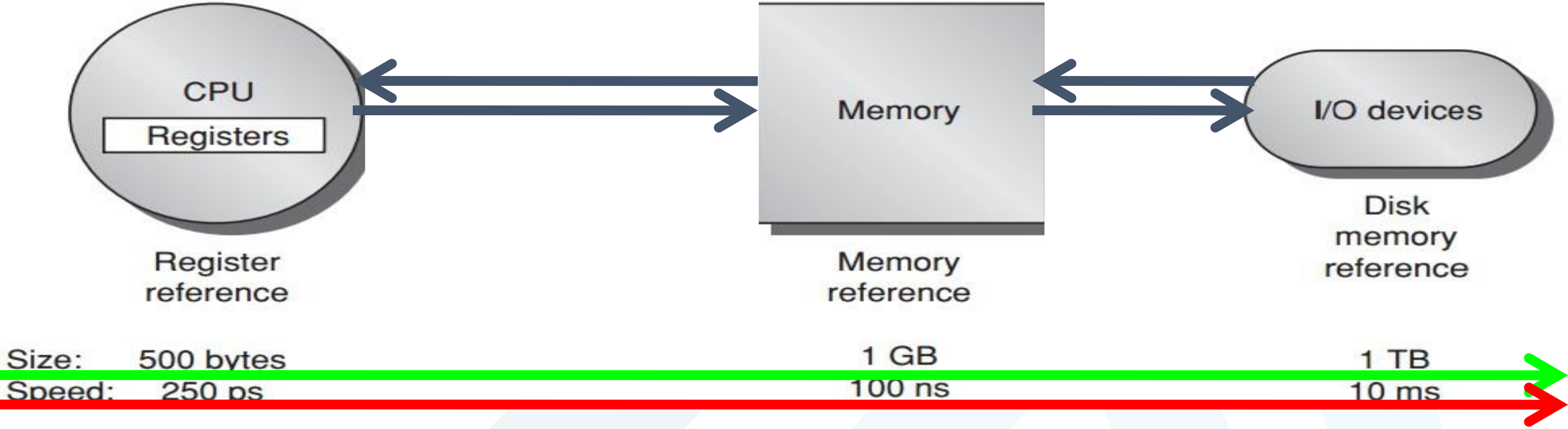
permanent storage التخزين الدائم



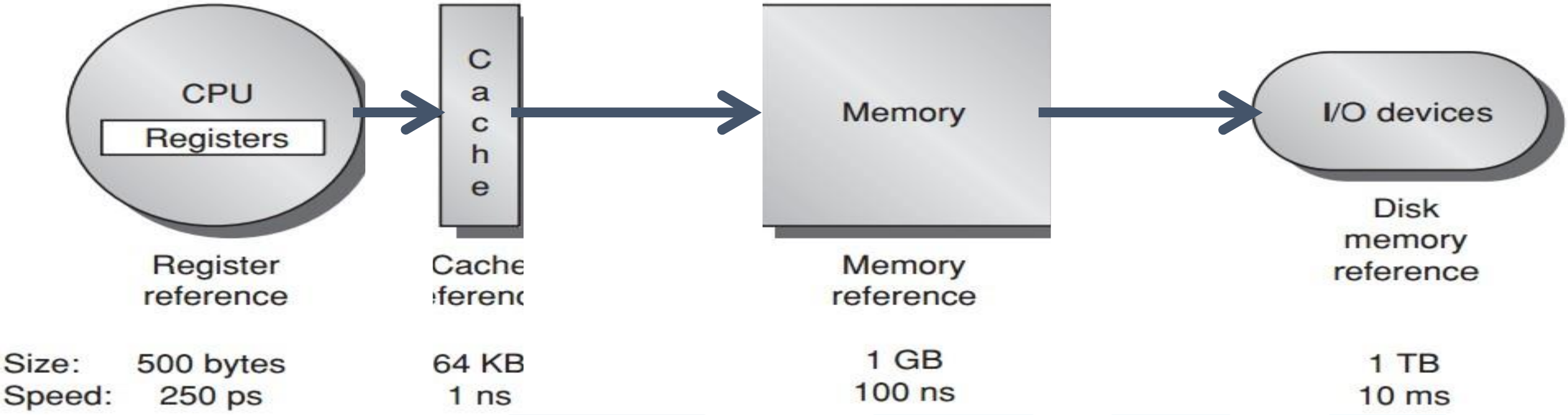
permanent storage التخزين الدائم



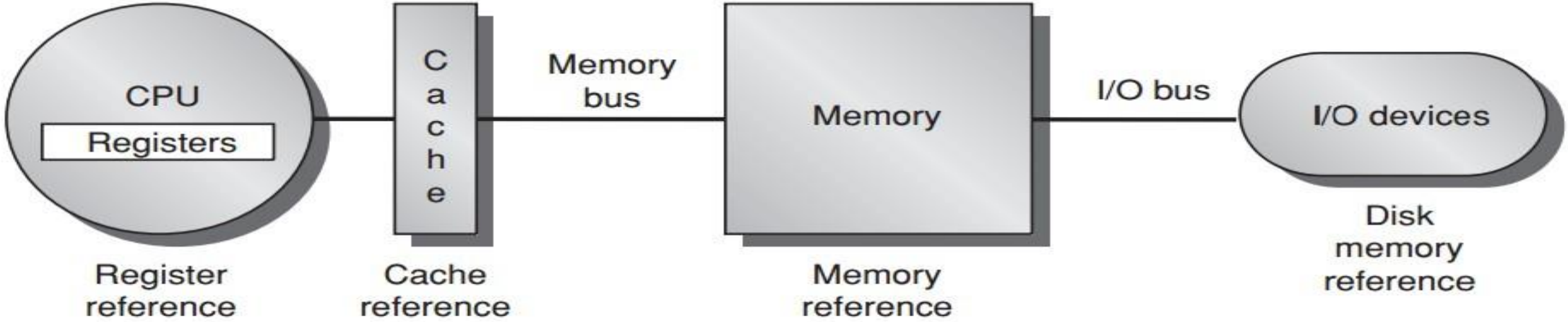
permanent storage التخزين الدائم



faster temporary storage التخزين المؤقت السريع



Memory Hierarchy هرمية الذاكرة



Size:	500 bytes	64 KB	1 GB	1 TB
Speed:	250 ps	1 ns	100 ns	10 ms

ماهي الذاكرة المخبأية

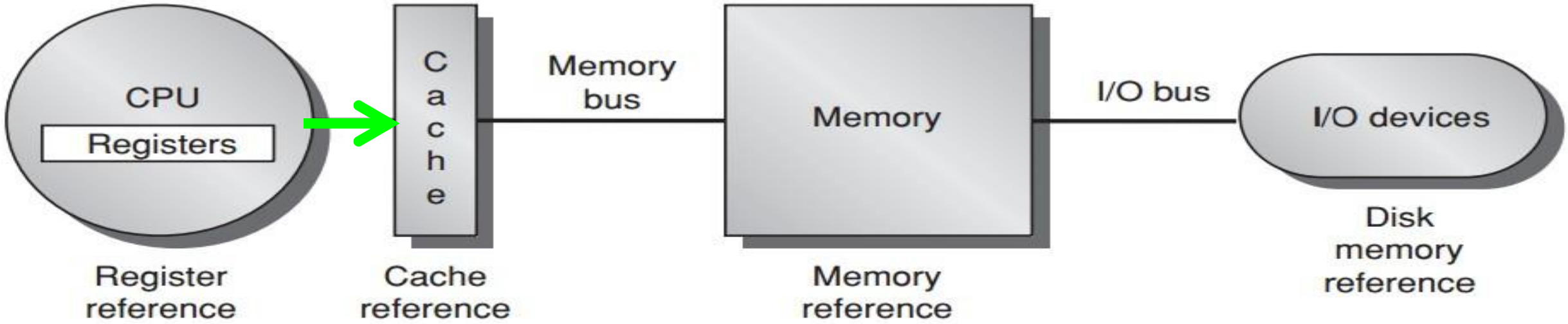
what’s cache?

نظرة عامة: Preview

- ماهي الذاكرة المخبئية؟ What’s cache?
- كيف ندخل ونخرج بيانات من الذاكرة المخبأية ؟ data in/out cache?
- كيف يمكننا جني فائدة أكبر من الذاكرة المخيأية؟

ماهي الكاش؟

الكاش Cache



- أعلى أو أول مستوي من مستويات هرمية الذاكرة يؤخذ بعسن الاعتبار عند مغادرة العناوين المعالج. The highest or first level of the memory hierarchy encountered once the addr leaves the processor
- تستخدم التخزين الوسيطي للبيانات كثيرة الاستخدام (بدلاً من الوصول إليها من الذاكرة الرئيسية)

إصابة/خطأ الوصول إلى الكاش Cache Hit/Miss

- عندما يتمكن/ لايتمكن المعالج من إيجاد البيانات المطلوبة في الكاش
- When the processor **can/cannot** find a requested data item in the cache

مجرى الكتلة/الخط Block/Line

- تضم تشكيلة ثابتة الحجم من البيانات كلمة البيانات المطلوبة، حيث يجري استحضارها من الذاكرة الرئيسية ووضعها في الكاش.

Cache Locality موضعية الكاش

- **Temporal locality موضعية مؤقتة**

need the requested word again soon تحتاج الكلمة المطلوبة فوراً

- **Spatial locality موضعية حيزية**

likely need other data in t,vhW غالباً ما تحتاج البيانات الأخرى في البلوك the block soon

خطأ الكاش Cache Miss

- يعتمد الزمن الذي يستغرقة خطأ الكاش على: Time required for cache miss depends on:

التأخير **Latency:** هو الزمن المستغرق لاسترجاع أول كلمة من البلوك. the time to : retrieve the first word of the block

عرض الحزمة **Bandwidth:** زمن استرجاع بقية كلمات البلوك the time to : retrieve the rest of this block

