الباب الثالث

التعليمات والعنونة

1-3 نظرة مختصرة للمكونات- انواع التعليمات

يمكن تقسيم مكونات الحاسب الالى إلى جزئين رئيسيين هما :
المكونات المادية : Hardware وتشمل جميع المكونات المادية والدو
المكونات البرمجية : Software وهي البرامج اللازمة لتشغيل الدوائر معينة .

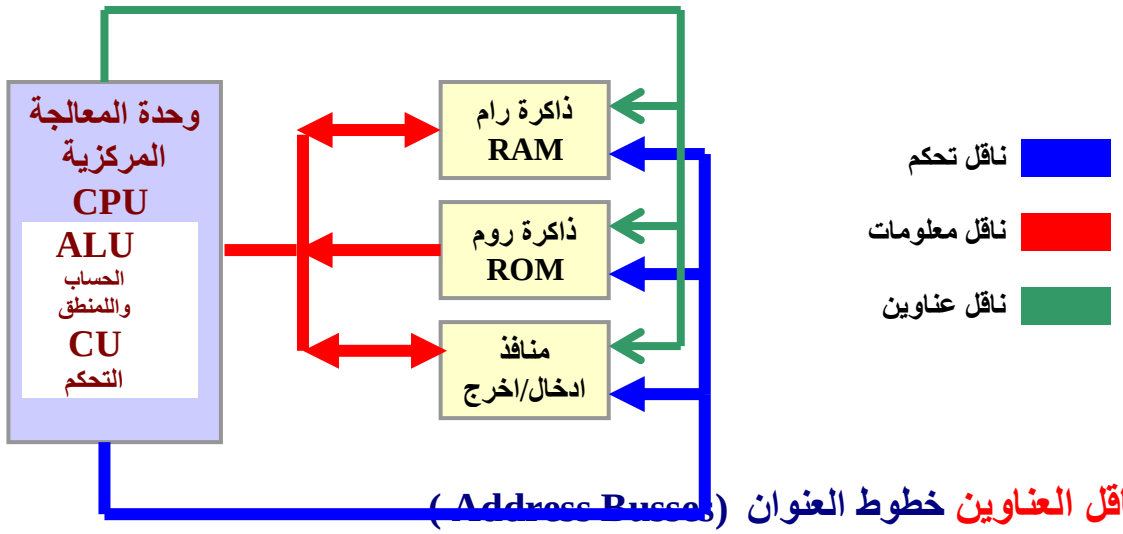
وحدة المعالجة المركزية CPU

يتكون الحاسب من الأجزاء الرئيسية التالية :

- وحدة المعالجة المركزية
- الذاكرة
- منافذ الإدخال والإخراج
- وحدات التخزين الخارجية

سنقوم بشرح هذه الأجزاء بالتفصيل ولكن دعنا نرى أولاً كيف يتم التخاطب بين هذه الأجزاء.

يكون التخاطب بين معالج البيانات و باقي الأجزاء عن طريق ما يسمى بالنواقل(Buses) كما هو موضح بهذا الشكل .



ناقل العناوين خطوط العنوان (Address Busses)

لنفرض أن المعالج لديه معلومة يريد أن يضعها في الذاكرة. فعليه أولا أن يختار الجزء من الذاكرة التي يريد أن يضعها فيه (أي عليه أن يحدد عنوان هذا الجزء) فيقوم بإرسال رمز العنوان عن طريق ناقل العناوين وعدد هذه الخطوط يحدد السعة القصوى للذاكرة التي يمكن توصيلها مع هذا المعالج الدقيق. لـ **n** خط يمكن توصيل ذاكرة حجمها 2^n بايت، وهذه الخطوط ذات اتجاه واحد (Unidirectional) دائماً خارجة من المعالج الدقيق .

ناقل المعلومات (البيانات) أو خطوط البيانات (Data Busses)

عندما يتحدد العنوان المطلوب في الذاكرة يقوم المعالج بإرسال المعلومة إلى الذاكرة عن طريق ناقل المعلومات.

