مدخل إلى البرمجة الصغرية

يتكـون البرنــامج Program (عنـد كتابـة برنـامج بلغـة مـا فإنـه بالإنجليزيـة يسـمى Code) مـن مجموعـة مـن التعليمـات Instructions، والتـي تكـون مخزنـة بشـكل دائـم ضـمن ذاكـرات ROM، (أو ذاكـرات RAM مـع بطاريـة) ، أو علـى أقـراص (مرنـة- صـلبة –ليزريـة)، ويـتم تشـغيلها مـن خـلال نقلهـا، مـن الأقـراص إلـى الـذاكرة RAM ، وتسـمى عمليـة النقـل هـذه بالشـحن Loding .

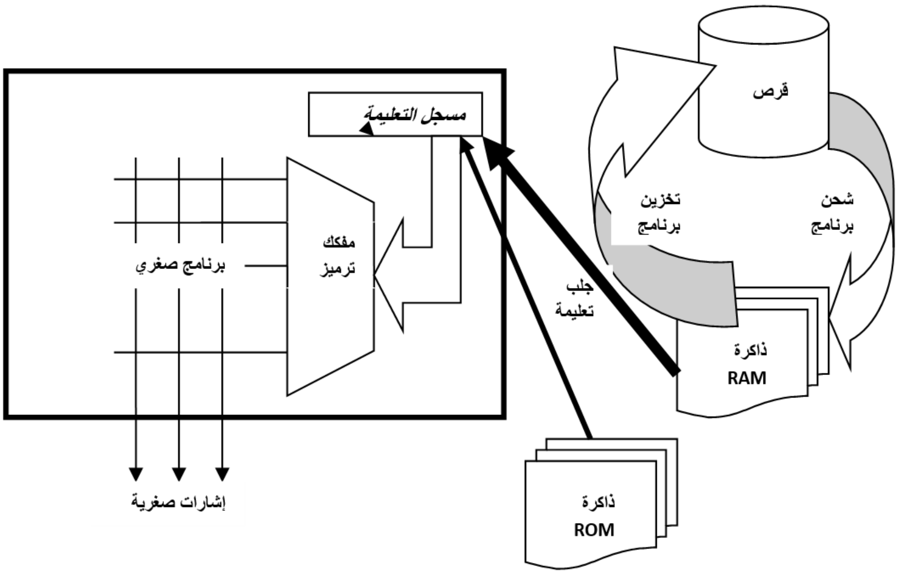
من سيئات الذواكر RAM ، أنها تفقد كل مـا مخزن منهـا عند انقطـاع التيـار الكهربـائي عنهـا ، أو حتـى انخفاضـه لحد معين، لهذا للاحتفاظ بالبيانات المخزنة أو البرامج لا بد من إعادة نقله للأقراص ، وتسمى عملية النقل هذه بالحفظ Saving.

نناقش الآن الحالتين الآتيتين:

1. **عدم وجود ذاكرة انتقالية (وميضه) Cash memory.**
2. **وجود ذاكرة انتقالية.**

1-2- حالة عدم وجود ذاكرة انتقالية

إن حالة عدم وجود ذاكرة انتقالية كانت موجودة في معالجات الحواسيب القديمة في الثمانينـات. عمومـاً يقوم الحاسوب بتنفيذ التعليمات تعليمة يليها تعليمة أخرى من بداية البرنامج حتى نهايته، وقد يكون هناك تعليمات قفز، و لتنفيذ تعليمة ما لابد من نقلها من الـذاكرة RAM إلـى داخـل المعـالج Microprocessor ، حيـث يـتم حفظهـا مؤقتـاً ضـمن مسـجل خـاص يـدعى بمسـجل التعليمـة Instruction Register ، وإختصـاراً IR (تـذكر أن هـذا المسـجل غيـر مـدرج مـع مسـجلات المعـالج الداخليـة لأن المبـرمج بلغـة التجميع Assembly Language لا يستطيع أن يتعامل مع هذا المسجل، بمعنى لا توجد أوامر للتعامل مع هذا المسجل) ، ويكون طول المسجل هذا متناسب مع طول التعليمة ، و هو مقسم إلى عدة أجزاء ، وكل جزء مخصص لعدد محدد من البتات، ويختلف ذلك من معالج لآخر، و لكن دائماً هنـاك قسم يسمى بقسم العنـوان ، و قسم آخر يسمى بقسم الترميز ، وهنـاك قسم ثالـث للعنونـة.