أهمية أداء الكاش

cache performance?

Cache Performance: Equations أداء الكاش: معادلة

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

Assumption:
Includes the time
to handle
a cache hit/miss

$$\begin{aligned}\text{Memory stall cycles} &= \text{Number of misses} \times \text{Miss penalty}\\ &= \text{IC} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss penalty}\\ &= \text{IC} \times \frac{\text{Memory accesses}}{\text{Instruction}} \times \text{Miss rate} \times \text{Miss penalty}\end{aligned}$$

$$\begin{aligned}\text{Memory stall clock cycles} &= \text{IC} \times \text{Reads per instruction} \times \text{Read miss rate} \times \text{Read miss penalty}\\ &\quad + \text{IC} \times \text{Writes per instruction} \times \text{Write miss rate} \times \text{Write miss penalty}\end{aligned}$$

Cache Miss Metrics قياس خطأ الكاش

- دورات تأخير الذاكرة **Memory stall cycles**

عدد الدورات التي يتأخر بها بالمعالج وهو بانتظار الوصول إلى الذاكرة. the number of cycles during processor is stalled waiting for a mem access

- نسبة الخطأ **Miss rate**

عدد مرات الخطأ مقسوماً على عدد مرات الوصول number of misses over number of accesses

- جزاء الخطأ **Miss penalty**

كلفة كل خطأ (عدد دورات الساعة الإضافية التي يتوجب انتظارها)

the cost per miss (number of extra clock cycles to wait)

Cache Performance: Example أداء الكاش: مثال

- **Example مثال**

حاسوب بعدد دورات لكل تعليمة CPI=1 عند إصابة الكاش a computer with CPI=1 when cache hit;

يتم تحميل وتخزين ٥٠% من التعليمات 50% instructions are loads and stores;

دورتي ساعة لكل وصول 2 cc per memory access;

نسبة الخطأ ٢% و نسبة الجزاء ٢٥دورة ساعة 2% miss rate, 25 cc miss penalty;

ماهي نسبة تحسين السرعة في حال جميع محاولات الوصول كانت ناجحة

Q: how much faster would the computer be if all instructions were cache hits?