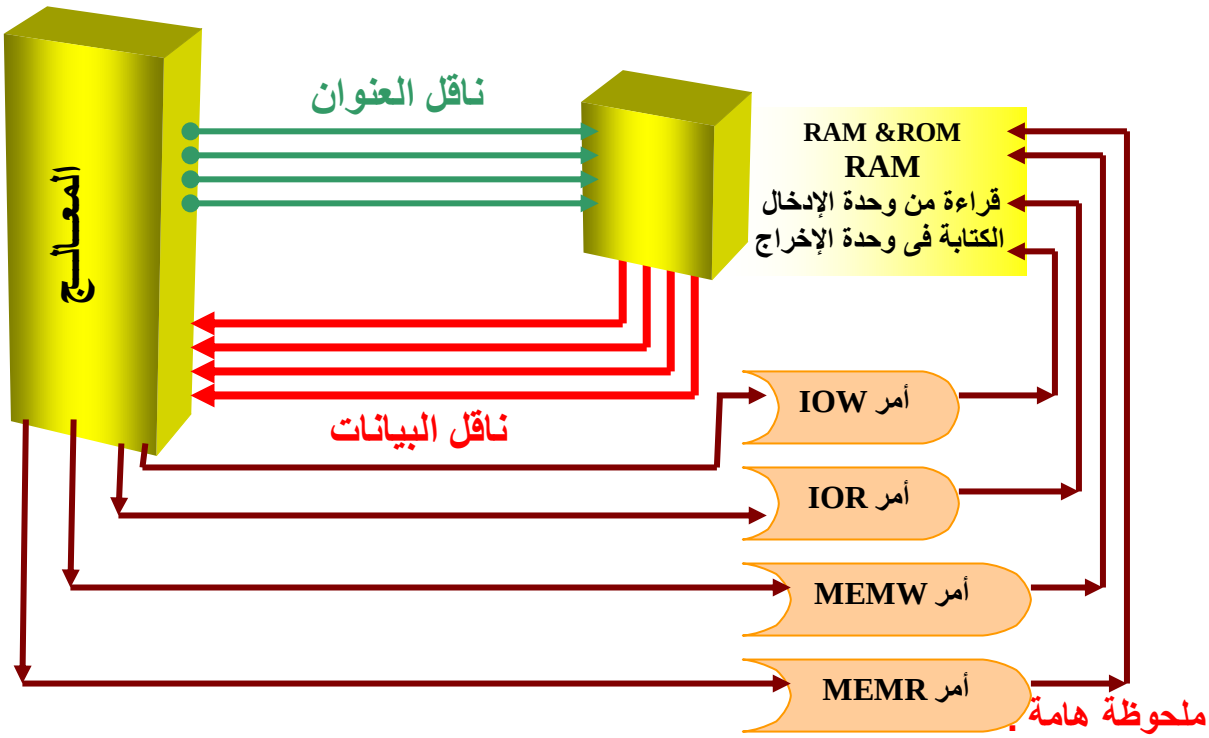
عدد هذه الخطوط يحدد عرض الموقع الذي يمكن نقله من و ألي المعالج الدقيق وهذه الخطوط ذات اتجاه ثنائي (Bidirectional)

ناقل التحكم او خطوط التحكم

يتكون ناقل التحكم من مجموعة من الأسلاك فعن طريقها يقوم المعالج بإبلاغ الأجزاء الأخرى متى يجب عليها أن تعمل مثل متى تقرأ أو تكتب وما إلى ذلك.

ولهذا الغرض للمعالج إشارة تخص القراءة (RD) وإشارة تخص الكتابة (WR) .
أما الإشارات التابعة لناقل التحكم وهى كالتالى :

الرمز	الاستخدام
MEMR	القراءة من ذاكرة الرام أو الروم (RAM-ROM)
MEMW	الكتابة فى الذاكرة الرام (RAM)
IOR	القراءة من وحدة إدخال
IOW	الكتابة فى وحدة إخراج



ملحوظة هامة :

لاحظ في الشكل السابق أن نواقل العناوين والتحكم لها اتجاه واحد فقط وهو من المعالج إلى الأجزاء الأخرى أي أن الأوامر تأتي عن طريقها من المعالج إلى الأجزاء الأخرى فقط.

