MANARA University

Lattakia

Faculty of Engineering

Department of Informatique

Computer Architecture

CPU Organisation and Operation

Memory Organisation

I/O Modules

Dr. H. ALAHMAD

March 2023

مدخل Preliminaries

يدرس منهاج بنيان الحواسيب تنظيم وبنية الحاسوب، ويعتبر مدخل لمقرر المعالجات الصغرية ولغة التجميع ويغطي المقرر مواضيع تنظيم وبنيان الحواسيب ويدرس الحاسوب كنظام هرمي من عناصر مختلفة ومكوناتها ومتطلبات مداخلها ومخارجها:

المعالج والذاكرة ووحدات الدخل/الخرج وطرق توصيلها المختلفة (بنية RISC/SISC المسالك أو الباص وتحكيم الباص) ودراسة محددات النظم الحاسوبية (سرعة المعالج وسرعة الذواكر وهرميتها وسرعة نقل البيانات وأسس نظم التشغيل).

توزع المحاضرات (3 ساعات معتمدة).

الامتحان:

تتوزع درجات المقرر على:

اختبار أول واختبار ثانٍ *30* درجة.

أعمال سنة *20* درجات

الكتاب المقرر *Textbooks*:

ليس هنالك كتاب واحد يغطي كل مناحي المقرر وسيتم اعتماد

– Computer Organization and Architecture: Designing for Performance (10th Ed) by
William Stallings, Prentice-Hall.

مراجعة النظم المنطقية

Review Logical Systems

الأسبوع الأول والثاني:

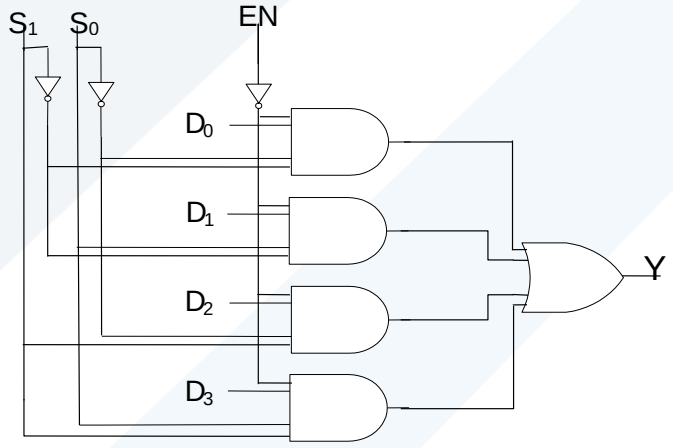
مراجعة الدارات المنطقية بشكل عام والتركيز على الدارات الأكثر استخداماً في بناء مكونات الحاسوب مثل النواخب MULTIPLEXERS ومفككات الترميز DECODERS والقلابات وبشكل خاص القلاب D الذي يعتبر كخانة ذاكرة لتخزين 1bit.

دراسة حاسوب افتراضي انطلاقاً من مرحلة التحليل إلى مرحلة التصميم

التوابع المنطقية التركيبية باستخدام النواخب MULTIPLEXERS:

تتوفر دارات النواخب بأشكالٍ متعددةٍ أهمها ناخب 1 من 4 4 to 1 MUX و ناخب 1 من 8 8 to 1 MUX وناخب 1 من16 16 to 1 MUX.

ويوضح الشكل (1) ناخب 1 من 4، وتظهر به خطوط الانتخاب S0 S1 وخطوط المعطيات D0 D1 D2 D3 وخط التمكين EN الذي عندما يكون فعّالاً.



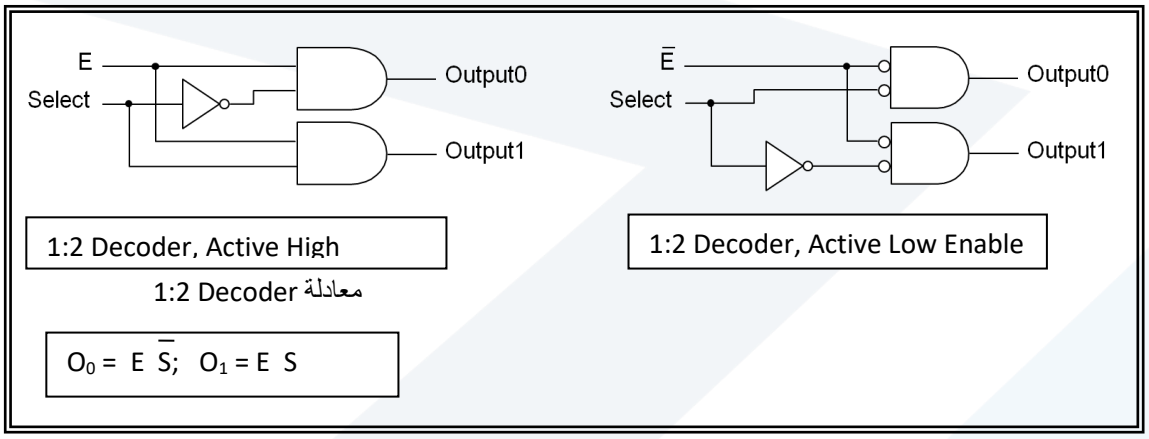
الشكل (1) ناخب 4 إلى 1 (4 TO 1 MULTIPLEXER)

وتستخدم النواخب بشكل خاص لاختيار مسجل من مجموعة مسجلات

التوابع المنطقية التركيبية باستخدام مفكك الترميز DECODER:

تتوفر دارات مفككات الترميز بأشكال متعددة أهمها مفككات 2×4 و 3×8 و 4×16 وتعطى معادلات مفككات الترميز باعتبار خط التمكين E كما يلي:

مفككات



الشكل (2) دارات مفكك الترميز DECODER

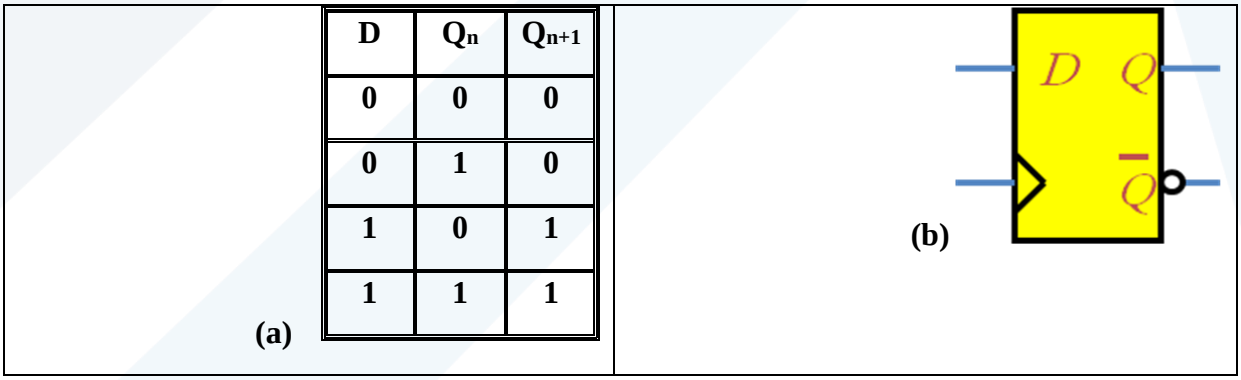
وتستخدم هذه الدوائر بشكل خاص لأجل اختيار أحد أسطر الذاكرة.

القلاّبات:

للقلابات أهمية خاصة في بناء الهيكلية الأساسية للذاكرة وبشكل خاص القلاب D الذي يعتبر كدارة تخزين (ذاكرة) لخانة ثنائية واحدة. يأخذ القلاب الشكل التالي:

القلاب نوع D:

يتميز بأنه يمكننا الحصول فقط على حالتين للخرج توافقان قيمة الدخل المطبقة على المدخل D كما يبين الشكل التالي:



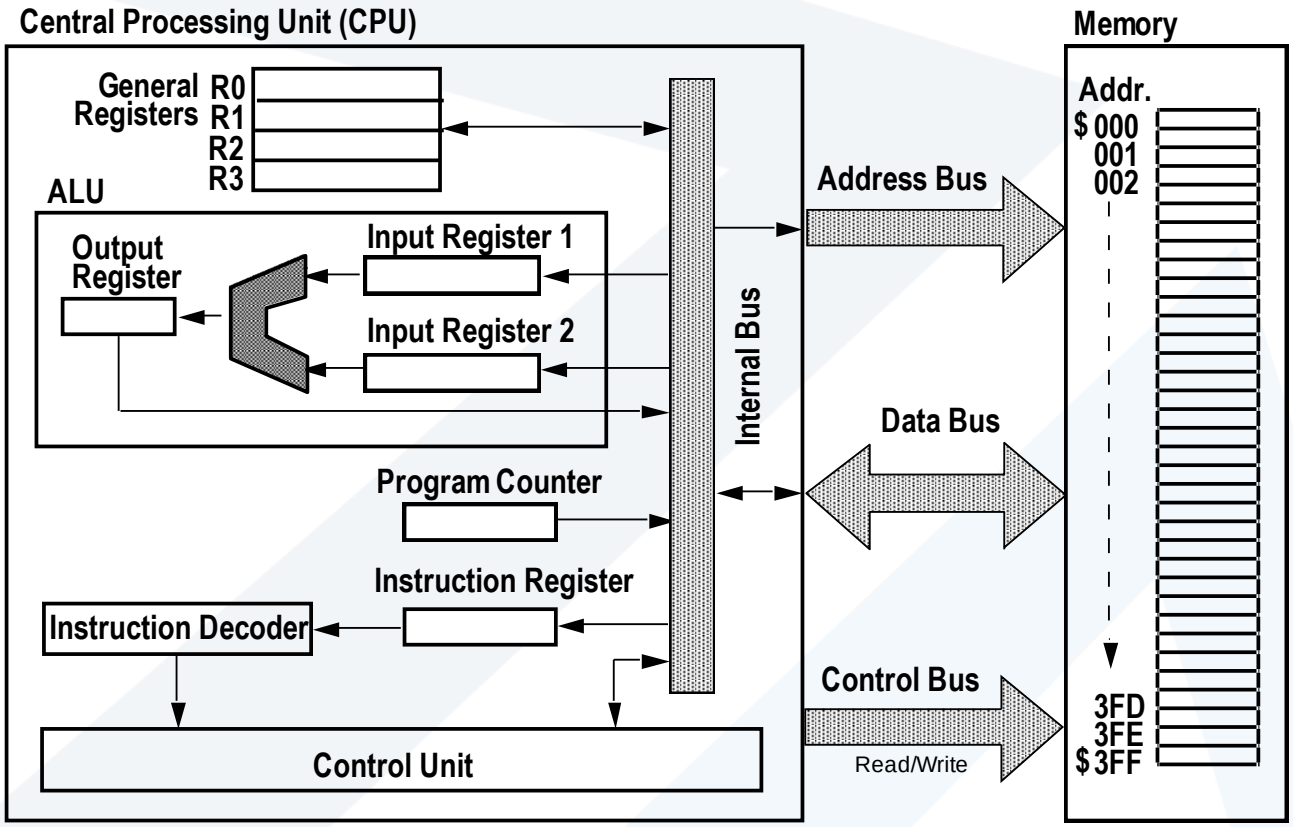
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

الشكل (3) القلاب D

(a) جدول الحالة للقلاب D
(b) رمز القلاب D

الحاسوب الافتراضي

من أجل توضيح فعاليات مكونات الحاسوب نبين فيما يلي بنية افتراضية لحاسوب مبسط يتبع بنية فون نيومان يتمتع بالشكل التالي:



الشكل (4) البنية العامة لحاسوب افتراضي

الأسبوع الثاني

تنظيم الحاسوب والعمليات الأساسية

CPU Organisation & Operation

The Fetch-Execute Cycle

تم تحديد الأسئلة التي يتوجب الإجابة عليها لتتبع عمل الـCPU من خلال مناقشة دورة جلب وتنفيذ التعليمة Fetch-Execute cycle[1].

[1] يطلق عليها أحياناً دورة الجلب-فك الترميز-التنفيذ Fetch-Decode-Execute Cycle.

Fetch-Execute Cycle دورة الجلب والتنفيذ	تطرح العديد من الأسئلة والمفاهيم الهامة مثل:
Fetch the *Instruction* جلب التعليمة	What is an Instruction? Where is the Instruction? Why does it need to be fetched? Isn't it okay where it is? How does the computer keep track of instructions? Where does it put the instruction it has just fetched?
Increment the *Program Counter* مزايدة عداد البرنامج	What is the Program Counter? What does the Program Counter count? Increment by how much? Where does the Program Counter point to after it is incremented?
Decode the Instruction فك ترميز التعليمة	Why does the instruction need to be decoded? How does it get decoded?
Fetch the *Operands* جلب الحد	What are operands? What does it mean to fetch? Is this fetching distinct from the fetching in Step 1 above? Where are the operands? How many are there? Where do we put the operands after we fetch them?
Perform the Operation تنفيذ عملية	Is this the main step? Couldn't the computer simply have done this part? What part of the CPU performs this operation?
Store the results تخزين النتيجة	What results? Where from? Where to?
Repeat forever التكرار الدائم	Repeat what? Repeat from where? Is it really an infinite loop? Why? How do these steps execute any instructions at all?

وهي الأسئلة التي يتوجب علينا الإجابة عنها من أجل أن نوضح سير العمل في الحواسيب.

تمثيل البرامج Representing Programs

يتوجب عند تنفيذ أية مهمة من المهمات الملقاة على عاتق وحدة المعالجة المركزية أن يتم تقسيمه إلى سلسلة من المهام الأبسط وإلى تعليمات لغة آلة. فإذا أخذنا بعين الاعتبار عملية مثل جمع عددين موجودين في الذاكرة[2], في مواقع يشار لها بـ B و C[3] وعلينا وضع النتيجة في موقع نشير له بالرمز A. أي:

$$A = B + C$$

[2] Let's assume they are held in two's complement form.

[3] A, B and C are actually main memory **addresses**, i.e. natural binary numbers.

يمكننا تجزيء المهمة إلى سلسلة من المهمات الأبسط المعبر عنها بتعليمات لغة التجميع **assembly instructions**، كما يلي:

Assembly Instruction	**Effect**
LOAD R2, B	Copy the contents of memory location designated by B into Register 2
ADD R2, C	Add the contents of the memory location designated by C to the contents of Register 2 and put the result back into Register 2
STORE R2, A	Copy the contents of Register 2 into the memory location designated by A.

نحتاج إلى ترميز كل تعليمة ثنائياً لكي تتمكن الـ(CPU) من تنفيذها على بنية افتراضية مبسطة.

البنية الافتراضية المبسطة Hypothetical Architecture

البنية الافتراضية المبسطة هي بنية تخيلية بالمواصفات التالية:

1024 x 16-bit words of RAM maximum. RAM is word-addressable.

4 general purpose registers R0, R1, R2 and R3. Each general purpose register is 16-bits (the same size as a memory location).

16 different instructions that the CPU can decode and execute, e.g. LOAD, STORE, ADD, SUB and so on. These different instructions constitute the **Instruction Set** of the Architecture.

The representation for integers will be two's complement.

نحتاج كمصممين أن نعرّف بنية تعليمة خاصة وسننطلق من مثال للتعبير عن بنية التعليمة لحاسوبنا الافتراضي.

بنية تعليمة الحاسوب الافتراضي Hypothetical Instruction Format

تعليمات الحاسوب الافتراضي هي 16-bits (بحيث تناسب كلمة الذاكرة الرئيسية fit into a main-memory word) وكل تعليمة مقسمة إلى حقول تعليمة ترمز المعلومات المختلفة التي يجب أن تقدم إلى الـCPU.

Field Name	**OPCODE**	**REG**	**ADDRESS**
Field Width	4-bits	2-bits	10-bits

حقل شيفرة التعليمة **OPCODE**[4] يعرّف عملية الـ CPU المطلوبة. ولمّا كان الحاسوب الافتراضي يدعم 16 تعليمة فإنه يحتاج إلى 4 خانات لتمثيل العملية، ونأخذ على سبيل المثال العمليات التالية:

0001 = LOAD 0010 = STORE 0011 = ADD 0100 = SUB

يرمز حقل المسجلات (المؤلف من خانتين)، أربع مسجلات هي:

00 = Register 0 01 = Register 1 10 = Register 2 11 = Register 3

وحقل العنوان يغطي 1024 موقع نحتاج منها إلى 10 خانات لتمثيل العنوان كما يلي:

فإذا خصصنا العناوين 200H، 201H ، و 202H، لـA و B و C يمكننا كتابة الترميز في المثال أعلاه كما يلي:

Assembly Instruction	**Machine Instruction**
LOAD R2, [201H]	0001 10 10 0000 0001
ADD R2, [202H]	0011 10 10 0000 0010
STORE R2, [200H]	0010 10 10 0000 0000

Memory Placement of Program and Data

ولكي ننفذ البرنامج على الحاسوب الافتراضي نحتاج إلى وضع التعليمات والبيانات في الذاكرة الرئيسية. ونضع التعليمات الثلاث أعلاه ابتداءً من العنوان 080H ونضع المتغيرات A و B و C في المواقع ذات العناوين 200H و 201H و 202H على التوالي ويصبح الوضع كما هو مبين:

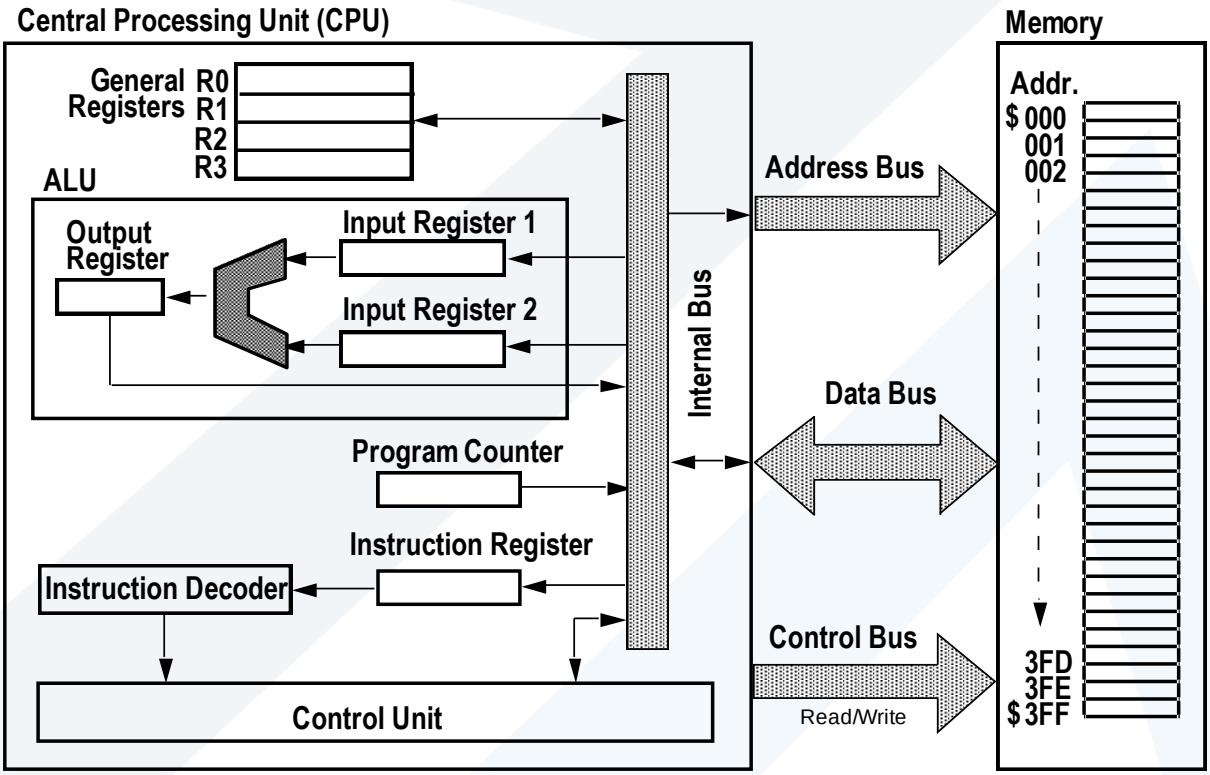
Memory Address in binary & hex	**Machine Instruction** OP Reg Address	**Assembly Instruction**
0000 1000 0000 0 8 0	$\boxed{0001}$ $\boxed{10}$ 10 0000 0001 1 A 0 1	LOAD R2, [201H]
0000 1000 0001 0 8 1	$\boxed{0011}$ $\boxed{10}$ 1000000010 3 A 0 2	ADD R2, [202H]
0000 1000 0010 0 8 2	$\boxed{0010}$ $\boxed{10}$ 10 0000 0000 2 A 0 0	STORE R2, [200H]

[4] Operation Code

Etc			Etc				Etc
0010	0000	0000	0000	0000	0000	0000	A = 0
2	0	0	0	0	0	0	
0010	0000	0001	0000	0000	0000	1001	B = 9
2	0	1	0	0	0	9	
0010	0000	0010	0000	0000	0000	0110	C = 6
2	0	2	0	0	0	6	

ويصبح السؤال الذي يتوجب علينا الإجابة عنه هو كيف ستنفذ CPU الحاسوب الافتراضي هذا البرنامج.