Cache Performance: Example أداء الكاش: مثال

- **Answer الجواب**

جميع محاولات الوصول ناجحة: always hit:

زمن تنفيذ الـCPU CPU execution time

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

$$= (\text{IC} \times \text{CPI} + 0) \times \text{Clock cycle}$$

$$= \text{IC} \times 1.0 \times \text{Clock cycle}$$

Cache Performance: Example أداء الكاش: مثال

- **Answer الجواب**

بوجود خطأ with misses:

Memory stall cycles دورات تأخير الذاكرة

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

$$= \text{IC} \times (1 + 0.5) \times 0.02 \times 25$$

$$= \text{IC} \times 0.75$$

CPU execution time$_{\text{cache}}$

$$= (\text{IC} \times 1.0 + \text{IC} \times 0.75) \times \text{Clock cycle}$$

$$= 1.75 \times \text{IC} \times \text{Clock cycle}$$

Cache Performance: Example أداء الكاش: مثال

- **Answer الجواب**

$$\frac{\text{CPU execution time}_{\text{cache}}}{\text{CPU execution time}} = \frac{1.75 \times \text{IC} \times \text{Clock cycle}}{1.0 \times \text{IC} \times \text{Clock cycle}}$$

$$= 1.75$$

Hit or Miss: خطأ أم إصابة
Where to find a block? أين نجد البلوك ؟

Block Placement توضع البلوك

- **التخطيط المباشر Direct Mapped**

مكان واحد فقط only one place

$$(\text{Block address}) \text{ MOD } (\text{Number of blocks in cache})$$

- **الترابط الكلي Fully Associative**

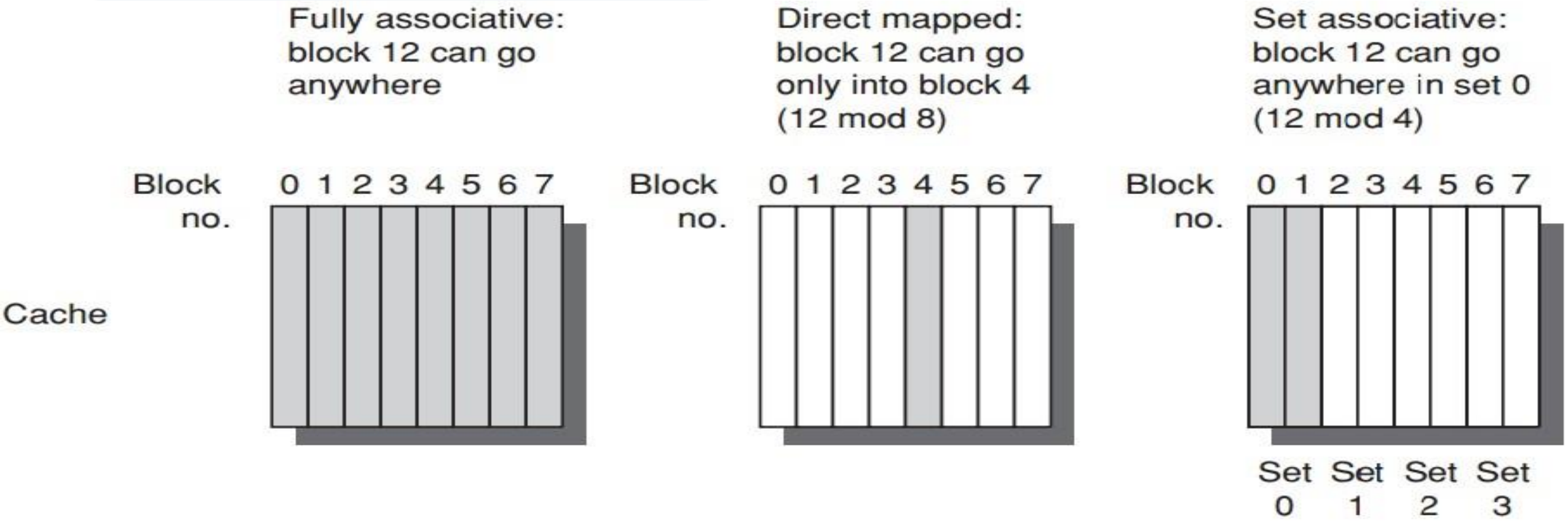
في أي مكان anywhere

- **ترابط مجموعة Set Associative**

أي مكان ضمن المجموعة الواحدة anywhere within only one set

$$(\text{Block address}) \text{ MOD } (\text{Number of sets in cache})$$

Block Placement توضع البلوك



Block Placement: Generalized تعميم توضيع البلوك

- **ترابط بـn طريق :n-way set associative**

يكون هنالك n بلوك في كل مجموعة n blocks in a set

- التخطيط المباشر هو ترابط بطريق وحيد Direct mapped = one-way set associative

أي هنالك بلوك واحد في كل مجموعة i.e., one block in a set

- ترابط كلي يعني m طريق ترابط مجموعة Fully associative = m-way set associative

أي جميع الكاش كمجموعة واحدة على الـn بلوك i.e., entire cache as one set with m blocks