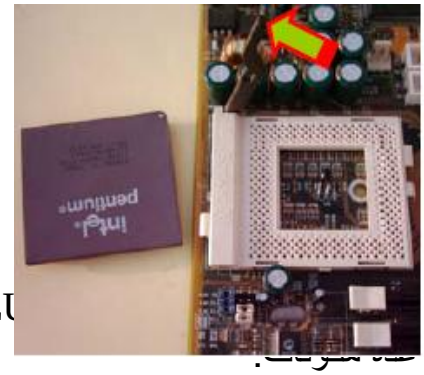
أما ناقل المعلومات فهو ذو اتجاه مزدوج حيث أن المعالج يحتاج إلى القراءة من هذه الأجزاء . كما

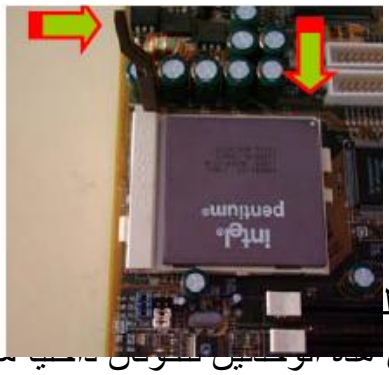
يحتاج إلى إرسال المعلومات إليها أيضاً (هذا الكلام صحيح فيما عدا ذاكرة الروم فإنه لايمكن الكتابة فيها وسنعرف لماذا عند تناول الذاكرة ، ولذلك نلاحظ أن السهم الخارج من الروم هو في اتجاه واحد فقط).

وحدة المعالجة المركزية : (CPU)

هو الملك المتوج على عرش الحاسب الآلى . حيث أنه لا يمكن أن تتحرك معلومة أو تخزن إلا عن طريق أمر من المعالج ، فهو العقل المفكر داخل الحاسب الالى كما انه يقوم بجميع العمليات الحسابية والمنطقية و كذلك جلب التعليمات من الذاكرة وفك تشفيرها إلى سلسلة من العمليات البسيطة ثم تنفيذ هذه العمليات بخطوات متسلسلة.

ولشدة أهميته للحاسب ففى كثير الأحيان يطلق اسمه على الحاسب ككل فمثلا يقال جهاز بنتيوم 4 - 2600 ميجا هيرتز ويقصد بهذا التعبير على الجهاز ككل رغم أن هذا ليس اسم الجهاز ولكنه اسم المعالج فقط .





يتكون المعالج ال ALU)ووحدة التحكم (CU) وكل من

الذاكرة الرئيسية Main Memory

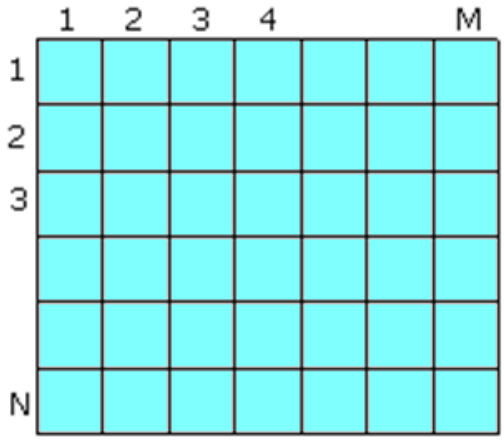
لتنفيذ العمل لابد من وجود البرنامج. و البيانات اللازمة في الذاكرة الرئيسية بصورة مؤقتة حتى تتم عملية المعالجة ويتم نقل النتائج إلى وحدات الإخراج وكما نعلم أن الذاكرة الرئيسية تتألف من مجموعة خلايا ثنائية على شكل مصفوفة وتحمل الخلايا في السطر الواحد نفس العنوان أو الموقع، ومسجل بيانات مؤقت يسمي مسجل الكلمة Word Register ووحدة تحكم محلية . و ترتبط الذاكرة مع وحدة التحكم عن طريق مسجل العنوان (باستخدام خطوط العنوان) وبعض إشارات التحكم لتحديد عملية القراءة أو الكتابة.

تركيب الذاكرة الرئيسية:

يجب تمثيل رموز البيانات بالنظام الثنائي حتى يستطيع الحاسب معالجة هذه الرموز لهذا فإن الذاكرة الرئيسية تتكون من عدد من الخلايا Cells قادرة على تمثيل الأرقام الثنائية (صفر أو واحد) لذا يمكن لهذه الخلايا أن تمثل بالمفاتيح حيث أن حالة المفتاح المغلق تمثل الواحد و المفتوح تمثل الصفر أو يمكن أن تمثل بواسطة الخلايا المغناطيسية التي يسري فيها التيار الكهربائي.فعند سريان التيار الكهربائي في الخلية المغناطيسية فأنه سوف يولد بها مجالاً مغناطيسياً و اتجاه هذا المجال يحدد الحالة التي تقع فيها الخلية.فإذا كان اتجاه المجال مع

عقارب الساعة فإن الخلية تقع في حالة الواحد و في حالة الصفر إذا كان المجال بعكس عقارب الساعة

ومهما كان نوع خلايا الذاكرة فإنه يمكن اعتبار الذاكرة لوحة الكترونية مكونة من خلايا بحيث تشكل هذه الخلايا المصفوفة والشكل التالي يوضح ذلك.



