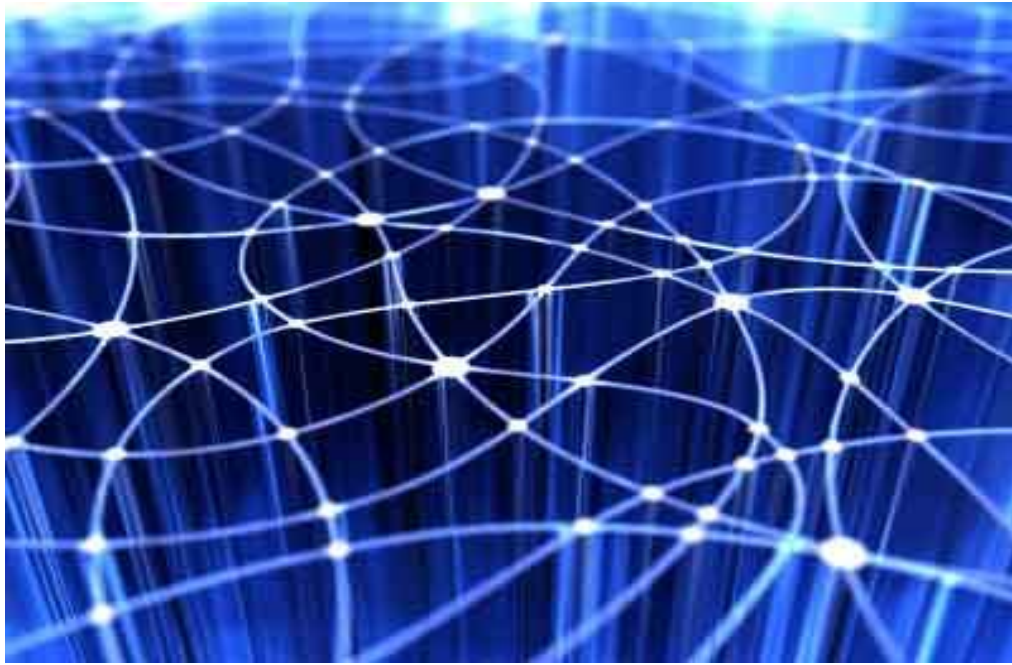




Computer Architecture IT222

By Arabic



ملتقى طلبة تقنية المعلومات – جامعة قاريونس

<http://www.itsf.ly>

المكتبة الالكترونية لكلية تقنية المعلومات

مقدمة عن الـ Computer Architecture تمهيداً للمنهج

هذا منهج مادة Computer Architecture بالعربي والمأخوذ من كتاب الدكتور وليام وهي المادة التي لم تعطى حقها في الكلية مطلقاً ، فأول ما يدخل علينا الأستاذ يقول لنا الكاش أو ترانزستور أو آدرس لعنوان الذاكرة أو الفلوتشارت و الخ ؛ بعد ذلك نصاب بالإحباط لأننا لا نعلم عن أي شي يتكلم و نجهل نقطة الانطلاقة من أين بدأت؟؟ ما هي الفائدة لو سألتها و ماذا يعني أركيكتشر وهو مع الأسف لا يعرف ؟ لكن ربما يقول تعمير و هو لا يدري تعمير عمارة و لا تعمير مستشفى و لا تعمير مبنى لكليتنا !! هذا مع الأسف إذا قال أركيكتشر ونطقها صح ! " مع كامل الاحترام للمختصين أمثال الدكتور منهم " عموماً كل هذه الأمور سنحاول توضيحها لتتضح الصورة وبعد ذلك سنتدرج في المنهج من الصفحة القادمة إن شاء الله ..

Architecture : تعني هندسة معمارية للحواسيب أي عبارة عن بناء الحاسب فمثلاً لو كان هناك مهندس معماري للمنازل راح يعطيك بناء و هندسة للمنزل الذي تريده و يرسمه لك على أوراق و يصمم لك المداخل و المخارج ؛ بالضبط كذلك الأمر في الهندسة المعمارية للحاسوب ، حيث سترى البناء الداخلي و الخارجي للحاسوب وكيف تدخل و تخرج منه البيانات و يعالجها لك لتراها .

Architecture & Organization : الهيكلية وتنظيم الحواسيب ، فالهيكلية عبارة عن الخواص المرئية للمبرمج وهي غالباً تعتمد على الأوامر التي يعطيها المبرمج للجهاز ، يعني شي غير ملموس لكن نراه ونستخدمه ، أما التنظيم فهي تهتم بالهاردوير .

وظيفة الهندسة المعمارية والتنظيم : هي رسم وبناء تركيب الحاسوب على ورق .

تختص المادة ببناء آلية الحاسوب ورسمها على ورق لرؤية كيفية انتقال البيانات وكيف نراها ما يعني أن الكتاب كله رسومات هندسية على ورق للحاسوب ومكوناته ووظائفه ، فالوظائف تعني حركة البيانات والتحكم فيها بالأسهم على حسب الأوامر "instructions" المعطاة .

بعض المصطلحات المهمة :

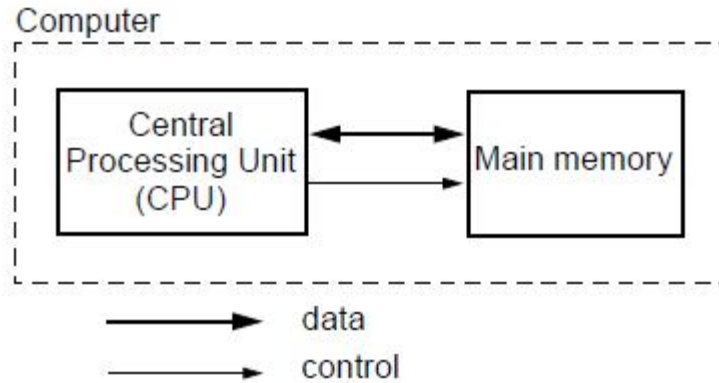
معناها	الكلمة	معناها	الكلمة
بيانات	Data	تعليمات / أوامر	Instructions
الذاكرة الثانوية	Secondary Memory	الذاكرة الرئيسية	Main Memory (M.M)
وحدة التحكم	Control Unit (CU)	وحدة المعالجة المركزية	Central Processing Unit (CPU)
ملحقات	Peripherals	وحدة الحساب والمنطق	Arithmetic Logic Unit (ALU)
دورة/حلقة	Cycle	أجهزة الإدخال والإخراج	Input/Output device (I/O)
نظام	System	عنونة	addressable
موقع	location	عنوان	address

Introduction about Computer Architecture

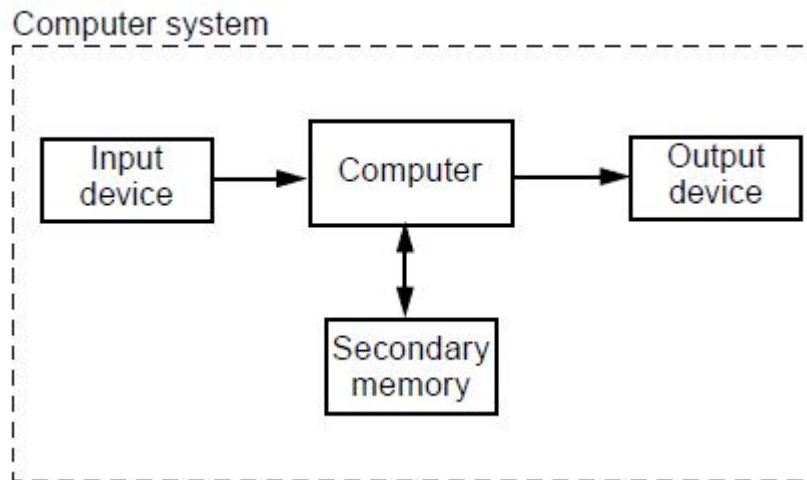
Chapter 1

مقدمة لهيكلية الحاسوب

A Computer: هو آلة لمعالجة البيانات تعمل تلقائياً تحت سيطرة قائمة من التعليمات (تسمى برنامج) مخزنة في ذاكرتها الرئيسية .



A Computer System: ويتكون من Computer وملحقاته .



Computer Peripherals: يشمل أجهزة الإدخال والإخراج والذاكرة الثانوية .

مبادئ هيكلية فون نيومان (1) (Von Numan Architecture):

فون نيومان عالم رياضي قام بتحديد القواعد الأساسية التي نستعملها في تصميم الحواسيب والمجمل في أربعة نقاط :

- 1- تخزين البيانات والتعليمات في الذاكرة الرئيسية (تخزين مفهوم البرنامج) .
- 2- محتوى الذاكرة عبارة عن عنوانة (addressable) تتم بواسطة الموقع بدون النظر لما هو مخزن بهذا الموقع .

3- يتم تنفيذ التعليمات بشكل تسلسلي (sequentially) ، من التعليمات الأولى للتالية بشكل منظم على حسب ترتيب موقعها ما لم يتم تعديل هذا النظام .

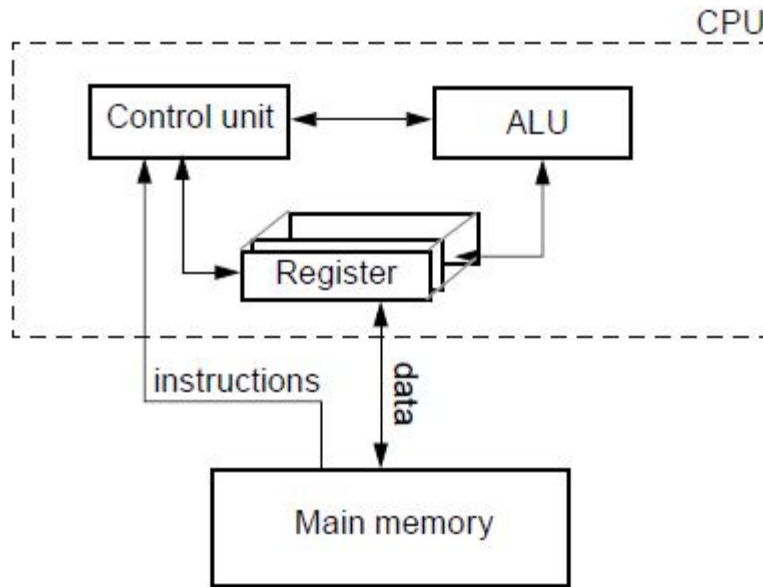
4- هيكلية الحاسوب هي :

- CPU "وحدة المعالجة المركزية" ويحتوي على CU "وحدة التحكم" والتي تتسق عملية تنفيذ التعليمات و ALU "وحدة الحساب والمنطق" والتي تجري العمليات الحسابية و المنطقية .
- Main Memory.

لقد تغيرت التقنيات المستخدمة في الحواسيب حيث بات بإمكانها حل مشاكل مختلفة عديدة تبعاً للبرنامج الذي ينفذها على عكس ما كان في بادئ الأمر عندما كان الحاسوب عبارة عن ذاكرة تحفظ المعطيات وغير قادرة على حل العمليات الحسابية.

هيكلية فون نيومان متعددة الأغراض (General Purpose Von Numan Architecture) :

إن الحواسيب الالكترونية المستخدمة حالياً هي حواسيب فون نيومان متعددة الأغراض والتي تؤدي برامج مختلفة على حسب البرنامج المنفذ لها .



1- تتمثل المهمة الرئيسية للـ CPU في تنفيذ التعليمات الآتية إليه من الذاكرة الرئيسية.

2- يتم إخبار الـ CPU عن وجوب أداء واحدة من العمليات الأساسية (قد تكون حسابية أو منطقية) وذلك لنقل البيانات من وإلى الذاكرة الرئيسية.

3- وحدة التحكم (CU) هي التي تقوم بتفسير/ترجمة التعليمات الآتية إليها من الذاكرة الرئيسية وهي أيضاً تقوم بإخبار العناصر الأخرى المختلفة عما يجب القيام به.

4- وحدة الـ CPU تتضمن مجموعة من السجلات "Records" التي تعتبر عادةً كأجهزة تخزين مؤقت للبيانات التي تتم معالجتها داخل الـ CPU .

تمثيل البيانات (Representation of data) :

يتم تمثيل جميع البيانات والمعلومات التحكمية " التعليمات " بداخل الحاسوب بالنظام الثنائي "binary system" والذي يستخدم فقط رمزين إما 0 أو 1 ؛ الرمزين 0،1 يتم تمثيلهما في الحاسوب بواسطة الإشارات الكهربائية حيث 0 تعني عدم وجود الإشارة ، 1 تعني وجودها.

كذلك البيانات الرقمية يتم تمثيلها باستخدام نظام الـ binary حيث يعتبر مكان القيم مرفوعاً كأس بالنسبة للأساس 2 كالتالي :

$$100101 = 1*2^0 + 0*2^1 + 1*2^2 + 0*2^3 + 0*2^4 + 1*2^5$$

$$10110 = 0*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4$$

وكذلك الأعداد الثنائية الـ binary تجمع وتطرح وتضرب وتقسم باستخدام ALU مباشرة دون الحاجة لتحويلها إلى النظام العشري الأولي "decimal system" : $100101 + 10110 = 111011$

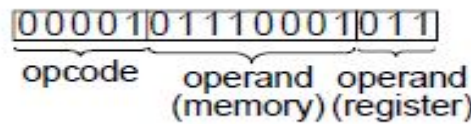
تعليمات الآلة (Machine Instruction) :

الـ CPU وحده هو من يمكنه تنفيذ تعليمات الآلة ، حيث أن كل كمبيوتر يملك مجموعة محددة من تعليمات الآلة التي تجعل الـ CPU قادراً على استلامها وتنفيذها .

يتم تمثيل تعليمات الآلة بسلسلة من الـ bits "الأعداد الثنائية" والمجموعة لمعرفة :

- 1- ما الذي ينبغي القيام به (Operation Code)
- 2- لمن تطبق العملية (Source Operands)
- 3- أين تذهب النتيجة (Destination Operand)
- 4- كيفية مواصلة العمل بعد الانتهاء

إنه من خلال تسلسل الثنائيات في الحاسوب يمكن تمثيل تعليمة معينة ، حيث كل تسلسل يقسم إلى ثلاثة أجزاء من خلال هذه الأجزاء يمكن معرفة ما هو المطلوب من هذه التعليمة .



الجزء الأول يعني نوع العملية وهو ما يسمى بـ opcode ، والجزء الثاني القيمة التي نحتاج أن نجلبها من الذاكرة لتدخل في التعليمة operand memory ، والجزء الأخير وهو المكان الذي سوف يوضع فيه الناتج . فمثلاً لتمثيل التعليمة التالية

$MOV\ y \rightarrow R1$ أي ضع القيمة y في السجل R1 وهذا يعني أنا نوع العملية هو نقل MOV وأن العنصر الذي سنجلبه من الذاكرة هو y وأن الوجهة سوف تكون R1 .

أنواع تعليمات الآلة (Types Of Machine Instructions) :

- 1- نقل البيانات من الذاكرة وسجلات الـ CPU.
- 2- العمليات الحسابية والمنطقية.
- 3- التحكم بالبرنامج .
- 4- النقل أثناء عمليات الإدخال والإخراج.

تنفيذ التعليمات (Instruction Execution) :

لتنفيذ عملية حسابية مثل $Z=(X+Y)*3$ فإن هذه العملية الحسابية سوف تقسم إلى أربع تعليمات ، حيث العملية الأولى هي وضع المتغير Y في أحد سجلات الـ CPU ولنفرض أنه R3 أي نقل القيمة Y إلى السجل R3 بهذا الشكل

MOV Y → R3

بعد ذلك وفي التعليمات الثانية نضيف المتغير X إلى القيمة الموجودة في R1 وبالتالي التعليمات هي

ADD X+R3 → R3

في التعليمات الثالثة نضرب القيمة 3 في السجل R1

MUL 3#R3→ R3

التعليمات الأخيرة هي نقل الناتج من R1 ووضعه في القيمة Z

MOV R3→Z

يتعرف جهاز الحاسوب على كل حرف أو رقم أو رمز في الكيبورد من خلال شفرة خاصة ، حيث أن :

يتم عنونة كل عنصر في هذه العملية التي وضعت في المثال السابق ، فالرقم 3 شفرته 00000011 والحرف X هو 01110000 والحرف Y هو 01110001 والحرف Z هو 01110010 ؛ كذلك العمليات لها شفرات خاصة حيث أن نوع العملية ADD هو 00011 و العملية MUL هي 00101 وعملية النقل من الذاكرة إلى سجل الـ CPU "MOV" هي 00001 بينما النقل العكسي من سجلات الـ CPU إلى الذاكرة "MOV" أخرى ولكن بالعكس هي 00010

كل الشفرات السابقة غير مطلوب من الطالب حفظها ولكنها لتسهيل طريقة شرح تنفيذ تعليمات الآلة داخل الحاسب الآلي .