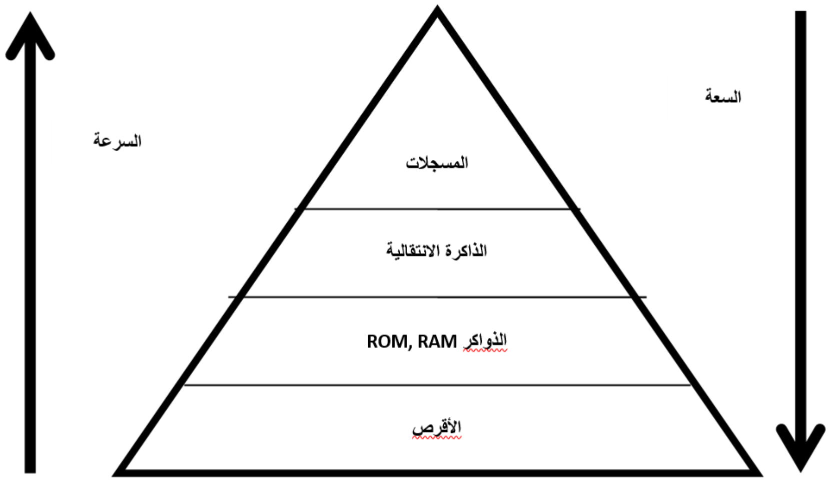
الشكل (1-2) تنفيذ تعليمة بدون وجود ذاكرة انتقالية

لكل تعليمة ترميز ما، و يتوضع هذا الترميز في قسم الترميز عند نقل التعليمة إلى مسجل التعليمة (هذا النقل يسمى بالجلب)، يتم فـك ترميـز التعليمــة بمسـاعدة مفكـك ترميـز Decoding ، ويكـون كـل ترميـز هـو عبـارة عـن عنـوان لبرنـامج صـغري

Microprogram (أو جزء منه) ، حيث يتم تنفيذ ذلك البرنـامج مـع كل تعليمـة ، وهذا التنفيذ يرافقه توليد مجموعـة محددة من النبضـات ترسـل إلـى خـارج المعـالج علـى دفعـات (كل دفعـة مـع نبضـة سـاعة) ، وتـؤمن تلـك النبضـات تنفيـذ التعليمـة، كمـا يبـين الشكل(2-1).