تنظيم وحدة المعالجة المركزية CPU Organisation



عداد البرنامج **Program Counter (PC)**: هو مسجل خاص يحمل التعليمة التالية التي يجب أن تجلب من الذاكرة is a special register that holds the **address** of the next instruction to be fetched from Memory. ومن أجل حاسوبنا الافتراضي فإن الـ PC بعرض 10-bits. وتجري مزايدته بمقدار طول التعليمة مع كل عملية من عمليات الجلب من الذاكرة الرئيسية main memory.

مسجل التعليمة **Instruction Register (IR)**: مسجل خاص يحمل التعليمة بعد جلبها من الذاكرة الرئيسية ومن أجل حاسوبنا الافتراضي فإن IR هو مسجل بعرض 16-bits لأن التعليمة هي بعرض 16-bit.

مفكك ترميز التعليمة **Instruction Decoder** هو من مكونات الـ CPU التي تفك ترميز وتترجم محتوى مسجل التعليمة **Instruction Register (IR)** حيث تجري قسمة التعليمة إلى حقول لتقوم وحدة التحكم Control Unit بترجمتها. ويعتبر مفكك ترميز التعليمة جزءاً من وحدة التحكمControl Unit.

وحدة التحكم **Control Unit**: مجموعة عمل ضمن الـ CPU مسؤولة عن تنظيم جميع فعاليات الـCPU. لوحدة التحكم اتصال مع جميع أجزاء الـCPU، وتضم دارات توقيت معقدة.

وحدة الحساب والمنطق **Arithmetic & Logic Unit (ALU)**: مجموعة عمل ضمن الـ CPU مسؤولة عن تنفيذ جميع العمليات الحسابية والمنطقية من جمع وطرح ومقارنة وربط منطقي بـAND/OR/NOT.

مسجلي الدخل أو المسجلات المؤقتة الخاصة بالـ **Input Registers 1 & 2 ALU** هي مسجلات خاصة تحتفظ بحدود دخل الـALU.

مسجل خرج الـ **Output Register ALU** هو مسجل خاص للاحتفاظ بنتيجة العملية باستكمال عملية الـ ALU، يجري نسخ النتيجة من مسجل خرج الـALU إلى الهدف النهائي (مسجل من مسجلات CPU العامة أو موقع ذاكرة أو جهاز I/O).

مسجلات أغراض عامّة **R0, R1, R2, R3** تحت تصرف المبرمج للاستخدام عند كتابة البرنامج. وبشكل عام يكثر المبرمج من استخدام هذه المسجلات لزيادة سرعة المعالجة. نفترض أن جميع هذه المسجلات ذات سعة واحدة وهي-16 bits.

للباص وظيفة اللاتصال بين مختلف الوحدات لتمرير البيانات وفقاً لعناوين معينة تح سيطرة مجموعة من إشارات التحكم يتألف باص الـCPU الداخلي من (باص عناوين وباص نيانات (معطيات) وباص تحكم the **address bus**, the **data bus**, and the **control bus**) يستخدم باص العناوين لإرسال عناوين الذاكرة الرئيسية من الـ**CPU** إلى الذاكرة الرئيسية عند رغبتها بقراءة بيانات منها أو كتابة بيانات فيها. أما باص البيانات فهو باص ثنائي الاتجاه يحمل البيانات أثناء عمليتي الكتابة أو القراءة.بينما يقوم باص التحكم بالدلالة على رغبة الـCPU بالقراءة أو الكتابة. وللتبسيط فقد اعتمد الحاسوب الافتراضي مسجلين خاصين هما مسجل عناوين لذاكرة**Memory Address Register (MAR)** ومسجل بيانات الذاكرة **Memory Data Register (MDR)**. حيث تخدم هذه المسجلات للتخزين الوسيطي لكل من العناوين والبيانات المتبادلة عبر الباص.

الأسبوع الثالث:

المقارنة بين البنية والتنظيم:

- البنية structure هي مجموعة الصفات التي تقع في نطاق رؤية المبرمج.
  - مجموعة التعليمات عدد الخانات المستخدمة لتمثيل البيانات آليات الإدخال/الإخراج تقنيات العنونة
  - مثل: هل هنالك تعليمة جداء
- التنظيم organization هو الكيفية التي تنفذ بها المميزات.
  - إشارات التحكم، الواجهات البينية، تكنولوجيا الذاكرة.

مثل: هل يوجد وحدة تحقيق جداء داراتية أو أن الجداء ينجز من خلال الجمع المتكرر.

مفهومي البناء structure والوظيفة Function

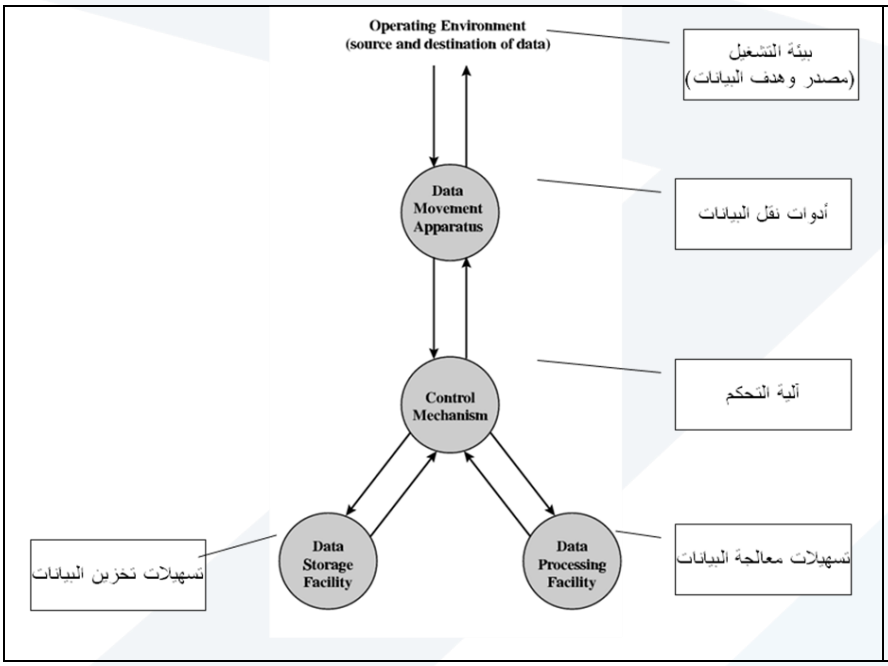
البناء هو الطريقة التي ترتبط بها المكونات مع بعضها البعض.

الوظيفة هي عملية المكوَّنات المنفردة المنظور إلها كجزء من البناء

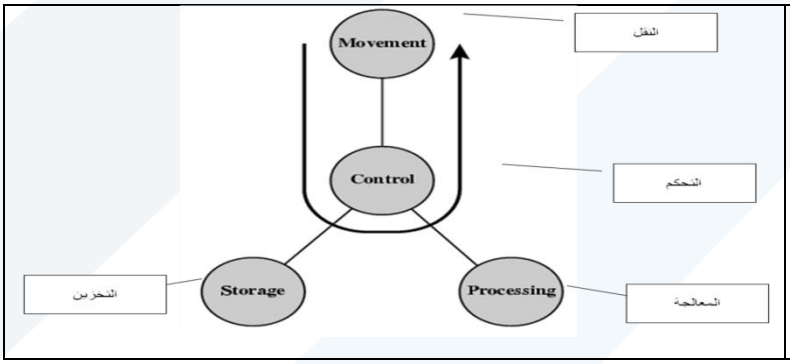
عمليات الحواسيب الرئيسية هي:

- معالجة البيانات
- تخزين البيانات
- نقل البيانات
- التحكم

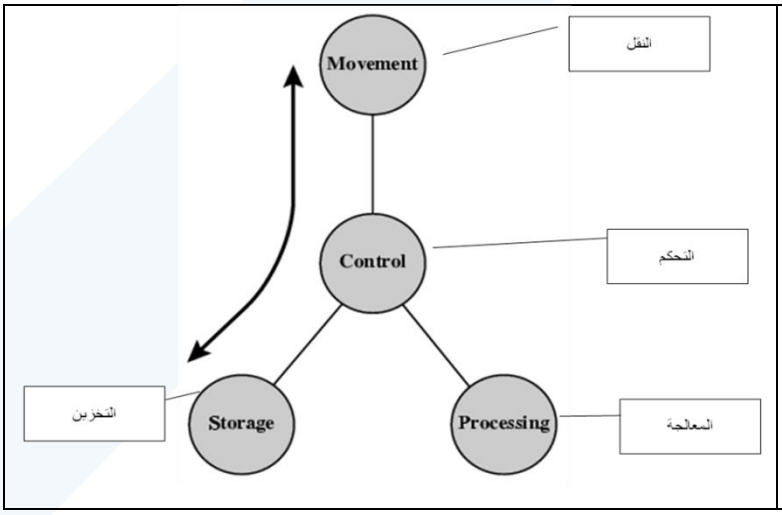
من وجهة النظر الوظائفية
Functional View

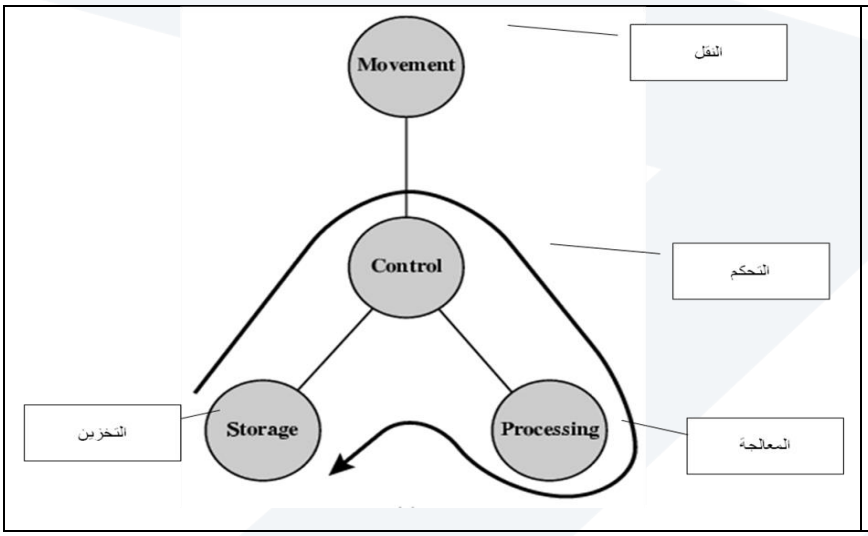


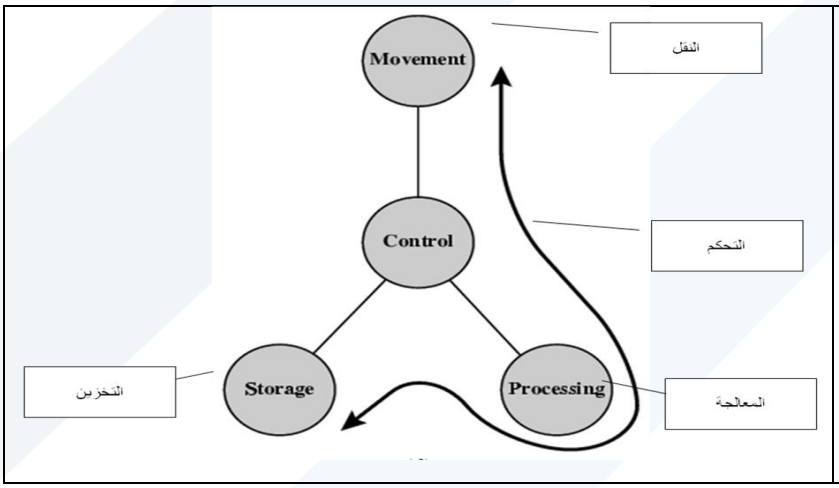
عملية (a) نقل البيانات
Operations (a) Data movement

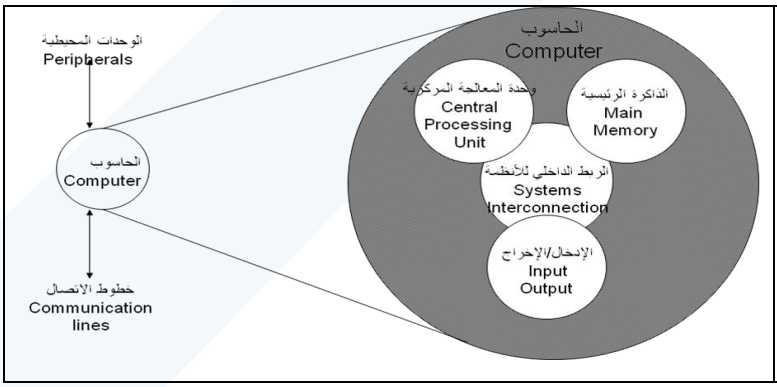


عملية (b) التخزين Operations (b) Storage

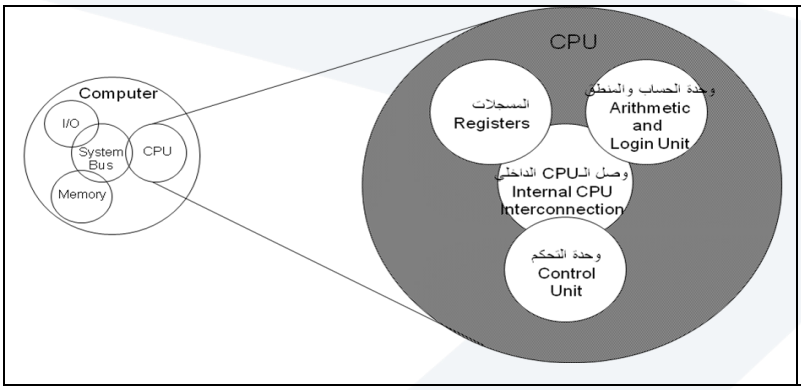


	عملية (c) المعالجة من وإلى التخزين Operation (c) Processing from/to storage

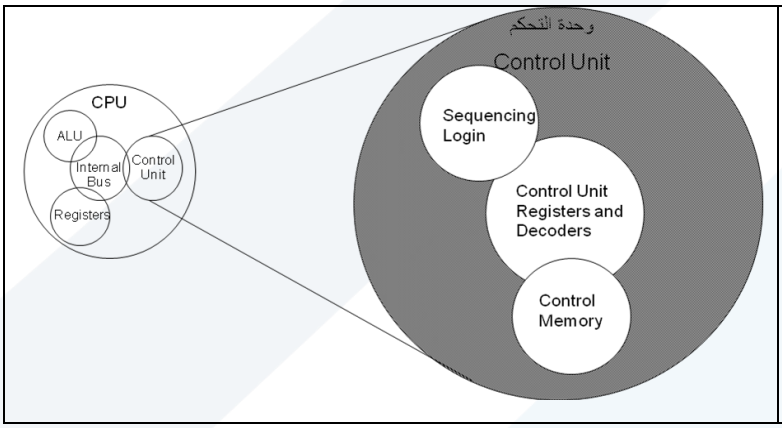
	عملية (d) المعالجة من وإلى الإدخال الإخراج Processing Operation (d) from storage to I/O

	البنيان –المستوي الأعلى - Structure Top Level

البنيان –وحدة المعالجة المركزية
Structure - The CPU



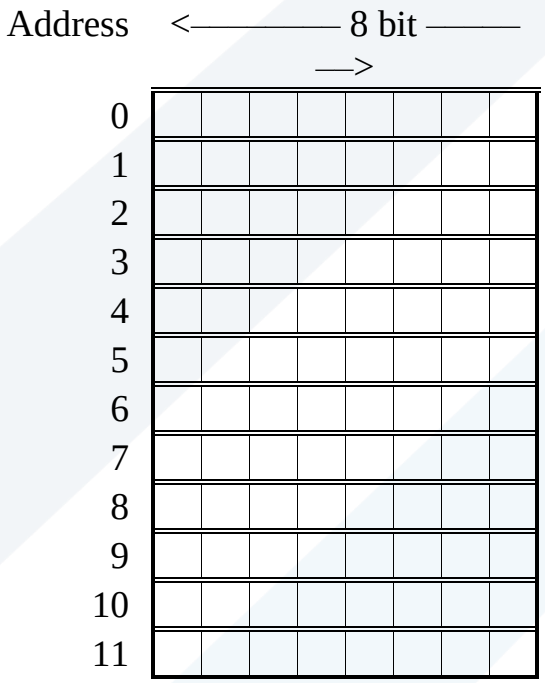
البنيان – وحدة التحكم Structure - The Control Unit

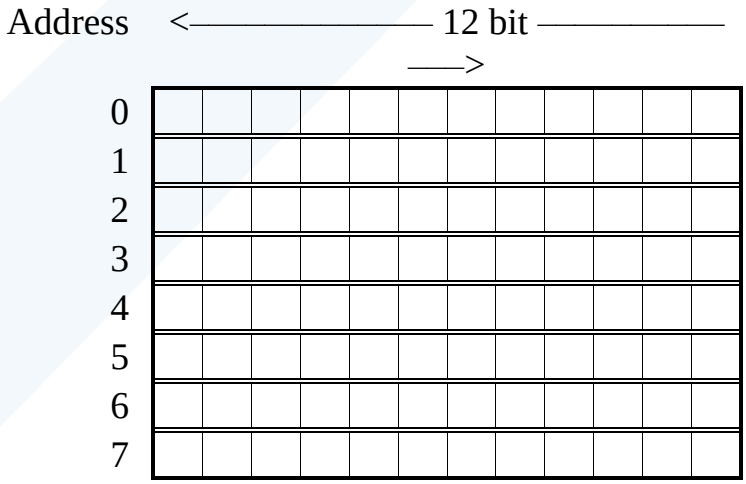


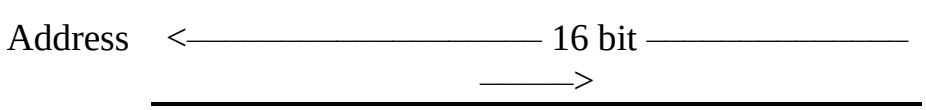
الأسبوع الرابع

تنظيم الذاكرة الرئيسية Main Memory Organisation

ينظر إلى الذاكرة كمصفوفة من الخانات الثنائية. يمثل كل صف موقع من مواقع الذاكرة وتقليدياً يساوي عدد الخانات كلمة من كلمات الذاكرة أو عدد من كلمات الذاكرة مثل (e.g. 2xWordsize) أو جزء من كلمة (e.g. half the wordsize).
وللتبسيط سنفترض أن البيانات في الذاكرة الرئيسية يمكن أن تقرأ أو تكتب في سطر من أسطر الذاكرة الرئيسية في كل عملية.ومن أجل الـ 96-bit memory يمكننا أن ننظم الذاكرة على شكل 12x8 bits أو 8x12 bits أو 6x16 bits أو 96x1 bits أو 1x96 bits. ولكل سطر رقم طبيعي هو العنوان يستخدم لانتخاب سطر الذاكرة .