بهذا الشكل لتمثيل التعليمات الأولى يكون الكود 000101110001001 حيث العملية هي MOV أي جلب القيمة Y من الذاكرة إلى سجل الـ CPU رقم 1 وشفرته 001. إذاً نفهم أن شفرة كل تعليمات تنقسم إلى ثلاثة أجزاء ، فالأول من اليسار إلى اليمين هو نوع العملية وحجمه 5bit والثاني العنصر المستخدم في الذاكرة وحجمه 8bit والآخر هو السجل المستخدم وحجمه 3bit . تعليمات الآلة ستكون متسلسلة بهذا الشكل

```

0000101110001011
Move   addr of Y  Reg 3

0001101110000011
Add     addr of X  Reg 3

00101000000011011
Mul     operand "3" Reg 3

0001001110010011
Move    addr of Z  Reg 3

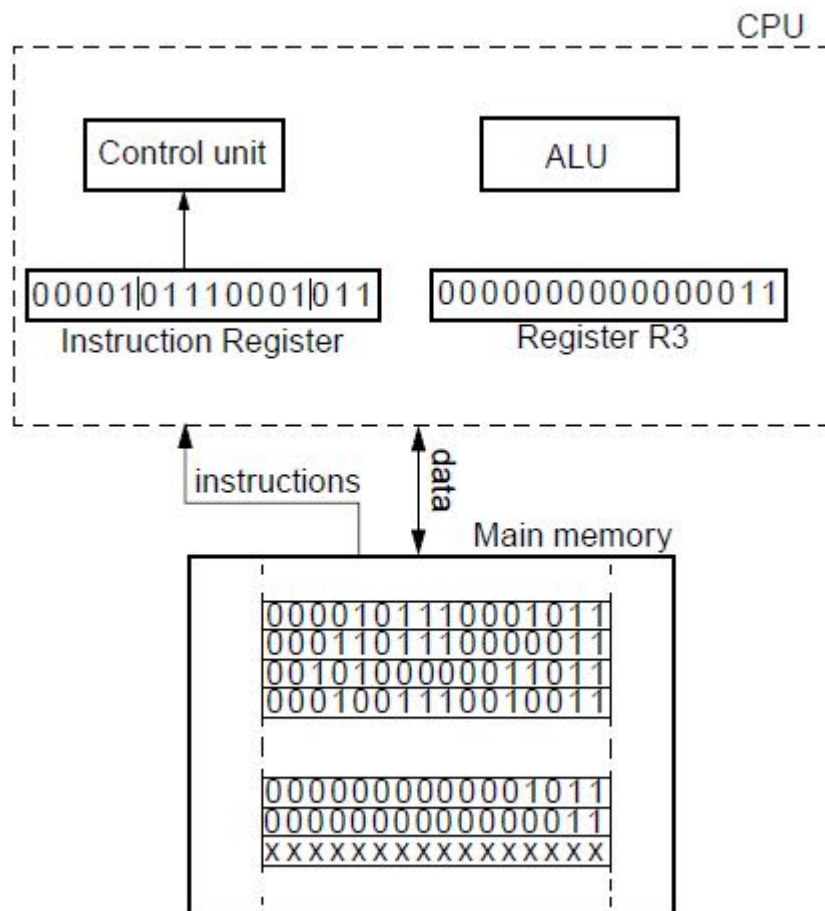
```

يتم تشفير قيمة X,Y,Z بأكواد مختلفة مخزنة في الذاكرة ؛ أي يتم وضع address لها فالذاكرة الرئيسية لا تخزن قيم حقيقية بل تخزن عناوين لهذه القيم كما تم الذكر في هيكلية فون نيومان فالقيم X,Y,Z يتم إعطاؤهم عناوين كالتالي :

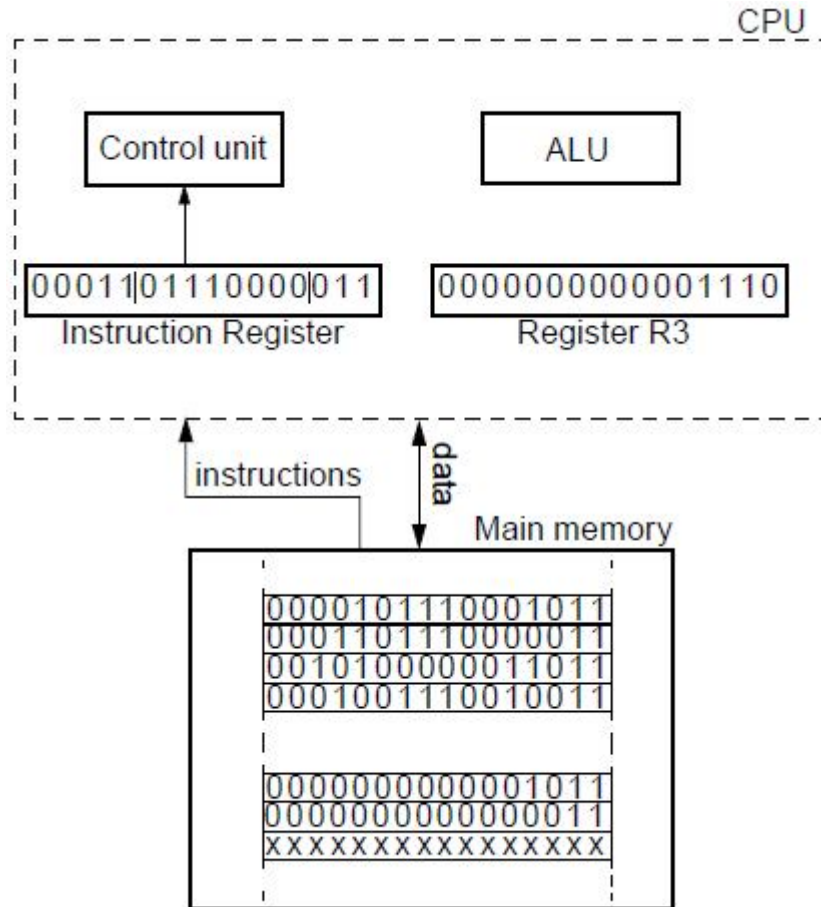
القيمة بالنظام الثنائي	بعد أن تم إعطاء عنوان لها
01110000 : X	00000000000000011
01110001 : Y	00000000000001011
01110010 : Z	00000000000101010

هذه العناوين تعطى للقيم بحسب موقعها في الذاكرة أي أنها ليست ثابتة بهذا الـ address .

مرحلة التعليمة الأولى :

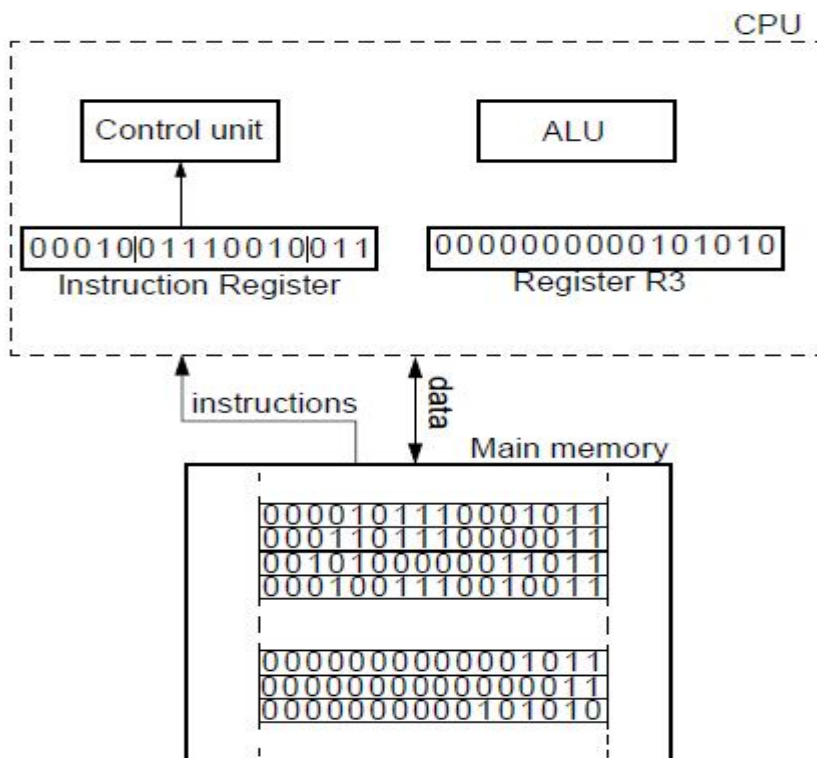
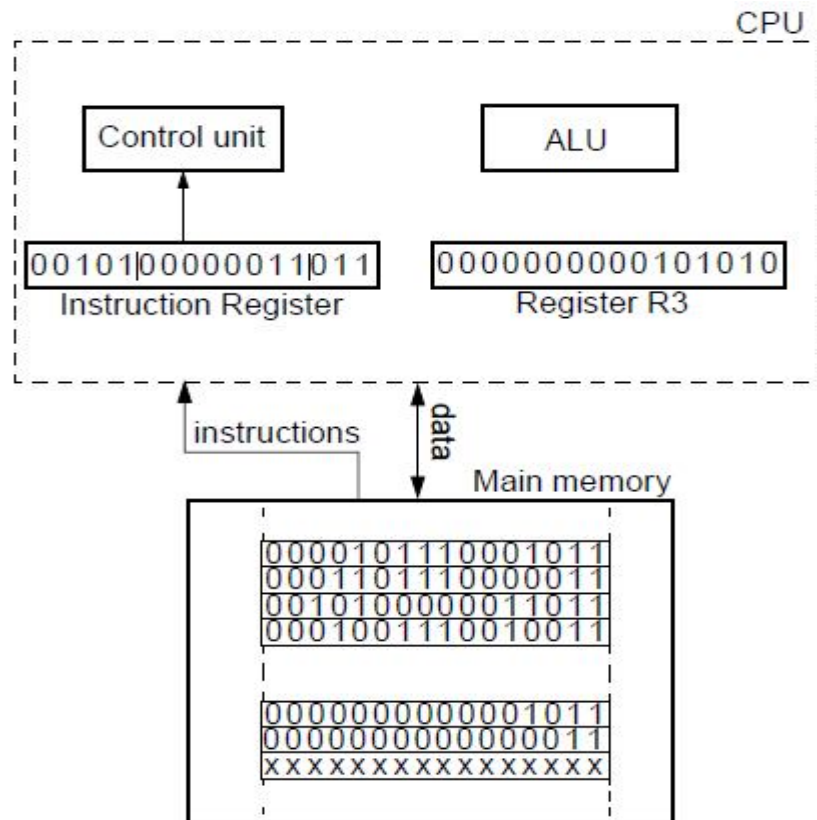


في المرحلة الأولى لحركة التعليمات ، تنتقل التعليمة الأولى من الذاكرة إلى الـ CPU " المعالج " فيتم تخزينها في سجل التعليمة IR " Instruction Register " والذي ينقلها إلى وحدة التحكم التي تفسر الغرض من هذه التعليمة فتعرف أن المقصود هو حساب وضع Y في السجل R3 وبالتالي يتم نقل التعليمة إلى ALU لتتم عملية الحساب هذه ، ونلاحظ هنا أن السجل R3 لازال مخزن فيه قيمته الأصلية رقم 3 .



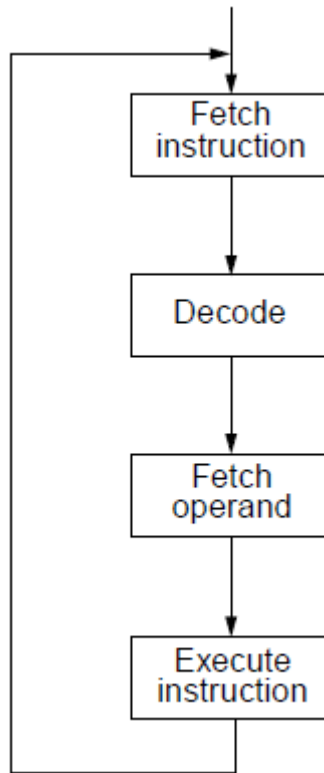
في هذه المرحلة وبعد أن قامت وحدة التحكم بنقل المطلوب من التعليم الأولى وهو إضافة القيمة Y إلى السجل R3 حيث $1110 = Y + R3$ وهي قيمة R3 الحالية التي توضع في R3 ؛ وكذلك في نفس الفترة التي يتم فيها عملية الحساب أو الإضافة بين R3 و Y فإنه يتم استدعاء التعليم الثانية من الذاكرة الرئيسية وتخزن بالـ IR وتنقل إلى وحدة التحكم ليفهم المغزى منها والذي يعني إضافة القيمة X إلى قيمة R3 وتخزين الناتج في R3 ، وبالطبع يلاحظ استمرار قيمة Z خالية بدون ناتج في الذاكرة الرئيسية لعدم الوصول إلى ناتج Z حتى الآن ، بعد ذلك يتم نقل الترجمة في المرحلة التي تليها إلى ALU ليتم إضافة X إلى قيمة R3 وتخزين الناتج في R3 .

و هكذا تتم الإضافة في كل مرحلة إلى أن يتم الوصول إلى قيمة Z النهائية باستخدام عمليات كل تعليمة (الجمع / الضرب)



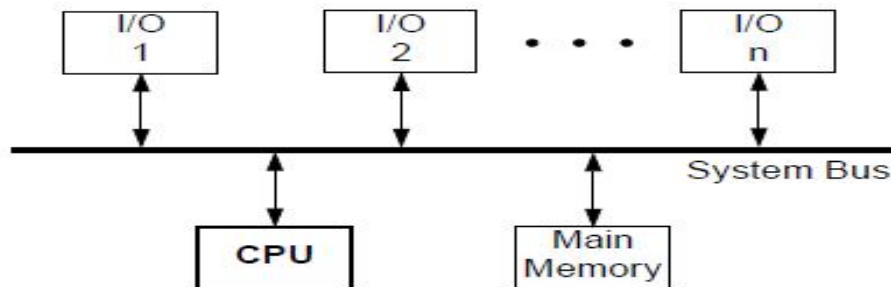
حلقة التعليم (The Instruction Cycle) :

يتم تنفيذ كل تعليمة كسلسلة من الخطوات ، وهي خطوات تتم في التعليمة الواحدة

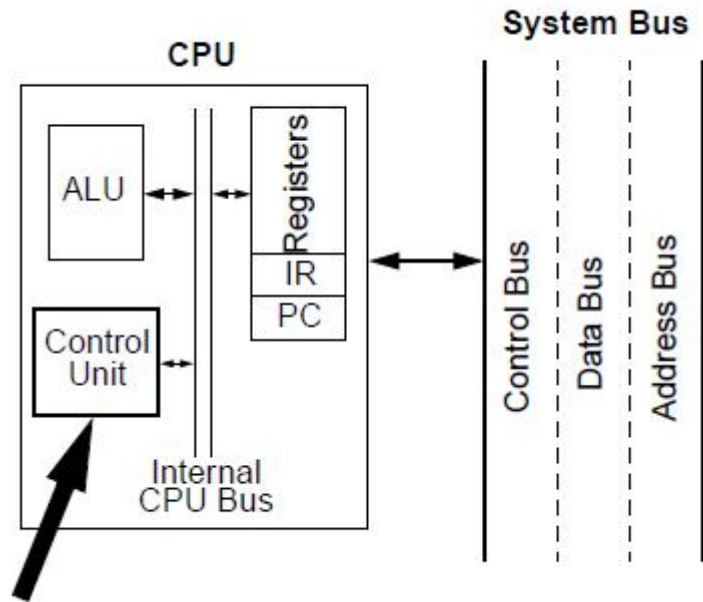


تطبق التعليمة دائماً على أربعة مراحل ؛ يتم في المرحلة الأولى جلب التعليمة من الذاكرة الرئيسية إلى المعالج CPU ، وبعد ذلك يتم تفسيرها وترجمتها لمعرفة الهدف منها في وحدة التحكم CU بداخل الـ CPU ، وفي المرحلة الثالثة يتم إحضار المعامل المطلوب استخدامه في التعليمة كقولنا $MOV\ Y \rightarrow R3$ أي أن المعامل هو Y والمطلوب هو إحضاره من الذاكرة ليتم جمعه مع قيمة R3 ، وفي المرحلة الأخيرة يتم تنفيذ هذه التعليمة وذلك بوضع قيمة الجمع بين Y و R3 في السجل R3 .

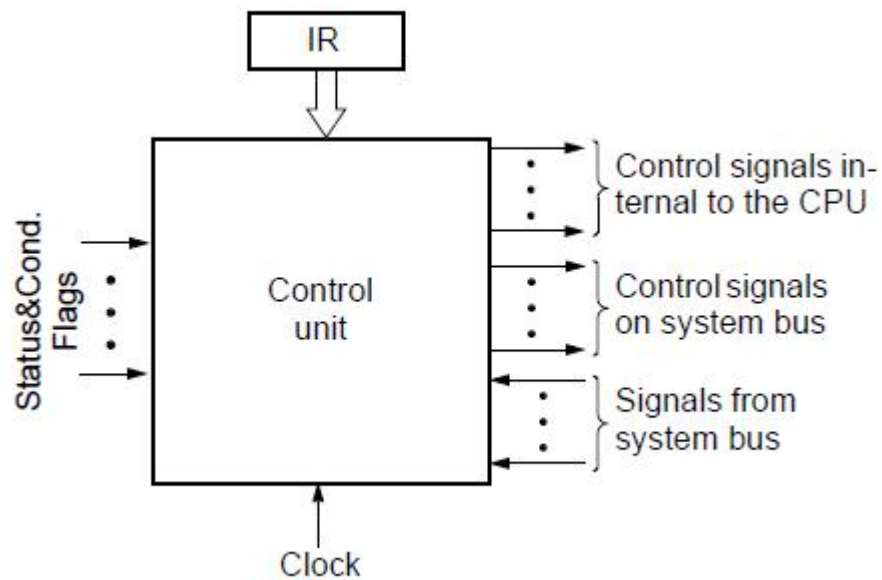
وحدة التحكم (The Control Unit) :



إن تحليل التعليمة كعملية الجمع بين R1,R2 يحتاج إلى وحدة تحكم وبشكل عام فإن الذي يتحكم في العمليات التزامنية أي الوقتية هو CPU ولكن بشكل خاص فإن الذي يتحكم في هذا التزامن هو Control Unit .