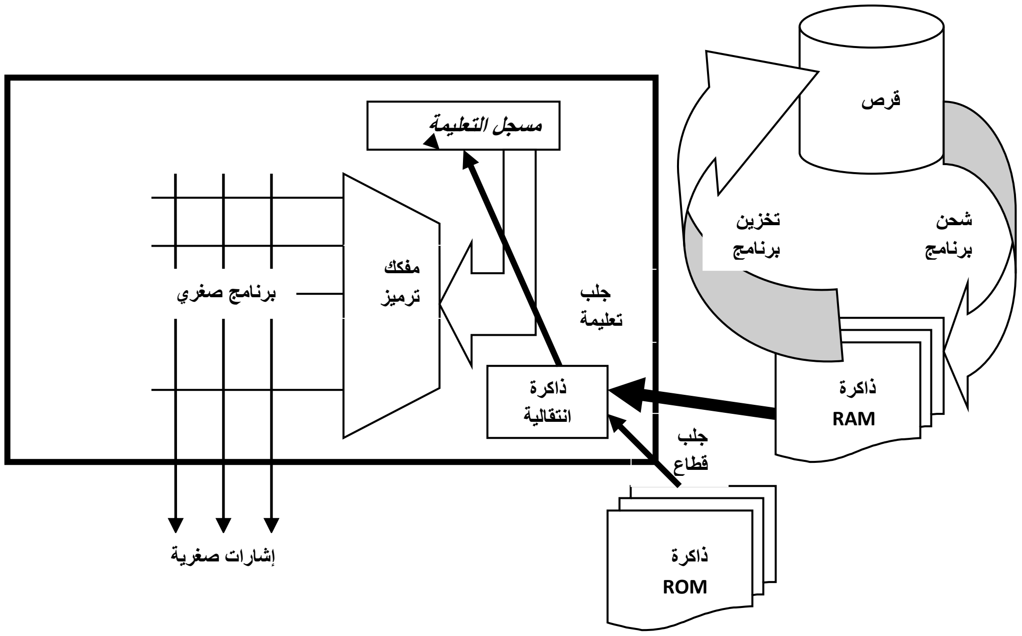
2-2- حالة وجود ذاكرة انتقالية

في المعالجات الحديثة توجد ذاكرة انتقالية وسيطة (أو أكثر) بين الذاكرة الخارجية (RAM, ROM)، و مسجل التعليمة، وهي سريعة ، ولكن سعتها قليلة إذا ما قورنت مع العناصر التخزينية الأخرى الموجودة ضمن بنية الحاسوب. لاحظ الشكل (2-2).



الشكل (2-2) العلاقة العكسية بين السعة التخزينية وسرعة الذاكرة.

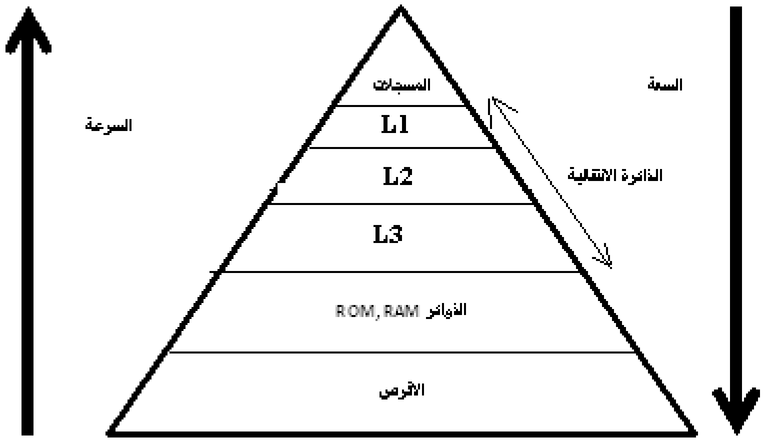
ويمكن إضافة الذاكرة إلى الشكل (2-1) ليصبح لدينا الشكل(2-3) وقد وضعت داخل شريحة المعالج. و عند وجود مستويات أخرى فإنها توضع خارج الشريحة.



الشكل (2-3) تنفيذ تعليمة مع وجود ذاكرة انتقالية.

2-2-1- هرمية الذاكرات الانتقالية

كما ذكرنا سابقاً فإن الذاكرة الانتقالية يمكن أن تتواجد على هيئة مستويات L1 تتواجد داخل شريحة المعالج و تتمتع بسرعة وصول كبيرة و سعة قليلة لتقليل التكلفة وهي بالتالي أعلى سرعة ولكنها أصغر سعة من الذاكرة الانتقالية L2 (الذاكرة الانتقالية الخارجية ذات المستوى الثاني) ومن L3 ذات المستوى الثالث (إن وجدت) و بدورها L2 أعلى منها سرعة من الذاكرة الانتقالية L3 ولكنها أصغر سعة . وهكذا يمكن تعديل الشكل(2-2) إلى الشكل(2-4).



الشكل(2-4) العلاقة العكسية بين السعة التخزينية ، وسرعة التسجيل في مستويات الذاكرة الانتقالية

2-2-2- مشكلة ترابط الذاكرات الانتقالية

تنشأ مشكلة ترابط الذاكرات المخبئة من وجود نسخ متعددة للبيانات وعملية نشرها على كافة الذاكرات المخبئة من مستويات مختلفة . نقول عن نظام ذاكري أنه مترابط إذا كانت جميع الذاكرات المخبئة تملك نفس القيمة لنفس العنصر، عندما يقوم أحد المعالجات بتغيير القيمة بإحدى هذه الذاكرات تصبح بالتالي غير مترابطة Incoherent، وتصبح هناك قيم خاطئة بالتالي نتائج نهائية خاطئة.
إذاً نحتاج لخوارزميات ترابط الذاكرة المخبئة لحل هذه المشكلة، وهناك عدة طرق لتحقيق الترابط.
في حال لدينا نظام بذاكرة مخبئة واحدة L1, يتحقق الترابط بين الذاكرتين الرئيسة والمخبئة من خلال الطريقتين التاليتين الموضحتين في الجدول (1):

