Chapter 4

Internal Memory الذاكرة الداخلية

Characteristics الخصائص

- الأداء Performance
  - النفاذ والدورة وزمن التراسل
    Access, Cycle, Transfer time
- النمط الفيزيائي Physical type
  - نصف ناقل أومغنطيسي أوضوئي
    , magnetic, optical
- الخصائص الفيزيائية
  - ماتطايرة أوقابلة للمحي , Volatile
    Erasable
- التنظيم
  - الترتيب الفيزيائي للخانات ضمن كلمات
    Physical arrangement of bits into words

- الموقع Location
  - ضمن الـCPU، داخلية، خارجية.
- السعة Capacity
  - حجم الكلمة وعدد الكلمات
- وحدة التراسل Unit of transfer
  - كلمة على الباص أوكتلة أوعنقود
    Word on bus, block, cluster
- طريقة النفاذ Access method
  - مباشر أوعشوائي أوتشاركي أو نسلسلي
    Direct, Random, Associative, Sequential

طرق النفاذ (١)
Access Methods (1)

- تسلسلي أوتتابعي Sequential
  - Start at the beginning and read through in order
  - يعتمد زمن النفاذ على موقع البيانات وعلى الموقع السابق.
  - مثل الشريط المغناطيسي tape
- مباشر Direct
  - لكل كتلة عنوان خاص.
  - النفاذ من خلال القفز إلى منطقة قريبة ومن ثم البحث تسلسلياً
  - زمن النفاذ يعتمد على الموقع والموقع السابق.
  - مثل القرص المغنطيسي disk.

طرق النفاذ (٢)
Access Methods (2)

- العشوائية Random
  - تحدد العناوين الإفرادية المواقع بدقة.
  - زمن النفاذ مستقل عن موقع النفاذ السابق.
  - مثل الـRAM.
- التشاركية Associative
  - تتوضع البيانات في موقع تحدده المقارنة مع محتوى جزء من القيمة المخزنة.
  - زمن النفاذ مستقل عن الموقع والنفاذ السابق.
  - مثل الذاكرة المخبأية cache.

هيكلية الذاكرة
Memory Hierarchy

- المسجلات Registers
  - مثل تلك التي في الـCPU.
- الذاكرة الرئيسية أو الداخلية Internal or Main memory
  - قد تضم مستوي أو أكثر من الـcache.
  - الـ"RAM"
- الذاكرة الخارجية External memory
  - التخزين الإضافي والاحتياطي Backing store

الأداء أو الإنجازية
Performance

- زمن النفاذ Access time
  - الزمن الفاصل بين تقديم العنوان والحصول على البيانات المطلوبة
- زمن دورة الذاكرة Memory Cycle time
  - الزمن الذي فد نحتاجه لإستعادة محتوى الذاكرة قبل النفاذ.التالي.
  - زمن دورة الذاكرة = النفاذ + الاستعادة

Cycle time is access + recovery

- معدل التراسل Transfer Rate
  - المعدل الذي يمكن أن يمكن أن تنتقل به البيانات.

الخصائص الفيزيائية
Physical Characteristics

- الاضمحلال Decay
- التطاير Volatility
- قابلية المحي Erasable
- استهلاك الطاقة Power consumption

الخط الأدنى
The Bottom Line

- How much? كم؟
  - الجواب: السعة Capacity
- How fast? مالسرعة؟
  - الجواب: الزمن كلفة Time is money
- How expensive? ما مدى ارتفاع الكلفة؟

- Tradeoffs among all of these: مقايضة بين العوامل التالية
  - أسرع =كلفة أكبر، أكبر = كلفة أقل (للخانة) ولكن أبطأ

Faster = More expensive, More = Less cost (per bit) but slower

  - الحل: هو هيكلة الذاكرة Solution : Memory Hierarchy

قائمة البناء الهرمي
Hierarchy List

- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape

- عند الانتقال من أعلى الهرم نزولاً
  - تتناقص كلفة الخانة cost per bit
  - تزجاج السعة Increasing capacity
  - يزيد زمن النفاذ Increasing access time
  - يتناقص تردد نفاذ المعالج إلى الذاكرة - الموضغية أو المرجعية frequency of access of the memory by the processor – locality of reference

إذا أردنا السرعة
So you want fast?

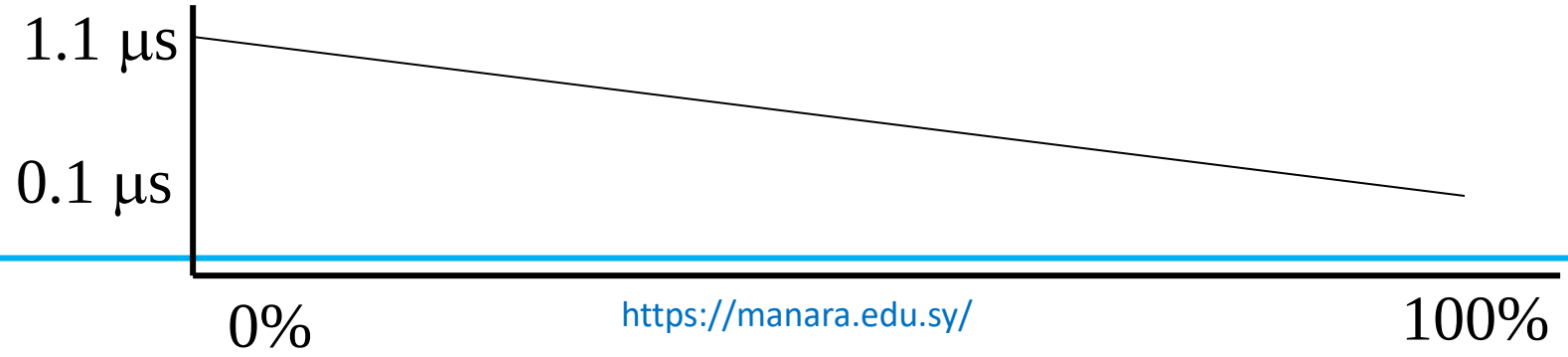
- يمكننا بناء حاسوب يستخدم فقط static RAM(مبين لاحقاً)
- سيحقق ذلك سرعة كبيرة جداً very fast
- لن يحتاج ذاكرة مخبأية need no cache
  - كيف نستطيع استخدام الذاكرة المخبأية.
- كلفة عالية جداً

الموضعية المرجعية
Locality of Reference

- الموضعية المؤقتة Temporal Locality
  - تصمم البرامج لجعل مرجعيتها نفس محموعة مواقع الذاكرة في الفترة الزمنية اللاحقة.
  - نتيجة ظهور الحلقات والتكرارات loops and iteration، يزيد استهلاك البرنامج للزمن اللازم لتنفيذ مقطع برمج ما.
- الموضعية الحيزية Spatial Locality
  - تصمم البرامج لجعل مرجعيتة مواقع الذاكرة المتجاورة حيزياً
  - ينتج عن ذلك الطريقة التي نجعل بها المواقع المتجاورة مرجعية مثل نسق النعليمات التي تشكل البرنامج.
- لايمكن دوماً الحفاظ على محلية المرجعية.

مثال الـCache
Cache Example

- لنأخذ على سبيل المثال ذاكرة مخبأية من المستوي الأول Level 1 cache تضم 1000 words بزمن نفاذ 0.1 µs. و المستوي الثاني Level 2 . هو ذاكرة بزمن نفاذ مقداره 1 µs.
- إذا تم إنجاز 95% من عمليات النفاذ في الـcache:
  - T=(0.95)*(0.1 µs) + (0.05)*(0.1+1 µs) = 0.15 µs
- إذا تم إنجاز 5% من عمليات النفاذ في الـcache:
  - T=(0.05)*(0.1 µs) + (0.95)*(0.1+1 µs) = 1.05 µs
- لذلك يجري السعي لزيادة احتمالية الإصابات cache hits



ذاكرة أنصاف النواقل
Semiconductor Memory

- RAM
  - تسمية خاطئة لأن جميع ذواكر أنصاف النواقل هي ذواكر نفاذ عشوائي random access
  - قراءة/كتابة Read/Write
  - متطايرة Volatile
  - تخزين مؤقت Temporary storage
  - نوعين ساكن أوديناميكي **Static** or **Dynamic**

Dynamic RAM

- تخزّن الخانات كشحن في المكثفات
- تعاني من تسرب الشحنة Charges leak
- تحتاج للإنعاش حتى عند عدم الاستخدام refreshing.
- بناء بسيط Simpler construction
- حجم صغير للخانة Smaller per bit
- أقل كلفة Less expensive
- تحتاج دارة إنعاش (refresh circuits (every few milliseconds
- أبطأ Slower
- تستخدم كذاكرة رئيسية Main memory

Static RAM

- تخزن الخانات كمفاتيح on/off باستخدام القلابات flip-flops.
- لايوجد تسرب في الشحنة No charges to leak
- لاحاجة للإنعاش No refreshing
- بنية أكثر تعقيداً More complex construction
- حجم أكبر للخانة Larger per bit
- أكثر كلفة More expensive
- لاتحتاج لدارات إنعاش Does not need refresh circuits
- أسرع Faster
- مخبأية Cache

ذواكر القراءة فقط
Read Only Memory (ROM)

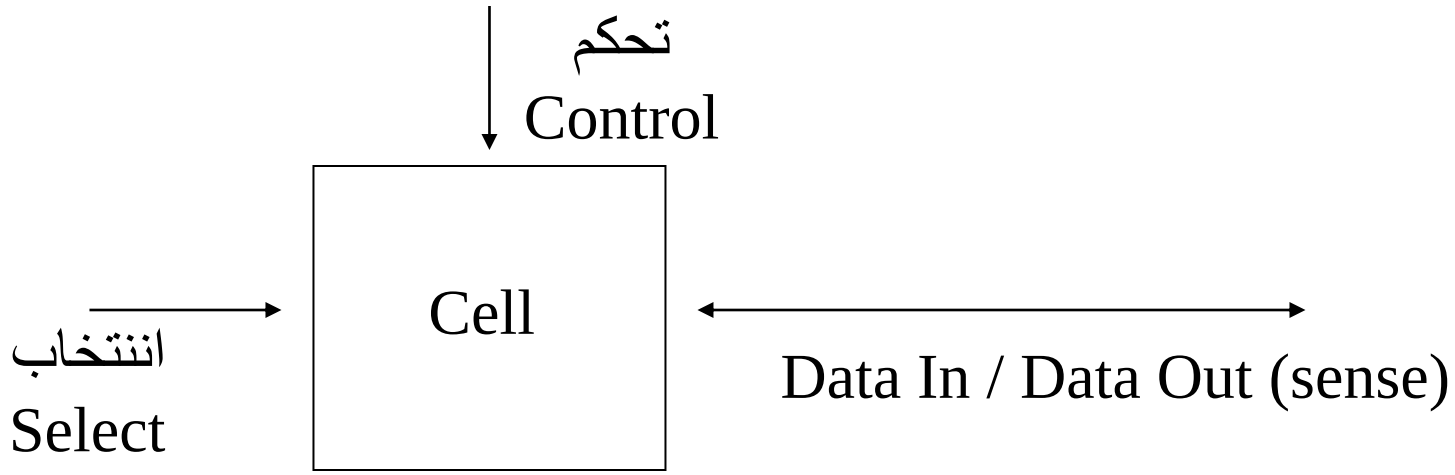
- تخزين دائم Permanent storage
- البرمجة الميكروية Microprogramming
- مكتبة البرامج الفرعية Library subroutines
- برمجيات النظم Systems programs (BIOS)
- الجداول الوظيفية Function tables