إن كيفية تواجد العناصر على سطح CPU وبداخله أمر مهم لمعرفة كيفية مسار البيانات في النظام بشكل صحيح ، كما وأن تنفيذ عملية (المزامنة) أي عملية التحكم تتم في CU .



التقنيات التي تنفذها وحدة التحكم :

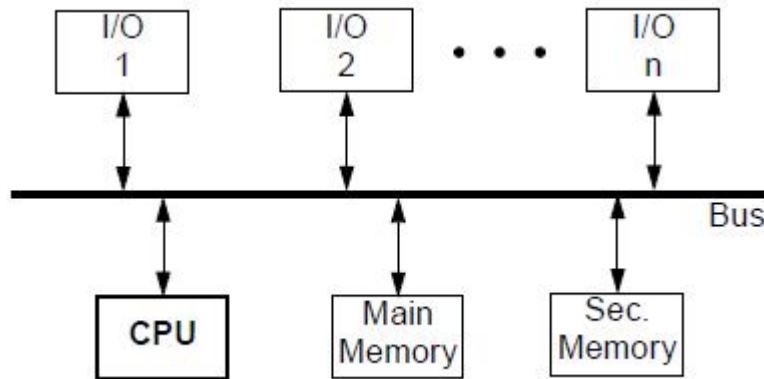
Hardwired Control -

Micro programmed Control -

نظام الحاسوب (The Computer System) :

يشكل الـ Main Memory & CPU نواة الـ Computer System ؛ كما وأن الذاكرة الثانوية و أجهزة الإدخال والإخراج تسمى ملحقات peripherals .

يتم التواصل بين المكونات المختلفة في أي نظام عادةً باستخدام Bus " ناقل " واحد أو أكثر .



الذاكر (Memories) :

تستخدم الذاكرة الرئيسية في تخزين البرامج والبيانات التي تتم معالجتها من قبل CPU .

الذاكرة الثانوية توفر تخزين على المدى الطويل لكميات كبيرة من البيانات والبرامج .

يمكن للـ CPU أن يدير البيانات والبرامج في الذاكرة الثانوية قبل تحميلها على الذاكرة الرئيسية .

أهم خصائص الذاكرة هي السرعة والحجم والتكلفة، وكذلك التكنولوجيا التي تستخدمها مفيدة جداً عند التنفيذ.

عادةً ما تكون الذاكرة الرئيسية سريعة في التنفيذ ولكنها محدودة الحجم، بينما الذاكرة الثانوية بطيئة نسبياً ولكنها ذات حجم كبير جداً .

الذاكرة الرئيسية (The Main Memory) :

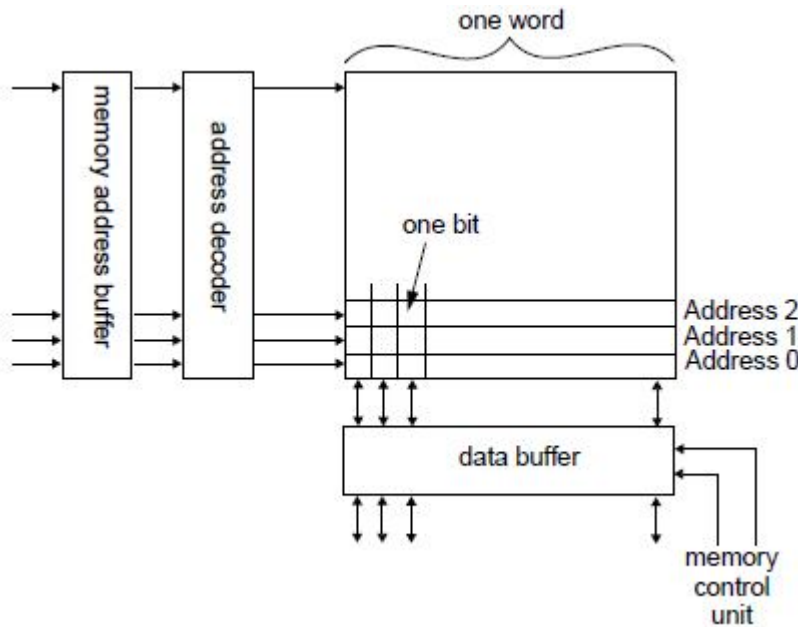
يمكن النظر إلى الذاكرة الرئيسية على أنها مجموعة من خلايا التخزين ، كل خلية منها تستخدم لتخزين كلمة "word" .

كل خلية تقوم بتعيين عنوان فريد من نوعه على شكل أرقام متسلسلة .

بالإضافة إلى خلايا التخزين يوجد مخزن عنوان الذاكرة " memory address buffer " - يعمل على تخزين العنوان الخاص بالكلمة المراد قراءتها أو كتابتها - وكذلك يوجد مخزن بيانات " data buffer " - لتخزين البيانات المراد كتابتها أو قراءتها - كذلك يوجد ما يسمى بـ " address decoder " مفسر العناوين و " memory control unit " وحدة التحكم بالذاكرة .

غالباً وعلى نطاق واسع ما يتم استخدام تقنية لتنفيذ الذاكر الرئيسية وهي ذواكر أشباه الموصلات .

نوع ذاكرة أشباه الموصلات الشائعة هي ذاكرة الوصول العشوائي (RAM) ؛ كما وأن المعلومات المخزنة في RAM " ذاكرة أشباه الموصلات " ستضيع عندما ينقطع التيار الكهربائي .



الذاكرة الثانوية (The Secondary Memory) :

القرص الصلب (Hard disk) :

يتم تسجيل البيانات على سطح القرص الصلب المصنوعة من المعدن المطلي بمادة مغناطيسية.

(disks " الأقراص " & drive " المحرك ") عادة ما يتم بناؤهما مع بعضهما البعض وتتم تغطيتهما في إناء وثيق لحماية الأقراص من الملوثات كجسيمات الدخان والغبار. كذلك عادةً ما تكون كل الأقراص مكدسة على عمود متحرك تشترك فيه كل الأقراص أيًا كانت وظيفتها القراءة أو الكتابة.

المزايا الرئيسية :

- الوصول المباشر " direct access " .

- سرعة في الوصول " fast success " حيث وقت البحث - seek time - تقريباً 10ms ومعدل نقل البيانات - data transfer rate - تقريباً 5MB/s .

- سعة التخزين كبيرة " large storage capacity " من 8 MB إلى عدة GB.

القرص المرن " Diskette " :

يتم تسجيل البيانات على سطح الـ Floppy disk المصنوع من مادة البوليستر والمغطى بمادة مغناطيسية. محرك قرصه المرن يجب أن يستخدم لإيصال البيانات المخزنة في الـ Floppy disk وهي تعمل مثل سجل القرص الدوار الموجود في الجراموفون .

المزايا الرئيسية :

- الوصول المباشر " direct access " .
- رخيص " cheap " .
- محمول " portable " وسهل الاستخدام .

الشريط المغنط " Magnetic tape " :

يتكون الشريط المغناطيسي من طبقة بلاستيكية مطلاة بأكسيد الحديد ، حيث يمكن لأكسيد الحديد أن يمغنط في اتجاهات مختلفة للتمكن من تمثيل البيانات بشكل أفضل .

يتم تشغيل الشريط المغناطيسي عن طريق مبدأ مماثل كما في حالة شريط المسجل .

المزايا الرئيسية :

- وصول تسلسلي " sequential access " حيث زومن الوصول يقدر ب 1-5 ثانية.
- قيمة تخزين عالية " High value of storage " 50MB/table .
- معقول الثمن " inexpensive " .
- يستخدم الشريط المغنط غالباً في عملية النسخ الاحتياطي أو في أعمال الأرشفة.

الذاكرة البصرية " Optical Memory " :

المزايا الرئيسية :

- سعة تخزين عالية " huge capacity " : 775MB/disk .
- معقولة الثمن " huge capacity " .
- قابلة للنقل " removable " .
- للقراءة فقط " read-only " .
- زمن وصول طويل " long access time " .

Typical Input Devices

Device	Main features	Advantages	Disadvantages
Keyboard	Like a typewriter	Efficient for inputting text	Relatively slow, speed depends on operator
Light pen	Point at screen	Easy to use	Needs much software to make it versatile
Mouse	Move around on desk	Efficient for icon-based input, and menu selection	Needs much software support
Joystick	Used for games and control	As above Fast	Needs much software support
Graphics tablet	Graphics input	Input picture and freehand sketch	Slow
Scanner	Copy pictures	Fast input of graphics	Bit-mapped graphics only
Voice input	User friendly	No hands needed	Limited vocabulary, Speech recognition software needed

Typical Output Devices

Device	Main features	Advantages	Disadvantages	Speed
Display Screen	Most versatile, both text and graphics	No waste of paper etc.	No hard copy	
Line printer	Impact printer, Very fast.	Can cope with high volume	Large versions are very noisy	up to 6000 cps
Dot matrix printer	Versatile text and graphics	Inexpensive	low quality and speed	up to 200 cps
Inkjet printer	Mechanically similar to above; dot produced by ejected ink droplet	small size; inexpensive	lower quality than laser printers	~20 line/sec
Laser printer	High quality text and graphics	Very fast, high volume	(used to be) expensive	20 000 line/min possible
Plotter	High quality graphics	large graphics output possible	Large machine, expensive	Pen up to 1 meter/s
Voice output	Natural for certain applications	Don't need to use eyes	Limited range of sounds	Normal speech

The Memory System

Chapter 2

نظام الذاكرة

مكونات نظام الذاكرة :

- Main Memory : 1- سريعة
- 2- ذات وصول عشوائي.
- 3- مرتفعة في الثمن
- 4- تقع بالقرب من CPU وليس بداخله.
- Secondary Memory : 1- بطيئة
- 2- ذات وصول مباشر.
- 3- رخيصة الثمن.
- 4- تتواجد بعيداً عن CPU.

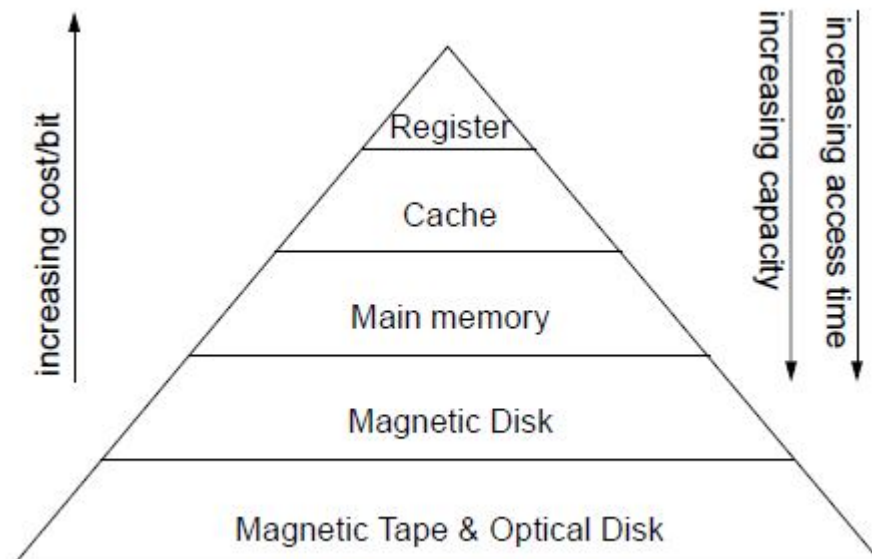
مشاكل بنظام الذاكرة :

دائماً ما تكون الحاجة إلى ذاكرة تناسب البرامج الكبيرة جداً ، وتعمل بسرعة مماثلة لسرعة المعالجات ، ولكن المشكلة الرئيسية هنا هي أن الذاكرة الكبيرة دائماً ما تكون بطيئة وبالتالي لا يمكنها مجاراة سرعة المعالجات لذلك يجب أن يكون حجم الذاكرة صغير ، كذلك عند تصغير حجم الذاكرة فإنه لن يكون بإمكانه أن تخزن البرامج الكبيرة ؛ حيث أنه الذاكرة تكون أبطأ كلما زاد حجمها ، وكذلك تزداد قيمة الذاكرة في السعر كلما كانت أسرع .

الحل لهذه المشكلة هو بناء نظام ذاكرة مركب، حيث يجمع بين الذاكرة السريعة الصغيرة والذاكرة الرئيسية البطيئة والكبيرة لتصبح كأنها ذاكرة كبيرة وسريعة.

إن المبدئين السابقين في الدمج بين الذاكرة الكبيرة البطيئة والصغيرة السريعة يمكن أن يحسن في نظام الحاسوب ليكون شكل هرمي يحتوي على عدة مستويات متضمنة الذاكرة الثانوية (قرص التخزين) والذاكرة الرئيسية Main Memory.

الشكل الهرمي للذاكرة : The Memory Hierarchy :



بعض خصائص نظام الذاكرة :

1. سجلات المعالج " Processor Registers " :

- عددهم 32 سجل في كل معالج ، حجم السجل الواحد 32bit ، أي أن حجم السجلات الكلي هو 128Byte.
- زمن الوصول = البعض من النانو ثانية.

2. الرقاقة الداخلية لذاكرة الكاش " on-chip Cache Memory " :

- السعة التخزينية = من 8 - 32KB .
- زمن الوصول تقريباً 10 نانو ثانية.

3. الرقاقة الخارجية لذاكرة الكاش " off-chip Cache Memory " :

- السعة التخزينية = عدة مئات من الكيلو بايتات.
- زمن الوصول = العشرات من النانو ثانية.

4- الذاكرة الرئيسية " Main Memory " :

- السعة التخزينية = العشرات من الميغا بايتات.
- زمن الوصول تقريباً 100 نانو ثانية.

5- القرص الصلب " Hard disk " :

- السعة التخزينية = عدة غيغا بايتات.
- زمن الوصول = العشرات من الملي ثانية.

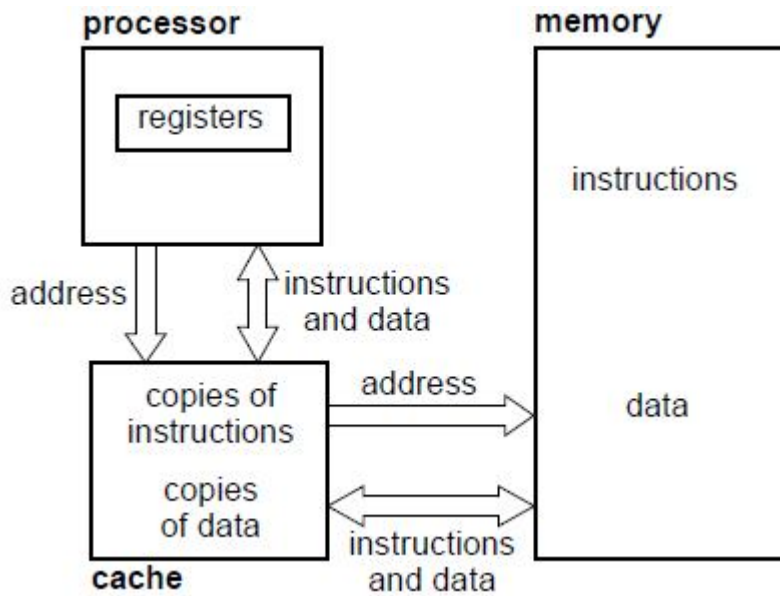
إن إمكانية توزيع البيانات والتعليمات عبر نظام الذاكرة المركب هو السبيل لنجاح المثلث الهرمي للذاكرة ؛ وذلك لاستغلال البطء الناتج من دوائر القاع في المثلث الهرمي التي عادةً ما تكون بطيئة في الاستجابة مع المعالج ، واستغلال هذا الوقت من خلال جعل المعالج متصلاً بدوائر القمة في المثلث الهرمي والتي تكون سريعة ، وذلك لمجاراة الـ CPU في سرعته .