***الكتابة خلال Write Through*: تعدل الذاكرة الرئيسة قيمها في كل لحظة يحدث فيها تعديل في الذاكرة المخبئة.**
***الكتابة عكسياً Write Back*: يحدث التعديل في الذاكرة الرئيسة فقط عندما يتم استبدال المقطع في الذاكرة المخبئة Cache.**

		Write-Through		Write-Back	
Serial	Event	Memory	Cache	Memory	Cache
1		X		X	
2	P reads X	X	X	X	X
3	P updates X	X'	X'	X	X'

الجدول(1) الكتابة عكسياً Write Back والكتابة خلال Write Through.

الحالات الأخرى لتحقيق الترابط تدرس في مقررات أخرى مستقبلاً.

3-2-2- تنظيم الذاكرة الانتقالية

يتم عادة تنظيم ذاكرة كل مستوى عن طريق تقسيمها إلى خطوط Lines و مجموعات Sets فمثلاً ذاكرة انتقالية سعتها 8kB[1] يتم تقسيمها إلى 128 خط و في كل خط 64B ، وهذه الخطوط يتم تقسيمها إلى 32 مجموعة و في كل مجموعة 4 خطوط. ويمكن استخدام مجموعة واحدة فقط من هذه المجموعات الـ 32 ويمكن الوصول إلى أي خط من خطوط المجموعة المستخدمة.

يتم أولاً تحديد المجموعة بدلالة العنوان ثم يأخذ خط واحد من خطوط المجموعة وفق الآتي:

(رقم المجموعة) = ((عنوان الذاكرة) / (سعة الخط)) % (عدد المجموعات).

حيث أن الرمز %: يشير إلى حاصل باقي قسمة الجواب الناتج ضمن القوسين على عدد المجموعات.

فمن أجل عنوان ذاكرة قدره 10000 يكون رقم المجموعة هو:

(رقم المجموعة) = ((10000) / (64)) % (32)= 28، أي رقم المجموعة هو 28 أي يتم الوصول و البحث في خطوط المجموعة 28 واحداً تلو الآخر.

وعند استخدام نظام العد الست عشري ومن أجل عنوان ذاكرة قدره Ox2710 يكون رقم المجموعة هو:

(رقم المجموعة) = ((Ox2710)/ (Ox40)) %(Ox20)= Ox1C، أي يتم البحث في خط سعته 64B أو Ox40 في نظام العد الست عشري أي من العنوان Ox2700 إلى العنوان Ox273F

عند عدم العثور على التعليمة المناسبة ضمن المجموعة يدعم القفز بخطوة طويلة مميزة Critical Stride تحسب وفق الآتي:

الخطوة المميزة= (سعة الذاكرة)/ (عدد الخطوط)=(عدد المجموعات)/(سعة الخط).

مثال(1):
بفرض لدينا معالج يحتوي على ذاكرة مخبئة بمستوى واحد فقط وإذا كان زمن الوصول إلى الذاكرة المركزية 200ns و إلى الذاكرة المخبئة هو 40ns وإذا كان احتمال العثور على تعليمة بشكل متوسط في الذاكرة المخبئة هو 90% أحسب زمن الاستجابة الظاهري للمعالج (زمن الوصول إلى التعليمة)، واحسب مقدار زيادة سرعة الوصول الوسطية مقارنة مع عدم وجود ذاكرة انتقالية.