أنواع الـROM
Types of ROM

- الكتابة أثناء التصنيع Written during manufacture
  - مرتفعة الكلفة للكميات الصغيرة Very expensive for small runs
- القابلة للبرمجة مرّة واحدة Programmable (once)
  - PROM
  - تحتاج لتجهيزات خاصة للبرمجة.
- المقروءة غالباً "Read "mostly
  - قابلة للمحي وإعادة البرمجة Erasable Programmable (EPROM)
    - تمحى بالأشعة فوق البنفسجية Erased by UV
  - القابلة للمحي كهربائياً Electrically Erasable (EEPROM)
    - تستهلك زمناً للكتابة أكبر من زمن القراءة longer to write than read
  - ذواكر الومضية Flash memory
    - يمكن محي الذاكرة بأكملها كهربائياً Erase whole memory electrically

تنظيم الشريحة
Chip Organization

- تحتوي خلية الذاكرة واحدة individual memory cell على خط انتخاب (فعال أو غير فعال) وخط تحكم (قراءة أو كتابة).

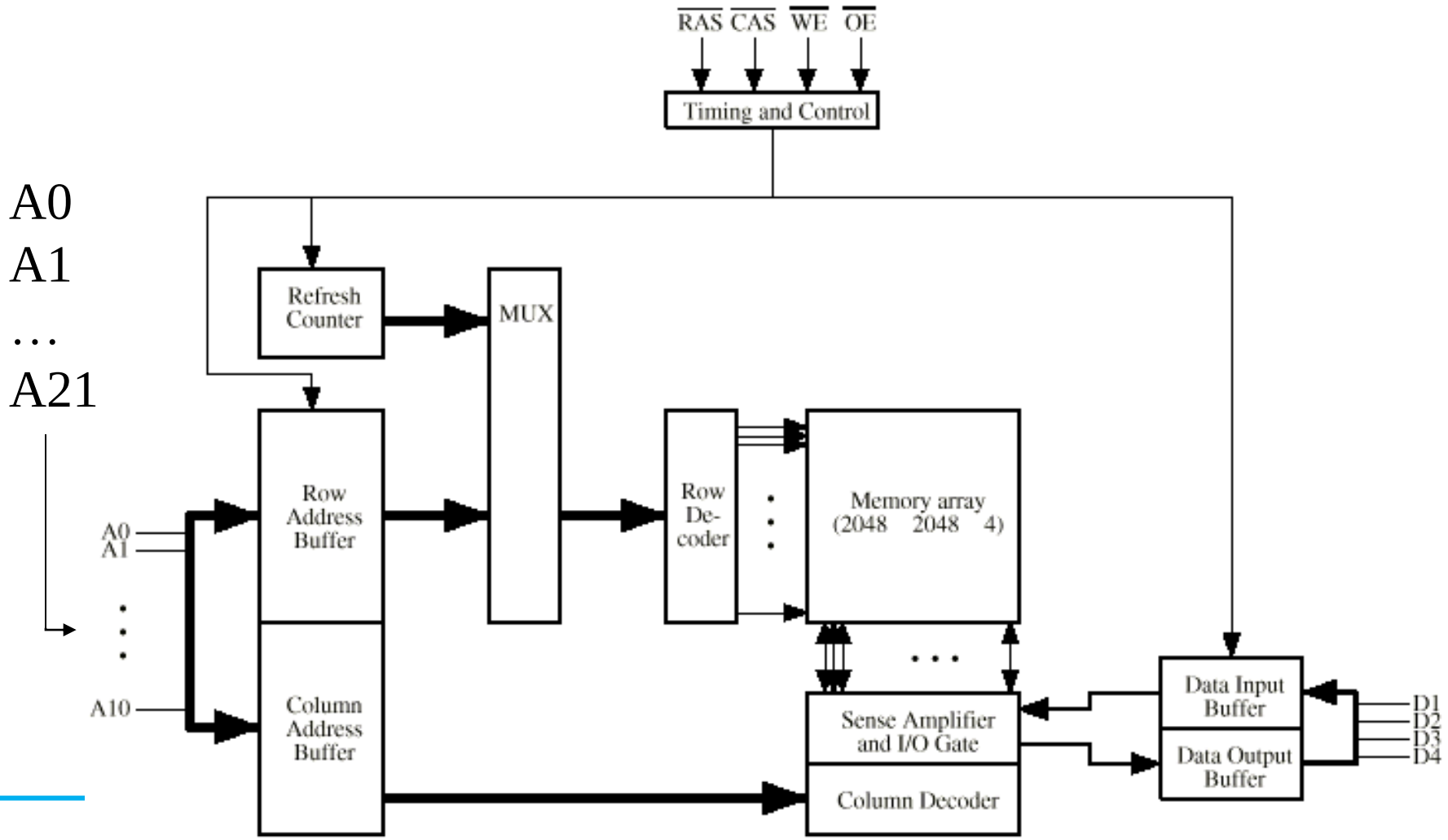


عمليات خلية الذاكرة Memory Cell Operations

التنظيم تفصيلياً
Organization in detail

- بعض الطرق الممكنة لتركيب شرائح 16Mbit
  - 1M من كلمات بعرض 16 bit
  - 16 شريحة كل منها 1Mbit، شريحة لكل خانة bit من عرض الكلمة المرغوبة.
  - أنساق 2048 x 2048 x 4bit array. تأخذ بعين الاعتبار عرض كلمة 4 bit، بحيث تغطى بـ 4,194,304 موقع قابل للعنونة.
    - تقليل عدد االأقطاب pinsالمستخدمة في العنونة.
    - تمزج عناوين الأسطر مع الأعمدة.
    - مثال: ١١قطب pins لعنونة (2^{11}=2048)، مرورين عبر كل نهاية للحصول على 22 bits (2^{22} = 4M)، لكل كلمة مؤلفة من 4 bit.
    - للنفاذ إلى الذاكرة نرسل بدايةً عنوان السطر (RAS)، ومن ثم نرسل عنوان العمود (CAS).مما يفعل خط الانتخاب SELECT line. نحتاج إلى ٤ خطوط من أجل خطوط Data In/Sense..
    - إضافة نهاية أو أكثر تضاعف مجال القيم فتزيد السعة ٤ مرات مع زيادة الأبعاد

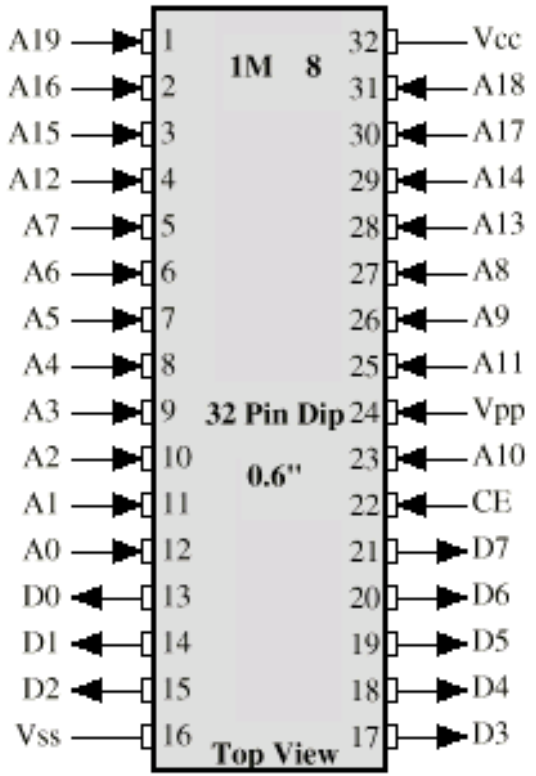
ذاكرة 16 Mb تقليدية DRAM (4M x 4)



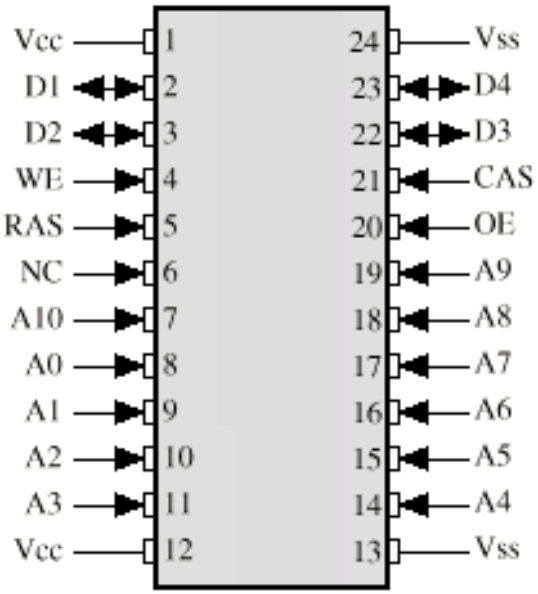
الإنعاش Refreshing

- دارة إنعاش مضمنة على الشريحة
- إلغاء تمكين الشريحة Disable chip
- العد عبر الأعمدة Count through rows
- قراءة و كتابة راجعة Read & Write back
- تستغرق وقتاً إضافياً Takes time
- تبطئ الأداء الظاهري apparent performance