الذاكرة الوسيطة " Cache Memory " :

ذاكرة الـ Cache تسمى الذاكرة الوسيطة أو المخبأة ، فهي مخبأة " مخفية " وذلك لأنه لا يمكن مشاهدتها إلا بعد أن نقوم بفتح الجهاز من الداخل ، و وسيطة لأنها تتواجد بين الذاكرة الرئيسية و المعالج ؛ وقد وضعت ذاكرة الكاش بين الذاكرة الرئيسية والمعالج للتمكن من مجاراة سرعة المعالج التي لا يمكن للذاكرة الرئيسية من تحقيق ذلك لأنها أبطأ من سرعة المعالج بسبب كبر حجمها ؛ قد صممت الكاش كذاكرة صغيرة وسريعة ليتم الدعم بشكل فوري بالبيانات عند طلبها من قبل المعالج ، حيث يحتفظ الكاش بنسخ من البيانات المنتقلة من الذاكرة الرئيسية إلى المعالج لإمكانية احتياج المعالج لهذه البيانات مرة أخرى ، وبضرب مثال على هذا نتخيل أنفسنا أننا ضيوف ضمن حفلة ما ، وأن الشخص المضيف مطلوب منه أن يقدم لنا المشروبات التي نطلبها تماماً دون تغيير ، وأن هناك لدى المضيف ثلاجة في المطبخ وعنده ثلاجة أخرى في غرفة التخزين الأبعد عن المطبخ ويوجد بالقرب من منزله محل سوبر ماركت ؛ فعلى اعتبار المشروبات المطلوبة هي المعلومات المطلوبة من المعالج ، والثلاجة في المطبخ هي الكاش ميموري الرقاقة الخارجية ، والثلاجة في غرفة التخزين هي كاش ميموري الرقاقة الداخلية، والسوبر ماركت هو الذاكرة الأساسية ؛ فإذا طلب شخص في الحفلة علبه كولا فعلى المضيف أن يذهب إلى الثلاجة في المطبخ ليرى إذا كانت موجودة لديه فإذا كانت داخل الثلاجة فيستطيع أن يلبي الطلب فوراً ، أما إذا لم تكن في ثلاجة المطبخ فعلى المضيف أن يذهب إلى الثلاجة في غرفة التخزين فإذا لم يجدها هناك أيضاً فعليه أن يذهب إلى السوبر

ماركت القريب من منزله ويحضرها من هناك وهذا سوف يأخذ وقت أطول فعليا ؛ فإذا اشترى المضيف صندوق كامل من علب الكولا من السوبر ماركت فهذا يضمن له أنه إذا طلب أحدهم علبة كولا أخرى فإنه يستطيع الحصول عليها من الثلاجة بدل الذهاب إلى السوبر ماركت مرة أخرى . وبنفس الطريقة عندما يطلب المعالج معلومة من الذاكرة الأساسية فإنه يتم جلب المعلومة والاحتفاظ بنسخة منها في الكاش وجلب بعض من المعلومات التي تليها ويأخذهم الكاش ميموري ليدعم بهم المعالج فوراً عند احتياجه إليهم بدلاً من رحيله إلى الذاكرة الرئيسية وأخذ للبيانات من هناك مما يجعل الوقت أطول .

ما ذكر كان مثلاً من خارج المنهج يوضح ماهية الكاش الذي يعمل على الاحتفاظ بنسخ من المعلومات التي استخدمت في الآونة الأخيرة من الذاكرة الرئيسية ؛ كذلك فإن الكاش يعمل بشفافية وهو من يقرر أي البيانات التي تبقى ويحتفظ بها لديه وأياً التي انتهت صلاحيتها فيتخلص منها .



كثيراً ما يطلب المعالج نفس المعلومات أكثر من مرة في أوقات متقاربة ما يجعل الكاش يحتفظ بنسخة هذه المعلومات التي يتوقع احتياجها من قبل المعالج ليدعمه بها حين احتياجه لها ، فعندما يريد المعالج جلب بيانات / تعليمات فإنه يبحث عنها أولاً في الكاش ميموري أولاً ، فإن لم يجدها أي فشل المعالج في إيجاد المعلومات التي يريدتها سمي ذلك بالـ **Miss** ، أما في حالة بحث عنها في الكاش ونجح المعالج في إيجاد المعلومات التي يريدتها سمي ذلك بالـ **Hit** .

Hit Rate " معدل الإصابة " : هو نسبة وجود البيانات في الكاش من عدد البيانات التي يحتاجها المعالج.

Miss Rate "معدل عدم الإصابة " : هو نسبة البيانات التي لم يتم إيجادها في الكاش من عدد البيانات التي يحتاجها المعالج.

كيف يتم عمل الكاش ؟

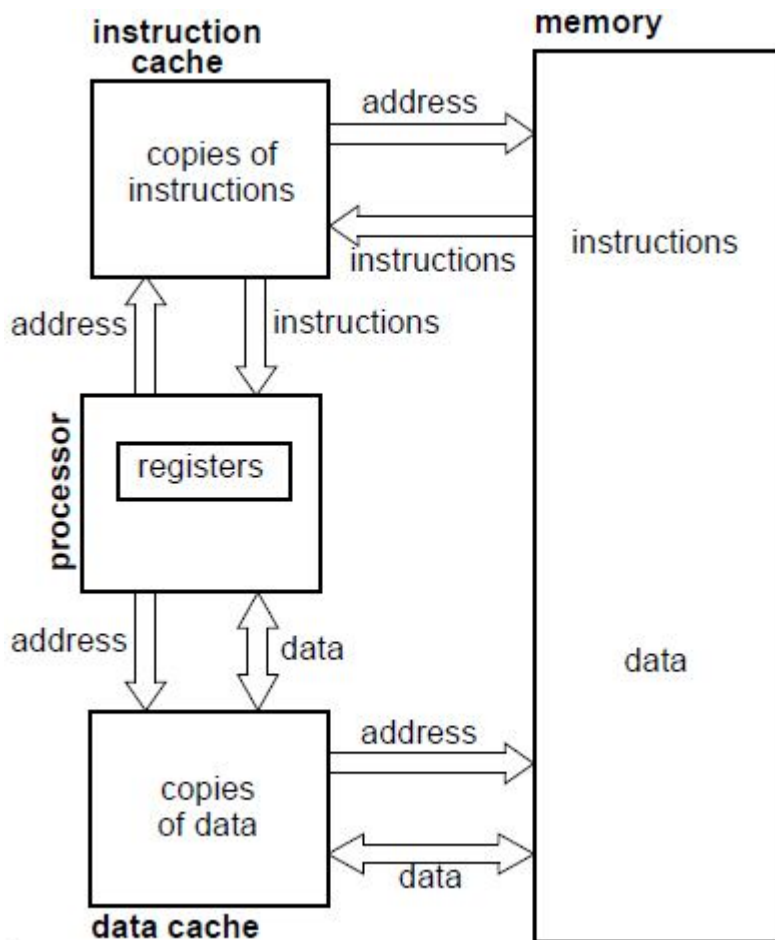
يعمل الكاش بمبدأ مرجعية المحل "**Locality**" وهي أن البيانات التي تمت الإشارة عليها مسبقاً في الذاكرة الرئيسية تميل إلى إمكانية الإشارة إليها مرة أخرى في وقت لاحق وهذا المبدأ يعتمد على عاملين اثنين هما :

Temporal Locality " التواجد الزمني المؤقت " : وفيه آخر تعليمة تم استدعاؤها من الذاكرة الرئيسية من المحتمل جداً أن يتم استدعاؤها مرة أخرى قريباً . والآخر هو **Special Locality** " التواجد المكاني المؤقت " : وفيه التعليمة القريبة من التعليمة المستدعاة من الذاكرة الرئيسية من المحتمل جداً أن يتم استدعاؤها هي الأخرى قريباً .

إن الـ Temporal يقترح أنه عند الحاجة لمعلومة من الذاكرة الرئيسية فإنه يتم الاحتفاظ بنسخة منها في الـ Cache وستبقى فيه إلى حين استدعاؤها مرة أخرى من قبل الـ CPU ، وإن لم يتم ذلك في وقت قريب فإن ذاكرة الكاش تتخلص من هذه المعلومة ، بينما الـ Special يقترح بأنه بدلاً من جلب المعلومة المطلوبة في حد ذاتها ، بل يتم جلب الـ Block كاملاً أي المعلومة المطلوبة والمعلومات المجاورة لها لإمكانية احتياجها هي الأخرى بعد استدعاء التعليمة السابقة وذلك لأنه قد تكون هذه التعليمات متتالية ومبنية على بعضها البعض في النظام .

المشاكل المتعلقة بالـ Cache Memory :

- كم عدد الـ Caches .
 - كيفية تحديد قراءة ما إذا كان الموجود هو miss أو hit .
 - إذا كان miss ولم يكن هنالك مساحة في الكاش يمكن أن نضيف فيها معلومات جديدة من الذاكرة الرئيسية كيف يقوم الكاش بعمليات استبدال المعلومات .
 - كيف يمكن المحافظة على الاتساق بين Cache و Main Memory أثناء عملية الكتابة (نسخ المعلومات للكاش) .
- حل هذه المشاكل هو عن طريق فصل البيانات والتعليمات في الكاش وهو بتقسيم ذاكرة الكاش الواحدة إلى قسمين ليكون جزءاً للتعليمات والآخر للبيانات ، أو باستخدام 2 Cache واحد للتعليمات والآخر للبيانات .



مزايا تقسيم الكاش :

- 1- إحداث اتزان أفضل في التفريق عند التحميل بين البيانات والتعليمات التي يتم إحضارها بناءً على آلية التنفيذ للبرنامج.
- 2- رخص التصميم والتنفيذ.

ميزة توحيد الكاش :

فض النزاع في الكاش بين عملية معالجة التعليمات ووحدات التنفيذ للتعليمات ، فالتعليمات المطلوب إحضارها يمكن أن تكون بياناتها تحت عملية تنفيذ في نفس اللحظة وهذا ما سيتم التطرق إليه في الفصل القادم وبعد ذلك يمكن الإجابة على السؤال التالي وهو كيف يمكن القضاء على ال Hazard ؟ - سؤال امتحان جزئي " دكتور أشرف " فصل الخريف 2010 -

تنظيم الكاش " Cache Organization " :

مثال : يوجد Cache حجمه 64KB والبيانات تنتقل بين ال Cache وال Main Memory بـ blocks حجمها 4B وال Cache شكله منظم على هيئة lines سعة كل line هو 4B وحجم ال Main Memory هو 16MB وكل بايت يتم عنوانه بواسطة 24bit ($2^{24} = 16M$) ، كما وأن ال Cache يتضمن 2^{14} من lines يساوي (16k) وال Main Memory يتضمن 2^{22} من ال blocks .

الأسئلة : - عندما يتم إحضار block من Main Memory إلى Cache ففي أي line سيوضع هذا ال block ؟

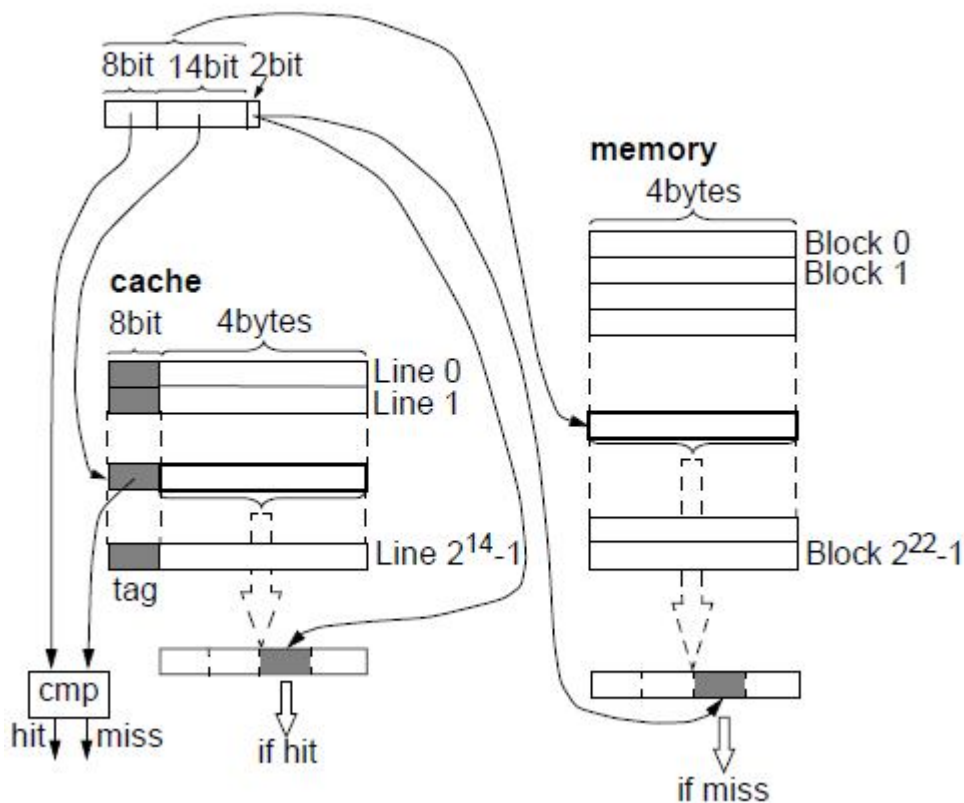
- عندما ننظر إلى محتوى عنوان معين بالذاكرة هل نقوم بالبحث في كل خطوط ال Cache ؟

- كيف لنا أن نحدد إذا ما تم إيجاد البيانات بال Cache " Hit " أو لم يتم إيجادها " Miss " .

للإجابة على كل هذه الأسئلة فإن ذلك لا يتم إلا بطريقة معينة لتخطيط ذاكرة Cache وهناك ثلاثة طرق لذلك

Associative Mapping (3**Set-Associative Mapping (2****Direct Mapping (1**

وقبل الغوص في هذه الأنواع سنقوم بتوضيح بسيط للمثال الذي ذكر أعلاه ؛ وللتوضيح فإن البيانات الموجودة في الذاكرة الرئيسية (M.M) هي نفسها مقسمة إلى blocks وال block كما تم تعريفه مسبقاً هو أكثر من كلمة (معلومة) واحدة بل هو مجموعة من الكلمات وقد فُرض في المثال أن حجم ال block في M.M هو 4Byte ، وبم أن حجم كل block في M.M هو 4B فإن ذاكرة ال Cache التي تعمل على الاحتفاظ بنسخ من البيانات المنقولة من M.M إلى المعالج - مع العلم بأن هذه البيانات المنقولة تنتقل على شكل blocks - هي من الطبيعي جداً أن تستقبل البيانات المنقولة من M.M على شكل blocks ، ولأن ال M.M يقسم يحتفظ ببياناته على شكل blocks حجم الواحد فيها 4B ، فإن ال Cache هو الآخر يكون مهيناً بتقسيمات على شكل خلايا كل خلية فيها سعتها 4B لأنها لن تستقبل غير هذا الحجم - وهو حجم ال block - فعلى هذا الأساس تسمى كل خلية باسم line . كذلك يوجد أمر آخر وهي عدد ال bits التي تعنون ، ومعنى كلمة تعنون أي التي تحدد أي ال block المطلوب جلبه من الذاكرة الرئيسية لل Cache ويحدد أيضاً أي التعليمات المطلوبة من قبل ال CPU بالتحديد في هذه اللحظة من خلال هذا ال block الذي يضم التعليمات الحالية وبعض التعليمات الأخرى ؛ كذلك تعمل هذه ال bits والتي عددها في هذا المثال 24b على تحديد أي من ال lines بالكاش التي يمكن أن تحتوي على ال block المطلوب سواء أن وجد بها ال block " hit " أم لا " miss " وتحديد التعليمات المطلوبة في حد ذاتها بداخل ال block أيضاً .



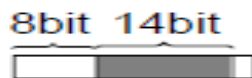
هذا النوع يسمى التخطيط المباشر وفي هذا المثال قد ذكر أن الـ Cache يتضمن 2^{14} من lines ما تساوي حجمه بالتمام وهو 16k ويمكن تفسير ذلك على أن الـ Cache خطوط تستقبل بها الـ blocks من الـ M.M عند حدوث حالة الـ Miss – أي طلب المعالج البيانات من الكاش ولم يجدها – وللتأكيد على 2^{14} هو أن :

$$16\text{kB} \div 1000 = 16384\text{B} = 2^{14}$$

ولأن الخط الأول كان يبدأ بـ 0 فلذلك فإن الخط الأخير سيكون $2^{14}-1$. وكذلك يمكن النظر إلى رسم الـ Cache فإنه يمكن التعرف على الخطوط " lines " التي تستقبل فيها الـ blocks من الـ M.M وعددها موضح بالشكل وفيه حجم كل line مساوياً لحجم كل block وهو 4B .

من المثال قد تعرفنا على أن العنوانه تتم بـ 24bit وكما قد ذكر بأن 22bit منها هي التي تحدد موقع الـ block في الـ M.M، وقد ذكر أيضاً أنه يتم استخدام 14b منها فقط لتحديد line في الـ Cache الذي يمكن أن يحتوي على block المطلوب .

إن كل line بالـ Cache يختلف تماماً عن line الآخر بسبب امتلاكه عنوان فريد من نوعه يجعله متميزاً عن الآخر ويسهل إمكانية التفريق بينهم ؛ وبالعودة إلى 24bit والتي تستخدم في العنوانه فإننا سنتعرف على أنها تتكون من ثلاثة حقول :



الحقل الأول وهو الأقل أهمية (في هذا المثال حجمه 2bit) وتعمل على تحديد الـ Byte بعينه المطلوب حالياً من قبل المعالج من بين الـ block كاملاً ؛ أما بالنسبة لبقية العنوان 22bit فهي تعمل بشكلين مختلفين ، حيث أن الـ 22bit تستخدم كاملاً لتحديد الـ block المطلوب في الـ M.M في حالة Miss أما في حالة Hit فإنه يقسم في الـ Cache إلى قسمين :

- الـ bits الأقل أهمية -في هذا المثال ذُكر بأن حجمه 14 - ومن خلالها يتم تحديد الـ line المطلوب أن يسلكه الـ block ليصل إلى M.M وهو الحقل الأوسط في العنوان ويسمى slot .

- الـ bits الأكثر أهمية - في هذا المثال ذكر بأن حجمه 8- ومن خلالها يتم اختيار الـ block المطلوب ليناسب الـ line المشابه له ؛ وبتوضيح أكثر فإنه عند استخدام slot الذي حدد الـ line الـ Cache الذي ستنقل من خلاله البيانات ، تعمل الـ 8bit بعد ذلك لتحديد وجود block بداخل الـ line أم لا وتسمى 8 bit بالـ Tag .