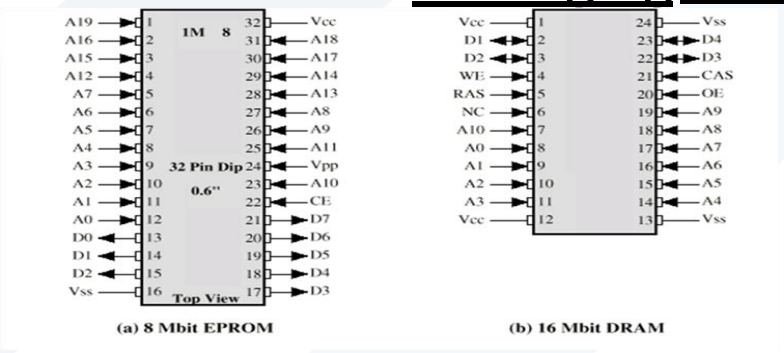






0																
1																
2																
3																
4																
5																

Packaging التغليف



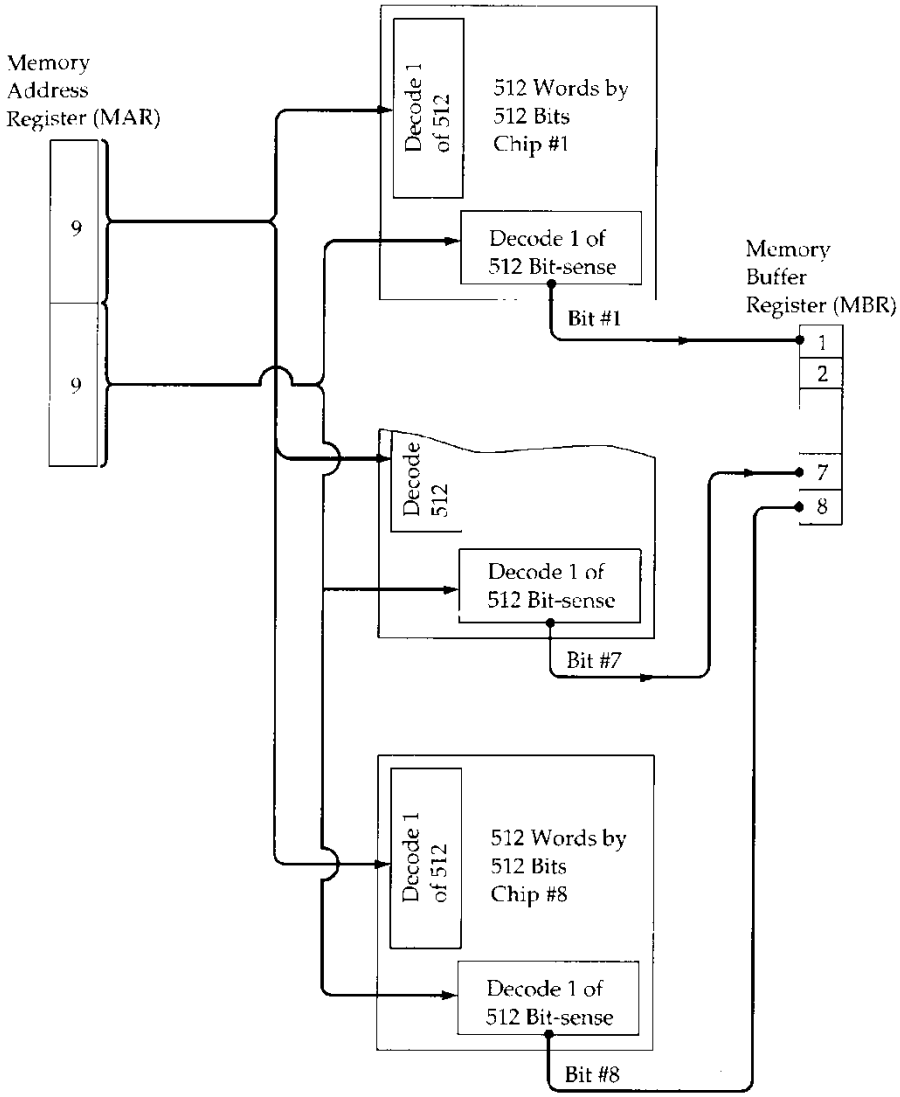
(a) 8 Mbit EPROM (b) 16 Mbit DRAM

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

تنظيم بديل باستخدام نماذج مرجعية كلماتها لـ256K 8 bit

— شريحة 256K 8لكل خانة من الكلمة المرغوبة المؤلفة من 8 bit.

○ تقدم عناوين 18 bit لكل 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 4 نماذج 256K x 8bit

— كيف يمكننا معرفة أي من النماذج الأربعة يحتوي البيانات التي تهمنا؟

— نحتاج 20 خط عنونة لتغطية 1M.

- 18 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.

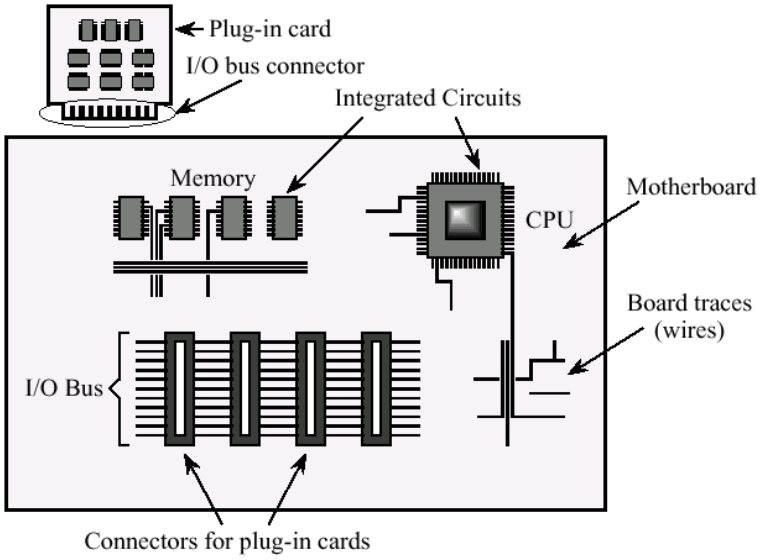
الأسبوع الخامس والسادس

مراجعة دورة التعليمة وبنية الباص العامة

البنية المبسطة للسكة أو الباصِ

Simple Bus Architecture

- **A simplified motherboard of a personal computer (top view):**



اللوحة الأم المبسطة للحاسوب الشخصي (منظورعلوي)

مراجعة البنية مفهوم البرنامج

Architecture Review - Program Concept

- النظم الموصولة داراتياً غير مرنة إذ أنها تحتاج إلى الكثير من الجهد لإعادة الوصل
- يمكن للبنى الداراتية العامة القيام بمهمات عديدة بفرض وجود إشارات تحكم صحيحة من خلال التزويد بمجموعة جديدة من الإشارات التحكمية بدلاً من إعادة الوصل
- وتأخذ الهيكلية البنية التالية

—

ماهو البرنامج

— برمجياً Software

- تتالي من الخطوات A sequence of steps
- في كل خطوة يجري إنجاز عملية حسابية أو منطقية،
- من أجل كل خطوة نحتاج إلى مجموعة جديدة من إشارات التحكم (تعليمة مثلاً)

وظيفة وحدة التحكم Control Unit

— هنالك شيفرة متميزة لكل عملية.

- مثل ADD أو MOVE

— قطاع داراتي hardware segment يستقبل الشيفرة ويصدر إشارات التحكم control signals.

عندها يكون لدينا حاسوب! We have a computer!

المكونات Components

— وحدة المعالجة المركزية Central Processing Unit

- وحدة تحكم Control Unit
- وحدة الحساب والمنطق Arithmetic and Logic Unit

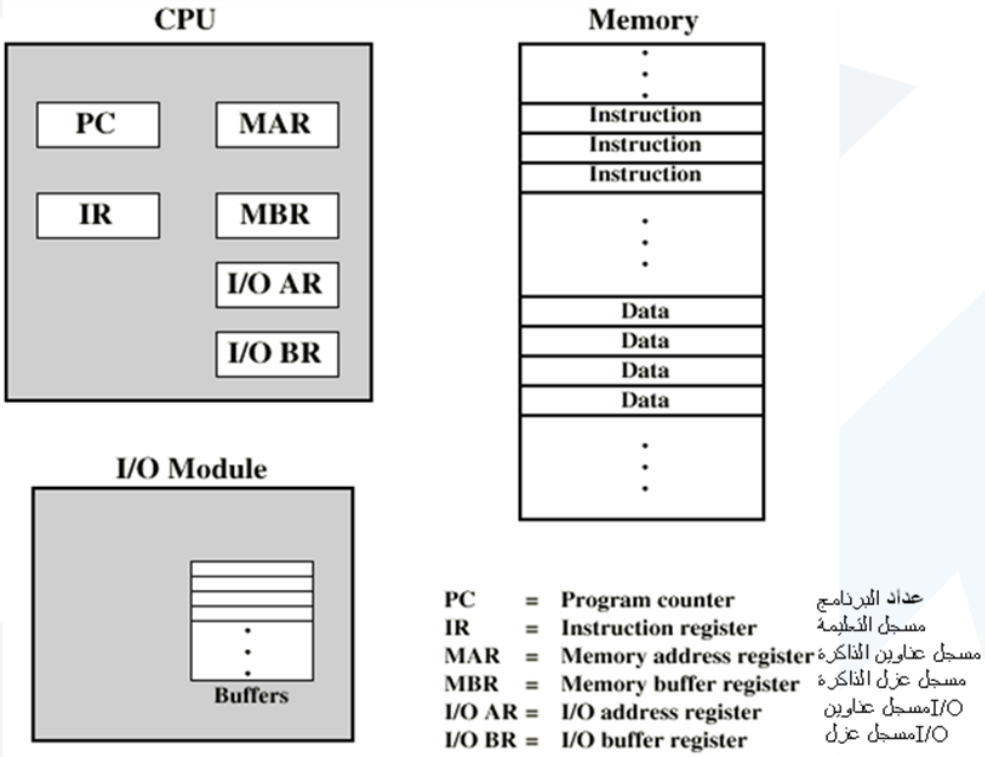
— نحتاج إلى تزويد الـCPU بالتعليمات instructions والبيانات data والحصول على النتائج.

- الإدخال/الإخراج Input/Output

— نحتاج إلى التخزين مؤقت للشيفرة code وللنتائج results.

- الذاكرة الرئيسية Main memory

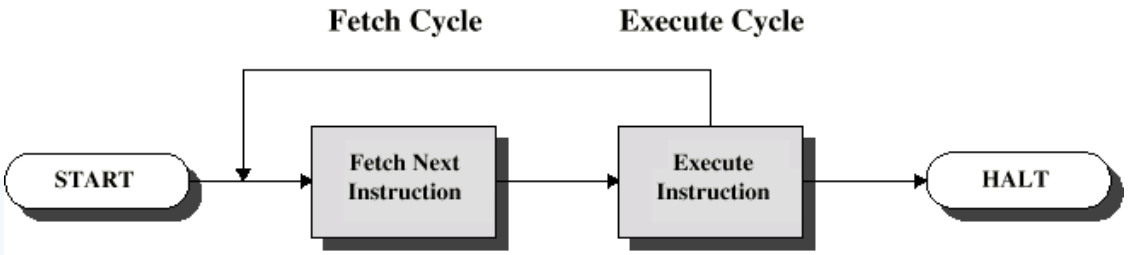
وندرس هذه المكونات من منظور داعم إذ تأخذ بنية الحاسوب الشكل التالي:



ويصبح لابد من دراسة الدورة المبسطة للتعليمة بشكل تفصيلي مع تحديد وظائف كل مرحلة.

دورة التعليمة المبسطة Instruction Cycle

- مرحلتين Two steps
  - الجلب Fetch
  - التنفيذ Execute



دورة الجلب Fetch Cycle

- يحمل عداد البرنامج (PC) بعنوان التعليمة **التالية** للجلب.
- يجلب المعالج التعليمة من الموقع المشار إليه بعداد البرنامج PC.
- مزايدة عداد البرنامج Increment PC
  - مالم يذكر شيء آخر.
- تحميل التعليمة في مسجل التعليمة IR Instruction Register.

يفسر المعالج التعليمة ويقوم بإنجاز الفعالية المطلوبة.

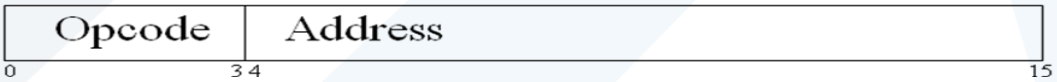
دورة التنفيذ Execute Cycle

- المعالج-الذاكرة Processor-memory
  - تبادل البيانات بين الـ CPU والذاكرة الرئيسية main memory.
- المعالج-I/O Processor I/O
  - تبادل البيانات بين الـ CPU نماذج I/O.
- معالجة البيانات Data processing
  - إنجاز بعض العمليات المنطقية أو الحسابية على البيانات.
- التحكم Control
  - تغيير تتالي التعليمات Alteration of sequence of operations
  - مثل القفز jump

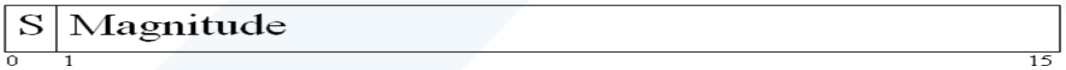
مزيج مما ذكر أعلاه Combination of above

آلية افتراضية Hypothetical Machine

صيغة التعليمة Instruction Format ومجال العنونة Address range



الصيغة الصحيحة Integer Format ومجال البيانات Data range



المسجلات Registers

PC = Program Counter, IR = Instruction Register, AC = Accumulator

قائمة جزئية من شيفرة العمليات Partial List of Opcodes

0001 = Load AC from Memory

0010 = Store AC to Memory

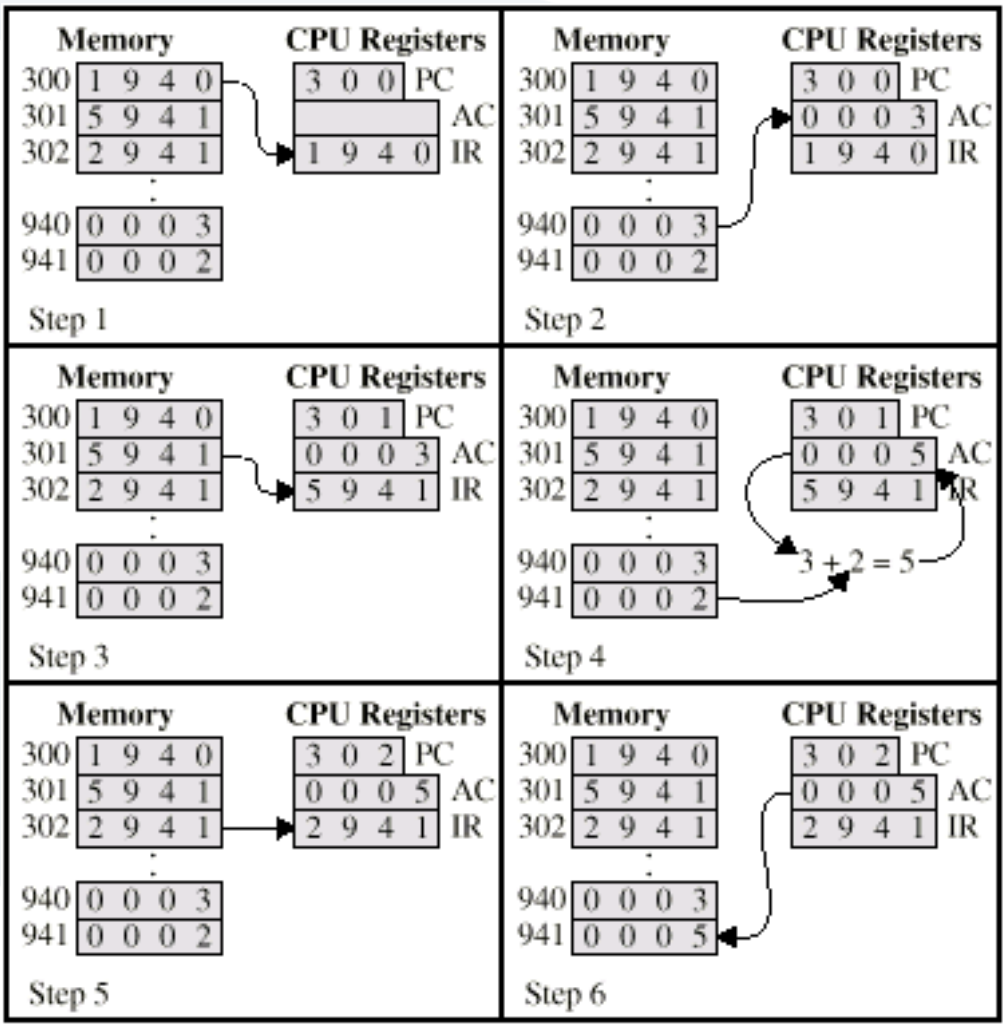
0101 = Add to AC from Memory

يظهر الشكل أدناه

المرحلة الأولى والثانية:تحميل التعليمة الأولى من الذاكرة في IR=1940 وتنفيذ عمليات تحميل المسجل AC بمحتوى موقع الذاكرة (0003) 940 وفي المرحلة الثانية: نتيجة تنفيذ التعليمة الأولى AC=0003.

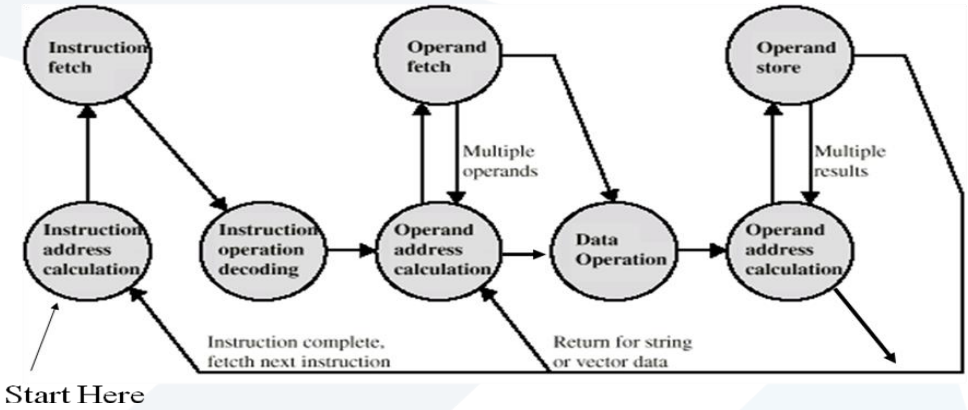
المرحلة الثالثة والرابعة: تحميل التعليمة الثانية في مسجل IR=5941 وتنفيذ جمع AC+[941] أي 2+3 ووضع الناتج في AC.