Packaging التغليف



(a) 8 Mbit EPROM

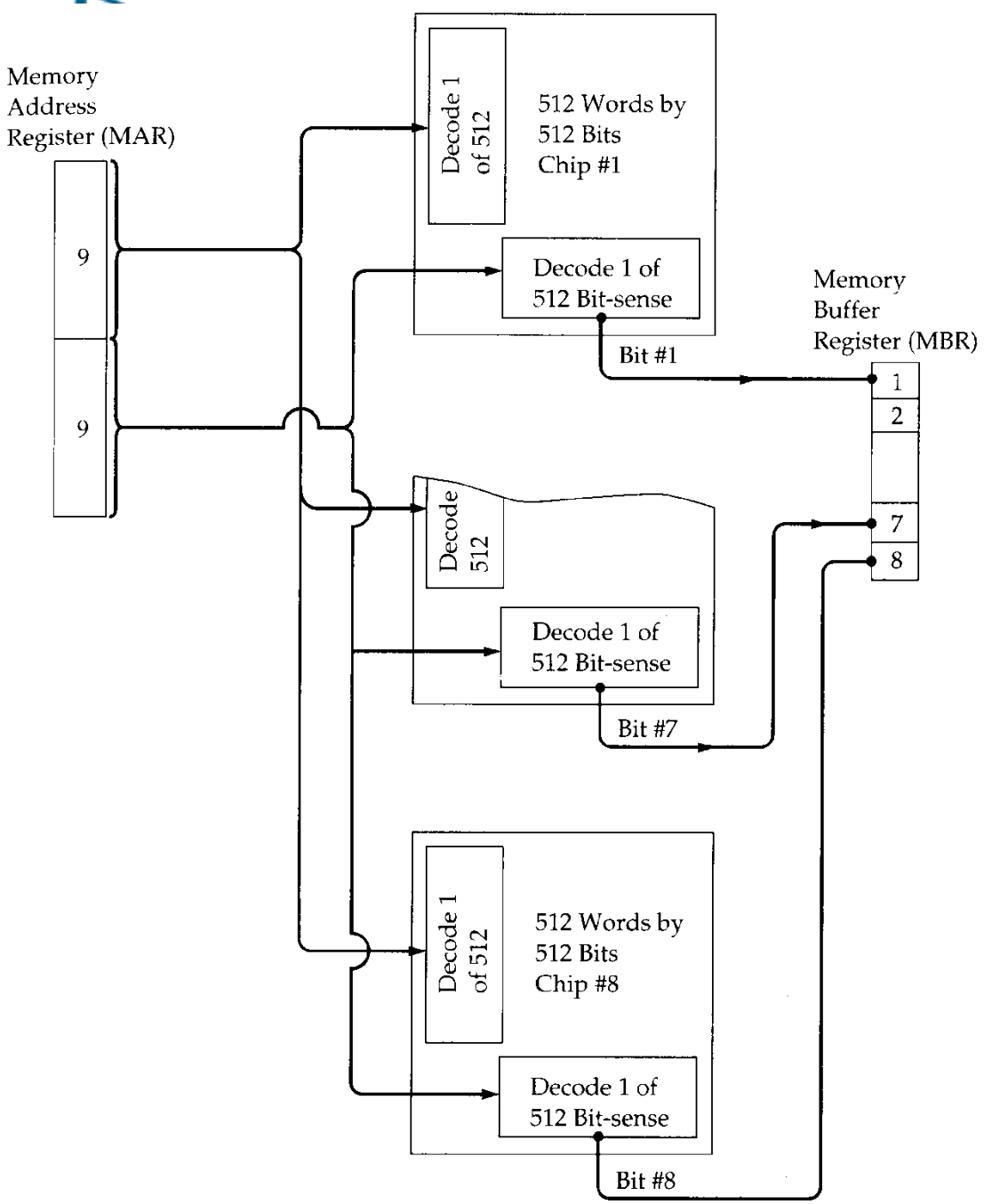


(b) 16 Mbit DRAM

CE = Chip Enable, Vss = Ground, Vcc=+V, OE = Output Enable, WE = Write Enable

تنظيم النموذج

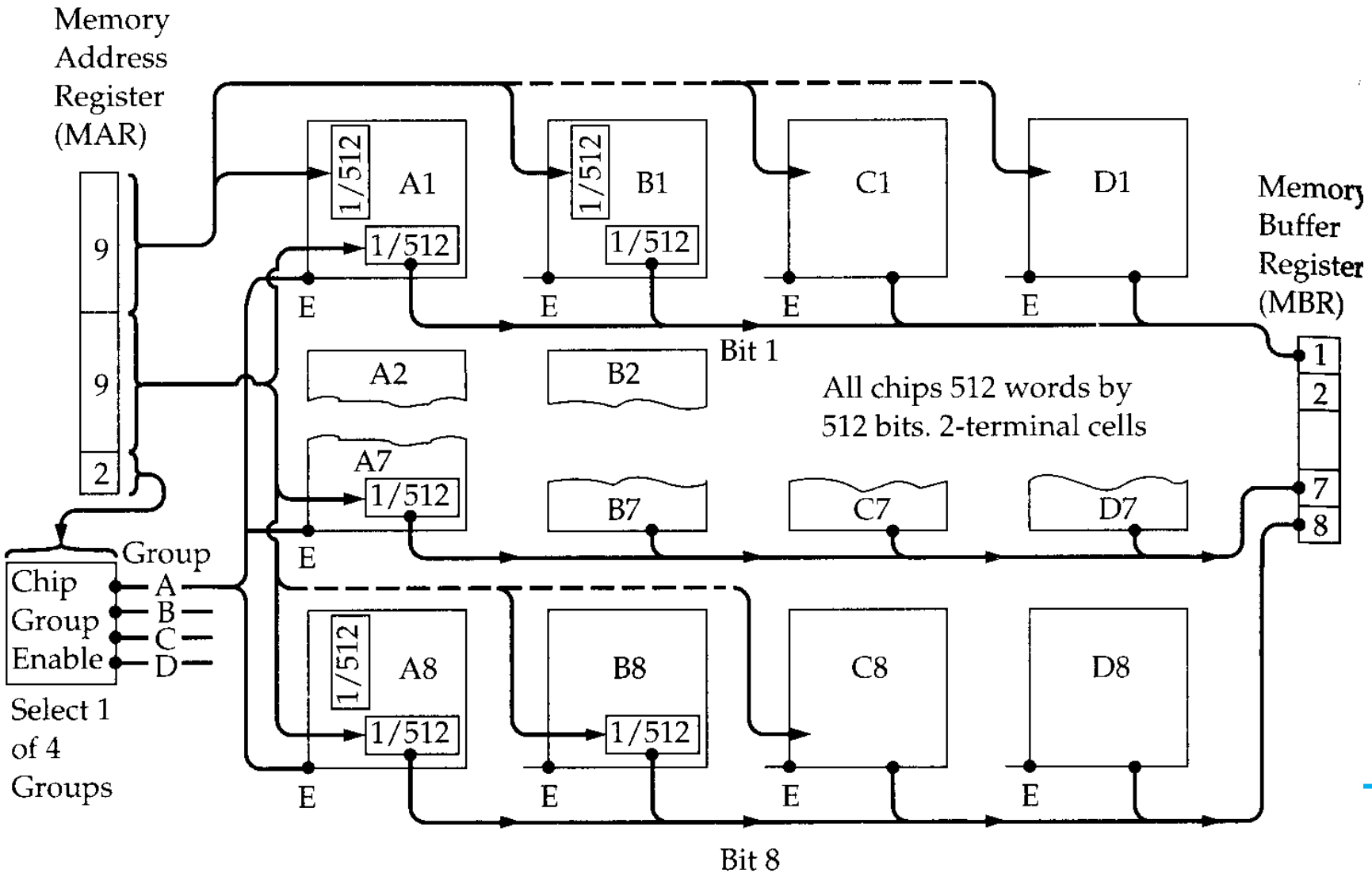
- تنظيم بديل باستخدام نماذج مرجعية كلماتها لـ256K 8 bit
- شريحة 256K 8لكل خانة من الكلمة المرغوبة المؤلفة من 8 bit.
- تقدم عناوين 18 bitلكل نموذج، وخانة خرج وحيدة. يتم توزيع البيانات على جميع الشرائح للكلمة واحدة.



تنظيم النموذج لذاكرة أكبر

- يمكن بدايةً تجميع نماذج موجودة مسبقاً للحصول على ذاكرة أكبر
- مثال ذواكر نظام 256K x 8bit
  - إذا أردنا الحصول على ذاكرة 1M يمكننا ربط ٤ نماذج256K x 8bit
  - كيف يمكننا معرفة أي من النماذج الأربعة يحتوي البيانات التي تهمنا؟
  - نحتاج 20 خط عنونة لتغطية 1M.
    - ١٨ خطاً لتغطية العناوين كما ذكر سابقاً
    - الخطين المتبقيين يستخدمان عادةً لتمكين الشرائح بحيث يجري تمكين شريحة نموذج واحدة فقط من أجل كل تركيبة من تركيبات الخطين.

تنظيم النموذج (٢)
Module Organization (2)

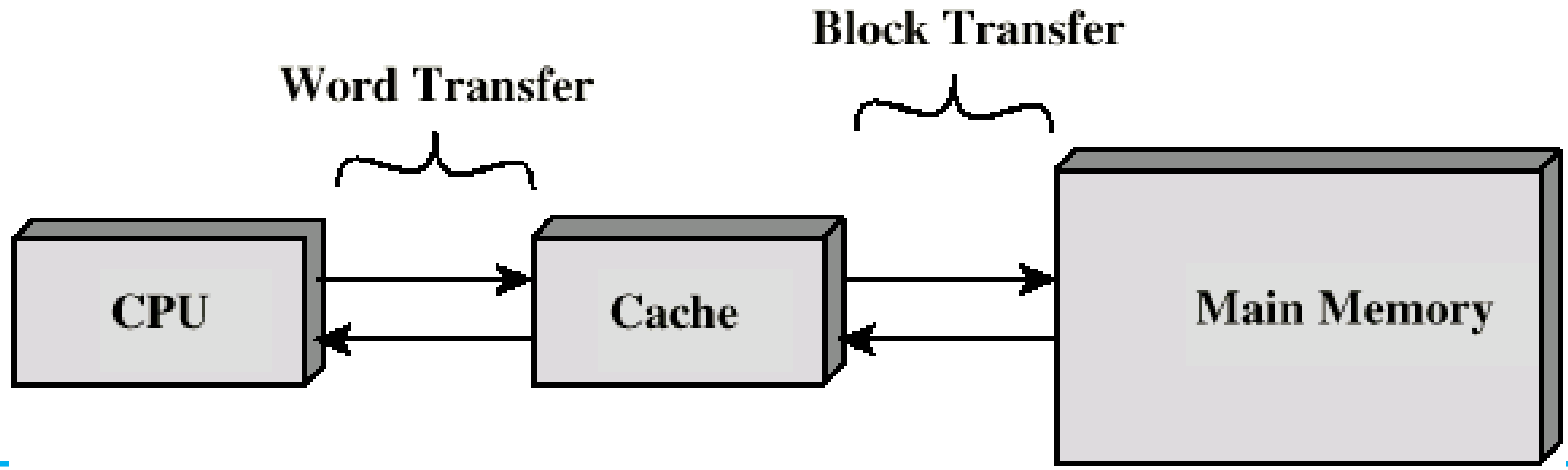


تصحيح الأخطاء
Error Correction

- الأخطاء الداراتية Hard Failure
  - عطل دائم Permanent defect
- الأخطاء اللينة أو البرمجية Soft Error
  - عشوائية وغير مخربة Random, non-destructive
  - لاعطل دائم للذاكرة.
- تعتبر تقنية شيفرة تصحيح خطأ هامينغ إحدى تقنيات كشف الأخطاء
  - شبيهة بخانة الإنجابية parity bit، لكنها تحتوي على معلومات كافية لتصحيح البيانات عند خطأ خانة واحدة.

الذاكرة المخبأية
Cache

- ذاكرة حجمها صغير لكن سرعتها كبيرة
- تتوضع بين الذاكرة الرئيسية و الـCPU.
- قد يجري وضعها على شريحة أو نموذج الـ CPU.