Block Identification تعريف البلوك

Block address		Block offset
Tag	Index	

- **عنوان البلوك: علام + دليل Block address: tag + index**

الدليل يحدد المجموعة Index: select the set

العلام هو خانة التوثيق + عنوان البلوك Tag: = valid bit + block address

يجري اختبار جميع البلوكات في المجموعة check all blocks in the set

- إزاحة البلوك هي عنوان البيانات المطلوبة ضمن البلوك Block offset: the address of the desired data within the block

- *الكاش كاملة التشاركية لاتعرف حقل دليل Fully associative caches have no index field*

Block Replacement استبدال البلوك

عند حدوث خطأ كاش يتوجب تحميل البيانات إلى بلوكك الكاش، والسؤال ي البلوكات علينا استبداله؟
upon cache miss, to load the data to a cache block, which block to replace?

- **توضيع البلوك في التخطيط المباشر Direct-mapped placement**

يمكننا استبدال بلوك وحيد فقط only one block can be replaced

Block Replacement استبدال البلوك

في حال تشاركية المجموعة والتشاركية الكلية Fully/set associative

- **عشوائي Random**

لسهولة البناء والتنفيذ simple to build

- **الأقل استخداماً في وقت الاستبدال LRU: Least Recently Used**

يستبدل البلوك الذي لم يستخدم لأطول فترة. the block that has been unused for the longest time;

تستخدم الموضعية الزمنية المؤقتة use temporal locality;

معقد / مرتفع الكلفة complicated/expensive;

- **أول من يدخل أو من يستبدل FIFO: first in, first out**

استراتيجية الكتابة Write Strategy

- **الكتابة العابرة Write-through**

تكتب البيانات في بلوك الكاش وفي نفس الوقت في بلوك الذاكرة ذات المستوي التالي

info is written to both the block in the cache and to the block in the lower-level memory

- **الكتابة الراجعة Write-back**

تكتب البيانات فقط في بلوك الكاش؛ وتكتب في الذاكرة الرئيسية فقط عند استبدال البلوك المعدل

info is written only to the block in the cache;

to the main memory only when the modified cache block is replaced;

Write Strategy استراتيجية الكتابة

خيارات الكتابة عند خطأ القراءة Options on a write miss

- **توزيع الكتابة Write allocate**

يجري الشروع بالكتابة عند خطأ الكتابة the block is allocated on a write miss

- **لا كتابة No-write allocate**

لا يؤثر خطأ الكتابة على الكاش write miss not affect the cache;

يجري تعديل البلوك في الذاكرة the block is modified in memory;

حتى يحاول البرنامج قراءة البلوك *until the program tries to read the block;*

Write Strategy: Example استراتيجية الكتابة : مثال

Example Assume a fully associative write-back cache with many cache entries that starts empty. Below is a sequence of five memory operations (the address is in square brackets):

```
Write Mem[100];
WriteMem[100];
Read Mem[200];
WriteMem[200];
WriteMem[100].
```

What are the number of hits and misses when using no-write allocate versus write allocate?

Write Strategy: Example

Example Assume a fully associative write-back cache with many cache entries that starts empty. Below is a sequence of five memory operations (the address is in square brackets):

```
Write Mem[100];   M
WriteMem[100];    M
Read Mem[200];    M
WriteMem[200];    H
WriteMem[100].    M
```

What are the number of hits and misses when using no-write allocate versus write allocate?

- No-write allocate: 4 misses + 1 hit

  cache not affected- address 100 not in the cache;

  read [200] miss, block replaced, then write [200] hits;

Write Strategy: Example

Example Assume a fully associative write-back cache with many cache entries that starts empty. Below is a sequence of five memory operations (the address is in square brackets):

```
Write Mem[100];   M
WriteMem[100];    H
Read Mem[200];    M
WriteMem[200];    H
WriteMem[100].    H
```

What are the number of hits and misses when using no-write allocate versus write allocate?

- Write allocate: 2 misses + 3 hits

Hit or Miss: كم تستغرق الإصابة والخطأ
How long will it take?

متوسط زمن الوصول إلى الذاكرة Avg Mem Access Time

- متوسط زمن الوصول إلى الذاكرة هو Average memory access time

=Hit time + Miss rate x Miss penalty

Avg Mem Access Time متوسط الوصول إلى الذاكرة

- **Example مثال**

16KB instr cache + 16KB data cache; **١٦ كيلوبايت كاش تعليمات و١٦ كيلوبايت كاش بيانات**

32KB unified cache; ٣٢ كيلوبايت كاش موحدة

36% data transfer instructions; ٣٦% من التعليمات هي تعليمات نقل بيانات

(load/store takes 1 extra cc on unified cache)

1 CC hit; 200 CC miss penalty;

Q1: split cache or unified cache has lower miss rate?

Q2: average memory access time?