Tag	Slot	Word
-----	------	------

قد ذُكر في هذا المثال بأن M.M يستخدم فقط 22 من 24 bit لعنونة الـ block المطلوب ومن هذا الرقم يمكن استنتاج أن عدد الـ bits التي نحتاجها لتحديد word - المعلومة الحالية التي يحتاجها المعالج من بين معلومات الـ block الكاملة - هو 2bit (24-22=2) ؛ وكما قد ذكر في المثال بأن الـ Cache يحتاج 14 bit لتحديد lines ، فلذلك فإن 8 bit = Tag (22-14=8) وهكذا يمكن تحديد Tag ، slot ، word في Direct Mapping .

أما في الحالة الأخرى فإنه يطلب المعالج البيانات من الـ Cache ، فتعمل الـ 14 bit بالبحث عن خط العنونة المطلوب الـ line فإذا أن تجده فارغاً وفي هذه الحالة تكون Miss فيقوم بطلب الـ block من M.M ليتم وضعه في هذا الـ line ، أو أن يجد الـ block المطلوب بداخله وتكون الحالة Hit فتذهب الـ 8 bit لتخبر الـ Compiler بوجود الـ block في الـ Cache فتعمل الـ 2 bit بعد ذلك لتحديد البيانات المطلوبة حالياً من قبل CPU في هذا الـ block الذي تم إيجاده ؛ أما في حالة وجد الـ 14 bit خط العنونة بدون الـ block المطلوب أي Miss فإنها تخبر الـ Compiler بنفسها عن عدم وجود الـ block لتنتقل الـ 8bit + 14bit إلى الـ M.M لتحديد الـ block المطلوب ، وتعمل 2 bit بعد ذلك على التعرف على البيانات المطلوبة حالياً من قبل المعالج.

مزايا Direct Mapping :

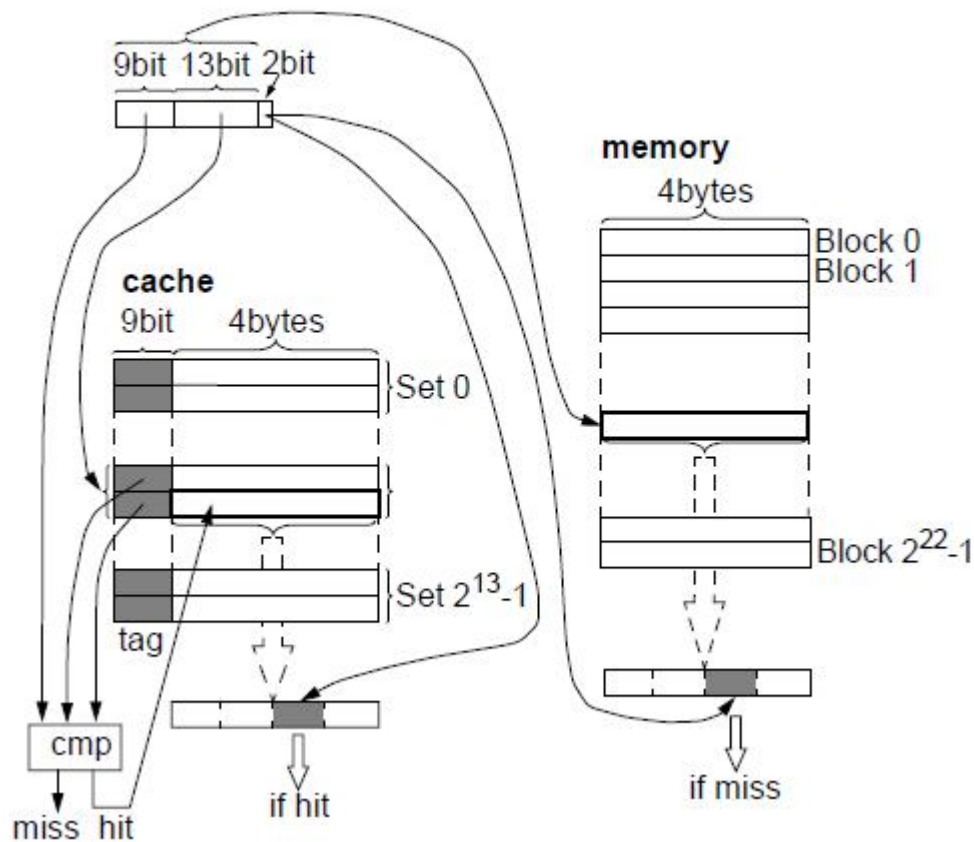
1- بسيط ورخيص.

2- حقل الـ Tag فيه صغير.

3- الوصول فيه سريع جداً .

عيب Direct Mapping :

هو أن للـ Data مكان واحد فقط في الـ Cache وفي حالة نريد جلب بيانات متتابعة فإنه يلزم جلب كل مرة Data واحدة وإزالتها ثم وضع الأخرى وإزالتها بعد ذلك وهكذا .

: Set-Associative Mapping -2Two-way set associative cache

هذا النوع يسمى التخطيط ذو المجموعات المترابطة إذا كانت الحالة Miss فإن الـ block يوضع في إحدى خطوط الـ Cache المنتمية لـ set معينة و التي قام بتحديد لها حقل الـ 13 bit في عنوان الذاكرة ، ويتم استخدام خوارزمية التبديل لتحديد أي خطوط الـ set التي يتم اختيارها لوضع block البيانات فيها. أما بالنسبة لعنوان الذاكرة هنا فإنه يفسر بثلاثة حقول كما في Direct Mapping ؛ على أي حال فإن الـ 13 bit هنا يتم استخدامها لتحديد الـ set المناسبة من Cache ، وبالمقابل فإن Tag يقوم بالتحقق لتحديد وجود Hit أو Miss ، فإن كان Hit يتم استخدام المنطق Logic لتحديد الـ line من بين الـ lines الـ set الموجودة بالـ Cache .

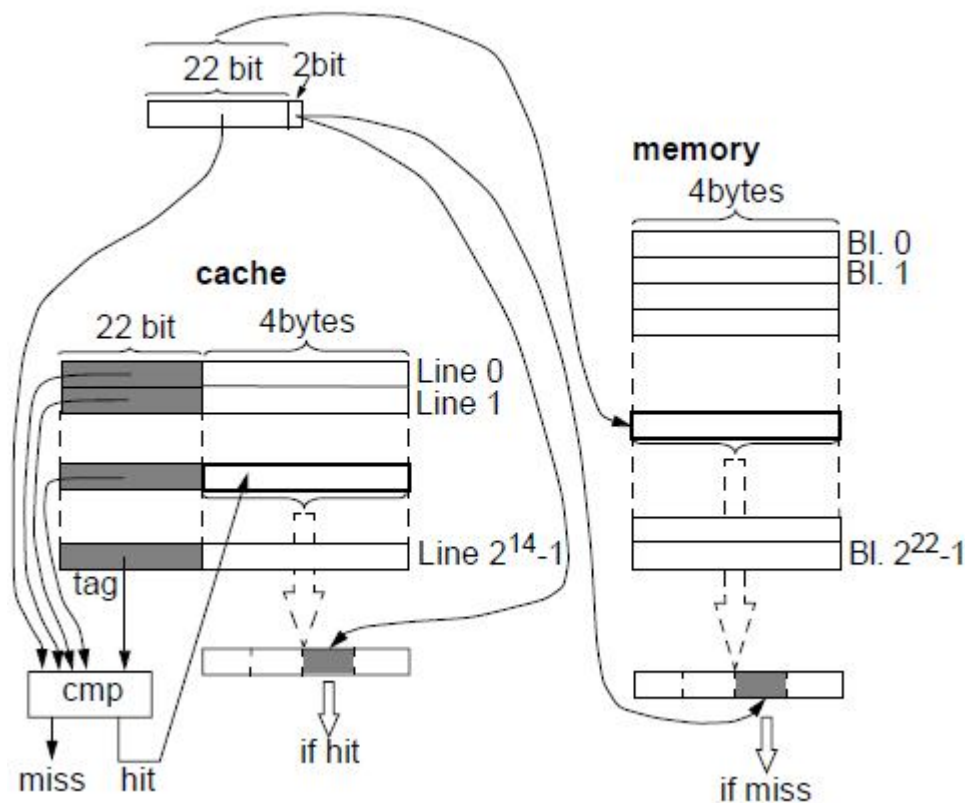
إن عدد الخطوط في المجموعة " set " يحدد من قبل المصمم ، فإن كان عدد خطوط المجموعة 2 كان ذلك ما يسمى بـ Two-way set associative mapping ، وإن كان عدد خطوط المجموعة 4 سمي ذلك Four-way set associative mapping .

مزايا الـ Set-Associative mapping :

- 1- حقل الـ tag صغير.
- 2- سرعة وصول.
- 3- بسيط نسبياً .

فتخطيط set-associative يحاول القضاء على العيب الرئيسي للـ Direct-mapping حيث يوجد مرونة نوعاً ما فيما يتعلق بعملية الاستبدال في الخطوط عندما تتم قراءة الـ Block الجديد في الكاش.

وتعتبر أجهزة الكاش المخططة بـ set-associative أكثر تعقيداً من المخططة بـ Direct .

: Associative Mapping -3

يسمى هذا النوع بالتخطيط المترابط الكامل ، وفيه إذا كانت الحالة miss فإن الـ block سيكون موضوعاً في واحد من الـ 2^{14} لخطوط الكاش ، وسيكون القرار لخوارزمية التبدل في اختيار الخط الذي سيستخدم.

بلوك الذاكرة يمكن أن يتم تعيينه لإحدى خطوط الكاش ، فإذا كان البلوك موضوعاً بالكاش في خط معين فإنه يتم تحديد هذا الخط عن طريق خوارزمية التبدل ، كذلك عنوان الذاكرة يفسر بحقلين فقط في منطق الكاش ، فالـ bits الأقل أهمية تحدد الـ byte داخل الـ block أي الكلمة المطلوبة بعينها في الجزء الداخلي للـ block .

كذلك يتم تفسير كل العناوين الباقية في منطق الـ Cache على أنها tag ، حيث أن كل الـ tags التي تقابل أي خط في ذاكرة الـ Cache يستوجب عليها القيام بعملية فحص منظم لتحديد ما إن كان الموجود هو hit أو miss ، فإن كان hit فإنه وباستخدام المنطق يتم معرفة الخط الحقيقي في الـ Cache .

مزايا الـ Associative mapping :

أحدث الـ associative مرونة هائلة فيما يتعلق بعملية استبدال الخطوط عند قراءة بلوك جديد في الذاكرة.

عيوب الـ Associative mapping :

- 1- معقد .
- 2- حجم الـ tag فيه كبير .
- 3- سرعة الوصول يمكن أن تحقق فقط باستخدام ذواكر الـ Associative عالية الأداء لأي كاش.

مسائل الـ Set-Associative Mapping :

يوجد بعض القوانين لحل المسائل وهي :

Tag : حجم البيانات المنتقلة (البلوك الواحد) بين الكاش والذاكرة الرئيسية يتم تسويته بالقوى المرفوعة لـ 2 والـ Tag يساوي تلك القوى ، وبتعريف آخر هو القوى المرفوعة لـ 2 التي تساوي حجم بلوك البيانات المنتقل بين الكاش و M.M . أي أن إذا كانت البيانات المنتقلة بين الكاش و M.M = 32؛ إذاً فإن $2^5 = 32$ فلذلك الـ Tag = 5

Slot: هو القوى للعدد 2 الناتجة من حاصل قسمة عدد lines على عدد مسارات التخطيط ways حيث أن lines هو النسبة بين قيمة حجم الكاش إلى حجم كل بلوك بيانات لهذا الكاش مع مراعاة توحيد الوحدات.

word : هو الباقي من عنوان الـ address المعطاة.

لإيجاد الـ Tag و Slot يتم استخدام هذه القوانين الثلاثة :

$$\text{Line} = \frac{\text{حجم الكاش}}{\text{عدد الكاشات} \times \text{حجم بلوك البيانات}} \quad (1)$$

$$\text{Line} = 2^{\text{slot}} \quad (2)$$

$$\text{Tag} = \frac{\text{حجم الـ word}}{2} \quad (3)$$

Ex : Given a computer system with the following cache architecture:

- Two on-chip Caches for data and instruction .
- Each cache of 8 Kbytes.
- Data transfer between cache and main memory is in blocks of 32 bytes.
- A main memory of 16 Mbytes, each byte is addressable by a 24-bit address.
- It uses 2-way set associative organization.

Draw the mapping function of this system, showing the detailed connection between the cache and the main memory.

الحل

Tag	line	block
-----	------	-------

$$\text{Tag} = n \ \& \ 32=2^n \quad \text{then} \quad 32 = 2^5 \quad \text{then} \quad \text{Tag} = 5$$

Slot : هو القوى للعدد 2 الناتجة من عدد lines ، وقانون lines هو حجم الكاش مقسوماً على حاصل ضرب حجم كل بلوك في عدد الكاشات المستخدمة في التخطيط .

$$\text{Lines} = \frac{8\text{KB} \times 1024}{32\text{B} \times 2\text{way}} = 128 \text{ Line}$$

$$128 = 2^7 \quad \text{then slot} = 7$$

أما بالنسبة للـ world فهو يساوي الباقي من العنوان الذي يساوي bit24 أي ([7+5]-24) = 12

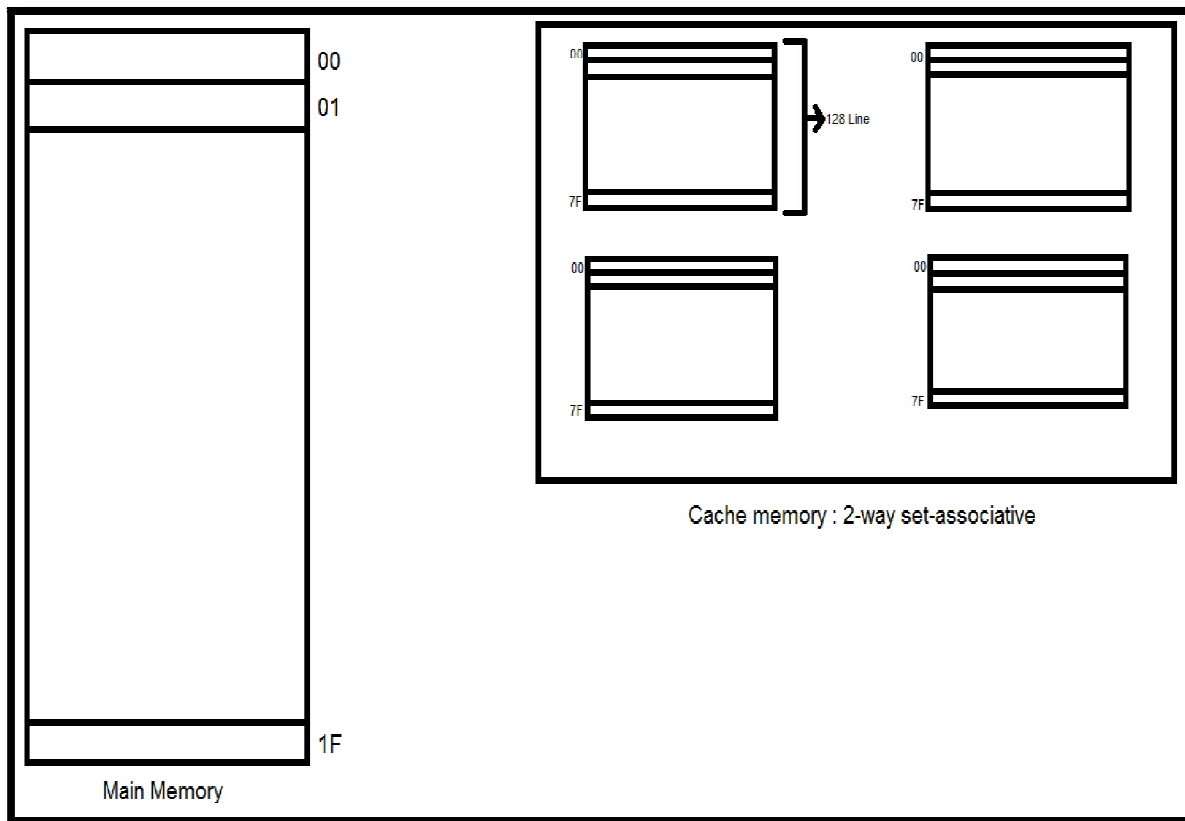
أما بالنسبة لعملية الـ Diagram أي الرسم فهو يكون كالتالي :

Tag = 5	Slot = 7	Word = 12
---------	----------	-----------

حيث أن slot تعنون ذاكرة الكاش ، و Tag تعنون الذاكرة الرئيسية .

1- slot = 7 يتم تمثيل الرقم 7 بسبعة ثنائيات للرقم 1 بهذا الشكل 1111111 وبعد ذلك يتم تحويله إلى النظام السادس عشر كالتالي : 1 1 1 1 1 1 1 ، وكذلك يتم تمثيل الـ Tag بثنائيات 11111 وتحويل للنظام السادس عشر لتكون 1F

إذا تعنون ذاكرة الكاش بـ 7F وعدد مسارات الكاش (Ways) هي 4 كما جاء بالسؤال و يتم عنوانه الـ M.M بـ 1F



إذا كان الكاش ميموري هو 2 way-set associative إذا يتم رسم Cache 2 وهنا بالفعل تم رسم Cache 2 ولكن أولى نقاط المسألة كانت تتحدث عن Two on-chip Caches for data and instruction أي أن الكاش المستخدم هو مقسم لذاكرتين واحدة للبيانات وأخرى للتعليمات ، ولأنه 2-way إذا فإنه يوجد ذاكرتين كاش مستخدمتين كلاً منهما مقسمة إلى قسمين واحدة للبيانات وأخرى للتعليمات .