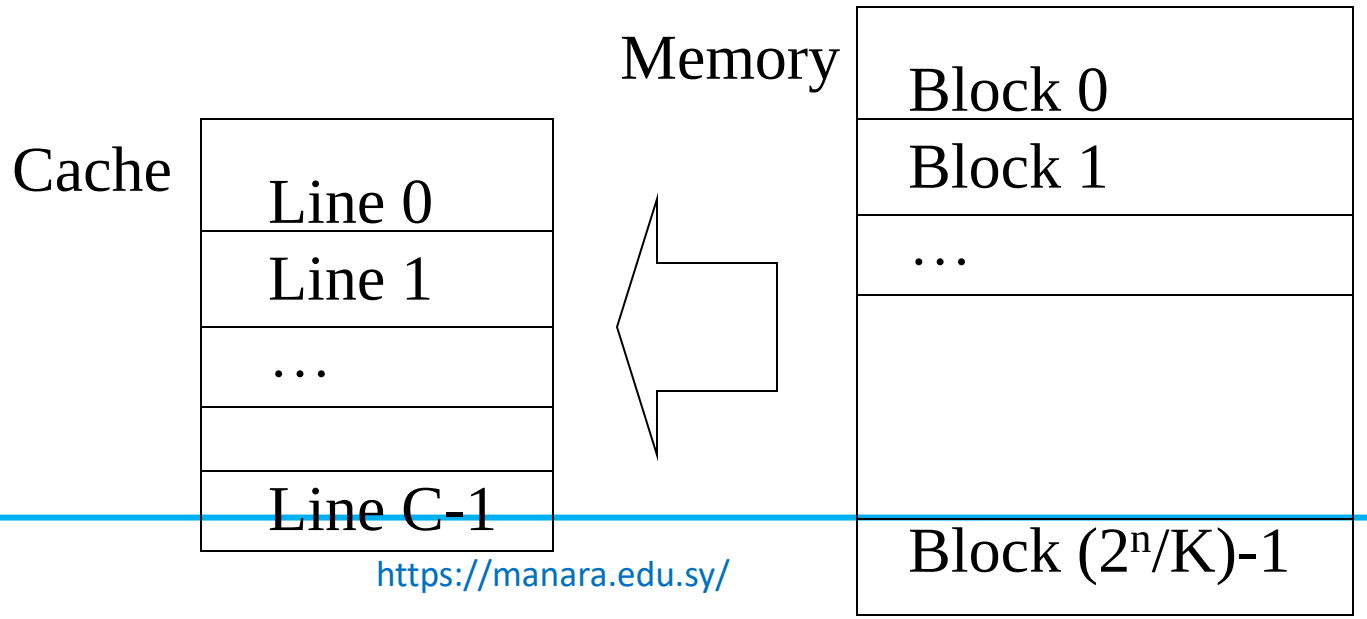


لمحة عامة عن عمليات الذاكرة المخبأية
Cache operation - overview

- تطلب الـ CPUمحتوى موقع ذاكرة ما.
- تختبر فيما إذا كانت الكاش تحتوي على البيانات المطلوبة.
- إذا كانت موجودة يتم الحصول على المحتوى بسرعة من الكاش.
- وإلا سنقرأ بلوك الذاكرة المقصود من الذاكرة الرئيسية إلى الكاش.
- ومن ثم ننقله من الكاش إلى الـCPU.
- تحتوي الكاش على علاّمات tags لتحدد أي بلوكات الذاكرة الرئيسية موجود حالياً في الكاش.

تصميم الكاش
Cache Design

- إذا كانت الذاكرة تحتوي على 2^n كلمة قابلة للعنونة.
  - يمكننا تقسيم الذاكرة إلى بلوكات كل منها يحتوي على K words. أي عدد البلوكات = 2^n / K
  - تتألف الكاش من C **lines** أو **slots**، **كل منها يضم** K words.
  - C << M
  - ويبقى السؤال عن كيفية مطابقة بلوكات الذاكرة مع خطوط الكاش؟



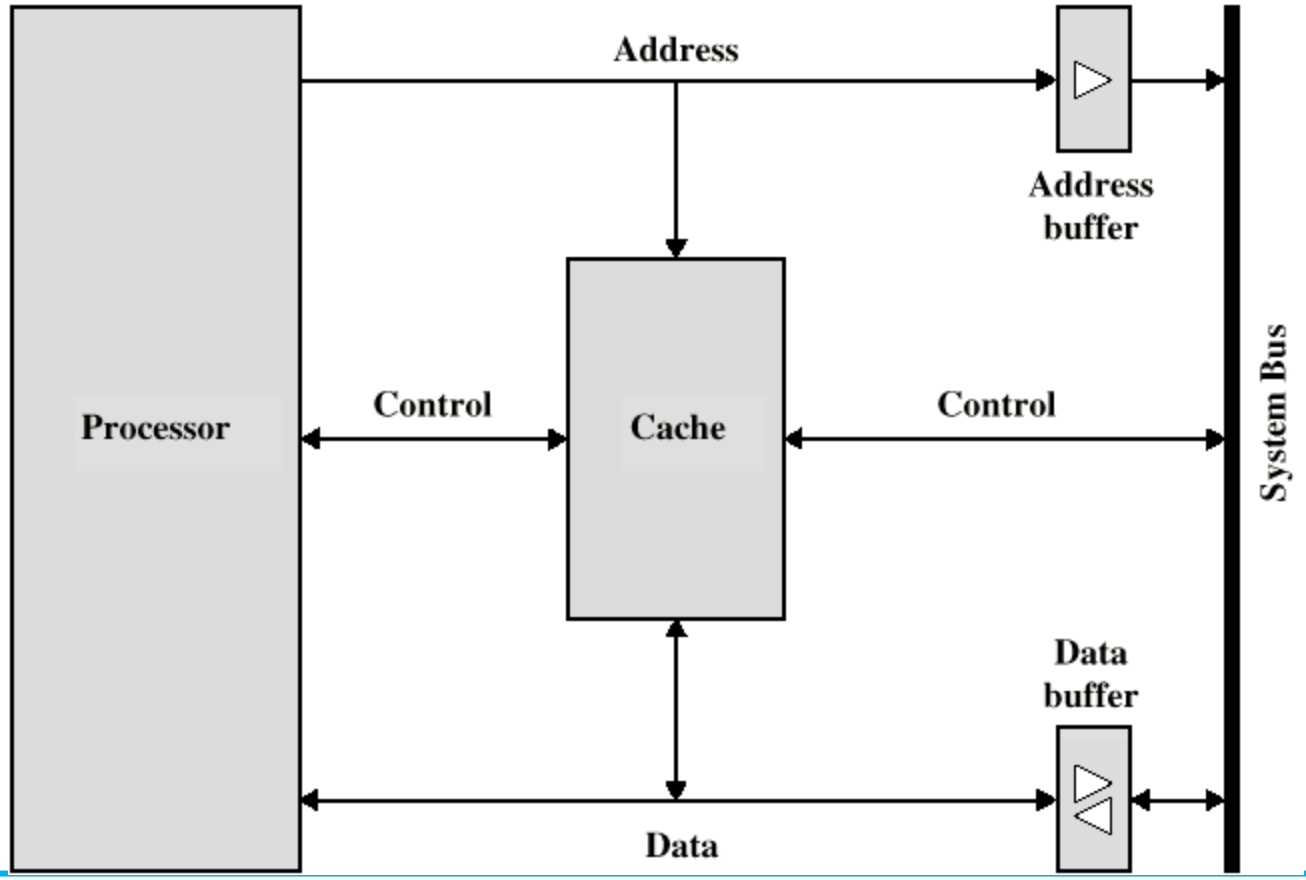
متغيرات تصميم الكاش
Cache Design

- الحجم Size
- تابع التخطيط Mapping Function
- خوارزمية الاستبدال Replacement Algorithm
- سياسة الكتابة Write Policy
- حجم البلوك Block Size
- عدد ذواكر الكاش Number of Caches

لماذا الحجم مهم
Size does matter

- الكلفة Cost
  - كلما كانت الكاش أكبر كانت الكلفة أكبر More cache is expensive
- السرعة Speed
  - كاش أكبر = سرعة أكبر More cache is faster (up to a point)
  - اختبار الكاش للبيانات يستغرق زمناً
    - إضافة كاش إضافية سيبطئ عملية البحث عن شيء ما في الكاش.

التنظيم النموذجي للكاش
Typical Cache Organization



توابع التخطيط
Mapping Function

- إذا استخدمنا التركيبة التالية للكاش:
  - كاش بسعة 64KByte.
  - علاقة السعة بين Cache line / Block هي 4 bytes
    - مثال عدد الخطوط للكاش هي (2^{14}) 16,385 خطاً كل منها 4 bytes
  - الذاكرة الرئسية بسعة 16MBytes
    - 24 bit address
    - (2^{24}=16M)
    - 16Mbytes / 4bytes-per-block → 4 MB of Memory Blocks
- يتوجب علينا أحياناً تخطيط الـ 4Mb من البلوكات in ذاكرة بسعة 16K من الخطوط في الكاش.يتوجب حينها أن نطابق البلوكات المتعدة إلى نفس الخط في الكاش

التخطيط المباشر
Direct Mapping

- تقنية التخطيط الأبسط تتم فيه مطابقة كل بلوك من الذاكرة الرئيسية بخط كاش وحيد.
  - مثلاً يجب أن يطابق البلوك الموجود في الكاش موقع محدد من الذاكرة.
- صيغة مطابقة بلوك الذاكرة إلى خط الكاش cache line:

- i = j mod c
  - حيث
  - i=Cache Line Number
  - j=Main Memory Block Number
  - c=Number of Lines in Cache