الحل:

زمن الاستجابة الظاهري: 200*0.1+40*0.9=56ns

مقدار زيادة سرعة الوصول الوسطية =200/56 = 3.57 مرة

مثال(2):
من أجل المعالج P4 لدينا ذاكرة انتقالية L1 سعتها 8kB يتم تقسيمها إلى 128 خط و في كل خط 64B، وهذه الخطوط يتم تقسيمها إلى 32 مجموعة و في كل مجموعة 4 خطوط المطلوب:

1. حدد رقم المجموعة الموافقة للعنوان 9984.
2. أحسب طول الخطوة الطويلة المميزة Critical Stride
3. إذا كان زمن الوصول إلى الذاكرة المركزية 350ns و إلى الذاكرة L1 هو 2ns و إلى الذاكرة L2 هو 19ns فإذا كان إحتمال العثور على تعليمة بشكل متوسط في الذاكرة L1 هو 60% في الذاكرة L2 هو 30% أحسب زمن الاستجابة

[1] - هذا موجود في الذاكرة L1 من المعالج بنتيوم4.

الظاهري للمعالج (زمن الوصول إلى التعليمة). و احسب مقدار زيادة سرعة الوصول الوسطية مقارنة مع عدم وجود ذاكرة انتقالية.

الحل:

1. يتم تحديد المجموعة بدلالة العنوان وفق الآتي:
(رقم المجموعة)= (عنوان الذاكرة)\ (سعة الخط)%(عدد المجموعات).

فمن أجل عنوان ذاكرة قدره 9984 يكون رقم المجموعة هو:

(9984)\ (64)%(32)=156%32= 28 أي رقم المجموعة هو 28

2. الخطوة الطويلة المميزة Critical Stride تحسب وفق الآتي:
الخطوة المميزة= (سعة الذاكرة) \ (عدد الخطوط في المجموعة)= (سعة الخط) * (عدد المجموعات)

الخطوة المميزة= 8k/4 = 2k أو الخطوة المميزة= 32*64= 2k

3. زمن الاستجابة الظاهري: 350*0.1+2*0.6+19*0.3=35+1.2+5.7=41.9ns
مقدار زيادة سرعة الوصول الوسطية = 350/41.9 = 8.3 مرات

مثال(3):

من أجل المعالج ثنائي النوى، لدينا ذاكرة انتقالية L1 سعتها 32kB يتم تقسيمها إلى خطوط بحيث يكون في كل خط 64B ، وهذه الخطوط يتم تقسيمها إلى مجموعات بحيث يكون في كل مجموعة 8 خطوط. المطلوب:

1. أحسب عدد الخطوط الكلي
2. أحسب عدد المجموعات الكلي
3. حدد رقم المجموعة الموافقة للعنوان 9984.
4. أحسب طول الخطوة الطويلة المميزة Critical Stride
5. إذا كان زمن الوصول إلى الذاكرة المركزية 185ns و إلى الذاكرة L1 هو 3ns و إلى الذاكرة L2 هو 14ns فإذا كان إحتمال العثور على تعليمة بشكل متوسط في الذاكرة L1 هو 60% في الذاكرة L1 هو 30% أحسب زمن الاستجابة الظاهري للمعالج (زمن الوصول إلى التعليمة). و احسب مقدار زيادة سرعة الوصول الوسطية مقارنة مع عدم وجود ذاكرة انتقالية.

الحل:

1. عدد الخطوط الكلي = سعة الذاكرة \ سعة الخط الواحد = 32k/64 = 64/ (32*1024)= 512
2. عدد المجموعات الكلي = عدد الخطوط الكلي \ عدد الخطوط في المجموعة الواحدة= 512/8 = 64
3. يتم تحديد المجموعة بدلالة العنوان وفق الآتي:
(رقم المجموعة)= ((عنوان الذاكرة)\ (سعة الخط))%(عدد المجموعات).

فمن أجل عنوان ذاكرة قدره 9984 يكون رقم المجموعة هو:

((9984) \ (64))%(64)=156%64= 28 أي رقم المجموعة هو 28

4. الخطوة الطويلة المميزة Critical Stride تحسب وفق الآتي:
الخطوة المميزة= (سعة الذاكرة) \(عدد الخطوط في المجموعة)= (سعة الخط) * (عدد المجموعات)

الخطوة المميزة= 32k/8 = 4k أو الخطوة المميزة= 64*64= 4k

5. زمن الاستجابة الظاهري: 185*0.1+3*0.6+14*0.3=18.5+1.8+4.2=24.5ns
مقدار زيادة سرعة الوصول الوسطية = 185/24.5 = 7.55 مرة.