المرحلةالخامسة والسادسة: تحميل التعليمة الثالثة في مسجل IR=2941 لتحميل محتوى AC في الموقع 941



تعديلات على دورة التعليمة

1. مثال بسيط Simple Example
   - إضافة واحد إلى عداد البرنامج Always added one to PC
   - جلب الحد بشكل كامل من خلال التعليمة Entire operand fetched with instruction
2. أمثلة أكثر تعقيداً More complex examples
   - احتمال الحاجة إلى حسابات عناوين التعليمات أكثر تعقيداً.
     - مثل معالجات 64 bit، بطول تعليمة متغير.
   - قد يأخذ تصميم مجموعة التعليمات بعين الاعتبار زيارة الذاكرة أكثر من مرة لجلب الحد.
     - خصوصاً عندما يزيد مجال عنوان الذاكرة عن طول كلمة البيانات.
   - قد يحتاج تخزين الحد أيضاً إلى زيارات متكررة للذاكرة، مثل حساب الموجهات (الأشعة) Vector calculation

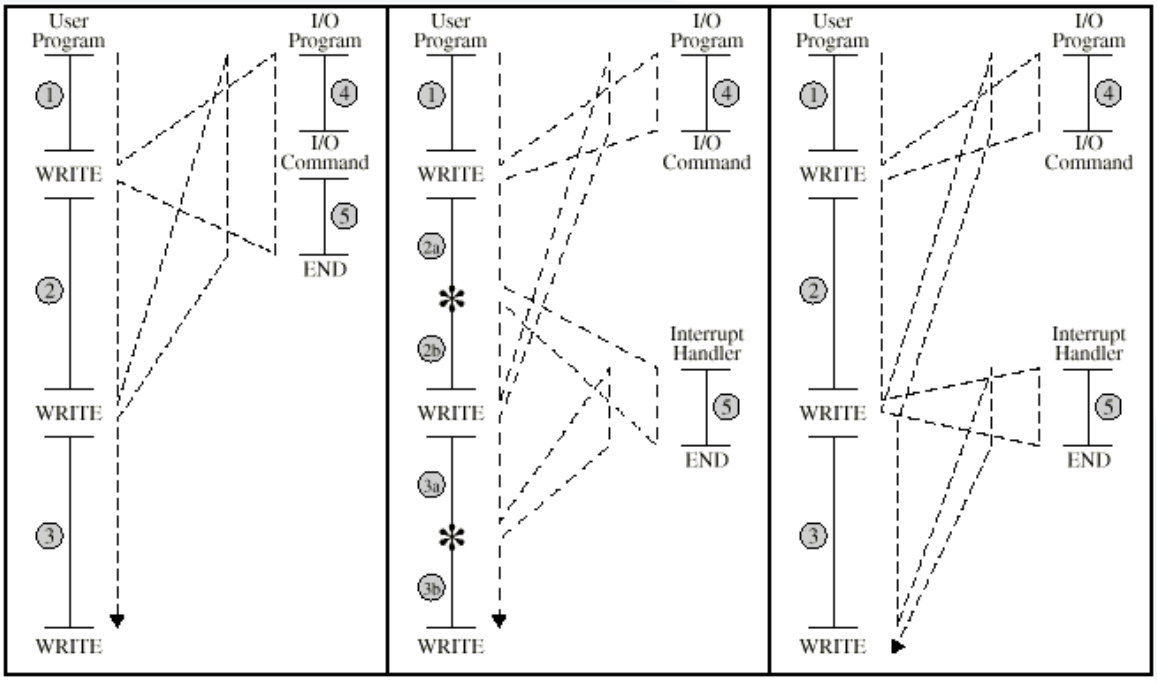


دورة التعليمة- مخطط الحالة
Instruction Cycle - State Diagram

مدخل إلى المقاطعة Introduction to Interrupts

1. سيكون هنالك الكثيرمما سيدرس لاحقاً بما يتعلق بالمقاطعة.
2. المقاطعة هي آلية يستطيع من خلالها نموذج ما أن يقاطع التتالي الطبيعي لعملية أخرى (مثلاً عمليات I/O).
3. هنالك أربع أصناف عامة للمقاطعات:
   - مقاطعة برنامج Program مثل الطفح overflow والتقسيم على صفر.
   - المؤقت Timer
     - تلك التي تتولد عبر موقت المعالج الداخلي.internal processor timer
     - تلك المستخدمة في تعدد المهام القسري pre-emptive multi-tasking
   - مقاطعات متحكم I/O I/O controller
   - عطل داراتي Hardware failure
     - مثل خطأ إنجابية الذاكرة memory parity error
4. تظهر فائدة المقاطعة بشكل خاص عندما يكون أحد النماذج أبطأ من النماذج الأخرى. مثل النفاذ إلى الأقراص (**بالميللي** ثانية) بينما النفاذ إلى الـCPU (**بالميكرو** ثانية).

أمثلة عن المقاطعة

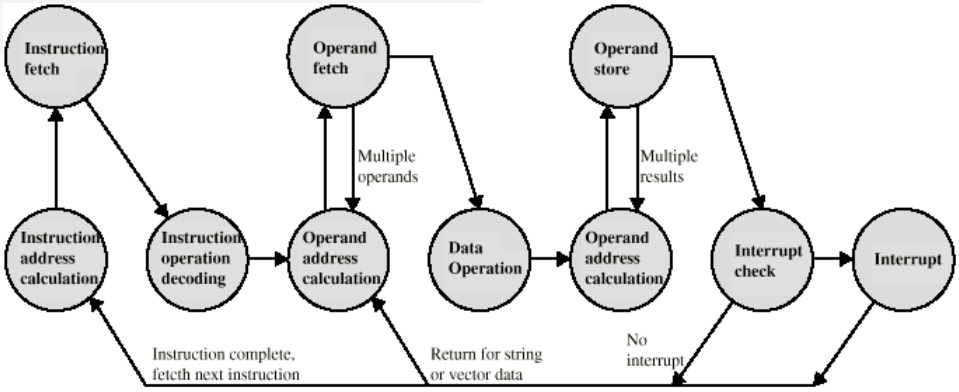


(a) No interrupts (b) Interrupts; short I/O wait (c) Interrupts; long I/O wait

دورة المقاطعة Interrupt Cycle

1. تضاف إلى دورة التعليمة Added to instruction cycle
2. يختبر المعالج وجود طلب مقاطعة Processor checks for interrupt
   - يستدل على الطلب بوجود إشارة مقاطعة Indicated by an interrupt signal
3. لاطلب، جلب التعليمة التالية If no interrupt, fetch next instruction
4. يوجد طلب: If interrupt pending
   - تعليق تنفيذ البرنامج الحالي Suspend execution of current program
   - احفظ السياق (what does this mean?) Save context
   - اضبط عداد البرنامج على عنوان بداية برنامج خدمة المقاطعة Set PC to start address of interrupt handler routine
   - عالج المقاطعة Process interrupt
   - استعد السياق وتابع Restore context and continue interrupted program

مخطط الحالة دورة التعليمة باعتبار المقاطعة

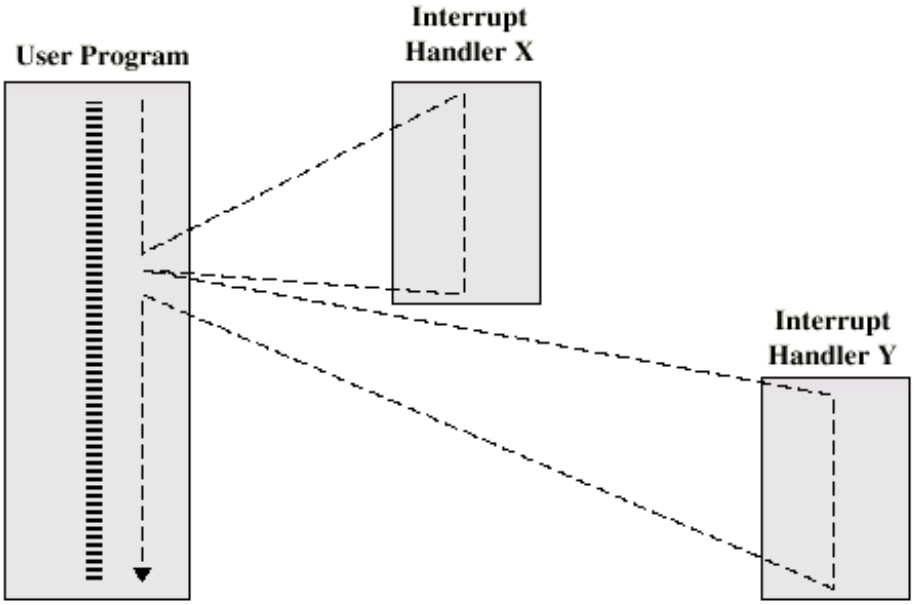


المقاطعات التعددية Multiple Interrupts

1. منع (إلغاء تمكين) المقاطعات Disable interrupts معالجة تسلسلية
   - يتجاهل المعالج المقاطعات الأخرى أثناء معالجته إحدى المقاطعات.
   - تبقى المقاطعات معلّقة ويجري اختبارها بعد الانتهاء من خدمة المقاطعة قيد المعالجة.
   - يجري خدمة المقاطعات بتسلسل الورود.
2. تعريف الأفضليات (الأولويات) المقاطعات المضمنةNested
   - يمكن مقاطعة المقاطعات ذات الأولوية الأدنى من قبل المقاطعات ذات الأولوية الأعلى.

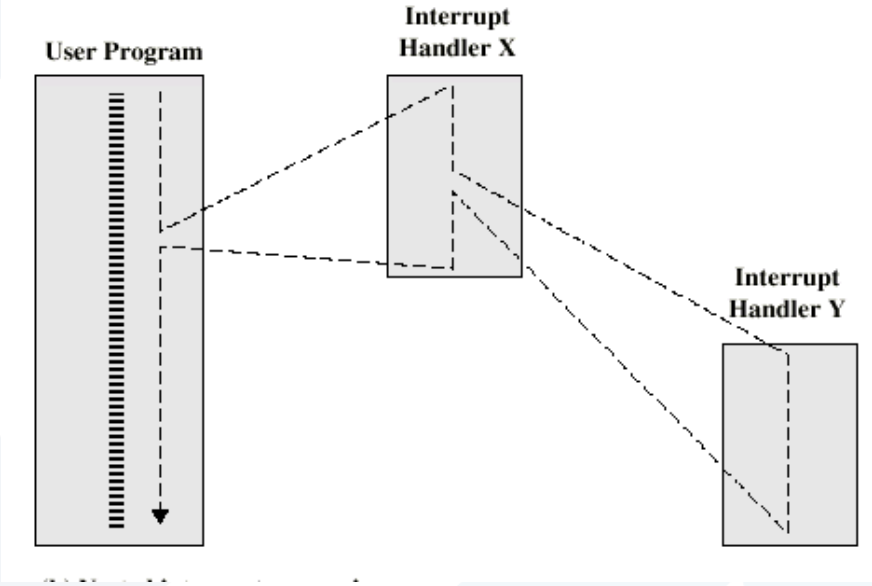
يعود المعالج إلى المقاطعة السابقة عند الانتهاء من خدمة المقاطعة ذات الأولوية الأعلى.

المقاطعات التعددية –تسلسلياً Multiple Interrupts – Sequential



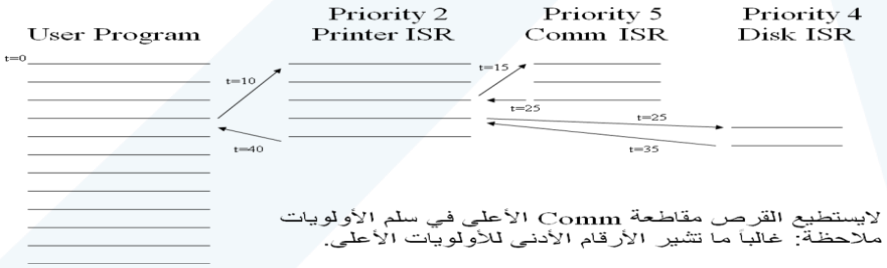
إلغاء تمكين المقاطعات واضح وبسيط Disabled Interrupts – Nice and Simple

المقاطعات التعددية –المضمنةMultiple Interrupts - Nested



كيفية التعامل مع عدد عشوائي من المقاطعات؟

عينة من التتالي الزمني لمجموعة مقاطعات



الوصل Connecting باستخدام الباص

1. يجب وصل جميع هذه الوحدات مع المعالج.
2. يختلف نمط الوصل باختلاف نمط الوحدة.
   - الذاكرة Memory
   - وحدات الـI/O Input/Output
   - وحدة المعالجة المركزية CPU

الوصل مع الذاكرة Memory Connection

- تتألف الذاكرة تقليدياً من N words بطول متساوي معنونة من 0 إلى N-1
- تستلم وترسل البيانات
  - إلى المعالج To Processor
  - إلى أجهزة I/O To I/O Device

- تستقبل عناوين المواقع Receives addresses (of locations)
- تستقبل إشارات التزامن Receives control signals
  - القراءة Read
  - الكتابة Write
  - التوقيت Timing

ربط الإدخال/الإخراج (1) Input/Output Connection(1)

- شبيه وظيفياً وشكلياً بالربط مع الذاكرة من وجهة النظر الداخلية.
- بدلاً من N words للذاكرة يكون لدينا M ports.
- الإخراج Output
  - لاستلام البيانات من الحاسوب Receive data from computer
  - لإرسال البيانات بإلى الوحدات المحيطية Send data to peripheral
- الإدخال Input
  - لاستقبال البيانات من الوحدات المحيطية Receive data from peripheral
  - إرسال البيانات إلى الحاسوب Send data to computer

ربط الإدخال/الإخراج (2) Input/Output Connection(2)

- لاستقبال إشارات التحكم control signals من الحاسوب
- إرسال إشارات التحكم control signals إلى الوحدات المحيطية
  - مثل القرص الدوار e.g. spin disk
- تستقبل العناوين من الحاسوب
  - مثل رقم البوابة المعرفة للوحدة المحيطية
  - ترسل إشارات مقاطعة (تحكم) interrupt signals (control)

ربط الـCPU ---- CPU Connection

- إرسال إشارات التحكم control signals إلى الوحدات الأخرى
- قراءة التعليمات والبيانات instruction and data
- كتابة البيانات (بعد المعالجة)
- استقبال ومعالجة المقاطعات

الباص أو السكة Buses

- هنالك عدد من نظم الوصل الداخلي. ونظام الباص **bus** هو الأكثر انتشاراً
- بنى الباص الأحادي والتعددي هي الأكثر شيوعاً
- مثل باص التحكم/العنوان/البيانات للحاسوب الشخصي PC Control/Address/Data bus (PC)

— الباص الأحادي – Unibus (DEC-PDP) عوضاً عن الباص المتوسط Omnibus.

ماهو الباص؟ What is a Bus?

— معبر اتصال ومرور يربط جهازين أو أكثر.

— عادة يستخدم البث broadcast

- الجميع ينصت، ويجب تقاسم وسط النقل.
- السيد Master -يمكنه القراءة والكتابة حصرياً ولا يوجد إلاّ سيد واحد
- الخادم Slave -بقية الأجهزة ويمكنه مراقبة البيانات ولطن لايمكنه توليدها.

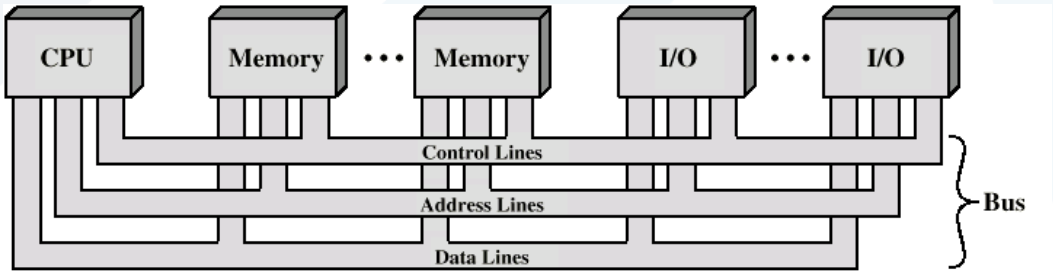
— غالباً يكون مجمعاً Often grouped

- عدد من الأقنية channels في باص واحد.
- مثال 32 bit data bus هو 32 قناة أحادية الخانة منفصلة.

— قد لاتظهر خطوط الطاقة أو التغذية.

— ثلاثة أنواع رئيسية للباص هي: data, address, control.

الشكل العام للربط باستخدام الباص Bus Interconnection Scheme



باص البيانات Data Bus

— يحمل البيانيات Carries data

- مع التذكير بأنه ليس هنالك فرق بين البانات والتعليمات عند هذا المستوي.

— يعتبر عرض الباص هو المحدد الأساسي للأداء performance

- 8, 16, 32, 64 bit
- ماذل لوأن عرض باص البيانات 8 bits بينما عرض التعليمة هو 16 bits?

ماذا لو أن باص البيانات كان 64 bits والتعليمات بطول 16 bits؟

باص العناوين Address bus

— لتحديد مصدر وهدف البيانات.

- تحدد العناوين بشكل عام عنوان أحد مواقع الذاكرة أو عنوان إحدى بوابات I/O.

- تحتاج الـ CPU لقراءة التعليمة (أو البيانات) من موقع محدد في الذاكرة.
- يحدد عرض الباص السعة الأعظمية للذاكرة في النظام.
  - لمعالج 8086 باص بعرض 20 خانة 20 bit address bus مقابل كلمات بعرض 16 bit word size في حيز قابل للعنونة بشكل مباشر بسعة 64k.
  - إلا أنه يستطيع عنونة حتى 1MB باستخدام نموذج الذاكرة المقسم إلى قطاعات segmented memory model
    - RAM: 0 – BFFFF, ROM: C0000 - FFFFF

يسمح نظام مثل DOS فقط باستخدام أول 640K بينما يخصص بقية الذاكرة للـBIOS وللمتحكمات البناء الصلب hardware controllers. ونحتاج لمدير للحيز الأعلى من الذاكرة ليكسر حاجز Needed High-Memory Manager to "break the 640K barrier" .