Size	Instruction cache	Data cache	Unified cache
8 KB	8.16	44.0	63.0
16 KB	3.82	40.9	51.0
32 KB	1.36	38.4	43.3
64 KB	0.61	36.9	39.4
128 KB	0.30	35.3	36.2
256 KB	0.02	32.6	32.9

Example: miss per 1000 instructions

مثال: الأخطاء لألف تعليمة

Avg Mem Access Time متوسط زمن وصول الذاكرة

- **Q1**

$$\text{Miss rate} = \frac{\dfrac{\text{Misses}}{\text{1000 Instructions}} / 1000}{\dfrac{\text{Memory accesses}}{\text{Instruction}}}$$

$$\text{Miss rate}_{\text{16 KB instruction}} = \frac{3.82/1000}{1.00} = 0.004$$

$$\text{Miss rate}_{\text{16 KB data}} = \frac{40.9/1000}{0.36} = 0.114$$

Overall miss rate $(74\% \times 0.004) + (26\% \times 0.114) = 0.0326$

$$\text{Miss rate}_{\text{32 KB unified}} = \frac{43.3/1000}{1.00 + 0.36} = 0.0318$$

متوسط زمن الوصول إلى الذاكرة
Avg Mem Access Time

- **Q2**

$$\text{Average memory access time}$$
$$= \%\ \text{instructions} \times (\text{Hit time} + \text{Instruction miss rate} \times \text{Miss penalty})$$
$$+\ \%\ \text{data} \times (\text{Hit time} + \text{Data miss rate} \times \text{Miss penalty})$$

$$\text{Average memory access time}_{\text{split}}$$
$$= 74\% \times (1 + 0.004 \times 200) + 26\% \times (1 + 0.114 \times 200)$$
$$= (74\% \times 1.80) + (26\% \times 23.80) = 1.332 + 6.188 = 7.52$$

$$\text{Average memory access time}_{\text{unified}}$$
$$= 74\% \times (1 + 0.0318 \times 200) + 26\% \times (1 + 1 + 0.0318 \times 200)$$
$$= (74\% \times 7.36) + (26\% \times 8.36) = 5.446 + 2.174 = 7.62$$

Cache vs Processor الكاش مقابل المعالج

- أداء المعالج Processor Performance

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

- يعزى متوسط زمن الوصول المنخفض إلى سرعة تنفيذ الـCPU للعمليات
- Lower avg memory access time may correspond to higher CPU time

Out-of-Order Execution

- in out-of-order execution, stalls happen to only instructions that depend on incomplete result;

  other instructions can continue;

  so less avg miss penalty

$$\frac{\text{Memory stall cycles}}{\text{Instruction}} = \frac{\text{Misses}}{\text{Instruction}} \times (\text{Total miss latency} - \text{Overlapped miss latency})$$

أمثلة أداء الكاش

How to optimize cache performance?

Average Memory Access Time =
Hit Time + Miss Rate x Miss Penalty

Average Memory Access Time =
Hit Time + Miss Rate x Miss Penalty

=Average Memory Access Time = متوسط معدل زمن الوصول

Hit Time + Miss Rate x Miss Penalty زمن الإصابة +نسبة الخطأ × عقوبة الخطأ

Average Memory Access Time =

Hit Time + Miss Rate x Miss Penalty

Larger block size; حجم بلوك أكبر

Larger cache size; حجم كاش أكبر

Higher associativity; تشاركية أعلى

Reducing Miss Rate تقليل نسبة الخطأ

٣ طبقات لـنسبة الخطأ / أسباب جذرية 3 categories of miss rates / root causes

- **Compulsory: إجبارية**

الإقلاع البارد/ فقدان المرجعية البدائية cold-start/first-reference misses;

- **Capacity السعة**

محدودية حجم الكاش cache size limit;

التخلي عن البلوك ومن ثم استعادته blocks discarded and later retrieved;

- **Conflict**

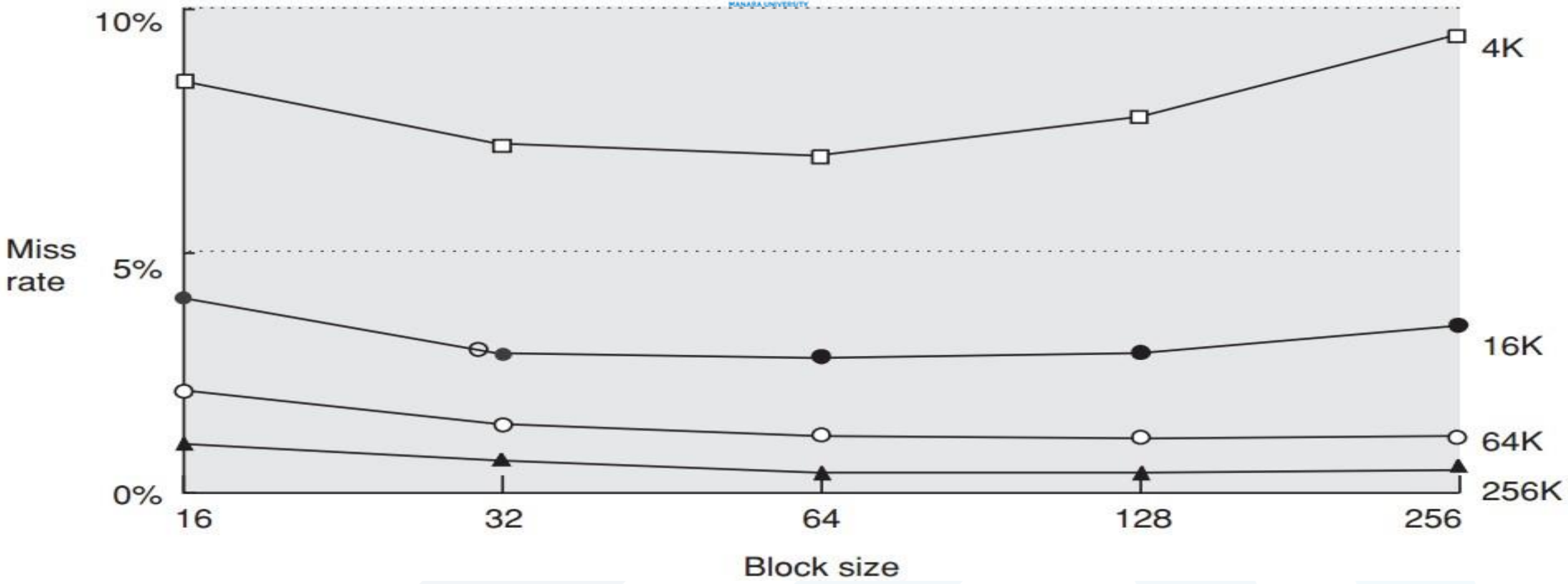
ضياعات التصادم والتنافس: التشاركية collision misses: associativity