مثال(4):

من أجل المعالج ثنائي النوى لدينا ذاكرة انتقالية L1 سعتها 32kB يتم تقسيمها إلى خطوط بحيث يكون في كل خط 64B ، وهذه الخطوط يتم تقسيمها إلى مجموعات بحيث يكون في كل مجموعة 8 خطوط. المطلوب:

1. أحسب عدد الخطوط الكلي .الحل: عدد الخطوط الكلي = سعة الذاكرة \ سعة الخط الواحد = 32k/64 = 512
2. عدد المجموعات الكلي الحل: عدد المجموعات الكلي = عدد الخطوط الكلي\ عدد الخطوط في المجموعة الواحدة= 512/8 = 64
3. حدد رقم المجموعة الموافقة للعنوان 10170.الحل: يتم تحديد المجموعة بدلالة العنوان وفق الآتي:

(رقم المجموعة)= ((عنوان الذاكرة)\ (سعة الخط))%(عدد المجموعات).

فمن أجل عنوان ذاكرة قدره 10170يكون رقم المجموعة هو:

((10170) \(64))%(64)=158 % 64 =30.9 أي رقم المجموعة هو 30

4. طول الخطوة الطويلة المميزة: تحسب وفق الآتي:

الخطوة المميزة= ((سعة الذاكرة) \ (عدد الخطوط في المجموعة)) = (سعة الخط) * (عدد المجموعات) أو

الخطوة المميزة= 32k/8 = 4k أو الخطوة المميزة= 64*64= 4k

5. إذا كان زمن الوصول إلى الذاكرة المركزية 185ns و إلى الذاكرة L1 هو 3ns و إلى الذاكرة L2 هو 14ns فإذا كان إحتمال العثور على تعليمة بشكل متوسط في الذاكرة L1 هو 65% في الذاكرة L2 هو 25% أحسب زمن الاستجابة الظاهري للمعالج (زمن الوصول إلى التعليمة). و احسب مقدار زيادة سرعة الوصول الوسطية مقارنة مع عدم وجود ذاكرة انتقالية. الحل: زمن الاستجابة الظاهري: 185*0.1+3*0.65+14*0.25=18.5+1.95+3.5=23.95ns

مقدار زيادة سرعة الوصول الوسطية = 185/23.95 = 7.7مرة

4-2- مراحل تنفيذ تعليمة

لنفرض لدينا برنامج أي ملف قابل للتنفيذ مخزناً على قرص صلب بالنقر على أيقونة الملف يتم التنفيذ حيث تجري داخل الحاسوب المراحل التالية:

- مرحلة الشحن Loading: حيث يقوم نظام التشغيل بنقل البرنامج من القرص إلى الذاكرة RAM (في بيئات لغات البرمجة يمكن ذلك بمساعدة الخيار open أو الزر <F3>) . يقوم بنقل البرامج برنامج آخر اسمه برنامج النقل وهو أحد أهم برامج نظم التشغيل.
- مرحلة جلب التعليمة Fetching : وهذه المرحلة موجودة في كافة أنواع وأجيال المعالجات الرقمية الدقيقة ولكن في المعالجات التي أقدم من المعالج 80486 كانت هذه المرحلة تتمثل في نقل التعليمة Instruction من ذاكرة البرنامج الرئيسية RAM إلى مسجل التعليمة Instruction Register الموجود داخل شريحة المعالج وفي المعالجات التي أحدث من المعالج 80486 (و المعالج 80486 ذاته) أصبحت هذه المرحلة تتمثل في نقل التعليمة Instruction من الذاكرة الانتقالية cache memory إلى مسجل التعليمة Instruction Register وكلاهما موجودين داخل شريحة المعالج .وهذا يعني أن هذه المرحلة أصبحت في الحواسيب الحديثة مرحلتين: آ- مرحلة نقل للتعليمات من الذاكرة

RAM إلى الذاكرة الانتقالية cache memory__ب- مرحلة نقل تعليمة من الذاكرة الانتقالية_إلى مسجل التعليمة_داخل المعالج_.