Ex : Given a computer system with the following cache architecture:

- a cache of 16 Kbytes is used.
- Data transfer between cache and main memory is in blocks of 32 bytes.
- A main memory of 16 Mbytes, each byte is addressable by a 24-bit address.
- It uses 4-way set associative organization.

Draw the mapping function of this system, showing the detailed connection between the cache and the main memory.

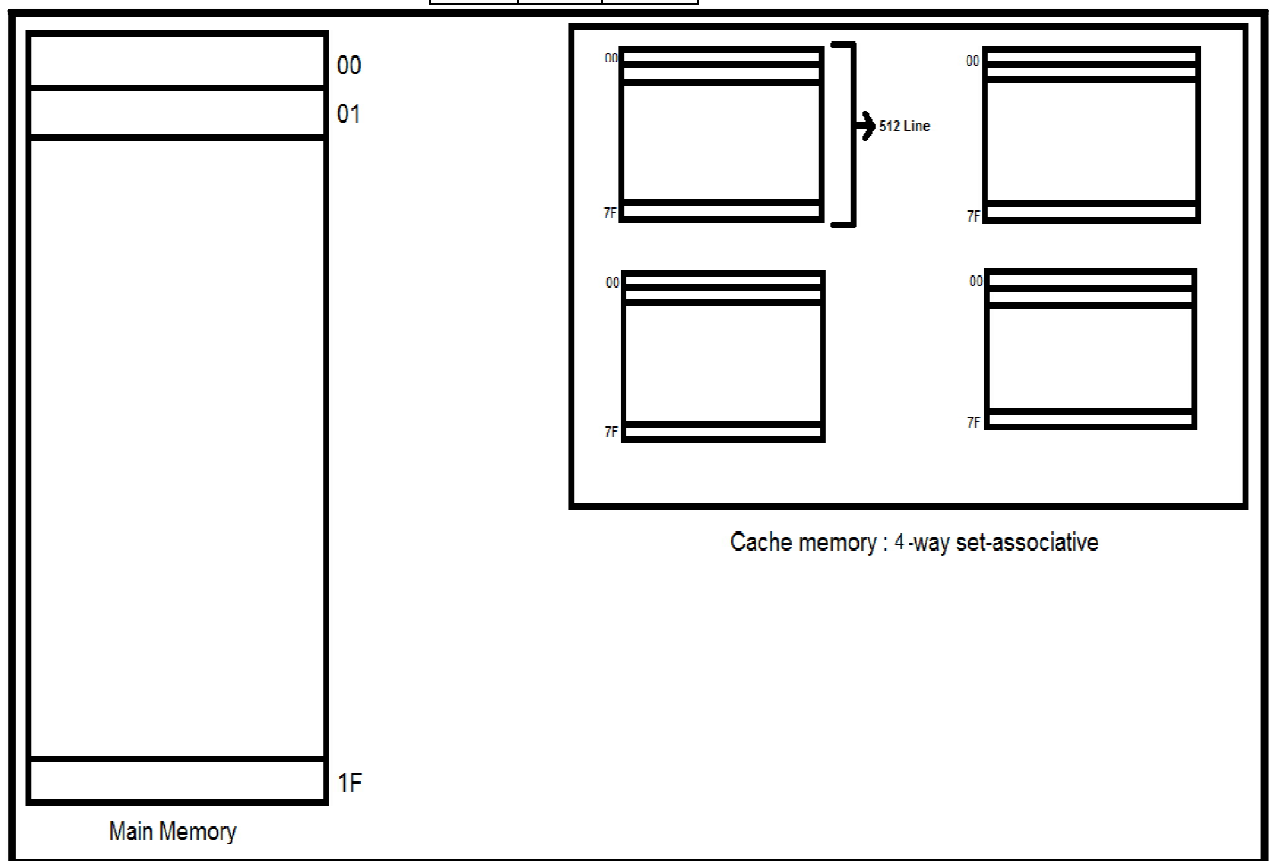
الحل

$$\text{Tag : } 32 = 2^5 \text{ then Tag} = 5$$

$$\text{Line : } \frac{16\text{KB} \times 1024}{32\text{ B}} = \frac{512}{4\text{ way}} = 128 = 2^7$$

$$\text{Then Slot} = 7$$

5	7	12
---	---	----



Replacement Algorithms " خوارزمية التبديل " :

عندما يتم وضع بلوك في الكاش ، فإن هذا البلوك سيخزن في إحدى خطوط الكاش التي لا بد من إحداث عملية استبدال فيها والمقصود بالاستبدال هو حذف البلوك من ذاكرة الكاش ووضع بلوك جديد بدلاً منه .

تستخدم هذه الخوارزمية في تخطيط الـ associative & set-associative لمعرفة في أي line بالكاش يمكن وضع الـ block الجديد ولكن لا يستخدم في الـ direct.

فعند الـ set-associative فإن الخطوط التي ترشح لعملية الاستبدال هي من بين خطوط الـ set أي أن الخيارات ستكون محدودة من بين خطوط المجموعة، بينما في associative فإن كل خطوط الكاش مرشحة لأن يتم إحداث عملية استبدال فيها.

1- Random Replacement " الاستبدال العشوائي " :

وفيها يتم اختيار أحد الخطوط المرشحة بشكل عشوائي ، أما الخوارزميات الأخرى فهي مبنية على معلومات متعلقة بأقدم استخدام للـ block في الكاش .

2- (LRU) Least Recently Used " أقل من استخدم أخيراً " :

يتم اختيار الخط المرشح الذي يحمل بلوك في الكاش أطول فترة ممكنة ولم يستخدم بعد.

3- (FIFO) First-In-First-Out " أول من يدخل هو أول من يخرج " :

يتم اختيار الخط المرشح الذي يحمل بلوك في الكاش أطول فترة ممكنة سواء أستخدم أو لم يستخدم.

4- (LFU) Least Frequently Used " أقل من تكرر استخدامه " :

يتم اختيار الخط المرشح الذي يحمل البلوك الذي تم اختياره كأقل واحد في عدد من المرات.

إن Replacement Algorithm لإدارة أي كاش يجب أن تنفذ في الهاردوير بشكل منظم لتظهر فاعليتها ، إن LRU هو الأكثر فاعلية وهو بسيط نسبياً في التنفيذ وفي إعطاء النتائج الجيدة ، كما وأن الـ FIFO هو بسيط في التنفيذ ، بينما الاستبدال العشوائي يعتبر الأبسط في التنفيذ وذو نتائج هائلة ومتميزة.

Write Strategies (سياسات الكتابة في الكاش):

من الضروري قبل الشروع في تنفيذ عملية استبدال في أحد خطوط الكاش التحقق أولاً إن كان قد ثمة تعديلات على هذا الخط أم لا ، بحيث أن البلوك الموجود في M.M و الذي سينتقل إلى خط كاش معين من المفترض أنهما يحملان ذات القيمة أي نفس العنوان ؛ فالمشاكل تكون موجودة عندما تكون هنالك تعديلات على خط الكاش قبل عملية الاستبدال (أي البلوك الموجود داخل خط معين بالكاش يُستبدل كل فترة) الحل هنا هو استخدام عدة استراتيجيات هي:

1- Write-through " الكتابة خلال " : عند الـ Hit فإنه تكتب المعلومات في كل من خط الكاش وفي البلوك الموجود بالـ

M.M في وقت واحد وذلك لتكون عملية الكتابة على الكاش والـ M.M متناسقة ومتتالية وعدم حدوث خطأ في الكتابة بالـ

M.M. وعند الكتابة فإن المعالجات دائماً ما تقوم بتبطين سرعتها لتتوافق مع سرعة الـ M.M.

2- Write-through with buffered write "الكتابة خلال باستخدام ذاكرة المخزن الاحتياطية :

هي مشابهة لسابقتها ولكن بدلاً من أن تبطئ المعالجات عملية كتابتها المباشرة على M.M فإنه يتم تخزين العنوان المكتوب والبيانات على ذاكرة ذات سرعة عالية في الكتابة اسمها Buffer ، حيث ذاكرة "write buffer" تنقل البيانات إلى M.M طالما المعالجات لازالت بحاجة إلى المعلومات المدعومة من M.M .

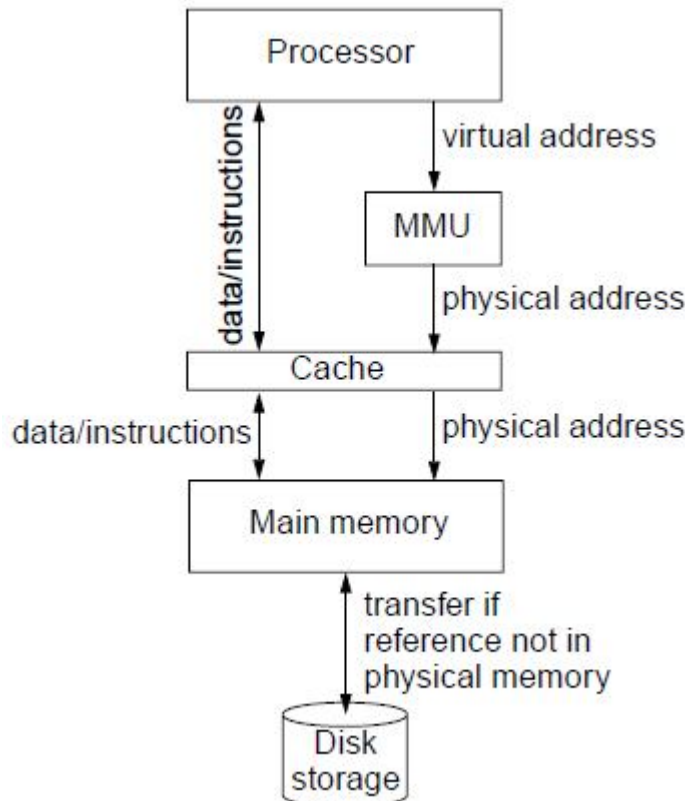
3- Copy-back " طريقة اكتب خلف " : يتم تحديث فقط العمليات المكتوبة على الكاش لكي تبقى متوافقة مع M.M فخطوط الكاش لها القدرة على أن تتذكر ما إذا تم تحديثها باستمرار ، فإن تم استبدال خط من الكاش فإن محتواه يتم نسخه وإرجاعه إلى الذاكرة ،

إن الأداء الجيد ما يحتاج أجهزة أكثر تعقيداً ، فمشاكل إحداث التوافق هي معقدة جداً وصعبة الحل في أنظمة المعالجات.

Virtual Memory " الذاكرة الافتراضية " :

المساحة التي نحتاجها غالباً لعنونة برنامج بالكامل قد تكون أكبر من تلك المتاحة في M.M ، لذلك يتم إدخال فقط جزء من البرنامج في M.M ويبقى باقي البرنامج مخزناً على ذاكرة ثانوية (HARD DISK) وعند الحاجة إليها يتم استبدالها تلك التي لم يعد لها حاجة في الـ M.M بتلك التي نحتاجها الآن من الذاكرة الرئيسية ، فتسترجع تلك إلى الذاكرة الثانوية ويتم جلب جزء آخر من البرنامج وهو الجزء الذي نحتاجه في هذه اللحظة ؛ إن حركة البرامج والبيانات بين M.M وذاكرة التخزين الثانوية تتم تلقائياً بواسطة أنظمة التشغيل ، هذه التقنيات تسمى Virtual Memory .

إن العنوان الثنائي الذي يتم إصداره من قبل المعالج يُعتبر افتراضياً أو تخيلياً حيث أن المساحة التخيلية للعنوان هي أكبر من تلك الفيزيائية " الحقيقية " المتاحة في M.M.



إذا قام العنوان التخيلي بالتأشير على برامج أو بيانات موجودة حالياً في ذاكرة حقيقية (M.M أو Cache) فإن الموقع المناسب يكون بالوصول المباشر باستخدام العنوان الفيزيائي (الموقع في الذاكرة ، ماعدا ذلك فإن الجزء الأول من البرنامج أو البيانات المقسمة هي من ستنتقل من الذاكرة الثانوية.

إن وحدة الهاردوير الخاصة وما تعرف بـ MMU أي Memory Management Unit هي من تعمل على ترجمة العنوان الافتراضي/التخيلي إلى وحدات فيزيائية .

يتم تقسيم المساحة الافتراضية للبيانات بالتساوي بحجم ثابت تعرف باسم pages ، كذلك تكون الـ M.M منظمة على شكل تسلسلي من الإطارات Frames ، وكل page يمكن أن يخصص لأي Frame متاح ، وتكون عملية التخزين بشكل منظم (Page Size = Frame Size).

يعتبر الـ page أساس وحدة المعلومات التي تتحرك بين M.M والذاكرة الثانوية باستخدام نظام الـ Virtual Memory .

يتألف البرنامج من مجموعة من الـ pages والتي تخزن على disk ، وفي أي وقت يتم تخزين البعض فقط من pages على M.M وهو ما يسمى بعملية الارتحال Demanding Paging ؛ حيث أن أنظمة التشغيل هي المسؤولة عن تحميل pages المستبدلة والتي لم تعد لها حاجة في بقائها على الـ M.M وجلب تلك التي نحتاجها في هذا الوقت.

عندما يؤثر CPU على موقع معين يطلب منه page غير موجودة في M.M ، فإن الـ page الموجودة على M.M يتم تحميلها إلى DISK وتستبدل بالـ page المطلوبة.

إن وصول الكلمة إلى الذاكرة يتضمن ترجمة للعنوان الافتراضي Virtual address إلى عنوان حقيقي Physical address

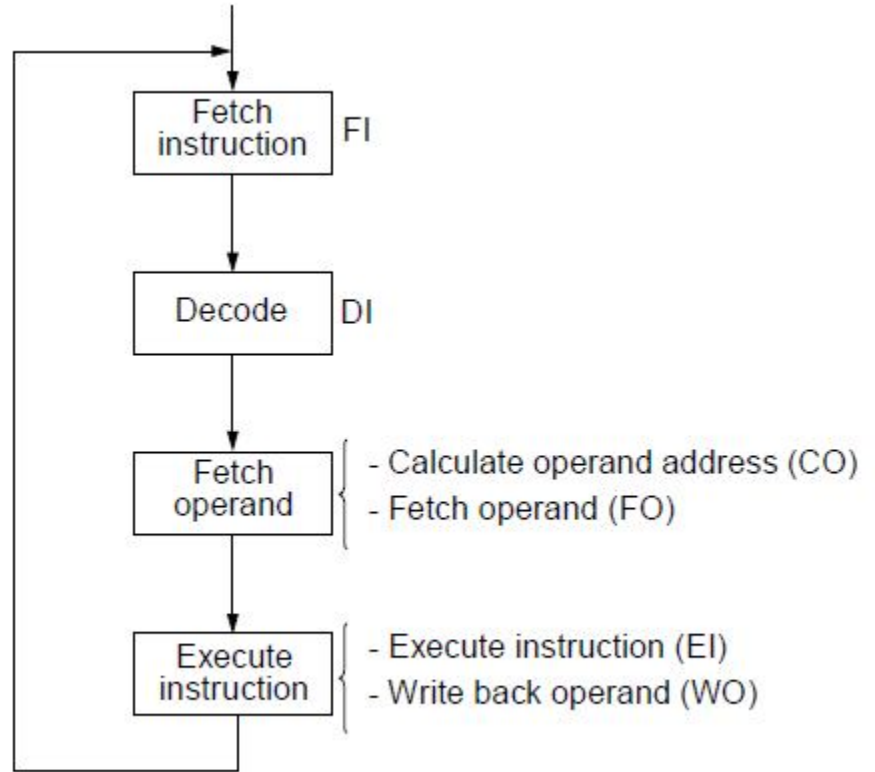
- Virtual address : Page number + offset

- Physical address : Frame number + offset

Instruction Pieplining I

Chapter 3

حلقة التعليم (The Instruction Cycle) :



إنه ولتنفيذ أي تعليمة لعملية حسابية مثلاً فإنها تمر بسنة مراحل ، وكما ونعلم بأن عمليات المعالجة تتم دائماً في CPU ، لذلك فإن المرحلة الأولى هي ما تسمى بـ FI أي إحضار التعليمة المطلوبة من M.M إلى CPU ، بعد ذلك يتم تحليل التعليمة وترجمتها لمعرفة المطلوب منها في وحدة التحكم بـ CPU وهذه المرحلة تسمى DI ؛ بعد ذلك يتم حساب عملية هذه التعليمة في وحدة الحساب والمنطق وهي ما تسمى بـ CO ، ثم استدعاء المعامل المطلوب إضافته في العملية الحسابية مثل قولنا MOV R1,X أي ضع X في السجل رقم 1 ، والمطلوب استدعاء قيمة X من الذاكرة الرئيسية ، ولكن MOV R1,R2 المطلوب هنا هو جلب قيمة R1 وذلك بالطبع لن يكون من M.M بل سيكون من السجلات ، وتسمى هذه المرحلة FO . المرحلة الخامسة هي إضافة ذلك المعامل إلى التعليمة لإظهار الناتج النهائي وهو ما يسمى بتنفيذ التعليمة EI، وأخيراً تخزين أو كتابة الناتج في M.M وهو WO.