باص التحكم Control Bus

- معلومات التحكم والتزامن Control and timing information
  - تحدد النماذج التي تستطيع استخدام البيانات وخطوط العناوين.
  - يتوجب على النموذج في حال إرسال البيانات أن (1) يحصل على السماح باستخدام الباص (2) ينقل البيانات التي قد يطلب التعامل معها في نفس الوقت نموذج آخر.
- خطوط التحكم القياسية Typical control lines:
  - Memory read
  - Memory write
  - I/O read
  - I/O write
  - Interrupt request
  - Interrupt ACK
  - Bus Request
  - Bus Grant
  - Clock signals

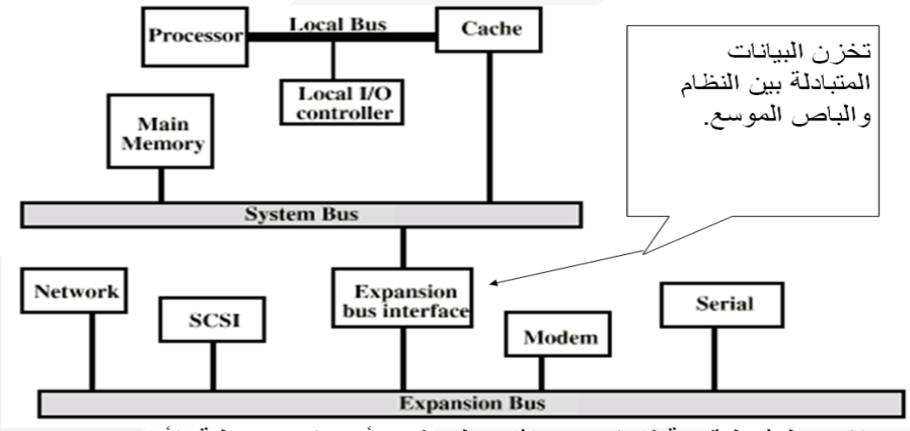
كيف يبدو الباص؟ What do buses look like?

- خطوط متوازية مطبوعة على لوحة الدارة.
- شريط كابلات Ribbon cables
- موصلات على اللوحة الأم Strip connectors.
  - z مثل الـPCI
- مجموعة من الأسلاك Sets of wires

محكومة بالتقارب الفيزيائي والتأخير الزمني وعدد المخارج fan out والتضعيف attenuation وجميعها معاملات مؤثرة ناجمة عن طول الباص.

مشكلة الباص الوحيد Single Bus Problems

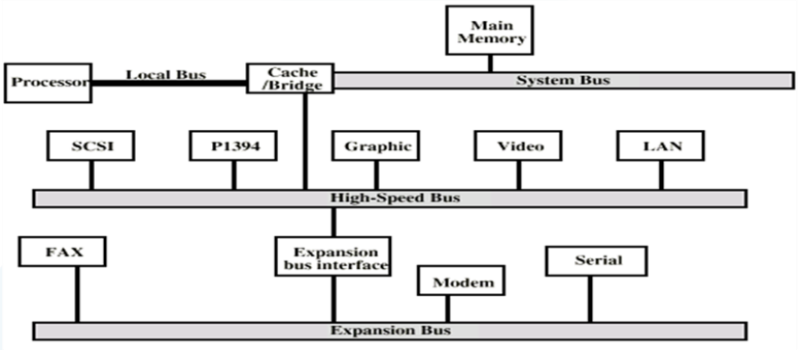
باصات ISA التقليدية (بوجود ذاكرة مخبأية) Traditional (ISA)(with cache)



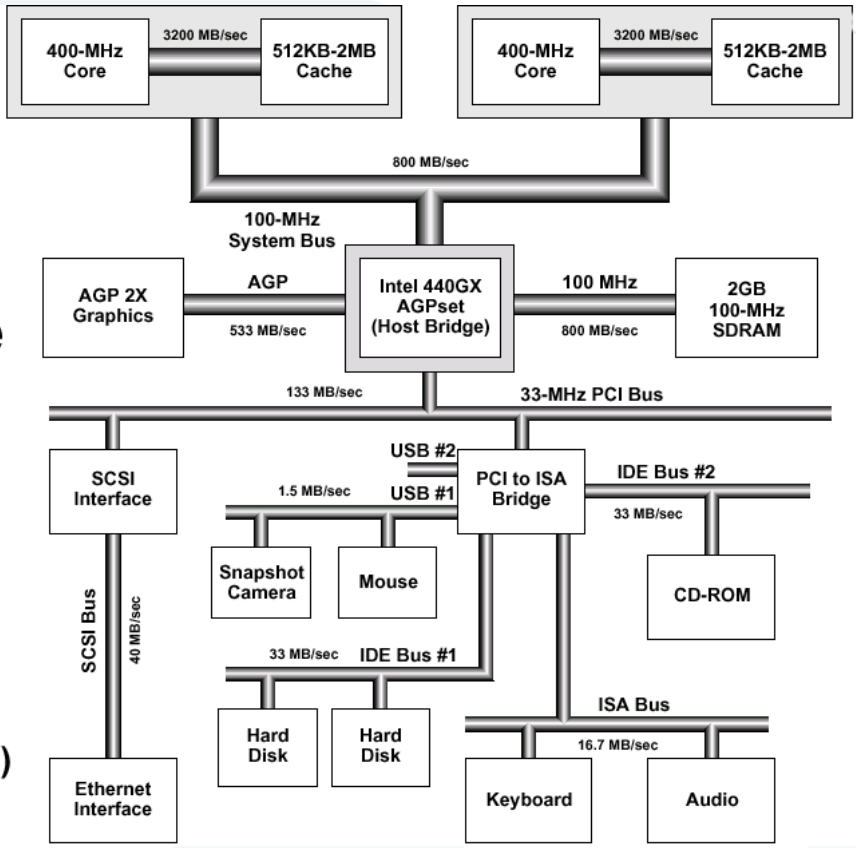
لايعود لهذه المقاربة فائدة ترجى إذا تم الربط مع أجهزة I/O عالية الأداء

باص عالي الأداء بنية الشرفة Mezzanine Architecture

لحنونة أجهزة I/O عالية السرعة بتغيير البنية.



Bridge Based Bus Architecture

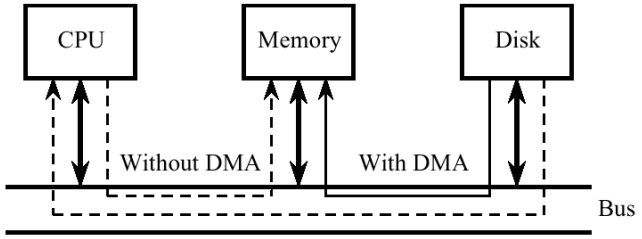


- Bridging with dual Pentium II Xeon processors on Slot 2.

(Source: http://www.intel.com.)

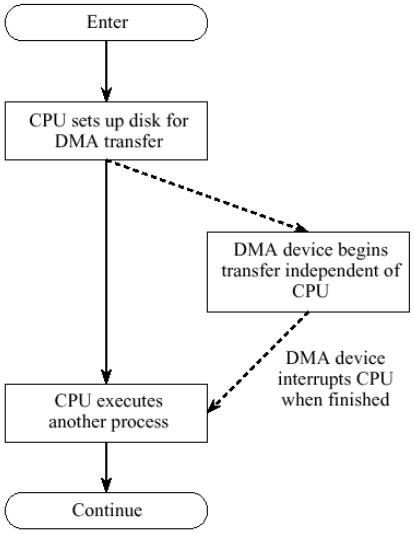
النفاذ المباشر إلى الذاكرة Direct Memory Access

DMA Transfer from Disk to Memory Bypasses the CPU



مخطط تدفقي لتبادل البيانات مع القرص باستخدام DMA

DMA Flowchart for a Disk Transfer



أنواع الباص Bus Types

- مخصص Dedicated
  - خطوط بيانات وعناوين مستقلة Separate data & address lines
- مزيج Multiplexed
  - خطوط مشتركة Shared lines
  - تأخذ بعين الاعتبار خطوط العناوين والبيانات المشتركة.
    - تحتاج إلى خطوط تحكم للمصادقة للبيانات و للعناوين.
    - تقسيم ومزج زمني.
  - الميزات : عدد خطوط أقل.
  - المساوئ Disadvantages
    - تحكم أكثر تعقيداً More complex control

تحديد (تحجيم) للأداء Ultimate performance

تحكيم أوتنظيم الباص Bus Arbitration

- قد يتنافس أكثر من نموذج على التحكم بالباص.
  - مثلاً: قد تحتاج نماذج I/O للتحكم بإرسال البيانات إلى الذاكرة والـCPU.
- إلاّ أنه لايسمح إلاّ لنموذج واحد بالتحكم بالباص في لحظة معينة.
  - يقرر المحكم من الذي سيستخدم الباص.

- يجب أن يكون التحكيم سريعاً وإلاّ فإن أجهزة I/O قد تفقد بياناتها.

يكون التحكيم مركزياً أو موزعاً

التحكيم المركزي Centralized Arbitration

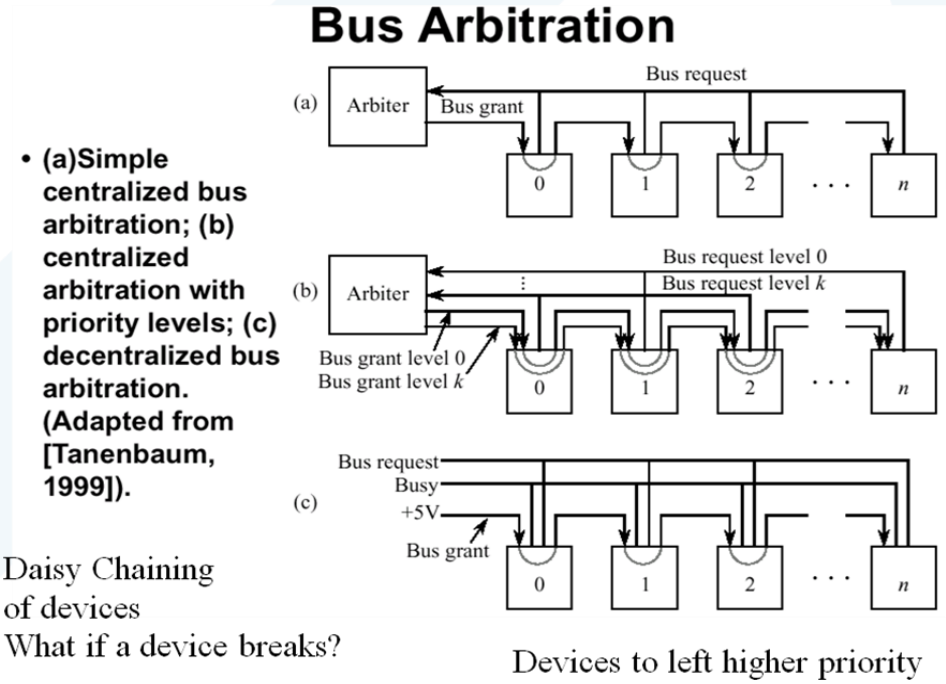
- جهاز واحد مسؤول داراتياً عن تحديد السماحيات للنفاذ إلى الباص.
  - متحكم الباص Bus Controller
  - المحكّم Arbiter

قد يكون جزءاً من الـ CPU أو مستقلاً.

التحكيم الموزع Distributed Arbitration

- لايوجد محكم منفرد No single arbiter
- يقوم كل نموذج بطلب الباص
- تقوم مجموعة منطقية مناسبة بالتحكم بالتوزيع على جميع النماذج التي تتشارك الباص.
- هدف التحكيم المركزي والموزع هو تحديد السيد master.
- مستقبل البيانات المتبادلة هو التابع slave.

هنالك العديد من خوارزميات التحكيم مثل round-robin و الأولويات priority....إلخ.



- **(a)Simple centralized bus arbitration; (b) centralized arbitration with priority levels; (c) decentralized bus arbitration. (Adapted from [Tanenbaum, 1999]).**

تطبيقات تحكيم الباص المركزي

التحكيم المركزي Centralized

- إذا أراد جهازٌ ما الباص يرسل طلباً assert bus request
- يقرر المحكم فيما إذا كان عليه إرسال إشارة قبول الطلب bus grant.

- يعمم قبول الطلب من خلال بث حلقي daisy-chain لجميع الأجهزة.
- إذا أراد جهاز الحصول على خدمات الباص فإنه يستخدمه ولا يمرر إشارة قبول الطلب bus grant إلى بقية الوحدات وبخلاف ذلك فإنه يمرره إلى مايليه من الأجهزة.
- كهربائياً كلما كان الجهاز أقرم من المحكّم كلما كانت له الأفضلية في الحصول على الخدمة.

التحكيم المركزي بمستويات أفضلية مختلفة Centralized with Multiple Priority Levels

من خلال إضافة أكثر من درجة أفضلية وأكثر من إشارة قبول مما يجعل النظام أكثر مرونة. لايقوم المحكم بإرسال إشارة قبول الطلب bus grant إلاّ إلى الخط ذي الأولوية الأعلى.

تطبيقات تحكيم الباص الموزع

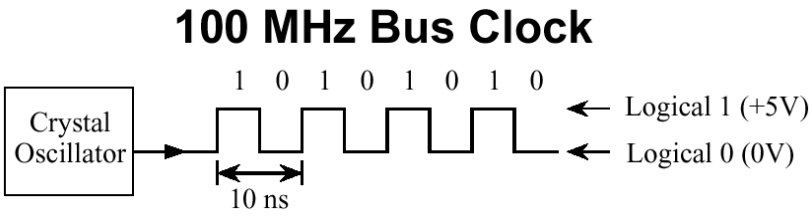
التحكيم الموزع Decentralized

- إذا كنت لاتحتاج الباص مرر إشارة قبول طلب الباص إلى التالي.
- للحصول على خدمات الباص تأكد من عدم مشغولية الباص وأرسل إشارة قبول الباص.
  - إذا كانت إشارة قبول الباص غير فعالة فلن يستطيع الجهاز أن يكون سيداً master ويمرر إشارة القبول إلى التالي.
  - إذا كانت إشارة القبول فعالة أرسل إشارة قبول سلبية إلى التالي.
- فعندما ينجلي الغبار لن يكون سوى جهاز وحيد حاصل على القبول.
- أكّد إشارة المشغولية busy وابدأ بالتراسل.

الجهاز التالي الذي يطلب بعدها االباص سوف يحصل عليه.

التزامن أو التوقيت Timing

- تنظيم ومواقتة الأحداث على الباص.
- المزامنة Synchronous
  - تحدد الأحداث بنبضات الساعة clock signals
  - خطوط نبضات الساعة جزء لايتجزأ من باص التحكم.
  - دورة الباص هي تتالي منفرد للواحد والصفر.
  - جميع الأجهزة تقرأ خطوط الساعة.
  - يحصل التزامن عادةً مع الجبهة الصاعدة.leading edge
  - عادة دورة واحدة للحدث.



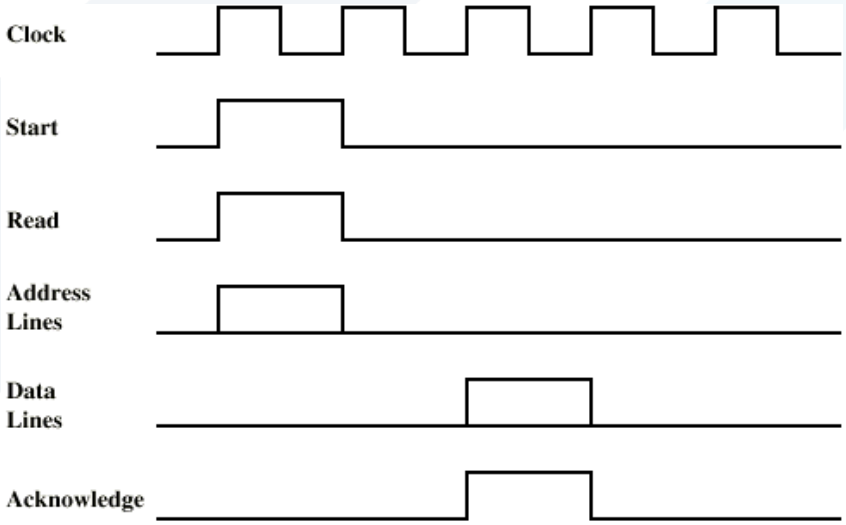
تعني 100 million cycles per second و

1 cycle in (1/100,000,000) seconds = 0.0000001s = 10 ns

في الحقيقة تقترب نبضات الساعة من شكل سن المنشار sawtoothed

مخطط التزامن الزمني

توقيت عملية القراءة Read Operation Timing



مساوئ التزامن Synchronous – Disadvantages

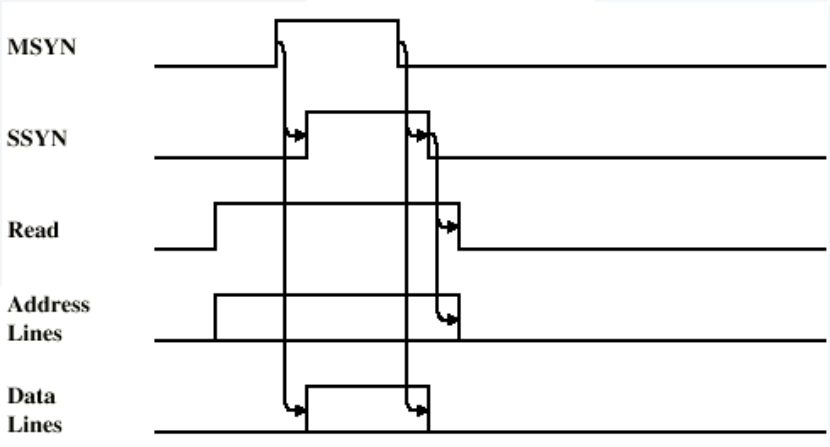
- رغم بساطة نبضات ساعة التواقت فإن لها بعض المساوئ.
  - يتم إنجاز جميع العمليات بعدد صحيح من نبضات الساعة فإذا أنجزنا عملية بـ 3.1 cycles فإنها ستأخذ بالمجمل 4 cycles.
  - بوجود مجموعة غير متجانسة من أجهزة سريعة وأخرى بطيئة فإنه يتوجب انتظار الجهاز الأبطأ
    - لاتستطيع الأجهزة السريعة أن تعمل بطاقتها العظمى فجميع الأجهزة مقيدة بالعمل بنسبة مبضات ساعة ثابتة محددة مسبقاً.
    - فأجهزة الذاكرة الأسرع من 10ns، لن تصل إلى سرعتها النهائية في ظل تردد ساعة من مرتبة 100Mhz clock.
- أحد الحلول هو باستخدام الباص غير المتزامن asynchronous bus

الباص غير المتزامن Asynchronous Bus

- لاتردد ساعة No clock

- يرتبط وقوع حدث على الباص بالأحداث السابقة واللاحقة.
- يحتاج ملاحظة الحالات بدقة وصعب كشف الأخطاء وإصلاحها إلاّ أنه يقدم أداء وإنجازية أعلى.
- يستخدم في عمليات التشبيك networking
  - تظهر المعوقات في انحراف وضياع التزامن.
  - وأحياناً يجري استخدام نظام ساعة خاص مثل شبكة الإثرنت Ethernet.

مخطط التوقيت غير المتزامن Asynchronous Timing Diagram



الامتحان في الأسبوع السادس

الأسبوع الثامن والتاسع

الذاكرة الداخلية

الخصائص Characteristics:

1. الأداء Performance
   - أزمان النفاذ والدورة والتراسل Access, Cycle, Transfer time
2. النمط الفيزيائي Physical type
   - نصف ناقل أومغنطيسي أوضوئي magnetic, optical.
3. الخصائص الفيزيائية
   - متطايرة أوقابلة للمحي Volatile , Erasable
4. التنظيم ونقصد به:
   - الترتيب الفيزيائي للخانات ضمن كلمات Physical arrangement of bits into words
5. الموقع Location
   - ضمن الـCPU، داخلية، خارجية.
6. السعة Capacity
   - حجم الكلمة وعدد الكلمات
7. وحدة التراسل Unit of transfer
   - كلمة على الباص أوكتلة أوعنقود Word on bus, block, cluster
8. طريقة النفاذ Access method
   - مباشر أوعشوائي أوتشاركي أو تسلسلي Direct, Random, Associative, Sequential.