- مرحلة فك ترميز التعليمة Instruction Decoding: بعد نقل التعليمة Instruction من الذاكرة الانتقالية cache memory إلى مسجل التعليمة Instruction Register يتم فك ترميزها بواسطة مفكك ترميز Instruction Decoder موجود ضمن شريحة المعالج، وهذا المفكك مرتبط من جهة خرجه مع ذاكرة البرامج الميكروية Microprograms الثابتة ROM الموجودة داخل شريحة المعالج أيضاً ويعتبر ترميز التعليمة هو عنوان بداية برنامج ميكروي خاص بتنفيذ تلك التعليمة (أي لكل تعليمة برنامج أو جزء من برنامج صغري) .
- مرحلة جلب المعاملات Getting of the Operands : لكل عملية من العمليات الحسابية والمنطقية معاملات أي معطيات تطبق عليها حتى يتم تنفيذها فمثلاً تعليمة الجمع تحتاج لمعاملين حتى يتم تنفيذها أي عددين وهذا ينطبق على التعليمات مثل تعليمة النقل MOV التي تحتاج أيضاً لمعاملين وبالتالي لا يمكن تنفيذ التعليمة من دون جلب المعاملات اللازمة لها والتي كما نعلم من المحاضرة1 لها ثلاثة مصادر وهي:
  - ذاكرة المعطيات Data memory RAM
  - بوابات الإدخال (I/O Ports) Input Ports
  - التعليمة ذاتها حيث في هذه الحالة يتم توفير الزمن اللازم للجلب وفي هذه الحالة يكون لدينا عنونة فورية Immediate Addressing للمعاملات أما في الحالتين الأولى والثانية فيكون لدينا عنونة مباشرة Direct Addressing للمعاملات إذا تم جلبها من مصدرها مباشرة وهذا يعني أن التعليمة تحتوي على العناوين المباشرة للمعاملات وهناك حالات عديدة للعنونة الغير المباشرة Indirect Addressing للمعاملات وهذا يعني أن التعليمة تحتوي على عناوين تحدد مكان تواجد العناوين المباشرة للمعاملات.
- مرحلة التنفيذ Executing : وتشمل هذه المرحلة التنفيذ الفعلي للتعليمة بعد فك ترميزها وجلب معاملاتها.
- مرحلة حفظ النتائج Writing of the back results : حيث يتم تسجيل نتيجة تنفيذ التعليمة في مسجلات المعالج الداخلية (في مسجل المراكم Accumulator) .

والسؤال هو لماذا لا ننقل التعليمة مباشرة من ذاكرة البرنامج الرئيسية RAM إلى مسجل التعليمة (كما في الحواسيب القديمة). الجواب هو في أن الذاكرة الانتقالية موجودة داخل المعالج[2] وهي أسرع من الذاكرة RAM ولكن سعتها أقل. في الحقيقة دائماً السرعة هي على حساب السعة لاحظ الشكل (2-2).

ويتم نقل التعليمات واحدة واحدة إلى مسجل التعليمة من الذاكرة الانتقالية وإذا كانت التعليمة غير موجودة في الذاكرة الانتقالية فإنه يتم البحث عنها في الذاكرة RAM ويتم نقلها إلى الذاكرة الانتقالية ولكن ليس لوحدها بل مع كامل القطاع Segment الموجودة فيه، الشكل(2-5) يوضح آلية جلب تعليمة إلى مسجل التعليمة[3].

وأشير أخيراً إلى أن المعالجات الحديثة تعتمد على تقنية المعالجة التدفقية المتعددة Multiple- Pipelining ، ولكن هنا لا بد من الإشارة إلى أنه توجد طريقتان لرفع سرعة المعالجة في المعالجات الرقمية الدقيقة Microprocessors الطريقة الأولى من خلال رفع تردد الساعة Clock frequency والطريقة الثانية من خلال تصميم المعالج بشكل يتم الاستفادة من كل نبضة من نبضات الساعة.

بالنسبة للطريقة الأولى ليست المشكلة في أن الشركات لا تستطيع تصنيع ساعات بترددات أعلى بل أنه في الواقع المشكلة هي في مواد ا أشباه النواقل التي ترتفع درجة حرارتها جدا وخصوصا عند وصول تردد الساعة إلى 400MHz حيث أن الدراسات والتجارب تجري منذ بداية التسعينات على مواد كريستالية بديلة تتحمل ترددات أعلى من 400MHz وهذا يعني الانتقال إلى مواد كريستالية بديلة بالنسبة لكافة عناصر الحاسوب مثل الذواكر وشرائح المعالجات الدقيقة وهذا مكلف نسبيا.