التخلي عن البلوك ومن ثم استعادته ضمن مجموعة a block discarded and later retrieved in a set;

Opt #1: Larger Block Size حجم بلوك أكبر

- **Reduce** compulsory misses **ينقص الضياعات الإجبارية**
- رفع المحلية الحيزية Leverage spatial locality

- **Increase** conflict/capacity misses **زيادة ضياعات التنافس / السعة**
- بلوكات أقل في الكاش Fewer block in the cache


10%
5%
0%
Miss rate
4K
16K
64K
256K
16
32
64
128
256
Block size

- **Example مثال**

Block size	Cache size 4K	16K	64K	256K
16	8.57%	3.94%	2.04%	1.09%
32	7.24%	2.87%	1.35%	0.70%
64	7.00%	2.64%	1.06%	0.51%
128	7.78%	2.77%	1.02%	0.49%
256	9.51%	3.29%	1.15%	0.49%

ليكن لدينا نسب الخطأ المدونة أعلاه given the above miss rates;

بفرض أن للذاكرة استهلاك فائض مقداره assume memory takes 80 CC overhead, 80 CC

وتحتاج إلى 2CC لتنقل ١٦ بايت delivers 16 bytes in 2 CC;

في أي حجم للبلوك يكون معدل زمن الوصول إلى الذاكرة أقل مايمكن لكل حجم للكاش **Q:** which block size has the smallest average memory access time for each cache size?

- **الجواب Answer**

avg mem access time

=hit time + miss rate x miss penalty

بفرض أن زمن الإصابه هو 1CC *assume 1-CC hit time

من أجل بلوك بحجم ٢٥٦ بايت في ذاكرة ٢٥٦ كيلوبايت for a 256-byte block in a 256 KB cache:

متوسط زمن الوصول avg mem access time

= 1 + 0.49% x (80 + 2x256/16) = 1.5 cc

- والجواب Answer

متوسط زمن الوصول إلى الذاكرة average memory access time

		Cache size			
Block size	Miss penalty	4K	16K	64K	256K
16	82	8.027	4.231	2.673	1.894
32	84	**7.082**	3.411	2.134	1.588
64	88	7.160	**3.323**	**1.933**	**1.449**
128	96	8.469	3.659	1.979	1.470
256	112	11.651	4.685	2.288	1.549

Opt #2: Larger Cache كاش أكبر

- **Reduce** capacity misses **تقلل سعة أخطاء الوصول**
- **Increase** hit time, cost, and **يزداد زمن الإصابة والكلفة واستهلاك الإستطاعة** power

Opt #3: Higher Associativity تشاركية أعلى

- **Reduce** conflict misses **تقلل أخطاء التنافس**
- **Increase** hit time **تزيد زمن الإصابة**

- **Example مثال**

بفرض أن التشاركية الأكبر تؤدي إلى زمن دورة ساعة أكبر assume higher associativity -> higher clock cycle time:

$$\text{Clock cycle time}_{2\text{-way}} = 1.36 \times \text{Clock cycle time}_{1\text{-way}}$$
$$\text{Clock cycle time}_{4\text{-way}} = 1.44 \times \text{Clock cycle time}_{1\text{-way}}$$
$$\text{Clock cycle time}_{8\text{-way}} = 1.52 \times \text{Clock cycle time}_{1\text{-way}}$$

بفرض أن زمن الإصابة هو 1CC وجزاء الخطأ 25CC وأن نسبة الخطأ كما في الجدول التالي: assume 1-cc hit time, 25-cc miss penalty, and miss rates in the following table;

- **Miss rates**

Cache size (KB)	Degree associative	Total miss rate	Miss rate components (relative percent) (sum = 100% of total miss rate)					
			Compulsory		Capacity		Conflict	
4	1-way	0.098	0.0001	0.1%	0.070	72%	0.027	28%
4	2-way	0.076	0.0001	0.1%	0.070	93%	0.005	7%
4	4-way	0.071	0.0001	0.1%	0.070	99%	0.001	1%
4	8-way	0.071	0.0001	0.1%	0.070	100%	0.000	0%
8	1-way	0.068	0.0001	0.1%	0.044	65%	0.024	35%
8	2-way	0.049	0.0001	0.1%	0.044	90%	0.005	10%
8	4-way	0.044	0.0001	0.1%	0.044	99%	0.000	1%
8	8-way	0.044	0.0001	0.1%	0.044	100%	0.000	0%
16	1-way	0.049	0.0001	0.1%	0.040	82%	0.009	17%
16	2-way	0.041	0.0001	0.2%	0.040	98%	0.001	2%
16	4-way	0.041	0.0001	0.2%	0.040	99%	0.000	0%
16	8-way	0.041	0.0001	0.2%	0.040	100%	0.000	0%
32	1-way	0.042	0.0001	0.2%	0.037	89%	0.005	11%
32	2-way	0.038	0.0001	0.2%	0.037	99%	0.000	0%
32	4-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
32	8-way	0.037	0.0001	0.2%	0.037	100%	0.000	0%
64	1-way	0.037	0.0001	0.2%	0.028	77%	0.008	23%
64	2-way	0.031	0.0001	0.2%	0.028	91%	0.003	9%
64	4-way	0.030	0.0001	0.2%	0.028	95%	0.001	4%
64	8-way	0.029	0.0001	0.2%	0.028	97%	0.001	2%
128	1-way	0.021	0.0001	0.3%	0.019	91%	0.002	8%
128	2-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	4-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
128	8-way	0.019	0.0001	0.3%	0.019	100%	0.000	0%
256	1-way	0.013	0.0001	0.5%	0.012	94%	0.001	6%
256	2-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	4-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
256	8-way	0.012	0.0001	0.5%	0.012	99%	0.000	0%
512	1-way	0.008	0.0001	0.8%	0.005	66%	0.003	33%
512	2-way	0.007	0.0001	0.9%	0.005	71%	0.002	28%
512	4-way	0.006	0.0001	1.1%	0.005	91%	0.000	8%
512	8-way	0.006	0.0001	1.1%	0.005	95%	0.000	4%