التخطيط المباشر بـC=4
Direct Mapping with C=4

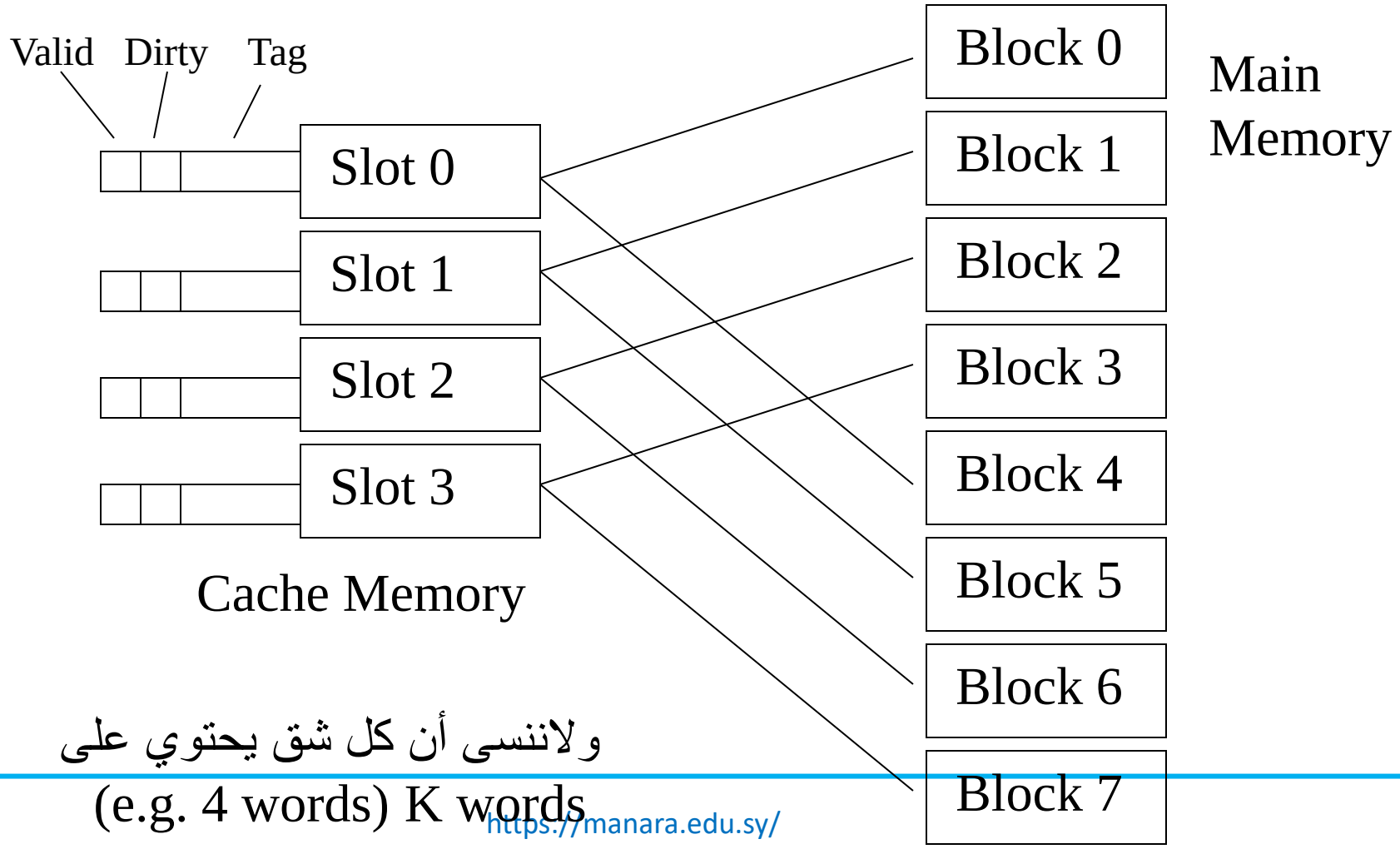
- إذا اقتصرنا في مثالنا على خط كاش بأربع شقوق (كل slot/line/block ليزال يضم 4 words):

Cache Line	Memory Block Held
0	0, 4, 8, …
1	1, 5, 9, …
2	2, 6, 10, …
3	3, 7, 11, …

- In general:

0	0, C, 2C, 3C, …
1	1, C+1, 2C+1, 3C+1, …
2	2, C+2, 2C+2, 3C+2, …
3	3, C+3, 2C+3, 3C+3, …

التخطيط المباشر بـC=4
Direct Mapping with C=4



ولاننسى أن كل شق يحتوي على
(e.g. 4 words) K words

Direct Mapping Address Structure بنية التخطيط المباشر

- Address is in two parts العنوان مكون من قسمين
  - w bits الأقل أهمية التي تعرّف كلمة وحيدة ضمن خط الكاش. خانات
  - s bits الأكثر أهمية التي تعرّف أحد بلوكات الذاكرة. خانات
- وتقسيم الخانات الأكثر أهمية MSBs إلى حقل خطوط الكاش cache line field r و العلاّم tag s-r (الأكثر أهمية most significant)

بنية عناوين التخطيط المباشر Direct Mapping Address Structure

V	D	Tag s-r	Line or Slot r	Word w
1	1	8	14	2

- ليكن لدينا عنوان مكون من 24 bit (للنفاذ إلى 16Mb)
- معرّف كلمة مؤلف من كلمة من 2 bit (بلوك من 4 byte)
- معرّف بلوك مؤلف من 22 bit.
  - علاّم من 8 bit أي (22-14=)
  - 14 bit لتحديد الشقوق أو الخطوط slot or line.
- ليس هنالك أي بلوكين في نفس الخط لهما نفس حقل العلاّم.
- نختبر محتوى الكاش من خلال إيجاد الخط واختبار العلاّم Tag.
- وبالتالي نحتاج إلى خانة المصادقةValid bit وخانة التلوث Dirty bit.
  - المصادقة Valid تبين فيما إذا كان الشق يضم بلوك ينتمي إلى البرنامج قيد التنفيذ.
  - التلوث Dirty تدل على أن البلوك قد تم تعديله عند وجوده في الكاش. مما يوجب الكتابة الراجعة في الذاكرة قبل إعادة استخدام الشق لبلوك آخر.

مثال تخطيط مباشر Direct Mapping لكاش 64K

ذاكرة الكاش

	Addr	Tag	W0	W1	W2	W3
0		00	F1	F2	F3	F4
1		1B	11	12	13	14
2						
3						
4						
5						
..						
..						
2^{14}-1						

الذاكرة الرئيسية

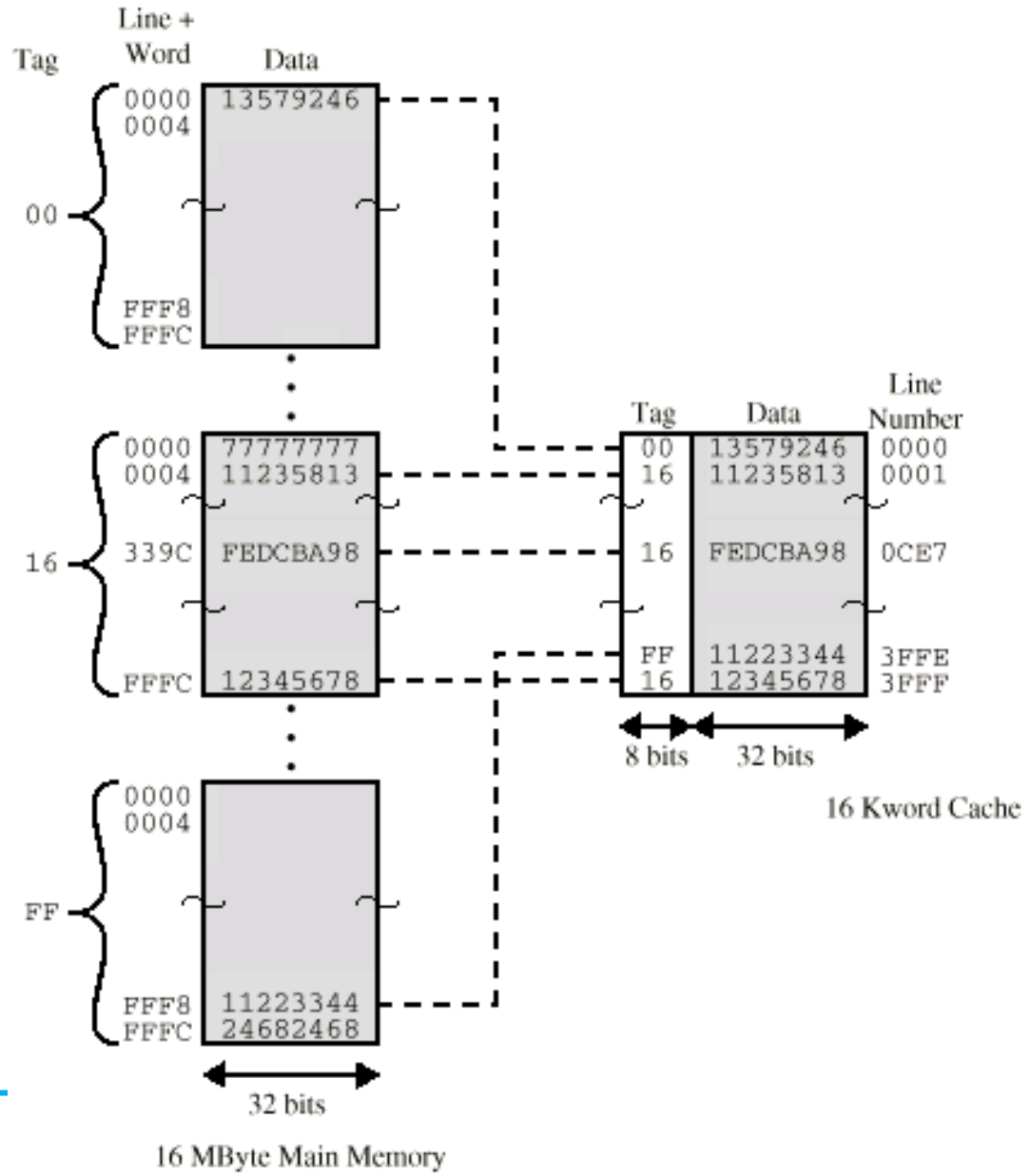
	Addr (hex)	Data
Line 0	000000	F1
	000001	F2
	000002	F3
	000003	F4
Line 1	000004	AB
	…	
Line 1	1B0004	11
	1B0005	12
	1B0006	13
	1B0007	14