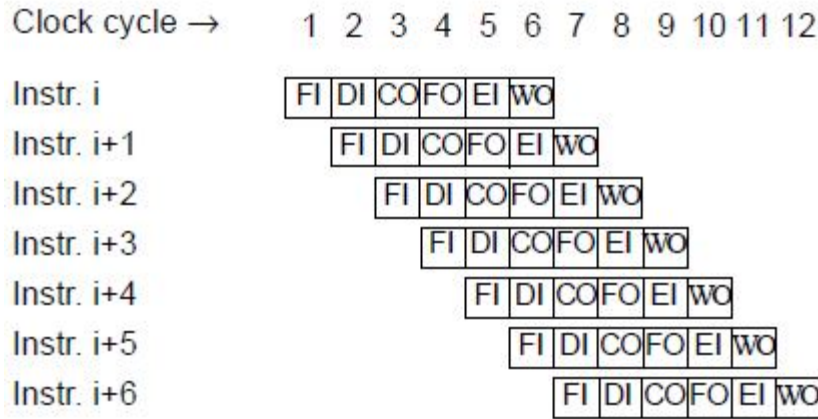
تقنية خطوط تجزئة تنفيذ التعليمات Instruction Pipelining :

هي تقنية تستخدم في صناعة المعالجات ، وتهدف إلى تحسين أداء المعالج وتسريع عمله بإعطائه نوعاً من أنواع التنفيذ التفرعي لمجموعة التعليمات الخاصة به .

يمكننا تشبيه تقنية Pipeline في المعالجات بغرفة لغسل الثياب حيث تمر عملية الغسيل بثلاث مراحل (الغسل والتجفيف والتوضيب) ، فعلى سبيل المثال أننا نريد أن نغسل 4 دفعات من الملابس بشرط أنه لا نبدأ في مرحلة غسل الدفعة الثانية قبل الانتهاء من الدفعة الأولى وتوضيبها ؛ بهذا الشكل سنأخذ وقت طويل في عملية الغسل ، ولكن إذا ما قمنا بعملية الغسل

بطريقة أخرى بحيث دخل الدفعة الثانية في مرحلة الغسل حالما تبدأ الدفعة الأولى بمرحلة التجفيف ؛ وعند انتقال الدفعة الأولى لمرحلة التوضيب فإنه تنتقل الدفعة الثانية إلى مرحلة التجفيف وتدخل الدفعة الثالثة في مرحلة الغسل ، وهكذا بهذا الشكل نكون قد قمنا بتقليص الوقت المستخدم في غسل الدفعات الأربعة من الملابس .

يشبه هذا تماماً تقنية Pipeline في المعالجات ، ففي المعالجات التي لا تستخدم تقنية Pipeline لا يتم البدء في تنفيذ التعليمة التالية من البرنامج إلا بعد الانتهاء تماماً من تنفيذ كافة العمليات التي يتطلبها تنفيذ التعليمة السابقة (FI,DI, ... ,WO) . يمكن تسريع المعالجات باستخدام تقنية Pipeline بتجزئة العمليات التي يقوم بها النظام بحيث يمكن لتعليمة جديدة أن تبدأ في التنفيذ حالما تنتقل التعليمة التي تسبقها من المرحلة الأولى إلى المرحلة الثانية ، وبهذا الشكل يتم تحسين سرعة أداء النظام.



مخاطر تقنية خطوط التجزئة Pipeline Hazards :

يوجد بعض المخاطر في تقنية pipeline التي تحدث في حالات معينة لتحول بين تنفيذ التعليمة الجديدة في لحظة معينة ، هذا التوقف يسمى stall ؛ عندما يحدث هذا stall في لحظة معينة لعملية في تعليمة ما ، فإنه لا يمكن أن تنفذ أي عملية جديدة لهذه التعليمة في هذه اللحظة ، مما يعني أنه عند توقف عملية عن التنفيذ بسبب مشكلة ما في لحظة معينة ، فإن هذا التوقف يؤثر على كل العمليات التالية ويسبب في توقفها عن العمل في هذه اللحظة .

أنواع الـ Hazards :

1- Structural Hazard.

2- Data Hazard.

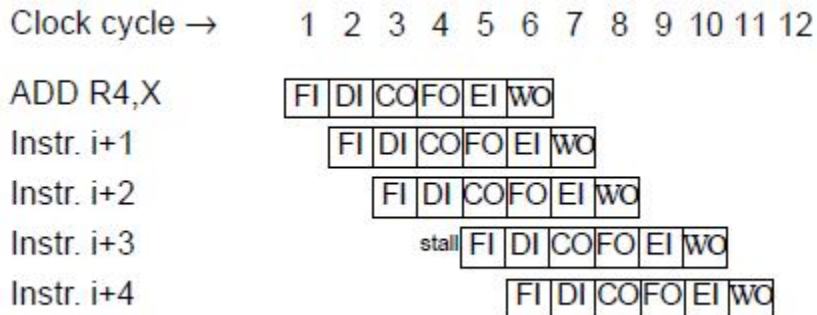
3- Control Hazard.

العوائق الإنشائية Structural Hazards :

تحدث المخاطر الإنشائية عندما يتم طلب معامل ما من نفس المكان في نفس اللحظة مما يسبب في حدوث stall للتعليمة الجديدة ، و stall تعني توقفها عن العمل واستمرارها في المرحلة التالية .

وعلى سبيل التوضيح نضرب بمثال لـ ADD R4,X في 5 مراحل متكررة ، يعني المطلوب هو إضافة قيمة X إلى السجل R4 بخمس مرات ، هنا نلاحظ بأنه عند جلب التعليمة من الذاكرة FI في المرحلة الأولى لن تكون هنالك مشكلة ، وفي اللحظة

الثانية عند جلب التعليمات الثانية المرحلة الثانية لن يحدث شيء ، وكذلك الأمر في الثالثة ، ولكن عند الوصول إلى المرحلة الرابعة وأراد المعالج جلب التعليمات من الذاكرة فإنه وفي نفس اللحظة يوجد عمل آخر وهو جلب المعامل X من الذاكرة في التعليمات الأولى ، ولذلك لا يمكن الوصول للذاكرة مرتين في نفس اللحظة ، فيحدث stall " توقف " للتعليمات الرابعة لتبدأ عملها في اللحظة الزمنية التالية .

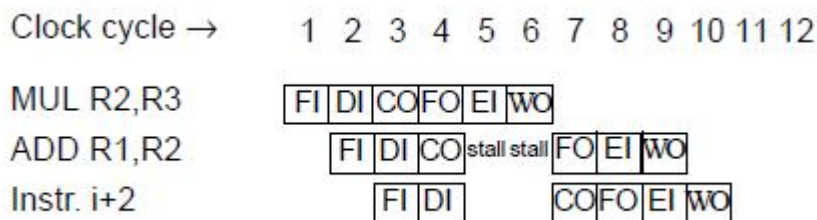


يمكن أن يتم تجنب Structural Hazard من خلال تقسيم الكاش إلى جزأين ، أحدهما للبيانات والآخر للتعليمات ؛ فعندما نحتاج لجلب المعامل يكون من ذاكرة وعندما نرغب في جلب التعليمات تكون من ذاكرة أخرى فلا يحدث Hazard .

عوائق البيانات Data Hazards :

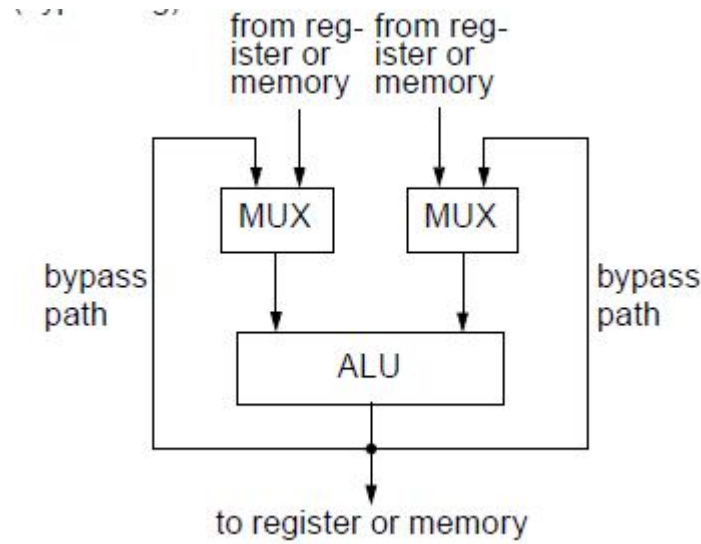
يحدث هذا النوع من Hazard عندما يكون لدينا تعليمتين مثلاً ، التعليمات التالية تريد أن تجلب معام (FO) لم يجهز بعد في التعليمات السابقة ولم يتم الانتهاء من تنفيذه (EI) وتخزينه (WO) في M.M .

فعلى سبيل المثال لدينا تعليمتين الأولى هي MUL R2,R3 أي $R2 \leftarrow R2 * R3$ والتعليمات الثانية هي ADD R1,R2 أي $R1 \leftarrow R1 + R2$ ، حيث سيحدث Hazard في التعليمات الثانية عندما يطلب في المرحلة الرابعة في هذه التعليمات جلب المعامل R3 أي القيمة الموجودة في السجل رقم 3 ولكن في التعليمات التي تسبقها نلاحظ أن قيمة R3 يجرى عليها عملية حسابية أي أنها في لحظة تغير حالية ولذلك قيمتها ليست معروفة بعد ، أي أنه عندما يرغب في جلبها من الـ M.M فإنه لن يجدها وذلك لأنه تجرى عليها عملية الآن ولم يتم تخزين هذه القيمة الجديدة في الـ M.M بعد ، مما يعني أن القيمة المطلوبة ليست موجودة حالياً في الذاكرة الرئيسية لأنها في طور التنفيذ (EI) ، في اللحظة الزمنية التي تليها يستمر الـ stall أيضاً وذلك لأنه يتم حالياً في هذه الفترة كتابة الناتج في الذاكرة الرئيسية وعند كتابة الناتج يمكن بعد ذلك في المرحلة الزمنية التي تليها أن يتم احضار القيمة للتعليمات التي انتظرت لفترتين زمنيتين كما بالشكل :



نلاحظ في الشكل السابق توقف التعليمات الثالثة على الرغم من عدم وجود أي Hazard ولكن قد ذكر مسبقاً بأنه عندما يكون هنالك stall فإن كل التعليمات التي تليها يجب أن تتوقف ولا تعمل في هذه اللحظة الزمنية .

بالنسبة لـ Data Hazard فيمكن تجنب حدوثه باستخدام تقنية تسمى بـ forwarding إعادة التوجيه (bypassing) من خلال التمرير أو التجاوز .



إذا اكتشف ALU بوجود Data Hazard ولكن لم يحدث هذا الـ Hazard بعد ، فإنه يختار النتيجة المحالة إليه على أنها هي المعامل الداخلة إليه في العملية ، وذلك بدلا من أن يتم قراءة القيمة سواء من الذاكرة أو من السجل .

Clock cycle → 1 2 3 4 5 6 7 8 9 10 11 12

MUL R2,R3

FI DI CO FO EI WO

ADD R1,R2

FI DI CO stall FO EI WO

بعد استخدام هذه الطريقة سيكون الخطأ هو عند التنفيذ فقط ، وبالتالي تم تخفيض الـ Hazard ليكون في دورة زمنية واحدة.

Q1: Mention different typed of Mass Storage ?

- 1- Magnetic Disk.
- 2- Floppy Disk.
- 3-Compact Disk.
- 4- Magnetic tape.
- 5- Magnetic Drums.

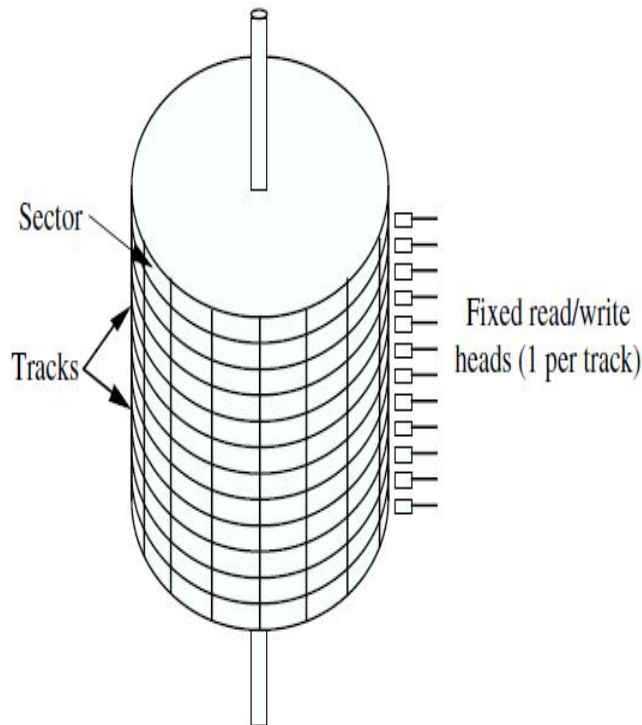
Q2:Draw the Structure of a Magnetic Disk & The Magnetic Drum,compare between them?

Figure 8-6 A magnetic drum unit.

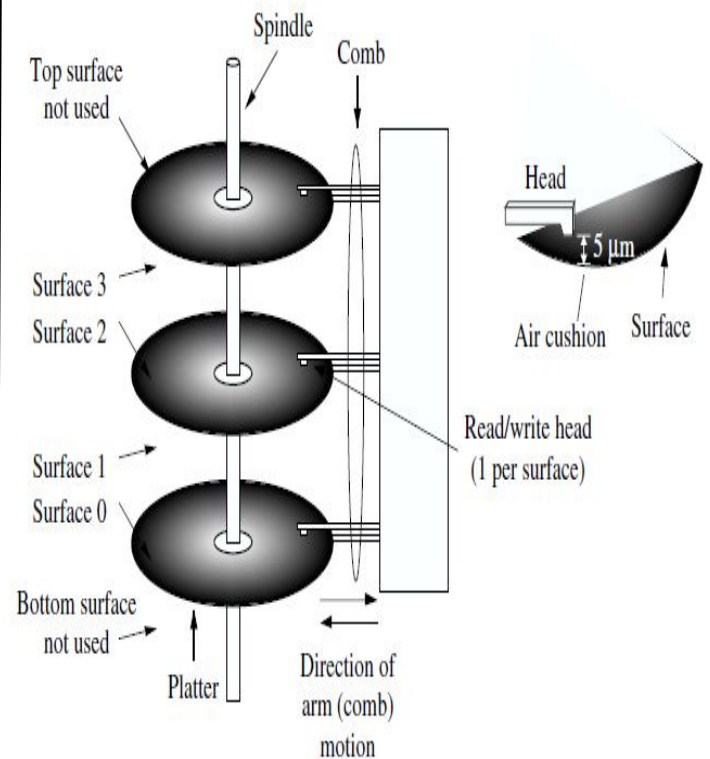


Figure 8-1 A magnetic disk with three platters.

Magnetic Drum	Magnetic Disk
لديه حركة واحدة فقط	لديه حركتين (دائرية وأخرى أفقية)
لا تأخذ وقت أطول (زمن القراءة) وبالتالي تكون أسرع	زيادة الحركة الميكانيكية فيه تأخذ وقت
السعة التخزينية المستخدمة مع Main Frame أكبر	السعة التخزينية أقل
السعر أكبر لأن سعته أكبر	السعر أقل

وجه الشبه : كلاهما يستخدمان ظاهرة المغنطة حيث يماغنط البيانات الموجودة بهما .

Q3: Show the Organization of disk platter with 1:2 inter leave factor ?

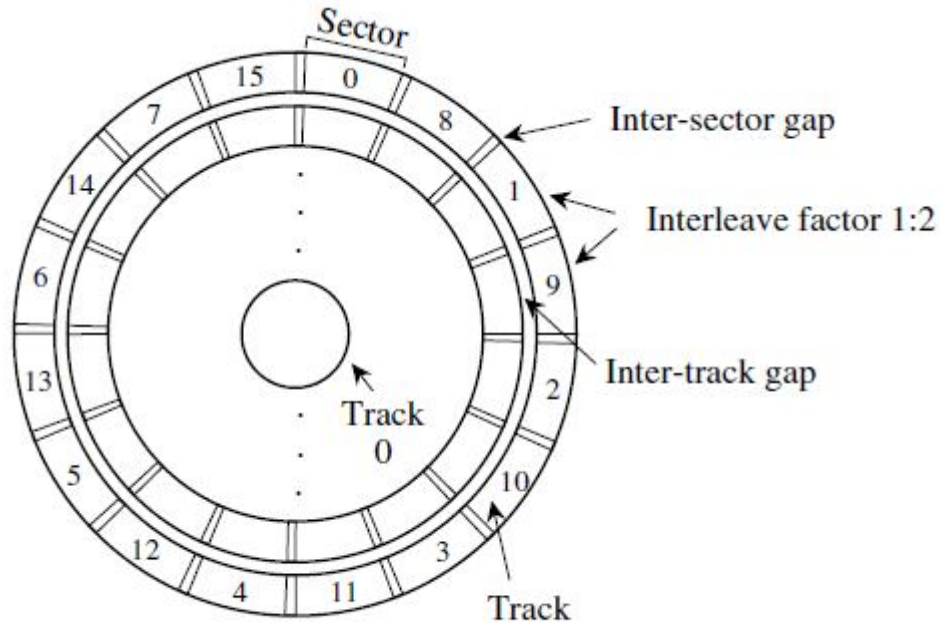


Figure 8-3 Organization of a disk platter with a 1:2 interleave factor.

Q4: Caculate the storage Capacity of disk given that :-

- The number of byte/sectors = 512 .
- The number of sector/trace = 1000.
- The number of platter surfaces = 8.
- The number of trace/platter = 5000.