جامعة المنارة
https://manara.edu.sy/

- **Question: سؤال**

في حجم للكاش for which cache sizes

تكون المقولات أدناه صحيحة are each of the statements true?

$$\text{Average memory access time}_{8\text{-way}} < \text{Average memory access time}_{4\text{-way}}$$
$$\text{Average memory access time}_{4\text{-way}} < \text{Average memory access time}_{2\text{-way}}$$
$$\text{Average memory access time}_{2\text{-way}} < \text{Average memory access time}_{1\text{-way}}$$

- **الجواب Answer**

من أجل كاش ٢٥٦ كيلوبايت بتشاركية ثمانية الطرق for a 512 KB, 8-way set associative cache:

avg mem access time

=hit time + miss rate x miss penalty

=1.52x1 + 0.006 x 25

=1.66

- الجواب Answer

معدل زمن الوصول للذاكرة average memory access time

	Associativity			
Cache size (KB)	One-way	Two-way	Four-way	Eight-way
4	3.44	3.25	3.22	**3.28**
8	2.69	2.58	2.55	**2.62**
16	2.23	**2.40**	**2.46**	**2.53**
32	2.06	**2.30**	**2.37**	**2.45**
64	1.92	**2.14**	**2.18**	**2.25**
128	1.52	**1.84**	**1.92**	**2.00**
256	1.32	**1.66**	**1.74**	**1.82**
512	1.20	**1.55**	**1.59**	**1.66**

Average Memory Access Time =
Hit Time + Miss Rate x Miss Penalty

Multilevel caches;

Reads > Writes;

Opt #4: Multilevel Cache الكاش متعددة المستويات

- **Reduce** miss penalty **تقلل جزاء الخطأ**

- **Motivation السبب**

faster/smaller cache to keep pace with the speed of processors? **كاش أسرع واصغر تتناسب مع سرعة المعالج أكثر**

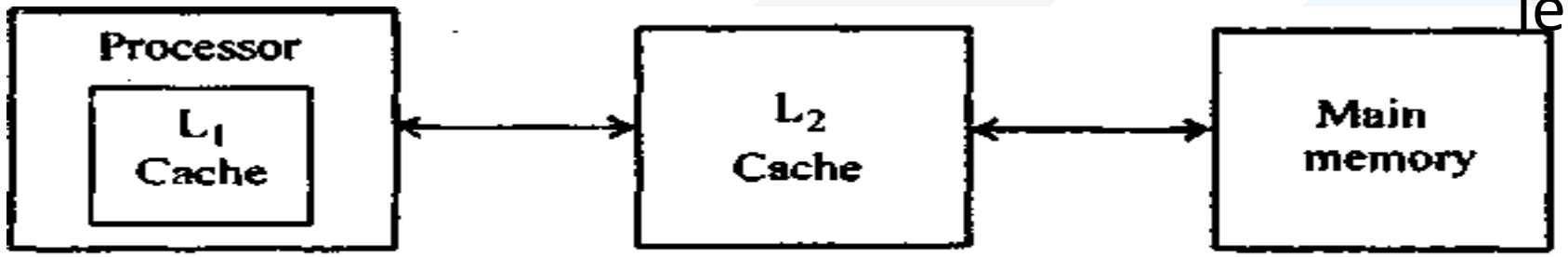
larger cache to overcome the widening gap between processor and main mem? كاش أكبر تردم الفجوة بين المعالج والذاكرة الرئيسية

Opt #4: Multilevel Cache الكاش متعددة المستويات

- **كاش بمستويين Two-level cache**

تضيف سوية أخرى للكاش بين الكاش الأولى والذاكرة الرئيسية Add another level of cache between the original cache and memory

- **L1 صغيرة كفاية لتوافق زمن CC مع المعالج السريع** **L1**: small enough to match the clock cycle time of the fast processor;
- **L2 كبيرة كفاية لتقلل الوصول إلى الذاكرة الرئيسية (لخفض جزاء الخطأ)** **L2**: large enough to capture many accesses that would go to main memory, lessening miss penalty



L_1 *and* L_2 *cache placement*

Opt #4: Multilevel Cache الكاش متعددة المستويات

- **Average memory access time** متوسط زمن الوصول للذاكرة

$=\text{Hit time}_{L1} + \text{Miss rate}_{L1} \times \text{Miss penalty}_{L1}$

$=\text{Hit time}_{L1} + \text{Miss rate}_{L1}$

$\times(\text{Hit time}_{L2}+\text{Miss rate}_{L2}\times\text{Miss penalty}_{L2})$

- **Average mem stalls per instruction** متوسط زمن الوصول للذاكرة لكل تعليمة

$=\text{Misses per instruction}_{L1} \times \text{Hit time}_{L2}$

$+ \text{Misses per instr}_{L2} \times \text{Miss penalty}_{L2}$

Opt #4: Multilevel Cache الكاش متعددة المستويات

- **نسبة الخطأ الموضعية Local miss rate**

عدد الأخطاء في الكاش مقسومة على عدد مرات الوصول إلى هذه الكاش

the number of misses in a cache
divided by the total number of mem accesses to this cache;
Miss rate$_{L1}$, *Miss rate*$_{L2}$