طرق النفاذ Access Methods:

1. تسلسلي أوتتابعي Sequential
   a. Start at the beginning and read through in order
   b. يعتمد زمن النفاذ على موقع البيانات وعلى الموقع السابق.
   c. مثل الشريط المغناطيسي tape
2. مباشر Direct
   a. لكل كتلة عنوان خاص.
   b. النفاذ من خلال القفز إلى منطقة قريبة ومن ثم البحث تسلسلياً
   c. زمن النفاذ يعتمد على الموقع والموقع السابق.
   d. مثل القرص المغنطيسي disk
3. العشوائية Random
   a. تحدد العناوين الإفرادية المواقع بدقة.
   b. زمن النفاذ مستقل عن موقع النفاذ السابق.

c. مثل الـRAM.

4. التشاركية Associative
   a. تتوضع البيانات في موقع تحدده المقارنة مع محتوى جزء من القيمة المخزنة.
   b. زمن النفاذ مستقل عن الموقع والنفاذ السابق.
   c. مثل الذاكرة المخبأية cache.

هيكلية الذاكرة Memory Hierarchy

1. المسجلات Registers
   - مثل تلك التي في الـCPU.
2. الذاكرة الرئيسية أو الداخلية Internal or Main memory
   - قد تضم مستوي أو أكثر من الـcache.
   - الـ"RAM"
3. الذاكرة الخارجية External memory
   - التخزين الإضافي والاحتياطي Backing store
4. الأداء أو الإنجازية Performance
5. زمن النفاذ Access time
   - الزمن الفاصل بين تقديم العنوان والحصول على البيانات المطلوبة
6. 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

7. معدل التراسل Transfer Rate
   - المعدل الذي يمكن أن يمكن أن تنتقل به البيانات.
8. الخصائص الفيزيائية Physical Characteristics
   - الاضمحلال Decay
   - التطاير Volatility
   - قابلية المحي Erasable
   - استهلاك الطاقة Power consumption

الخط أو الحد الأدنى The Bottom Line

1. كم؟ How much?. الجواب: السعة Capacity

2. مالسرعة؟ How fast? الجواب: الزمن كلفة Time is money
3. ما مدى ارتفاع الكلفة؟ How expensive? الجواب: مقايضة بين العوامل التالية: Tradeoffs among all of these

- 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 Example Cache

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.

1. إذا تم إنجاز 95% من عمليات النفاذ في الـcache:
   - $T=(0.95)*(0.1\ \mu s) + (0.05)*(0.1+1\ \mu s) = 0.15\ \mu s$
2. إذا تم إنجاز 5% من عمليات النفاذ في الـcache:
   - $T=(0.05)*(0.1\ \mu s) + (0.95)*(0.1+1\ \mu s) = 1.05\ \mu s$

لذلك يجري السعي لزيادة احتمالية الإصابات cache hits

ذاكرة أنصاف النواقلSemiconductor Memory

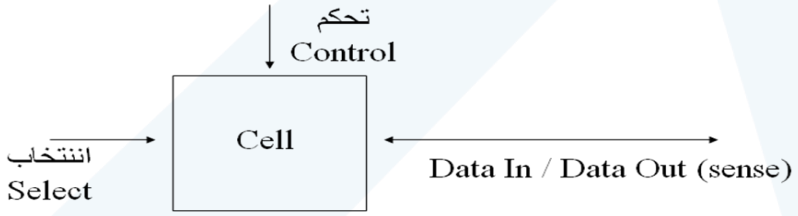
- RAM
  - تسمية خاطئة لأن جميع ذواكر أنصاف النواقل هي ذواكر نفاذ عشوائي random access
  - قراءة/كتابة Read/Write
  - متطايرة Volatile
  - تخزين مؤقت Temporary storage

- 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 (every few milliseconds) refresh circuits
  - 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 الـTypes of ROM
  - ROM الكتابة أثناء التصنيع Written during manufacture
    - مرتفعة الكلفة للكميات الصغيرة Very expensive for small runs
  - PROM القابلة للبرمجة مرّة واحدة Programmable (once)
    - 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).



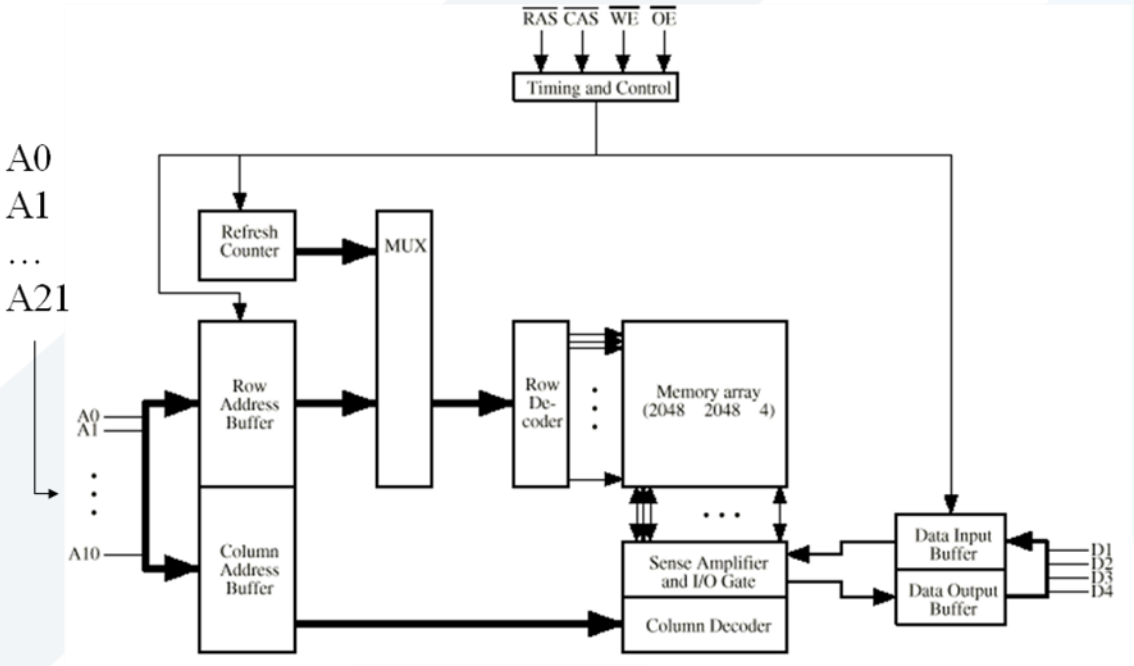
Memory Cell Operations عمليات خلية الذاكرة

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 الأقطاب pinsالمستخدمة في العنونة.
    - تمزج عناوين الأسطر مع الأعمدة.
    - مثال: 11قطب pins لعنونة (2^{11}=2048)، مرورين عبر كل نهاية للحصول على (2^{22} = 4M) 22 bits، 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 4 خطوط من أجل خطوط Data In/Sense..
- إضافة نهاية أو أكثر تضاعف مجال القيم فتزيد السعة 4 مرات مع زيادة الأبعاد

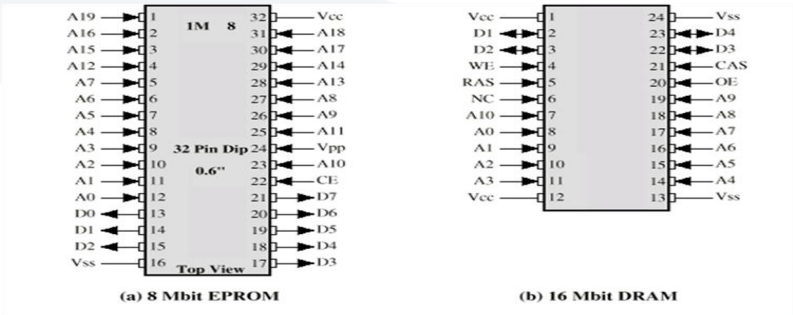
ذاكرة 16 Mb تقليدية DRAM (4M x 4)



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

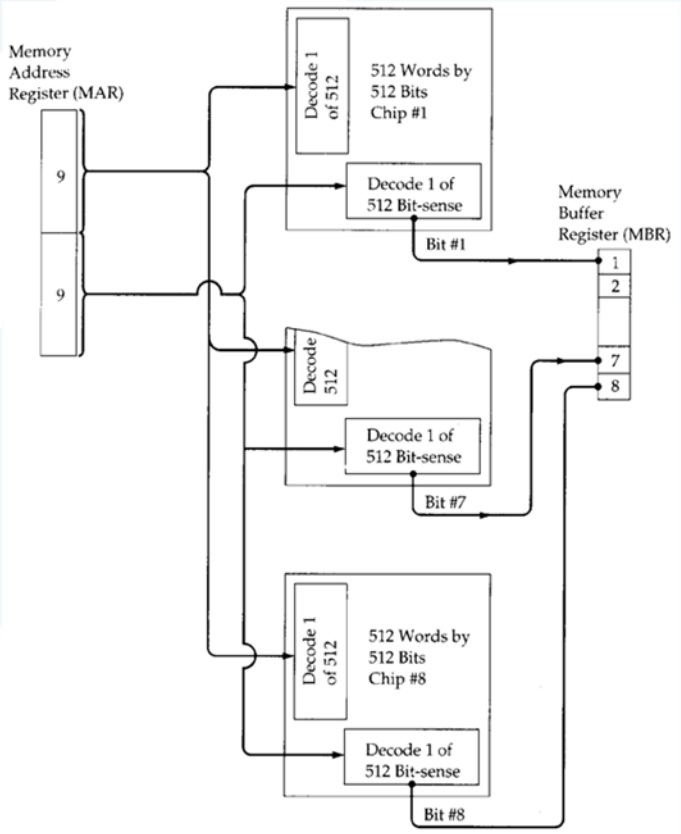
التغليف Packaging



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.

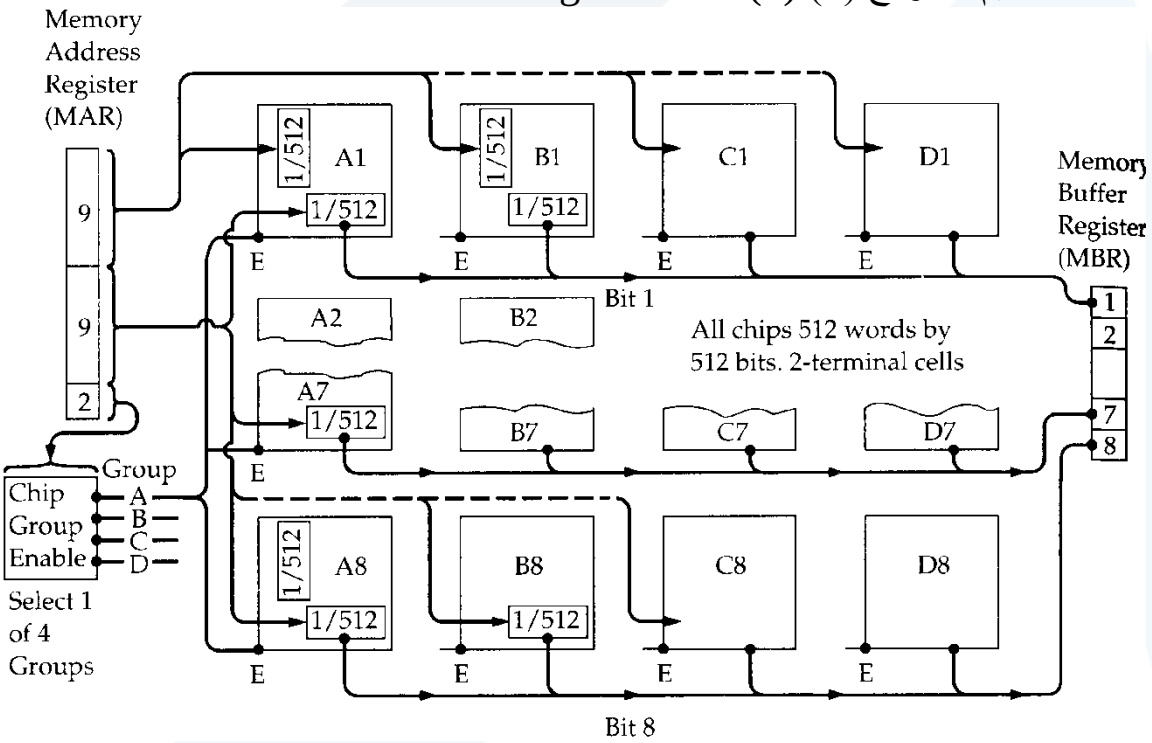


تنظيم النموذج لذاكرة أكبر

- يمكن بدايةً تجميع نماذج موجودة مسبقاً للحصول على ذاكرة أكبر
- مثال ذواكر نظام 256K x 8bit
- إذا أردنا الحصول على ذاكرة 1M يمكننا ربط 4 نماذج 256K x 8bit

- كيف يمكننا معرفة أي من النماذج الأربعة يحتوي البيانات التي تهمنا؟
- نحتاج 20 خط عنونة لتغطية 1M.
  - 18 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.

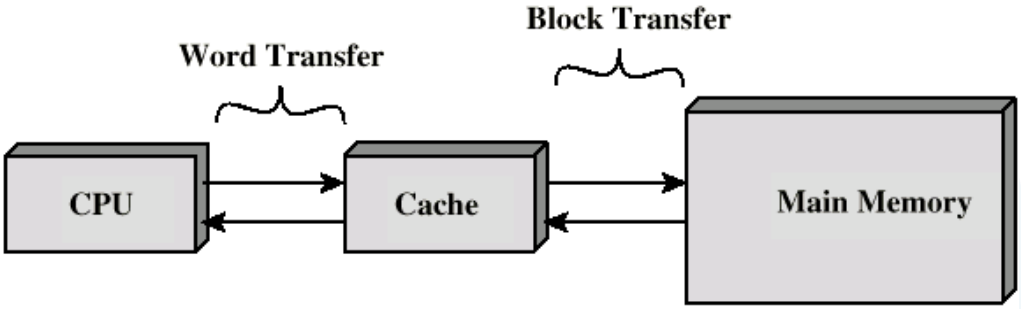
- تنظيم النموذج (2) Module Organization (2)



- 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.

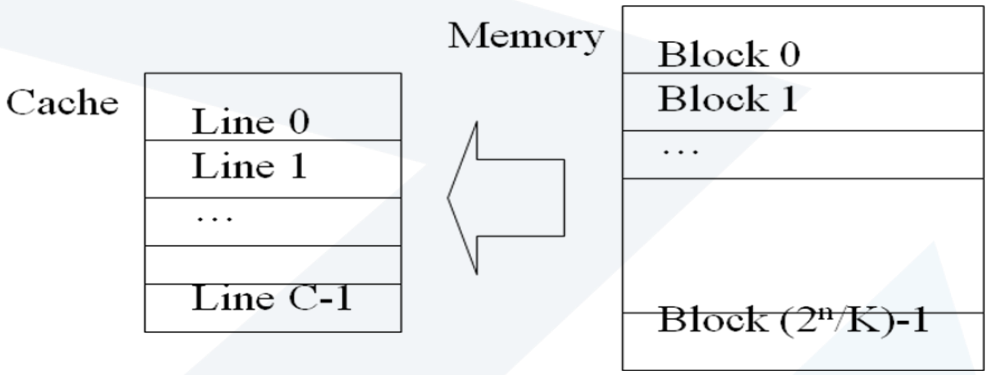


لمحة عامة عن عمليات الذاكرة المخبأية Cache operation - overview

- 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؟



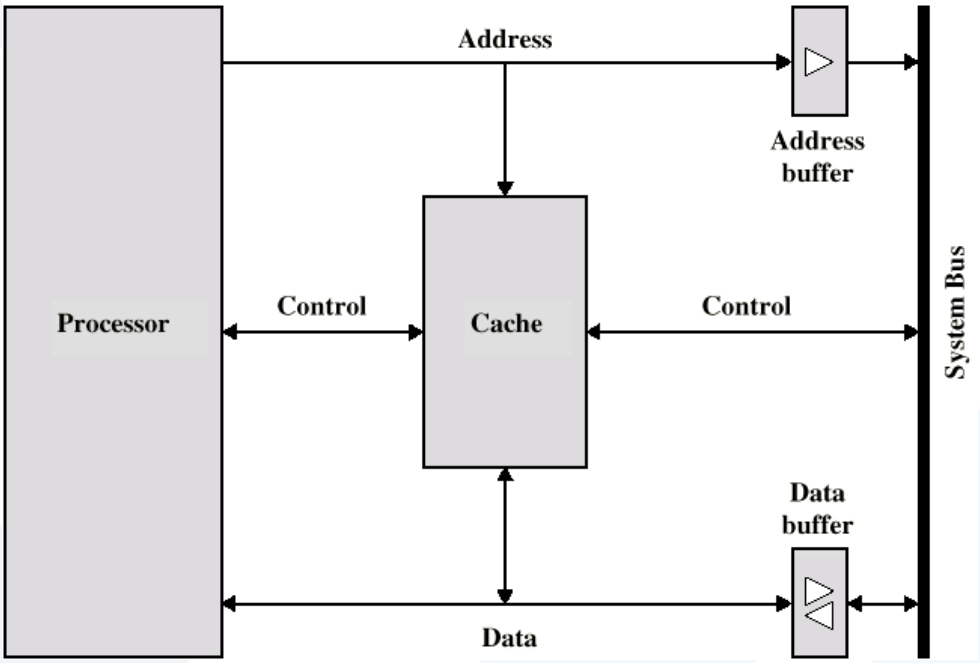
متغيرات تصميم الكاش Cache Design

- 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 الرئيسية 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 في 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

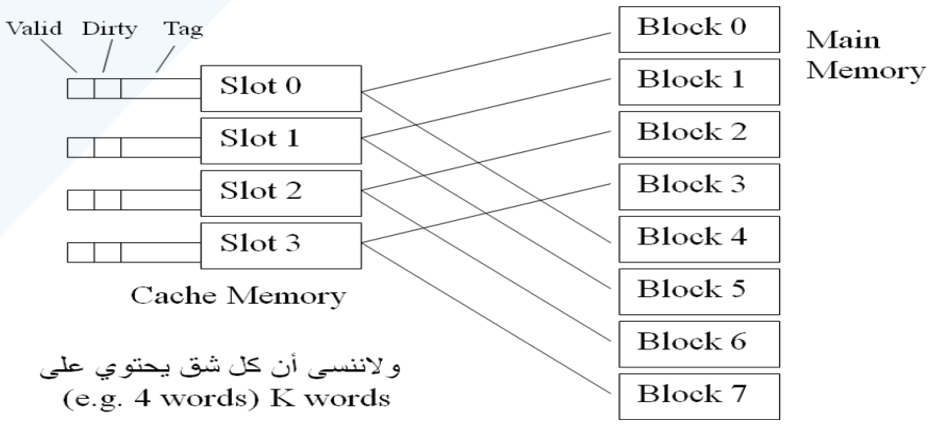
- إذا اقتصرنا في مثالنا على خط كاش بأربع شقوق (كل slot/line/block ليزال يضم 4 words):

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

- وشكلٍ عام:

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

التخطيط المباشر بـC=4 Direct Mapping with C=4



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

بنية عنوان التخطيط المباشر Direct Mapping Address Structure

- العنوان مكون من قسمين Address is in two parts
  - خانات w bits الأقل أهمية التي تعرّف كلمة وحيدة ضمن خط الكاش.
  - خانات s bits الأكثر أهمية التي تعرّف أحد بلوكات الذاكرة.
  - 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 MappingAddress 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 المصادقة Valid 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					