الحل

$$C = N \times S \times T \times P = [512 \times 1000 \times 5000 \times 8] / (1024 \times 1024) = 19 \text{ G Byte}$$

Q5: Define : Seek time & The Rotational latency & Transfer time & burst time ?

- Seek time : هو الزمن اللازم لحركة الرأس المستخدم في القراءة والكتابة إلى المكان الذي يصل إليه.
- Rotational latency : هو الزمن اللازم لحركة القرص للوصول إلى القطاع "Sector" المطلوب.
- Transfer time : هو زمن النقل من القطاع "Sector" في الـ Disk Platter بمجرد أن يتم وضع القطاع تحت الرأس.
- Burst time (Rotational time) : هو معدل تخزين البيانات في الذاكرة عندما تبدأ عملية القراءة أو الكتابة، وتسلاوي سرعة دوران القرص \times سعة كل مسار " track ".

Q6: State the factors that effects the speed of hard disk data transfer ?

Transfer time -3

Rotational latency -2

Seek time - 1

Q7: Access time of Floppy disk is slower than a hard disk ... Why ?

لأن القرص المرن Floppy disk لا يمكنه الدوران بسرعة كما في القرص الصلب Hard disk فسرعة آلية الدوران المثالية للقرص المرن 300 Rpm فقط ، ويمكن أيضاً أن تكون متنوعة كحركة الرأس من مسار إلى مسار لتحسين معدل نقل البيانات ، إن بطء سرعة الدوران تعني أن زمن الوصول للقرص المرن هو 250-300 ms مما يقرب 10 مرات بأبطأ من الأقراص الصلبة ، السعات تختلف لكن تصل لمدى 1.44MB .

Q8: Why does the hard disk Organization use interleaving machnism ?

لتحسين السرعة ، وللقراءة من الهارديسك ، ولحركة الـ platter .

Q9: What does the Master Control Block (MCB) Contain ?

1- قسم التهيئة : ويختص بتحديد المعلومات المتعلقة بتقسيم القرص مثل عدد الأسطح ، وعدد القطاعات لكل سطح.

2- قسم الملفات : حيث يتم تخزين اسم الملف أولاً مع قائمة القطاعات التي تتألف منها وخصائص هذا الملف مثل تاريخ إنشاؤه وتاريخ آخر تعديل عليه .

3- قسم البلوكات الحرة : تخبر بموقع البلوكات التي لم تستخدم " free " ليتم استخدامها .

4- قسم البلوكات الرديئة : حيث تخبر بمواقع البلوكات التي لم تستخدم " free " ولكنها رديئة لكي لا يتم استخدامها .

Q10: How can we find a bad sectors in a hard disk ?

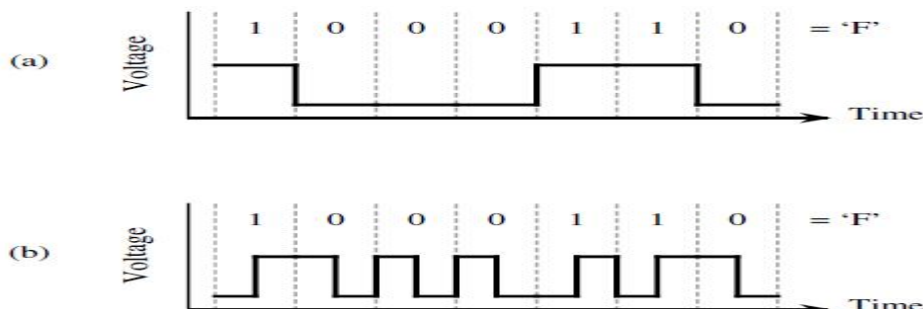
إن الـ Bad Sector ينتج أثناء التصنيع ويتم استخدام طريقة المغنطة لتمثيل الرقم 1 ، وهنا فإن جودة الهارديسك نفسه في استقبال عملية المغنطة هي التي تؤثر في الـ Bad Sector.

فعند وجود رقم 1010101 ولكنه أثناء القراءة تمت قراءته 1110101 أي يوجد هنالك اختلاف ، فإن بهذا يتم التأشير على هذا الـ Sector بأنه غير قابل للاستخدام .

وللتغلب على هذه المشكلة يجب إعادة تقسيم الهارديسك ، فإعادة التقسيم تحل بعض مشاكل الـ Bad Sector وليس كلها وذلك لأن طريقة التقسيم تتغير.

Q11: Show the Manchester Encoding of the bit sequence 1000110 = 'F' ?

النقطة من الأسفل لأعلى تمثل بـ 1 ، والنقطة من أعلى لأسفل تمثل بـ 0 وتكون القراءة من منتصف الـ Phase .



Q12: What is Video Display ? Explain how it work ?

Video Display : يتكون من جهاز عرض مضيء مثل (CRT) " أنبوب أشعة كاثود " أو من (LCD) " العرض بالبلورات السائلة " ومجموعة من الدوائر الكهربائية التحكمية.

فكرة عمل CRT : فكرة عملها هي انطلاق الإلكترونات من خلف الشاشة إلى أن تصل إلى سطح العرض المبطن بطبقة من الفسفور ، ولعرض صورة على الشاشة فإن شعاع الإلكترون يسبح خلال خط أفقي مبتدئاً من أعلى الشاشة من اليسار إلى اليمين فيسبب توهجاً في طبقة الفسفور ، وتختلف شدة سطوعه باختلاف جهد الإلكترون الكهربائي ، وعندما يصل الشعاع إلى نهاية الخط يتوقف للحظة تسمى " فترة الخمول الأفقية " ، ثم يعاد إعداد المغناطيس كي يبدأ برسم الخط السفلي الجديد ؛ وتعاد هذه العمليات مسببة رسم خط بعد خط على الشاشة حتى تمتلئ ، وهنا يتوقف الشعاع للحظة تسمى " فترة الخمول الرأسية " ويعاد إعداد المغناطيس كي تعاد كل العملية من جديد فترسم صورة أخرى على الشاشة مبتدئة من الركن العلوي الأيسر . أما التلفزيون الملون فهو لا يختلف كثيراً عن تلفاز الأبيض والأسود إلا أن به ثلاث مدافع للإلكترونات بدلاً من مدفع واحد ، وكذلك النقاط وحيدة اللون في طبقة الفسفور التي تغلف زجاج الشاشة من الداخل تستبدل بنقاط ثلاثية اللون هي (الأحمر ، الأخضر ، الأزرق) ، وبخلط هذه الألوان الثلاثة بنسب متفاوتة يمكننا الحصول على جميع الألوان الأخرى ، حيث أن هذا الخلط يتم عن طريق تغيير كثافة كل لون من هذه الألوان على طبقة الفسفور.

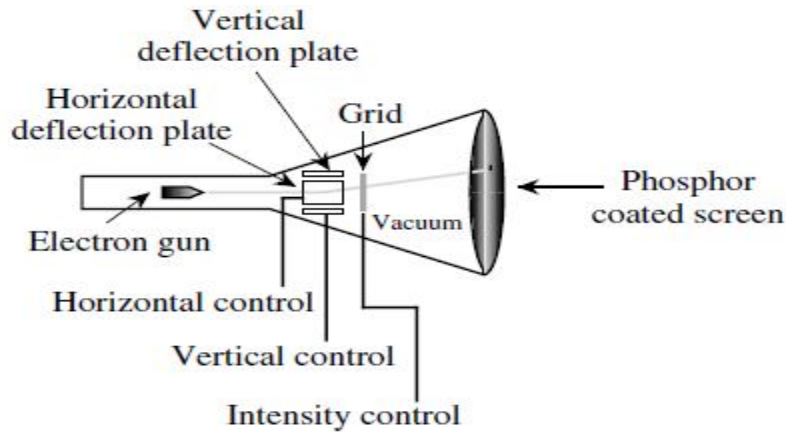


Figure 8-16 A CRT with a single electron gun.

Q13: What are the components of the Bus lines ?

يتكون الناقل من أجزاء فيزيائية مثل الموصلات والأسلاك وبروتوكولات النقل ، فالأسلاك يمكن أن تقسم إلى مجموعات منفصلة (Control, address, data, power) الأجهزة تتقاسم المجموعة المشتركة من الأسلاك ولكن فيها جهاز واحد فقط هو من يمكنه الإرسال خلال الناقل في أي لحظة زمنية معينة ؛ وحين ذلك يستوجب على بقية الأجهزة الاستماع وعدم الإرسال ، ولكن عادة جهاز واحد فقط هو من يستقبل ، فلذلك جهاز واحد فقط يمكن أن يكون master " السيد " والأجهزة الأخرى هي slaves " التابعة " ؛ حيث أن الـ master هو من يتحكم بالـ Bus ويمكن أن يكون المرسل أو المستقبل.

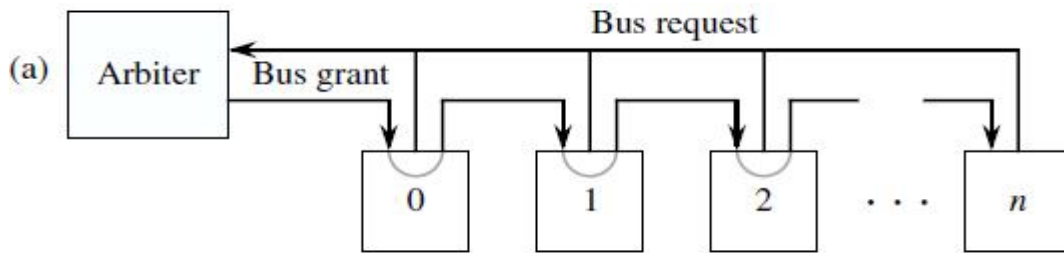
Q14: What's the advantage of using Bus ? and disadvantage ?

الميزة : أن استخدام الناقل يزيل الحاجة لربط كل جهاز مع الأجهزة الأخرى ، مما يجنب تعقيد الأسلاك " تشابكها " ومن شأن ذلك أن يهيمن بسرعة على تكلفة مثل هذا النظام. **العيب** : تشمل البطء الناتج من آلية (Master/Slave) وعدم وجود قابلية لأحجام كبيرة بسبب القيود المفروضة على الزمن.

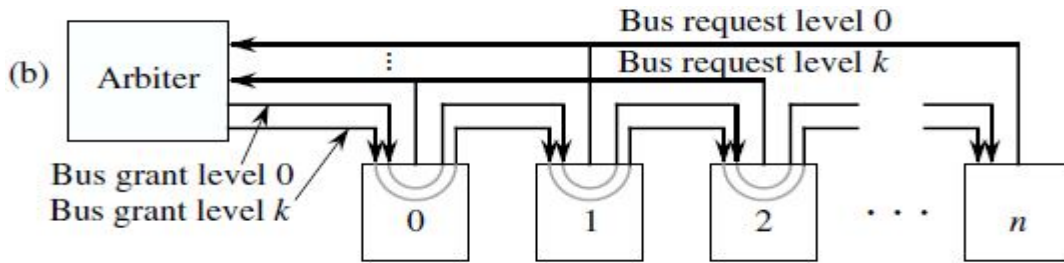
Q15: What's the Bus Arbitration types ? Explain and draw How different types are working ?

الأنواع هي Centralized و Decentralized (distributed) .

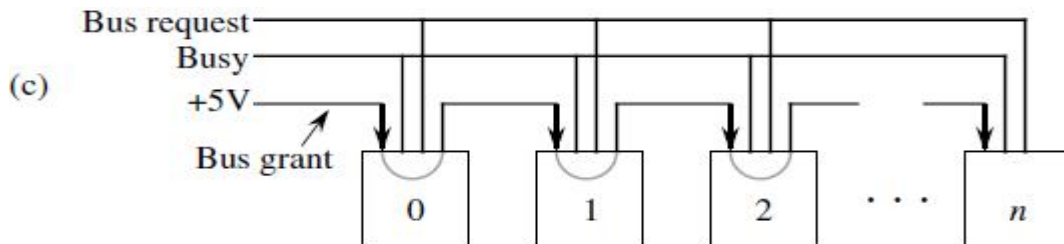
كيفية العمل : الأجهزة من 0 إلى n كلها تتصل بنفس الناقل " BUS " وكذلك كلهم يشتركون في خط طلب النقل الذي يعود لل Arbiter . فعندما يرغب جهاز أن يكون Bus master فعليه أن يؤكد على خط طلب النقل ، وحينما يرى Arbiter النقل هذا فإنه يحدد إعطاء الـ (Grant " إذن النقل ") أم لا ، فإذا وجدت إمكانية لإعطاء Grant إذاً فإن Arbiter يجب عليه أن يؤكد على خط السماح بالنقل " Bus Grant " وخط السماح بالنقل هذا سيكون على شكل سلسلة متعاقبة من الجهاز الأول للجهاز التالي. أول جهاز يرى الموافقة على الـ Bus Grant فإنه قد يرغب أن يكون master ، ففي هذه الحالة فإن كل ما عليه هو أن يأخذ التحكم ولا يقوم بنشر Grant للأجهزة التي تليه ، أما إذا الجهاز لا يريد ذلك فإنه وببساطة ما عليه إلا أن يمرر الـ Grant للجهاز الذي يليه ، وفي هذه الطريقة الجهاز الأقرب لل Arbiter ستكون له الأولوية أكثر من تلك البعيدة .



في بعض الأحيان الأولوية المطلقة حسب الترتيب قد لا تكون مناسبة ، حيث الأجهزة القريبة من الـ Arbiter لها أولوية على البعيدة ، ولذلك لزيادة الأولوية للأجهزة البعيدة عن Arbiter يمكن أن نعين لها خط طلب نقل بمستوى آخر ، فتكون الأولويات وفقاً لخط طلب النقل من حيث الجهاز الأقرب لل Arbiter بالنسبة له .



أما في الـ Decentralized لا يوجد Arbiter مركزي ، فالجهاز الذي يريد أن يصبح Buss master عليه أن يؤكد في خط طلب النقل ، ومن ثم يتفحص الناقل إذا ما كان مشغولاً ، فإن كان كذلك فإن طلبه لن يؤكد ويرسل الجهاز 0 للجهاز الذي يليه في السلسلة المتعاقبة للدلالة على انشغال الخط ، وعدم التأكيد ، وإن كان الناقل مشغولاً أو الجهاز لم يرد النقل فبكل بساطة يتم نشر الـ Grant للجهاز الذي يليه



Q16 : a) Define what we mean by a daisy chined ?

تعني السلسلة المتعاقبة للأجهزة من 0 إلى n في ال Bus Arbitration.

b) What is the disadvantage of a centralized bus arbitration?

العيب هو وجود Central arbiter مما يعني أنه عند التحكم يجب المرور به شخصياً ومنه يعطي الموافقة ، وهذا الأمر يأخذ وقتاً واستخدامه يكون جيد فقط في الأجهزة القليلة ، ولكن عند العدد الكبير من الأجهزة فإن استخدام Decentralized أفضل.

c) Define what we mean by a bus Arbitration problem ?

هو الذي جاء ليحل مشكلة ما إذا كانت هنالك أكثر من جهاز يريد أن يكون Bus Master في نفس اللحظة ، ولذلك فإن ال Buss Arbitration Problem هو الذي يتخذ القرار لأي جهاز يكون Bus Master.

Q17: Given a system that uses are of arbitration scheme imagine a situation in which N card slots are used and then card M is removed where $M < N$. what will be happened ?

طالما أن كل bus request line متصل مباشرة بكل الأجهزة في المجموعة ، و bus grant line ممرر خلال كل جهاز في المجموعة ، إذاً فإن الناقل عندما يطلب من جهاز رقم موقعه أكبر من M ، فإنه سوف لن يرى تأكيد ال bus grant line والذي يمكن أن يسبب في تدهم النظام ، وهذا يجعل المشكلة صعبة في التحديد ، وذلك لأن النظام يمكن أن يستمر في التنفيذ إلى ما لا نهاية من دون أخطاء إلى أن يصل إلى أرقام الأجهزة الأخيرة .

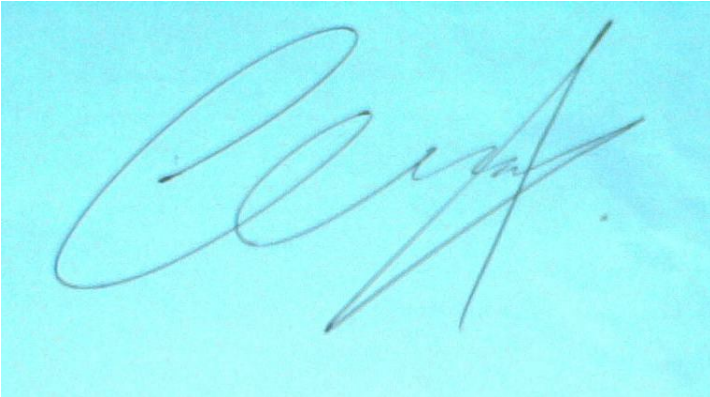
لذلك عند إزالة بطاقة من البطاقات العليا ، يجب أن يتم وضعها لتملأ المنفذ المفقود ، أو أن يتم استخدام بطاقة وهمية تستبدل مكان البطاقة المزالة لتكمل بها ال bus grant line ، كما وأن الأجهزة السريعة يجب أن تكون له الأولوية الأكبر من البطيئة في عملية الترتيب ، ولذلك يجب أن تكون أقرب لل Arbiter في Centralized أو قريبة من البداية في Decentralized.

Q18: Mention the Methods used to transfer data between I/O devices and memories ?

Direct Memory Access (DMA) -3

Interrupt driven I/O -2

Programmed I/O - 1



لا تنسونا من صالح دعائكم.. أخوكم محمد حسين