1B0007 = 0001 1011 0000 0000 0000 0111

Word = 11, Line = 0000 0000 0000 01, Tag= 0001 1011

مثال تخطيط مباشر

المثال الأصلي
لكاش من64K
ب 4 words
لكل بلوك



فرضيات ونتائج التخطيط المباشر
Direct Mapping pros & cons

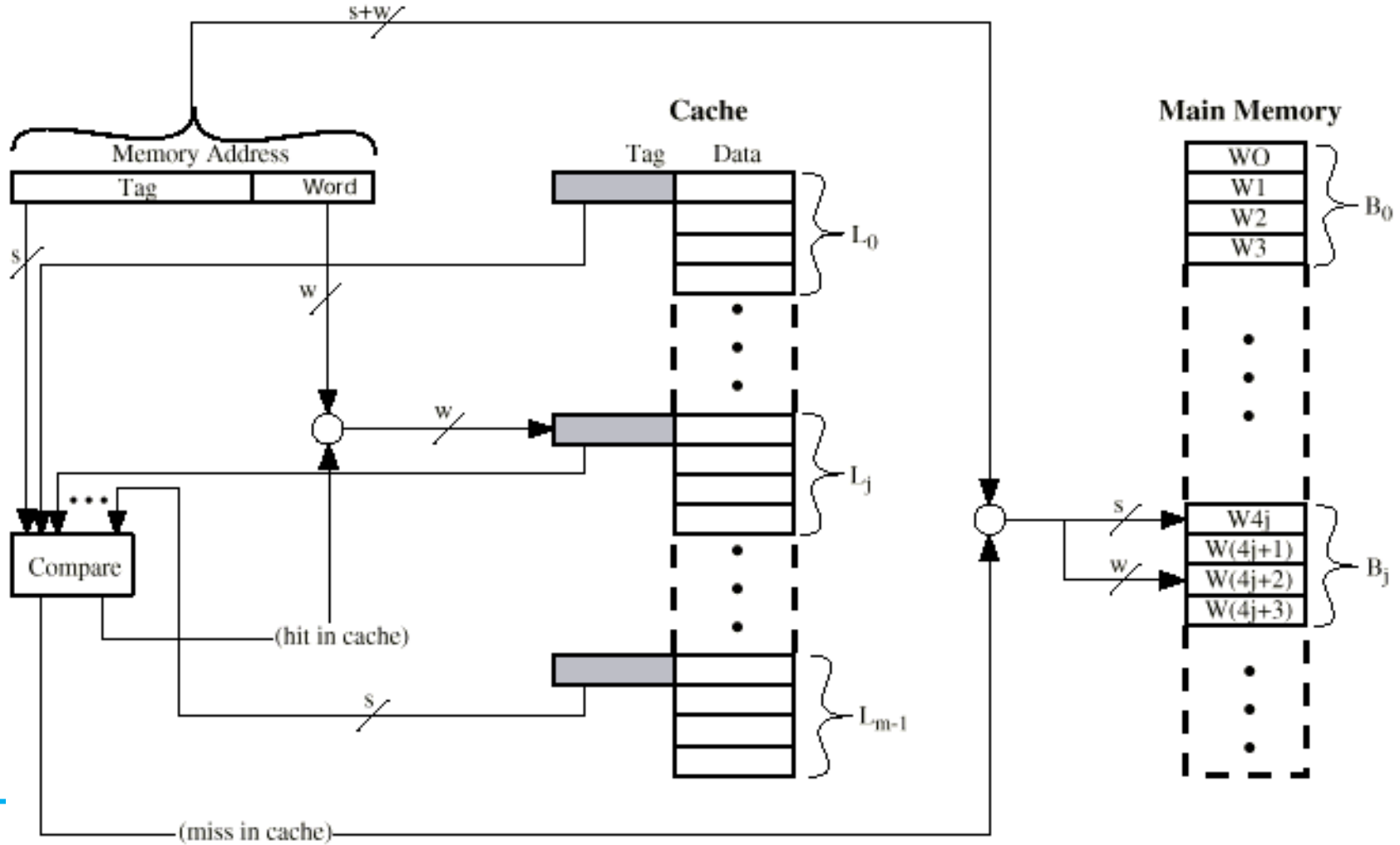
- البساطة Simple
- رخيصة الثمن Inexpensive
- موقع ثابت لبلوك ما Fixed location for given block
- إذا احتاج البرنامج بشكل تكراري للنفاذ إلى بلوكين يقعان وفقاً لتخطيط الذاكرة في نفس الخط line، ترتفع نسبة خطأ الكاش إلى مستوى عالي جداً وهذا مايطلق عليه تسمية بالنفايات أو **thrashing**

التخطيط كامل التشاركية
Fully Associative Mapping

- يمكن للتخطيط كامل التشاركية أن يتغلب على مشاكل التخطيط المباشر.
  - يمكن تحميل أي بلوك من الذاكرة الرئيسية في أي خط من خطوط الكاش.
  - تترجم عناوين الذاكرة كعلاّم tag و كلمة word.
  - يعرّف العلاّم بشكل مفرد بلوك في الذاكرة.
  - يجري اختبار كل علاّم خط لاختبار التطابقmatch.
  - وبالتالي تظهر الحاجة لخانة التلوث Dirty و خانة المصادقة Valid (ليست في المثال)
- إلاّ أن البحث ضمن الكاش يصبح أكثر كلفةً!
  - نحتاج أمثلياً إلى دارات تستطيع اختبار تطابق جميع العلاّمات في نفس اللحظة
  - نحتاج إلى عدد كبير من الدارات وهذا يؤدي إلى كلفة عالية.
- نحتاج إلى سياسة استبدال تسمح برمي أية بيانات من الكاش (يدرس لاحقاً)

Fully Associative Cache Organization

تنظيم الكاش كامل التشاركية

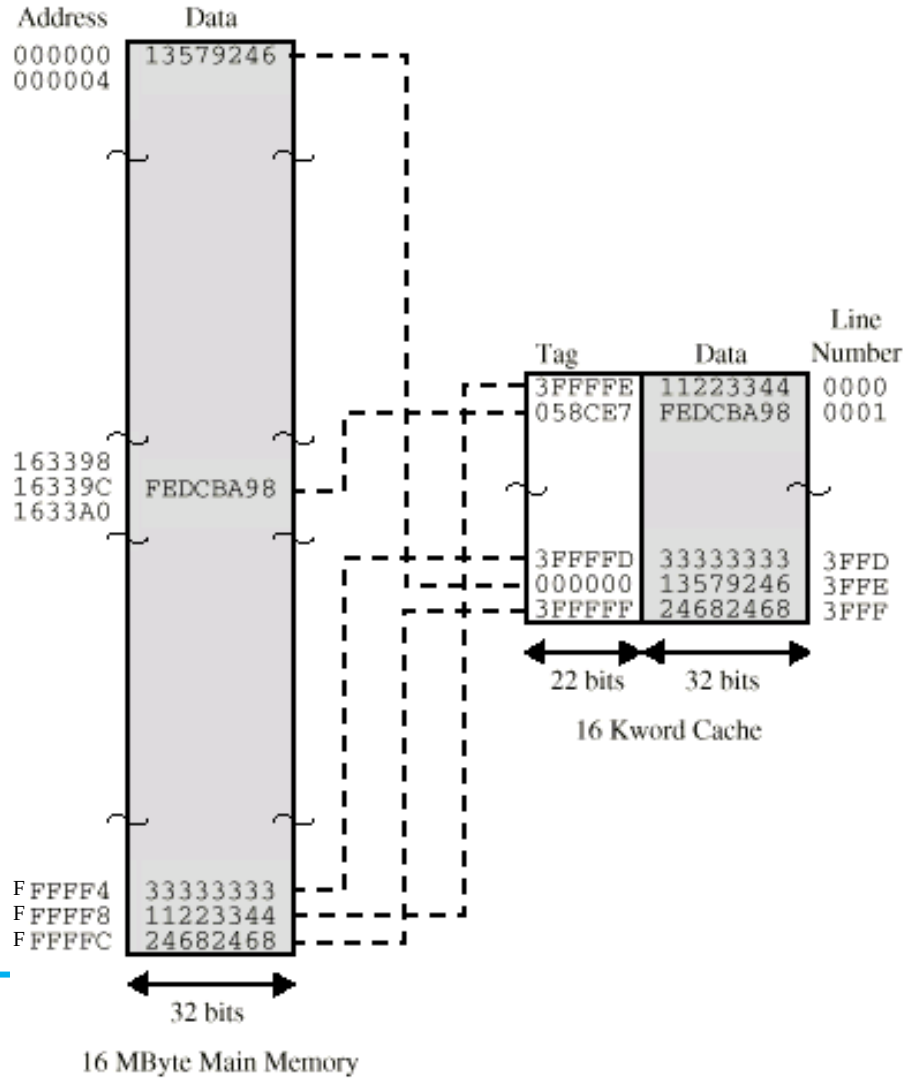


Associative Mapping بنية عنوان التخطيط التشاركي Address Structure

Tag 22 bit	Word 2 bit

- علاّم 22 bit مخزّن من أجل كل بلوك بيانات من 32 bit.
- مقارنة حقل العلاّم مع مدخل العلاّم في الكاش لاختبار الإصابة hit.
- تحدد الخانتين الأقل أهمية من العنوان أي الكلمات المؤلفة من 8 bit هي المطلوبة من قبل بلوك البيانات المؤلف من 32 bit.
- مثال:
  - Address: FFFFFC = 1111 1111 1111 1111 1111 1100
    - Tag: Left 22 bits, truncate on left:
      - 11 1111 1111 1111 1111 1111
      - 3FFFFF
  - Address: 16339C = 0001 0110 0011 0011 1001 1100
    - Tag: Left 22 bits, truncate on left:
      - 00 0101 1000 1100 1110 0111
      - 058CE7