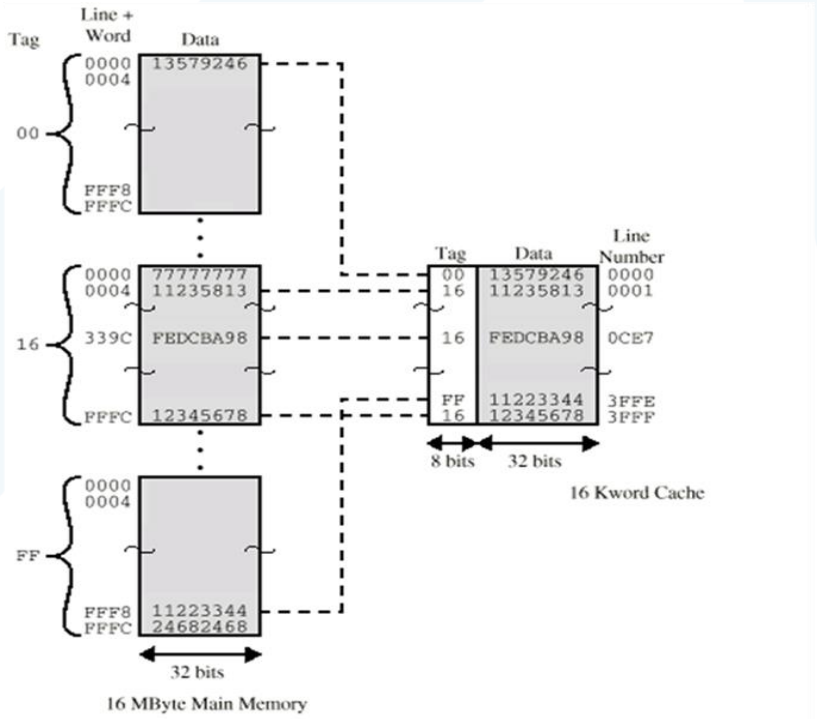
الذاكرة الرئيسية

	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

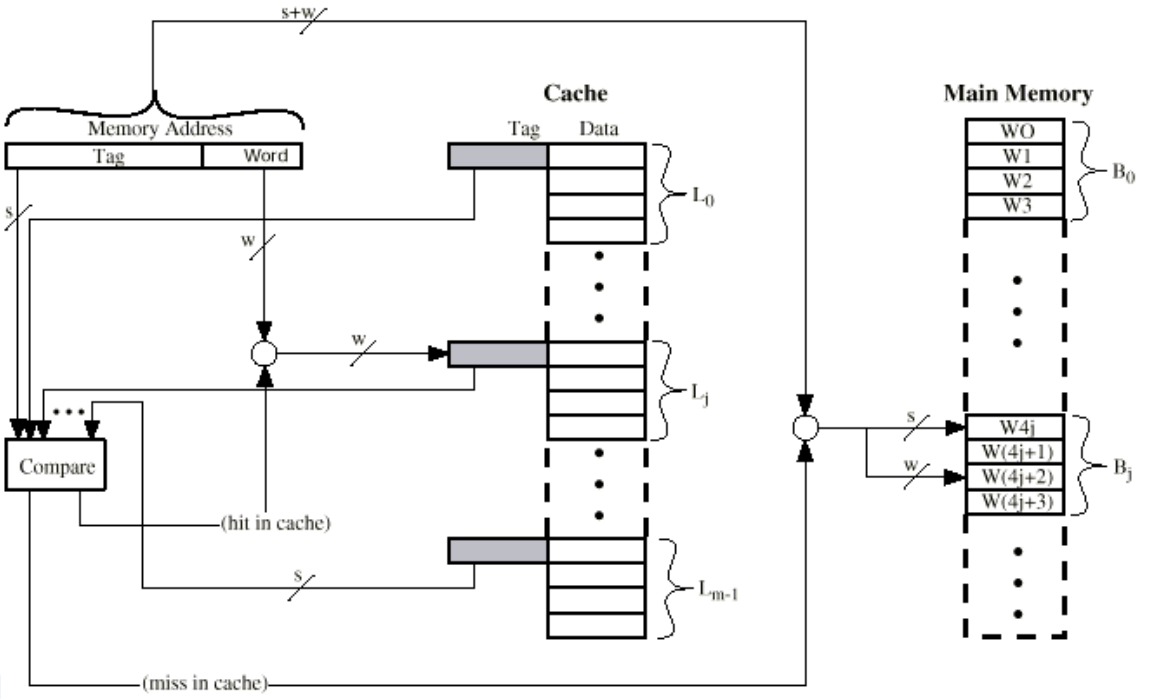
- البساطة Simple
- رخيصة الثمن Inexpensive
- موقع ثابت لبلوك ما Fixed location for given block

- 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 علي به 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)

تنظيم الكاش كامل التشاركية Fully Associative Cache Organization

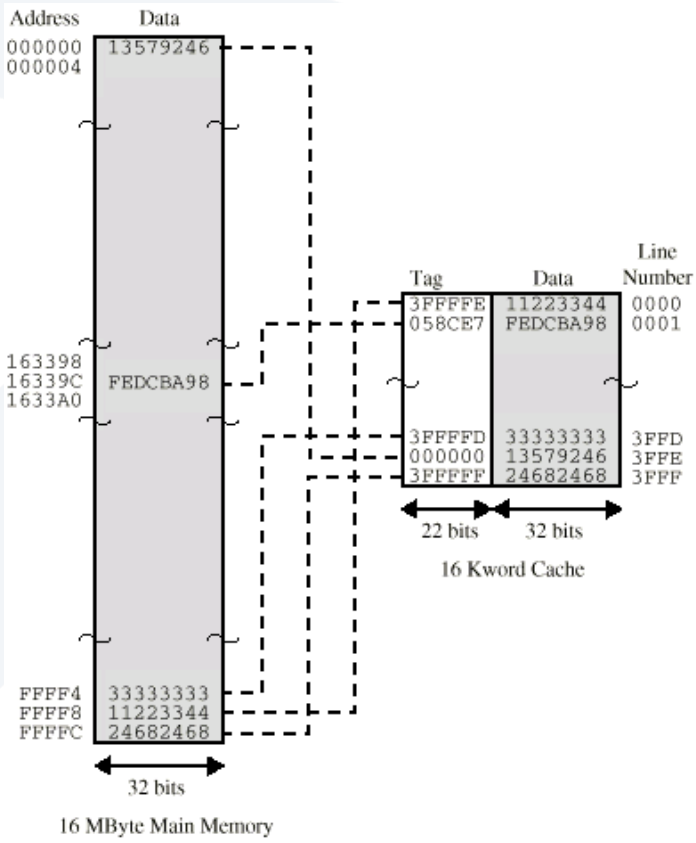


بنية عنوان التخطيط التشاركي Associative Mapping Address Structure

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

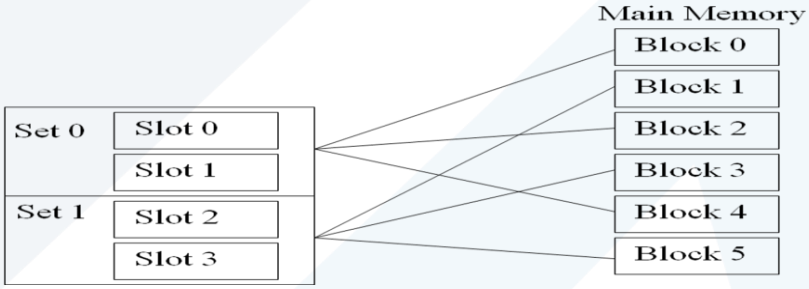
مثال التخطيط التشاركي Associative Mapping Example



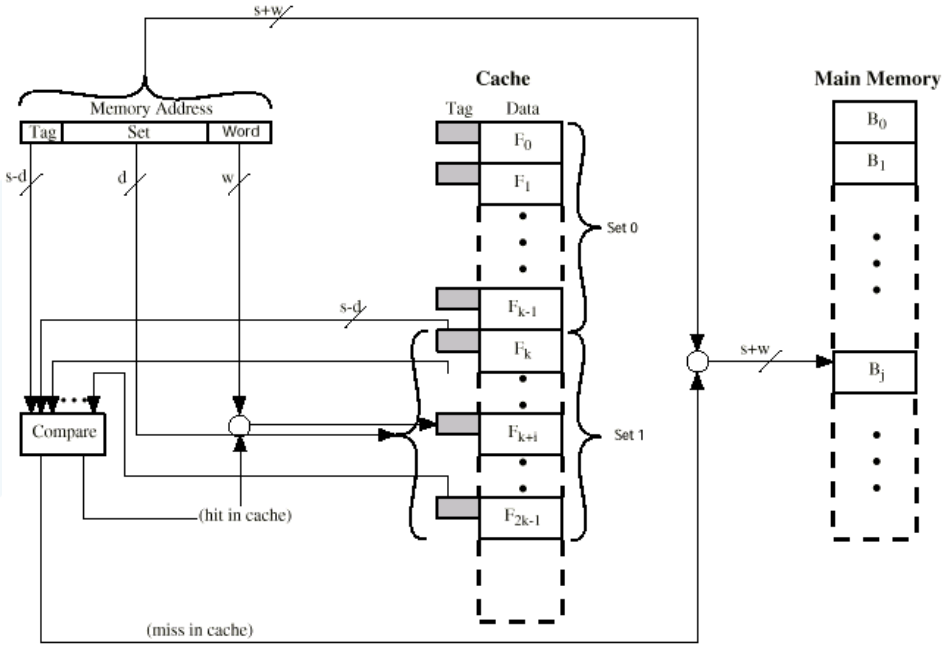
- تخطيط تشارك المجموعة Set Associative Mapping
- حل وسط بين التخطيط التشاركي الكلي و التخطيط المباشر.
  - تقسم الكاش إلى عدد من المجموعات sets.
  - يخطط كل بلوك أي خط في المجموعة المحددة.
  - تحتوي كل مجموعة على عدد من الخطوط.
    - نستخدم التخطيط المباشر لتحديد أي مجموعة من مجموعات الكاش توافق مجموعة 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



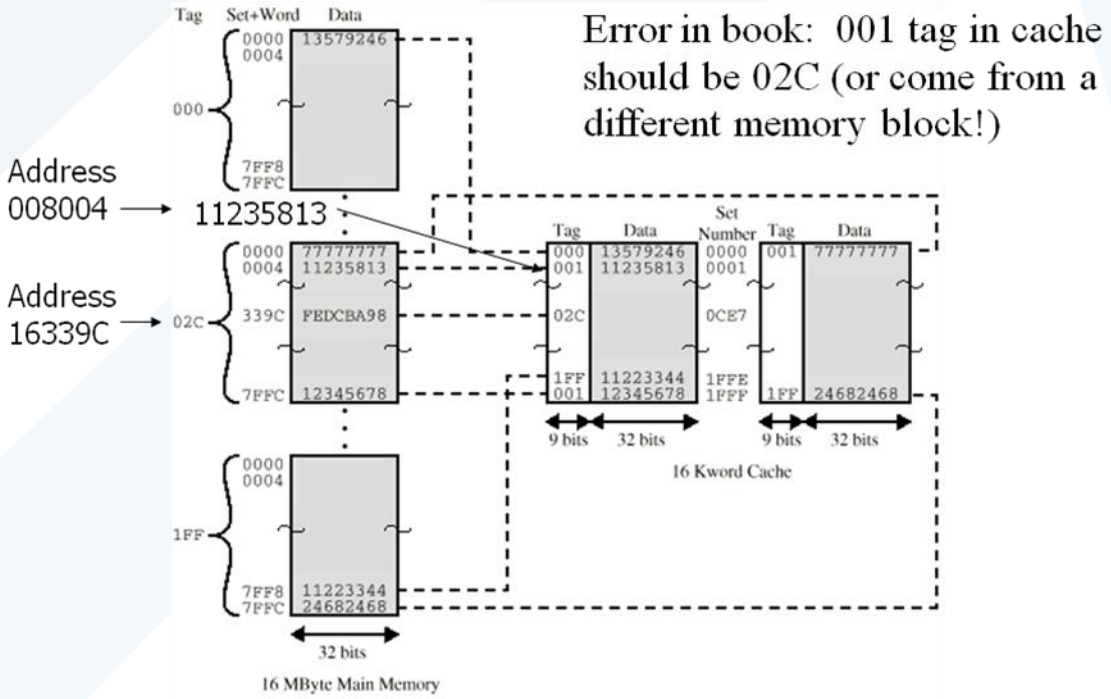
تنظيم الكاش لتشاركية المجموعة باتجاهين
Two Way Set Associative Cache Organization



العنوان في تخطيط المجموعة التشاركي Set Associative Mapping Address Structure

- 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



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 (1) للتخطيط المباشر Direct Replacement Algorithms (1) mapping

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

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

- 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 الذاكرة 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 قداراتي 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**

(Num times referenced words are in cache)

Hit Ratio =----------

(Total number of memory accesses)

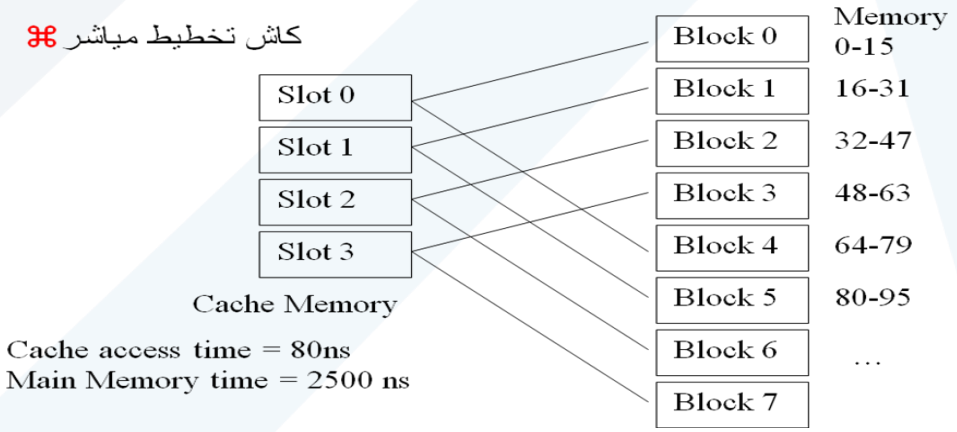
(# hits)(TimePerHit)+(# misses) (TimePerMiss)

Eff. Access Time = ----------

(Total number of memory accesses)

مثال عن أداء أو إنجازية الكاش Cache Performance Example

- كاش تخطيط مباشر



مثال عن أداء أو إنجازية الكاش Cache Performance Example

- بتنفيذنا لبرنامج في المواقع 48-95 لمرة واحدة فإنه ينفذ من 15-31 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.

الأسبوع العاشر والحادي عشر

الذاكرة الخارجية External Memory

الذاكرة الخارجية External Memory

أنواع الذاكرة الخارجية Types of External Memory

1. القرص المغنطيسي Magnetic Disk
1. RAID

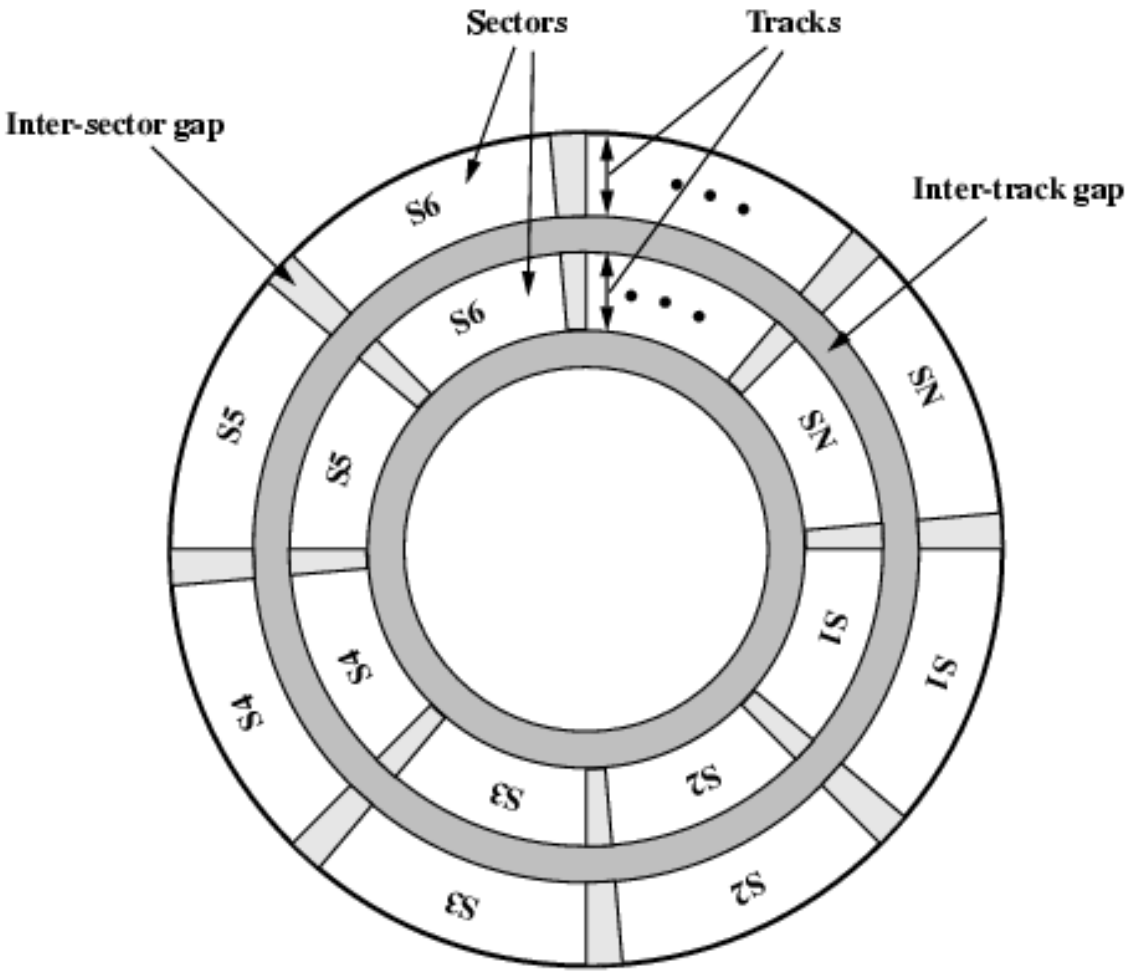
2. القابل للاستبدال Removable
2. الضوئي Optical
1. CD-ROM
2. CD-Writable (WORM)
3. CD-R/W
4. DVD
3. الشريط المغناطيسي Magnetic Tape

1. القرص المغناطيسي Magnetic Disk
   - قرص معدني أو بلاستيكي مكسو بمادة ممغنطة (أكسيد الحديد صدأ)
   - أشكال المنتج Range of packaging
     - القرص اللين Floppy
     - قرص وبنشستر الصلب Winchester hard disk
     - القرص الصلب القابل للإزالة Removable hard disk

تنظيم البيانات والتهيئة
Data Organization and Formatting

- مسالك أو حلقات متحدة المركز Concentric rings or tracks
  - فراغات بين المسالك Gaps between tracks
  - تقليل الفراغات لزيادة السعة Reduce gap to increase capacity
  - نفس عدد الخانات في كل مسلك (اختلاف كثافة التغليف) Same number of bits per track (variable packing density)
  - سرعة زاوية ثابتة Constant angular velocity
- تقسم المسالك إلى قطاعات Tracks divided into sectors
- حجم أصغري للبلوك في قطاع واحد Minimum block size is one sector
- قد يكون هنالك أكثر قطاع لكل بلوك May have more than one sector per block