- **نسبة الخطأ العامة Global miss rate**

عدد الأخطاء في الكاش مقسومة على عدد مرات الوصول إلى الذاكرة من قبل المعالج

the number of misses in the cache
divided by the number of mem accesses generated by the processor;
Miss rate$_{L1}$, *Miss rate*$_{L1}$ *x Miss rate*$_{L2}$

- **مثال Example:**
- **١٠٠٠ طلب للذاكرة نتج عنها ٤٠ خطأ للـL1 و ٢٠ خطأ للـL2.**

1000 mem references -> 40 misses in L1 and 20 misses in L2;

جزاء الخطأ في L2 هو 200CC miss penalty from L2 is 200 cc;

زمن الإصابة في L2 هو 10CC hit time of L2 is 10 cc;

زمن الإصابة في L1 هو 1CC hit time of L1 is 1 cc;

معدل 1.5 طلب كل تعليمة 1.5 mem references per instruction;

السؤال ١. نسب خطأ مختلفة **Q: 1.** various miss rates?

٢. متوسط زمن الوصول **2.** avg mem access time?

٣. متوسط زمن الانتظار مقاساً بدورة ساعة لكل تعليمة **3.** avg stall cycles per instruction?

- **Answer**

1. various miss rates? نسب خطأ مختلفة

L1: local = global أي L1 موضعية وعامة

40/1000 = 4%

L2:

local: 20/40 = 50%

global: 20/1000 = 2%

- **Answer الجواب**

2. avg mem access time? متوسط زمن الوصول .

average memory access time

$=\text{Hit time}_{L1} + \text{Miss rate}_{L1}$

$x(\text{Hit time}_{L2}+\text{Miss rate}_{L2} x \text{Miss penalty}_{L2})$

$=1 + 4\% \times (10 + 50\% \times 200)$

$=5.4$

- **Answer الجواب**

3. avg stall cycles per instruction?

٣. متوسط زمن الانتظار (الإعاقة) مقاساً بدورة ساعة لكل تعليمة

average stall cycles per instruction

متوسط زمن الانتظار (الإعاقة) لكل تعليمة

$=\text{Misses per instruction}_{L1} \times \text{Hit time}_{L2}$

$+ \text{Misses per instr}_{L2} \times \text{Miss penalty}_{L2}$

$=(1.5\times40/1000)\times10+(1.5\times20/1000)\times200$

$=6.6$

إعطاء الأولوية للكتابة فوق إخطاء القراءة
Opt #5: Prioritize read misses overwrites

- **تقلل جزاء الخطأ** **Reduce** miss penalty

- بدلاً من انتظار خطأ القراءة حتى يفرغ عازل الكتابة instead of simply stall read miss until write buffer empties,

اختبر محتوى عازل الكتابة check the contents of write buffer,

اترك خطأ القراءة يستمر let the read miss continue

إذا لم يكن هنالك تناقض أو تنافس على عازل الكتابة وتوفرت ذاكرة النظام

if no conflicts with write buffer & memory system is available

إعطاء الأولوية للكتابة فوق إخطاء القراءة
Opt #5: Prioritize read misses overwrites

- **Why**

```
SW R3, 512(R0)    ;M[512] ← R3      (cache index 0)
LW R1, 1024(R0)   ;R1 ← M[1024]     (cache index 0)
LW R2, 512(R0)    ;R2 ← M[512]      (cache index 0)
```

افترض من أجل تتالي الشيفرة أعلاه أن الكاش تخطيط مباشر وكتابة عابرة وأنها تطابق ٥١٢ و١٠٢٤ لنفس البلوك

for the code sequence, assume a direct-mapped, **write-through** cache that maps 512 and 1024 to the same block;

لايجري اختبار عازل الكتابة رباعي الكلمة للتأكد من أخطاء القراءة

a four-word write buffer is **not** checked on a read miss.

R2.value ≡ R3.value ?

Average Memory Access Time =
Hit Time + Miss Rate x Miss Penalty

Avoid address translation

during indexing of the cache

تجنب ترجمة العناوين أثناء فهرسة الكاش

تجنب ترجمة التعليمة أثناء فهرسة الكاش

Opt #6: Avoid address translation during indexing cache

- **عنونة الكاش** **Cache** addressing

عنوان افتراضي – كاش افتراضية virtual address – *virtual cache*

عنوان فيزيائي – كاش فيزيائية physical address – *physical cache*

- **معالج/برنامج – عنوان افتراضي** **Processor/program** – virtual address
- معالج يعني ترجمة عنوان أي كاش Processor -> address translation -> Cache

كاش افتراضية أو فيزيائية virtual cache or physical cache?

تجنب ترجمة التعليمة أثناء فهرسة الكاش

Opt #6: Avoid address translation during indexing cache

- **فهرس افتراضي وعلام فيزيائي** **Virtually indexed, physically tagged**

إزاحة الصفحة لفهرسة الكاش page offset to index the cache;

عنوان فيزيائي لمطابقة العلام physical address for tag match;

- من أجل كاش التخطيط المباشر For direct-mapped cache,

لايجوز أن تكون أكبر من حجم الصفحة it cannot be bigger than the page size.

- *Reference: CPU Cache*

http://zh.wikipedia.org/wiki/CPU%E9%AB%98%E9%80%9F%E7%BC%93%E5%AD%98

جَامعة
المَنارة
MANARA UNIVERSITY