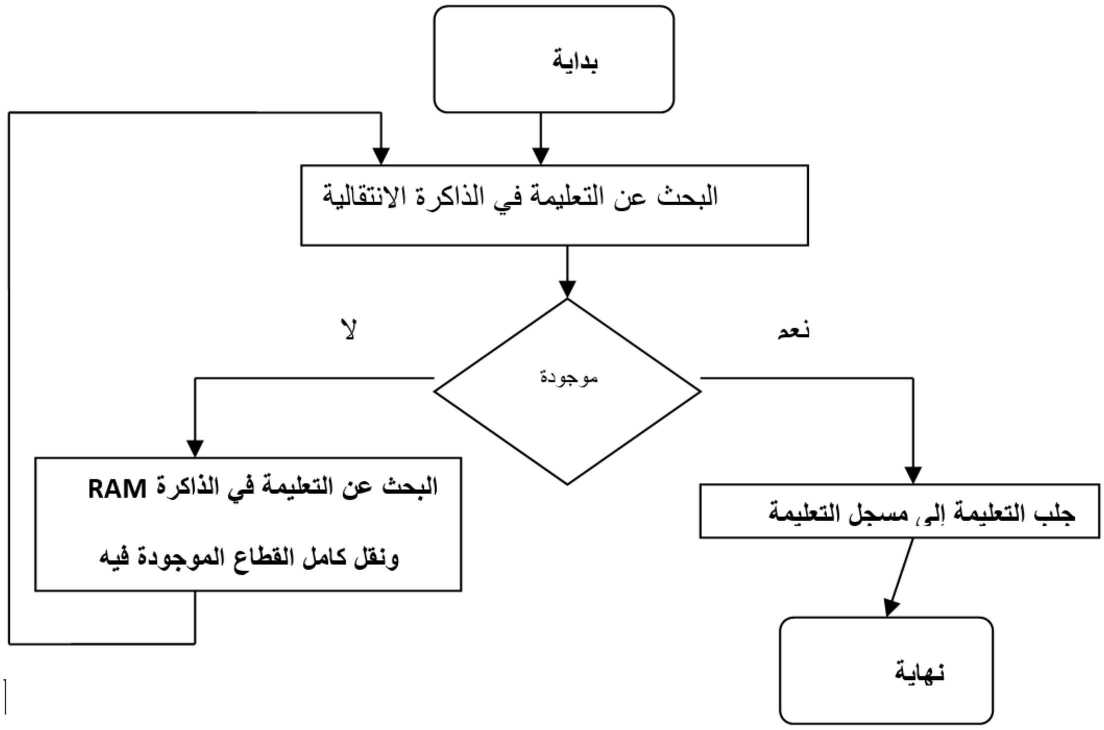
بالنسبة للطريقة الثانية من خلال استخدام المعالجة الضخية Pipelining حيث تجزأ التعليمة Instruction إلى عدة أجزاء تنفذ كل منها على معالج بحيث يكون عدد الأجزاء مساو لعدد المعالجات المرتبطة مع بعضها البعض بشكل ضخي وهي موجودة داخل

---

[2] - نقصد هنا الذاكرة L1

[3] - نفترض هنا وجدود الذاكرة L1 فقط

شريحة المعالج الرئيسي وقد تم استخدام ذلك في المعالج 80486 لأول مرة من قبل شركة Intel في عام 1989 ثم بعد ذلك في معالجات Pentium بزيادة عدد المعالجات المرتبطة مع بعضها البعض بشكل ضخي مع كل نسخة جديدة فمثلاً في معالجات Pentium2 وصل عددها إلى 14 بينما هي في المعالج 80486 فقط 5 والتالي تجزأ التعليمة إلى خمسة أجزاء وهذا يعني أنها تنفذ في خمسة مراحل، كما أشرنا سابقاً.



الشكل (2-5) خوارزمية جلب التعليمة إلى مسجل التعليمة