مثال التخطيط التشاركي
Associative Mapping Example

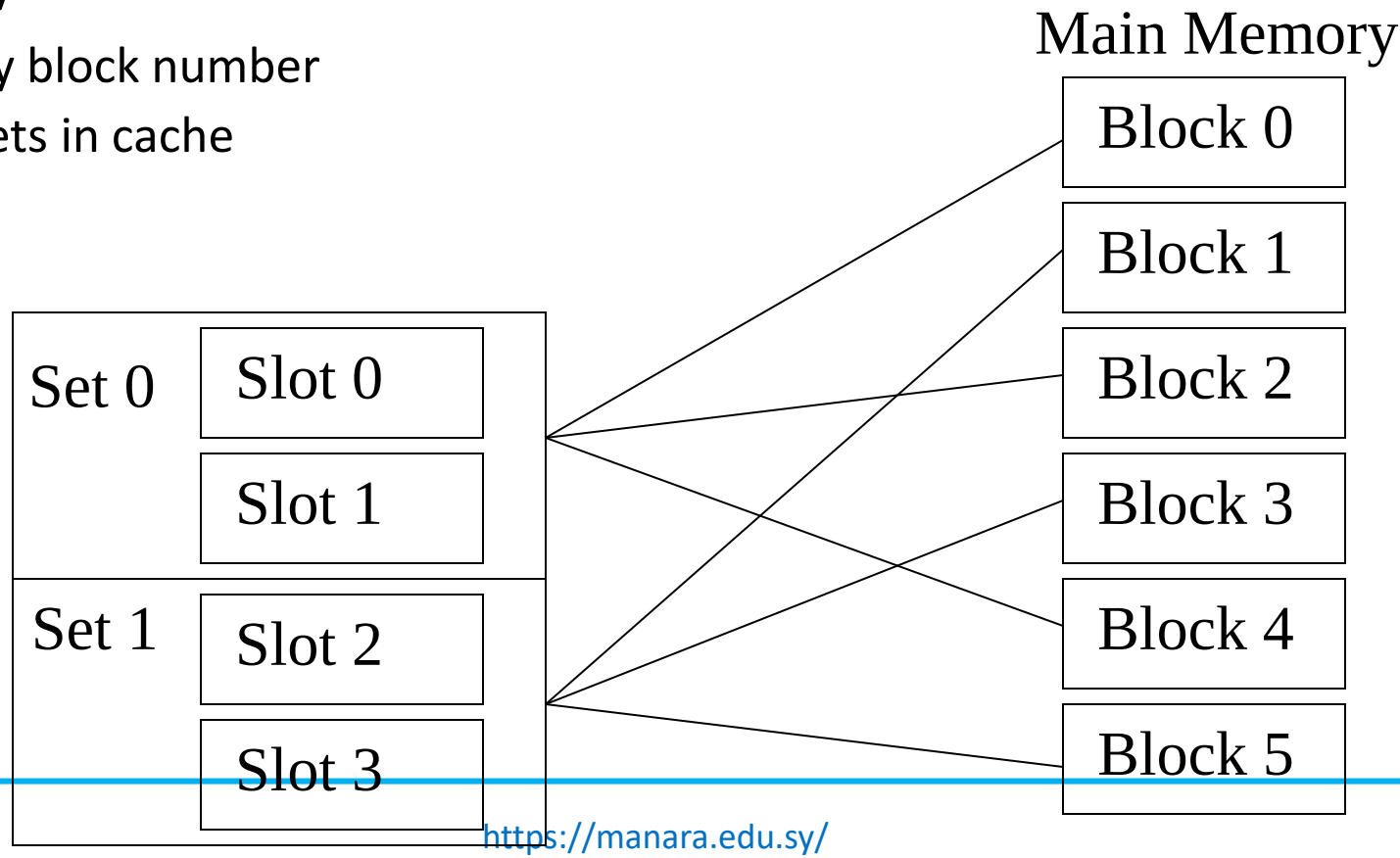


تخطيط المجموعة التتشاركي
Set Associative Mapping

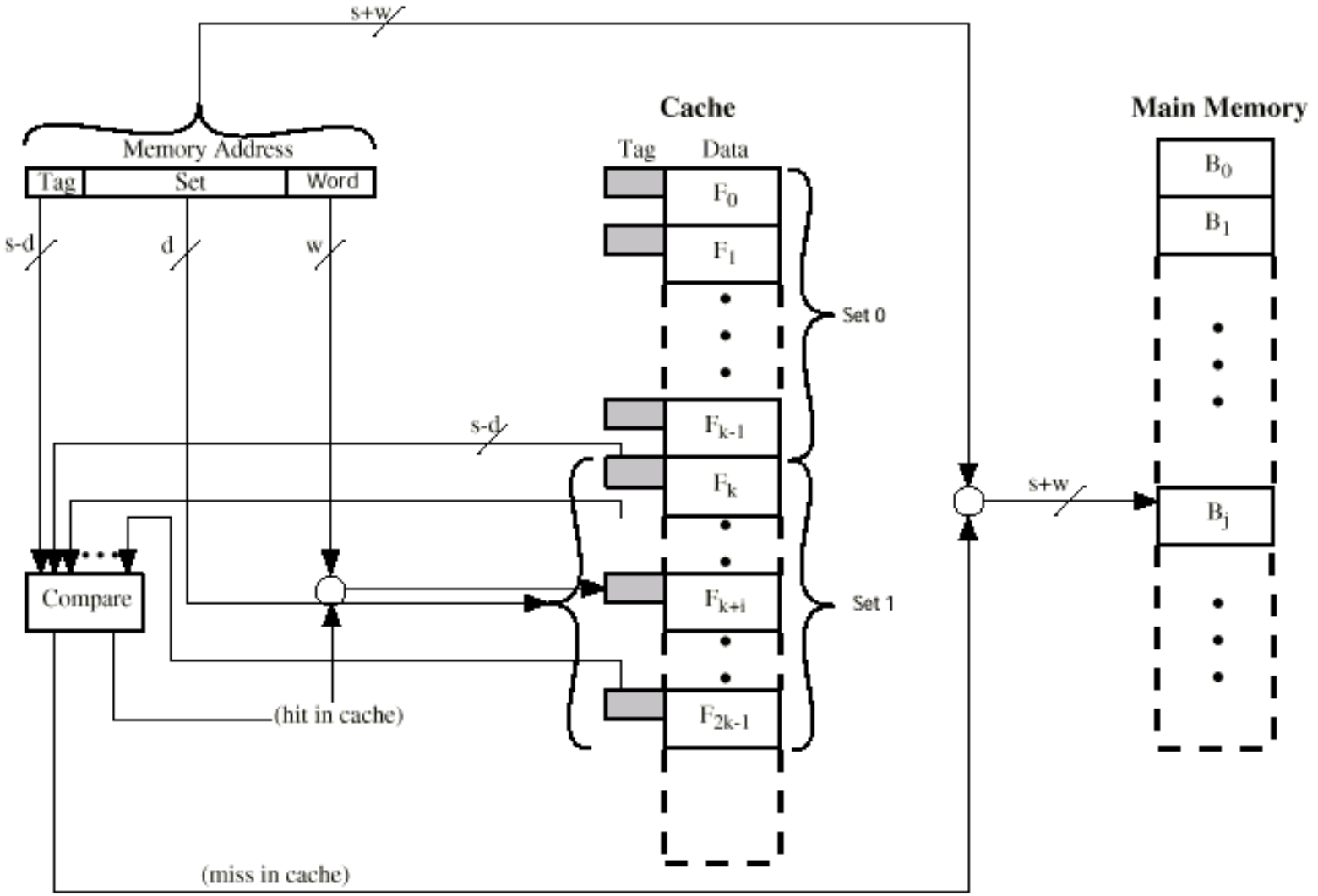
- حل وسط بين التخطيط التشاركي الكلي و التخطيط المباشر.
  - تقسم الكاش إلى عدد من المجموعات sets.
  - تحتوي كل مجموعة على عدد من الخطوط.
  - يخطط كل بلوك أي خط في المجموعة المحددة.
    - نستخدم التخطيط المباشر لتحديد أي مجموعة من مجموعات الكاش توافق مجموعة من الذاكرة.
    - عندئذٍ يمكن أن يتوضع بلوك الذاكرة في أي خط من خطوط تلك المجموعة.
  - مثال خطين لكل مجموعة 2 lines per set
    - تخطيط تشاركي باتجاهين 2 way associative mapping
    - يمكن للبلوك المعني أن يكون في أحد الخطين في المجموعة المحددة.
  - مثال k خط لكل مجموعة K lines per set
    - تخطيط تشاركي ب k اتجاه K way associative mapping
    - يمكن أن يكون البلوك المعني في أي من الـk خط من خطوط المجموعة المحددة.
    - ويكون من الأسهل البحث في نفس اللحظة في مجموعة بدلاً م البحث في جميع الخطوط.

تخطيط المجموعة التشاركي
Set Associative Mapping

- لحساب رقم مجموعة الكاش:
  - SetNum = j mod v
    - j = main memory block number
    - v = number of sets in cache



تنظيم الكاش لتشاركية المجموعة باتجاهين
Two Way Set Associative Cache Organization

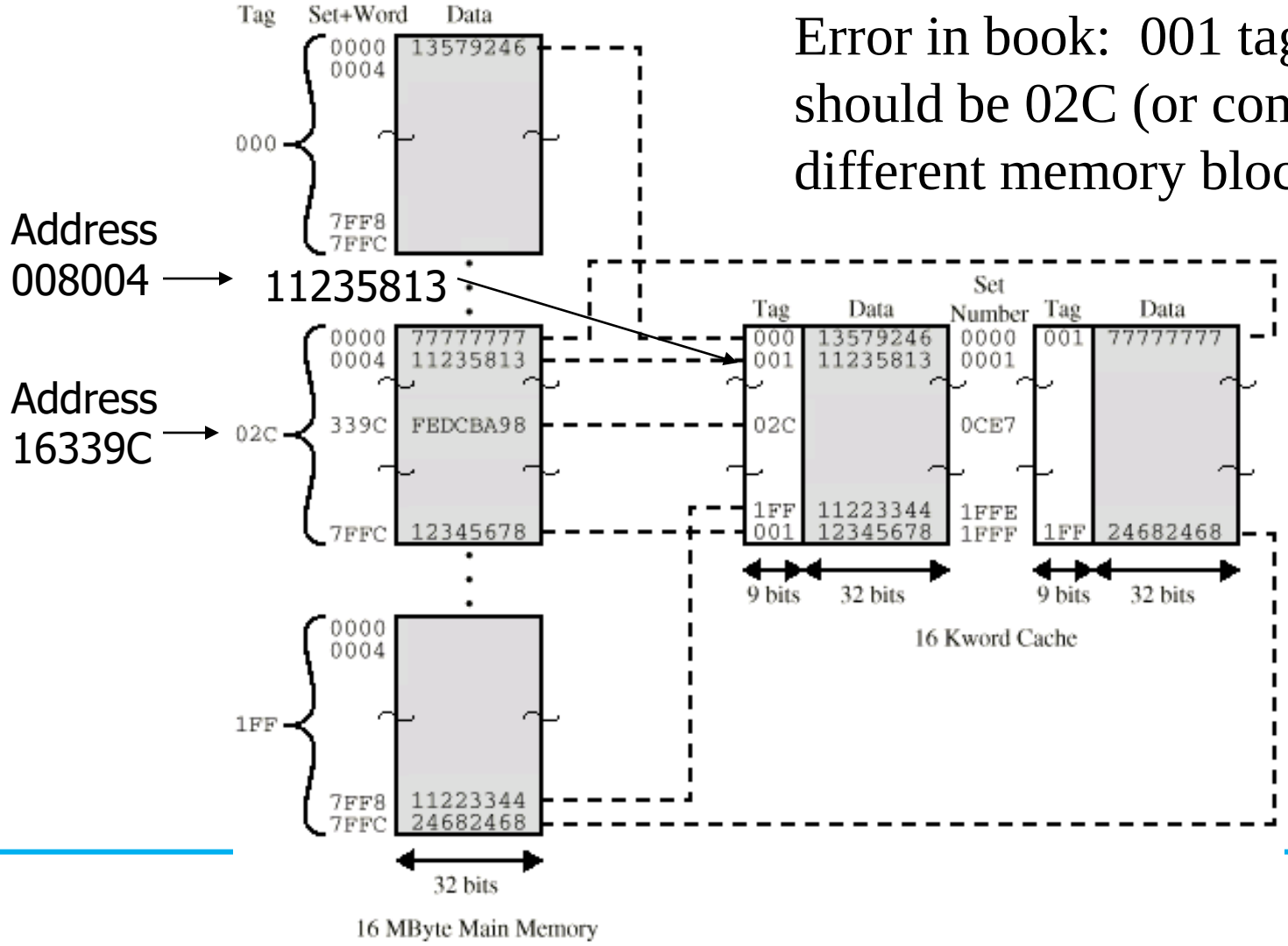


بنية العنوان في تخطيط المجموعة التشاركي
Set Associative Mapping Address Structure

Tag 9 bit	Set 13 bit	Word 2 bit

- E.g. given a 13 bit set number for 24 bit address
- Use set field to determine cache set to look in
- Compare tag field of all slots in the set to see if we have a hit, e.g.:
  - Address = 16339C = 0001 0110 0011 0011 1001 1100
    - Tag = 0 0010 1100 = 02C
    - Set = 0 1100 1110 0111 = 0CE7
    - Word = 00 = 0
  - Address = 008004 = 0000 0000 1000 0000 0000 0100
    - Tag = 0 0000 0001 = 001
    - Set = 0 0000 0000 0001 = 0001
    - Word = 00 = 0

مثال تخطيط مجموعة تشاركي ثنائي الاتجاه
Two Way Set Associative Mapping Example



Error in book: 001 tag in cache should be 02C (or come from a different memory block!)