قرص البياناتDisk Data Layout



رؤوس الأقراص الثابتة والمتحركة Fixed/Movable Head Disk

- الرؤوس الثابتة Fixed head
  - رأس قراءة/كتابة لكل مسلك One read write head per track
  - الرؤوس مركبة على أذرع ثابتة Heads mounted on fixed ridged arm
- الرؤوس المتحركة Movable head
  - رأس قراءة/كتابة لكل جانب One read write head per side
  - مركبة على ذراع متحرك Mounted on a movable arm

التخزين الضوئي Optical Storage CD-ROM

- يستخدم أساساً من أجل الصوتيات Originally for audio
- 650Mbytes تعطينا تسجسل صوتي لأكثر من 70 دقيقة 650Mbytes giving over 70 minutes audio

- مغطاة بالبولي كربون بغطاء عالي الانعكاس (عادة ألمنيوم) Polycarbonate coated with highly reflective coat, usually aluminum
- تخزن البيانات ضمن فرض Data stored as pits
- تقرأ من قبل الليزر المنعكس Read by reflecting laser
- كثافة تغليف ثابتة Constant packing density
- سرعة خطية ثابتة Constant linear velocity

سرعة الـCD-ROM CD-ROM Drive Speeds

- السرعة الصوتية المفردة Audio is single speed
  - سرعة خطية ثابتة Constant linier velocity
  - $1.2\ ms^{-1}$
  - مسار حلزوني بطول 5.27 كم Track (spiral) is 5.27km long
  - يعطينا زمنياً 4391 seconds = 73.2 minutes
- السرعات الأخرى هي سرعات تعددية Other speeds are quoted as multiples
- e.g. 24x
- الإطار المتعارف عليه هو السرعة القصوى للسواقة

الـCD-ROM ماله وماعليه CD-ROM for & against

ماله:

- سعة كبيرة (?) Large capacity
- سهل التصنيع بكميات كبيرة Easy to mass produce
- قابل للإزالة Removable
- متين Robust

ماعليه:

- مرتفع الثمن للتشغيل لفترة صغيرة Expensive for small runs
- بطيء Slow
- قراءة فقط Read only

وسائط التخزين الضوئي الأخرى Other Optical Storage

- الـCD القابل للكتابة CD-Writable
  - حلزوني WORM
  - كلفة مقبولة Now affordable
  - متوافق مع سواقات CD-ROM Compatible with CD-ROM drives
- CD-RW
  - قابل للمحي Erasable
  - أرخص Getting cheaper
  - متوافق مع معظم سواقات CD-ROM. Mostly CD-ROM drive compatible

ماذا يعني اسم DVD - what's in a name? DVD

- قرص فيديوي رقمي Digital Video Disk
  - يستخدم للأفلام Used to indicate a player for movies
    —يشغل أقراص فيديوية فقط Only plays video disks
- قرص متعدد الأغراض Digital Versatile Disk
  - يستخدم للدلالة على السواقة Used to indicate a computer drive
    —يقرأ القرص ويشغل الأقراص الفيديوية Will read computer disks and play video disks

تقنية الـDVD – technology DVD

- متعدد الطبقات Multi-layer
- سعة عالية جداً Very high capacity (4.7G per layer)
- يستوعب فلم طويل على قرص وحيد Full length movie on single disk
  - باستخدام ضغط MPEG. Using MPEG compression
- تم تقييسها مؤخراً Finally standardized (honest!)
- تحمل الأفلام الترميز المحلي Movies carry regional coding
- المشغلات تشغل الأفلام الصحيحة Players only play correct region films
- لايمكن إصلاحها Can be "fixed"

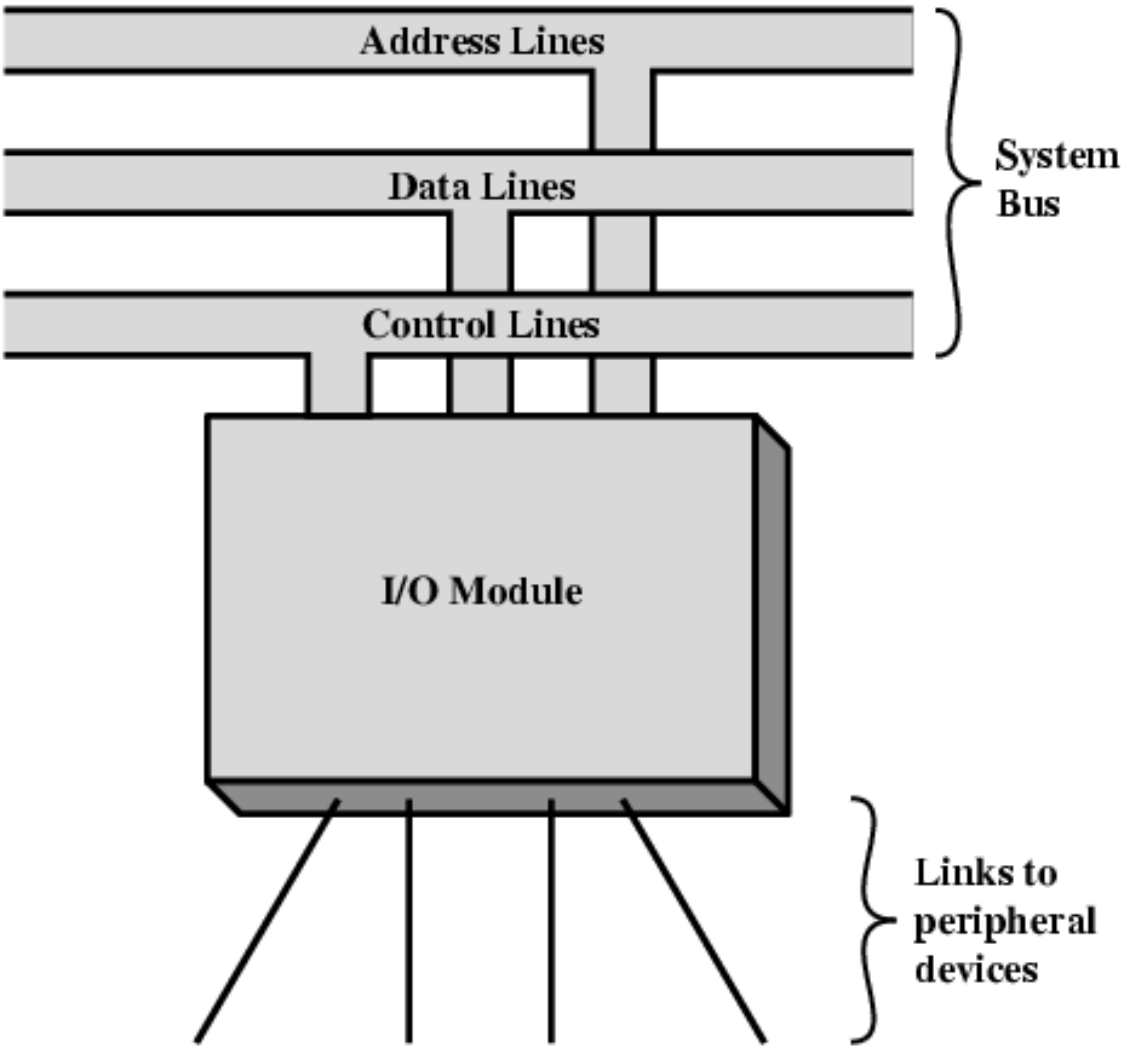
الإدخال/الإخراج Input/Output

مشاكل الإدخال/الإخراج Input/Output Problems

- التنوع الكبير لوحدات I/O Wide variety of peripherals
  - تباين حجم البيانات النتبادل Delivering different amounts of data
  - اختلاف سرعة النقل At different speeds
  - اختلاف صيغة البينات In different formats
- أبطأ من الـCPU والـRAM All slower than CPU and RAM.
- تظهر الحاجة لنماذج I/O. Need I/O modules

نموذج I/O Input/Output Module

نموذج I/O الأساسي Generic Model of I/O Module



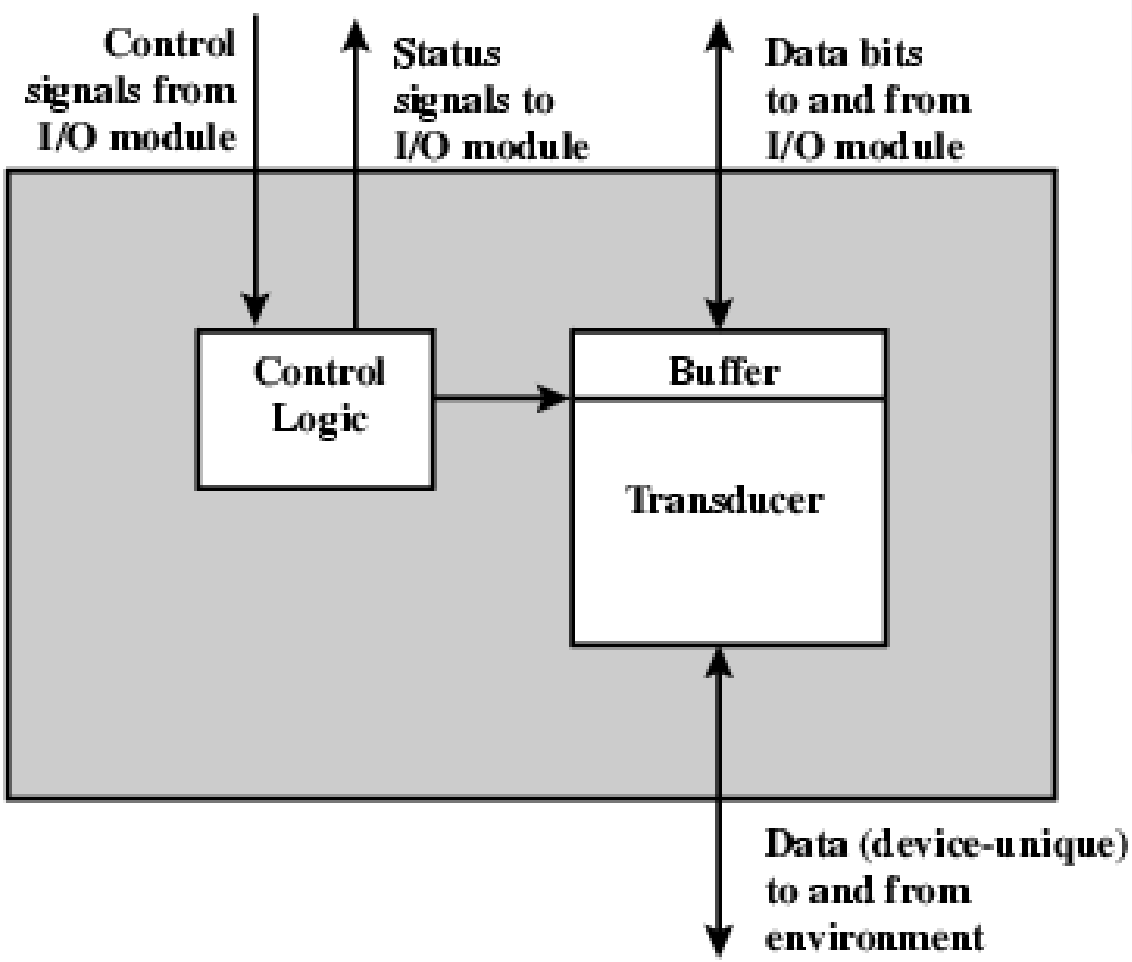
الأجهزة الخارجية External Devices

- المقروءة من قبل المستخدم Human readable
  - Screen, printer, keyboard
- المقروءة من قبل الآلة. Machine readable
  - المراقبة والتحكم Monitoring and control
- الاتصال Communication
  - عبر المودم Modem

بطاقة الشبكة Network Interface Card (NIC)

المخطط الصندوقي للجهاز الخارجي

External Device Block Diagram



وظائف نموذج I/O I/O Module Function

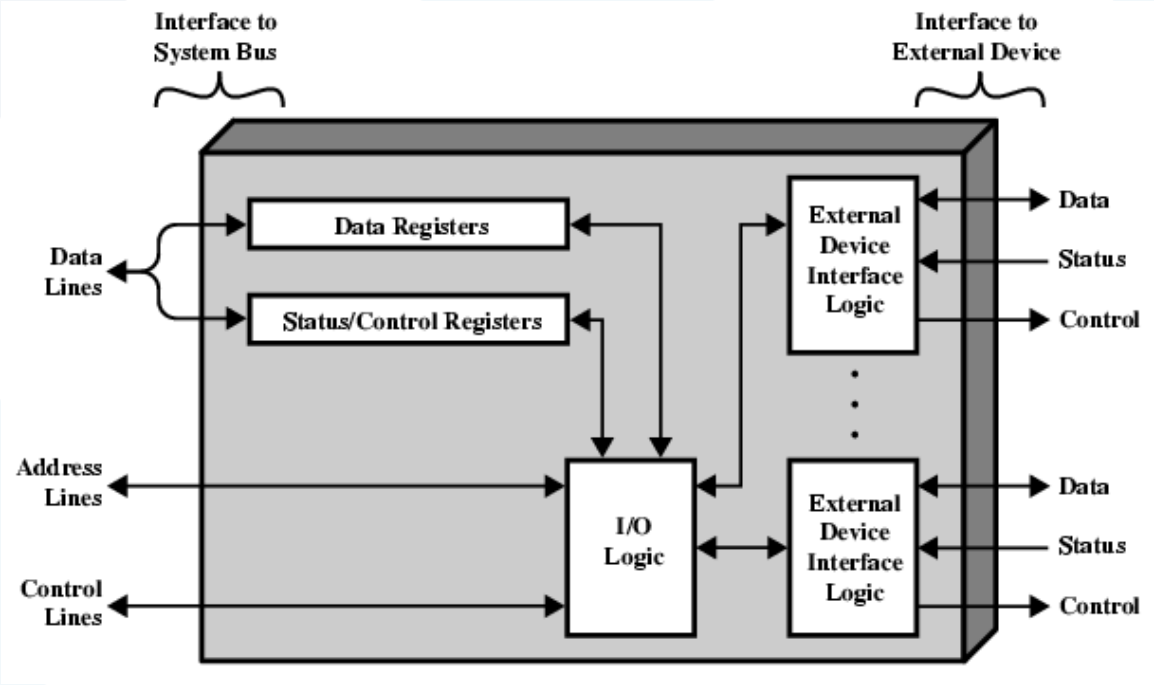
- التوقيت والتحكم Control & Timing
- الاتصال مع الـCPU CPU Communication

- الاتصال مع الجهاز الخارجي Device Communication
- التخزين الوسيطي Data Buffering
- كشف الأخطاء Error Detection

مراحل التعامل مع الـI/O I/O Steps

- تختبر الـCPU حالة نموذج وجهاز I/O
- CPU checks I/O module device status
- يعيد نموذج I/O الحالة I/O module returns status
- في حال توفر الجاهزية تطلب الـCPU الشروع بتبادل البيانات
- If ready, CPU requests data transfer
- يستقبل نموذج I/O بيانات من الجهاز الخارجي. I/O module gets data from device
- ينقل نموذج I/O البيانات للـCPU I/O module transfers data to CPU
- Variations for output, DMA, etc.

مخطط نموذج I/O I/O Module Diagram



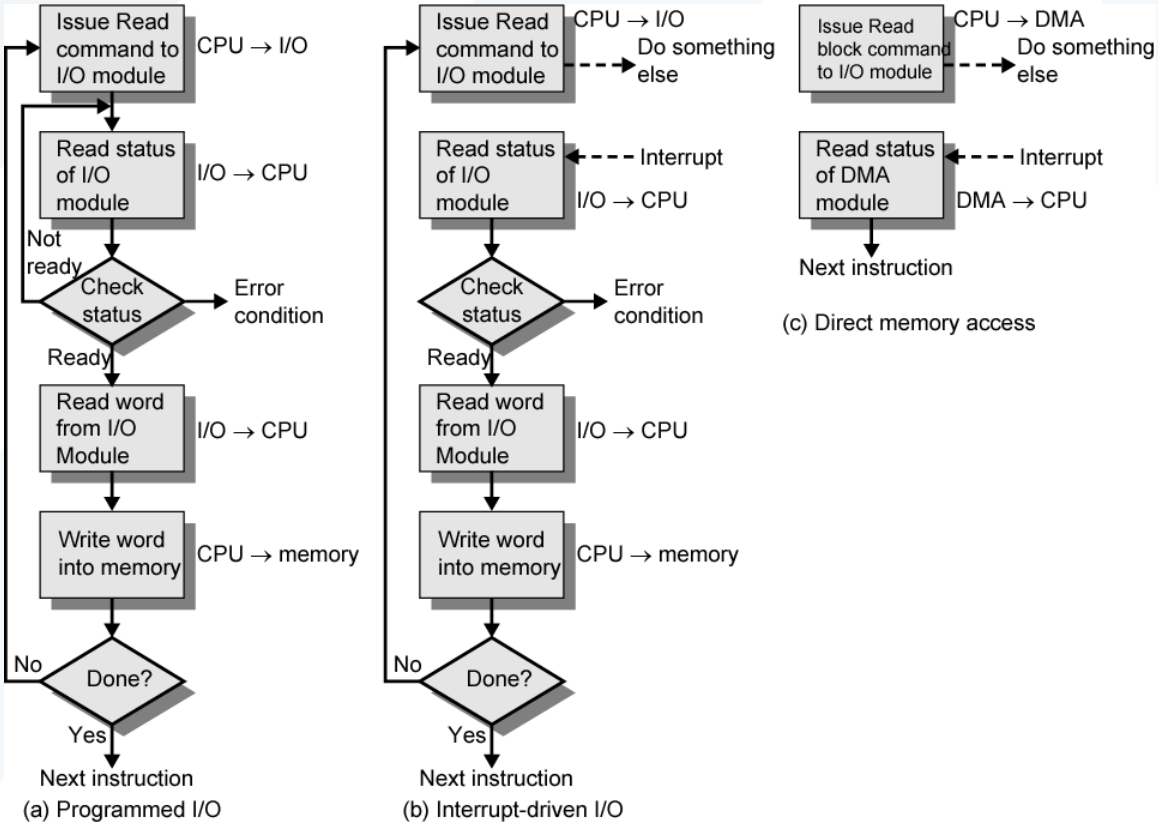
أحكام وقرارات I/O I/O Module Decisions

- إظهار خصائص الجهاز للـCPU Hide or reveal device properties to CPU
- دعم جهاز وحيد أو تعددي multiple or single device
- برمجة وظائف الجهاز أو تركها للـCPU. Control device functions or leave for CPU
- إضافةً إلى بعض قرارات نظامالتشغيل Also O/S decisions
—نظام التشغيل Unix -على سبيل المثال يتعامل مع كل شيء كملف file.

تقنيات الإدخال الإخراج Input Output Techniques

- المبرمج Programmed
- المقاد بالمقاطعة Interrupt driven
- النفاذ المباشر إلى الذلكرة Direct Memory Access (DMA)

التقنيات الثلاث لإدخال بلوك من البيانات
for Input of a Block of Data



الإدخال/الإخراج المبرمج Programmed I/O

- تتحكم الـCPU بشكل مباشر بالـI/O من خلال: CPU has direct control over I/O
  - تحسس الحالة Sensing status
  - أوامر القراءة الكتابة Read/write commands
  - نقل البيانات Transferring data
- تنتظر الـCPU نموذج الـI/O لإتمام العملية CPU waits for I/O module to complete operation
- تضيع وقت الـCPU Wastes CPU time