تشاركية المجموعة بـK اتجاه
K-Way Set Associative

- تقدم تشاركية المجموعة باتجاهين Two-way set associative أداءً وإنجازية أفضل من التخطيط المباشر direct mapping .
  - يتوجب فقط على شق إضافي أن يتحمل مشكلة النفايات problem
- بينما تقدم تشاركية المجموعة بأربع اتجاهات تحسينا طفيفاً على التشاركية باتجاهين.
- وتأثير الزيادة في عدد المجموعات ضعيف التأثير رغم أنه يزيد في كلفة البنية الداراتية.

خوارزميات الاستبدال (١) للتخطيط المباشر
Replacement Algorithms (1)
Direct mapping

- No choice دون خيارات
- Each block only maps to one line مطابقة كل البلوك مع خط وحيد
- Replace that line استبدل ذلك الخط

خوارزميات الاستبدال (٢) للتشاركية ولتشاركية المجموعة
Replacement Algorithms (2)
Associative & Set Associative

- يجب تطبيق خوارزمية تنفذ داراتياً لزيادة السرعة.
- استبدال الأقل استخداماً Least Recently used (LRU)
  - مثلاً في تشاركية المجموعة باتجاهين، أي البلوكين سنقوم باستبداله؟
    - لكل شق يكون لدينا خانة إضافية (خانة الاستخدام) تأخذ قيمة الواحد عندما يتم النفاذ إلى هذا الشق وإلا تأخذ قيمة الصفر.
  - لأكثر من اتجاهين، نحتاج إلى مميز زمني لكل شق (عالي الكلفة)
- أول من يطلب أول من يستبدل First in first out (FIFO)
  - استبدال البلوك الأقدم في الكاش.
  - سهل التطبيق كعازل دائري أو حلقي circular buffer.
- أقل البلوكات من حيث تكرار استخدامه Least frequently used
  - استبدال البلوك ذي أقل نسبة إصابة fewest hits.
  - نحتاج إلى عداد لحساب عدد الإصابات counter to sum number of hits.
- العشوائي Random
  - غالباً بنفس جودة LFU وسهل التطبيق.

سياسة الكتابة
Write Policy

- لايجوز الكتابة فوق أي من بلوكات الكاش مالم يجري تحديث الذلكرة الرئيسية. فإذا كان هنالك على سبيل المثال خانة تلوث فعالة "dirty" ، فإننا نحتاج لحفظ هذا البلوك قبل الكتابة فوقه.
- وهذا مايمكن أن يسبب مشكلة كبيرة
  - قد يكون لأكثر من CPUs كاش خاصة بها
    - ماذا لو حاولت الـ CPU قراءة البيانات من الذاكرة؟ قد لاتكون هذه البيانات مشروعة وصحيحة إذا إذا قام أحد المعالجات الأخرى بتغيير الكاش الخاصة به من هذا الوقع.
    - تسمى هذه المشكلة مشكلة تماسك الكاش **cache coherency** problem
  - قد تستطيع الـ I/O أيضاً عنونة الذاكرة الرئيسية مباشرةً.

الكتابة العابرة
Write through

- التقنية الأبسط لمعالجة تشكلة تماسك الذاكرة – حيث تذهب جميع عمليات الكتابة إلى الذاكرة الرئيسية إضافة إلى تلك التي في الكاش.
- علينا مراقبة حركة بيانات الذاكرة الرئيسية في حال تعدد الـ CPUs (snooping) لنحافظ على محلية وحداثة الكاش المحلية بالنسبة لكل CPU في حال قامت CPU أخرى بحيازة نسخة من موقع الذاكرة المشترك في الكاش الخاصة به.
- بسيط إلا أن حركة البيانات فيه كبيرة. Simple but Lots of traffic
- يبطأ عمليات الكتابة Slows down writes

- الحلول الأخرى: الذاكرة غير القابلة للتخزين الوسيطي noncachable memory، استخدام البنية الداراتية للحفاظ على التماسك.

الكتابة الراجعة
Write Back

- تحصل جميع عمليات التحديث مبدئياً في الكاش فقط.
- يجري تصفير خانة التلوث بعد تحديث بلوك الكاش.
- إذا توجب استبدال بلوك ما، فلا تتم كتابته في الذاكرة الرئيسية إذا كانت خانة التلوث فعالة
- يمكن أن تخرج الكاشات الأخرى من المزامنة out of sync.
- إذا توجب على الـ I/O أن ينفذ إلى ذاكرة رئيسية غير مصادق عليها فإن عبور الكاش قد يكون أحد الحلول المطروحة.
  - تعقيد ةداراتي Complex circuitry
- فقط ~15% من الذاكرة يسمح باعتبارها للكتابة.

أداء أو إنجازية الكاش
Cache Performance

- هنالك معياران لتقييس أداء الكاش هما نسبة الإصابة **hit ratio** وزمن النفاذ الفعال **effective access time**

- Hit Ratio = (Num times referenced words are in cache)

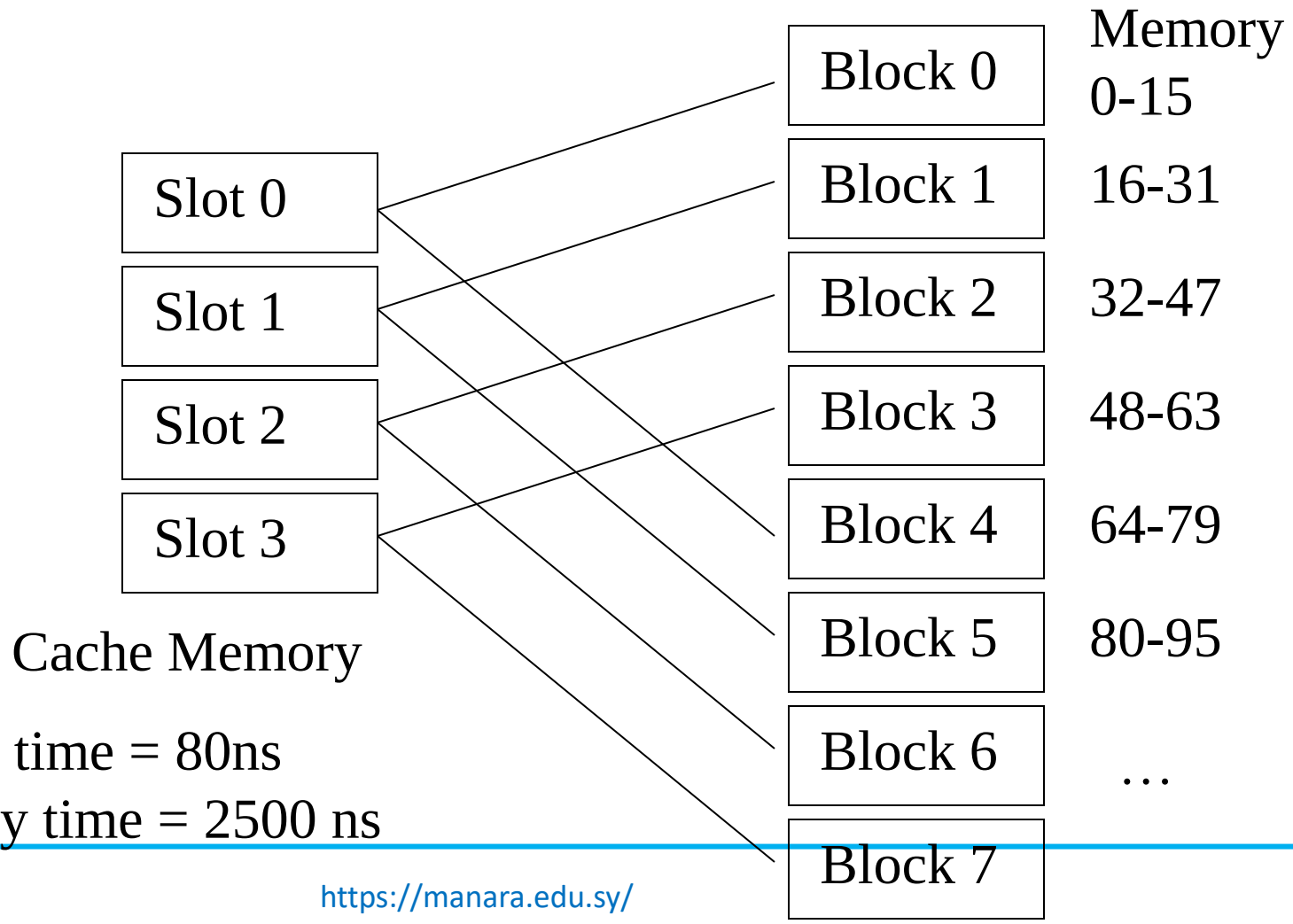
  (Total number of memory accesses)

Eff. Access Time = (# hits)(TimePerHit)+(# misses) (TimePerMiss)

(Total number of memory accesses)

مثال عن أداء أو إنجازية الكاش
Cache Performance Example

- كاش تخطيط مياشر



Cache access time = 80ns
Main Memory time = 2500 ns

مثال عن أداء أو إنجازية الكاش
Cache Performance Example

- بتنفيذنا لبرنامج في المواقع 95-48 لمرة واحدة فإنه ينفذ من 31-15 في حلقة قبل عشر مرات قبل أن يخرج.

Event	Location	Time	Comment
1 miss	48	2500ns	Memory block 3 to cache slot 3
15 hits	49-63	80ns×15=1200ns	
1 miss	64	2500ns	Memory block 4 to cache slot 0
15 hits	65-79	80ns×15=1200ns	
1 miss	80	2500ns	Memory block 5 to cache slot 1
15 hits	81-95	80ns×15=1200ns	
1 miss	15	2500ns	Memory block 0 to cache slot 0
1 miss	16	2500ns	Memory block 1 to cache slot 1
15 hits	17-31	80ns×15=1200ns	
9 hits	15	80ns×9=720ns	Last nine iterations of loop
144 hits	16-31	80ns×144=12,240ns	Last nine iterations of loop
Total hits = 213		Total misses = 5	

مثال عن أداء أو إنجازية الكاش
Cache Performance Example

- Hit Ratio: 213 / 218 = 97.7%
- Effective Access Time: ((213)*(80ns)+(5)(2500ns)) / 218 = 136 ns

- نلاحظ أن نسبة الإصابة عالية، وزمن النفاذ الفعال في هذا المثال هو 75% أطول من زمن نفاذ الكاش الناجم عن الزمن الطويل المصروف خلال خطأ الكاش.
- ماهو تتالي النفاذ لبلوكات الذاكرة الرئيسية الذي يقدم نتائج أسوأ من ذلك.

جَامعة
المَنارة
MANARA UNIVERSITY