تمثل أسطر هذه المصفوفة المواقع في الذاكرة . فالأعمدة فتمثل الكلمات ويختلف طول الكلمة (عدد الخلايا في السطر) من حاسب لآخر فقد تحتوي الكلمة على أربعة خلايا "بت" bit أو ثمان أو ست عشرة .

ولتحديد كلمة ما في الذاكرة لابد من تحديد موقعها Address الممثل بالسطر فالكلمة الثالثة يتم الوصول إليها عن طريق العنوان 3 وهكذا .ومن هنا يتبين لنا تركيب الذاكرة الرئيسية حيث تضم هذه الذاكرة الأجزاء الرئيسية التالية:

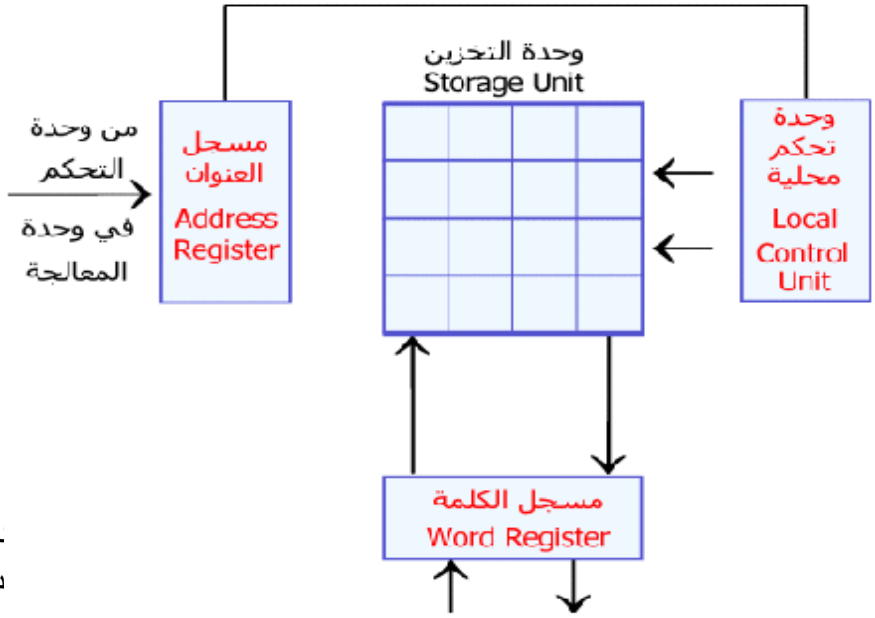
1. **مسجل العنوان address register** حيث يخزن في المسجل عنوان الكلمة المراد الوصول إليها (رقم السطر في المصفوفة)

2. **مسجل الكلمة word register** حيث يسجل في المسجل الكلمة المراد الوصول إليها (الأعمدة المشار إليها بالعنوان المسجل في مسجل العنوان)

3. **وحدة تحكم محلية Control Unit** للإشراف على عمليات الوصول إلى الكلمات (القراءة والكتابة)

4. **وحدة التخزين Storage Unit والممثلة بالمصفوفة نفسها (خلايا الذاكرة).**

والشكل يوضح تركيب الذاكرة الرئيسية:



يلعب مسج ـ الخلايا) يحدد عدد
المواقع es ت فإن عدد المواقع
يساوي أي 256 موقعاً. أما مسجل الكلمة فيحدد طول الكلمة التي يمكن تخزينها في الذاكرة.

مثال: إذا علمت أن طول مسجل العنوان يساوي **8** خلايا وطول مسجل الكلمة **4** خلايا فاحسب حجم الذاكرة:

الحل:

عدد المواقع(Addresses) $= 2^8 =$ 256 موقعاً

طول الكلمة = 4 بت

حجم الذاكرة = 256 × 4 = 1024 بت

حجم الذاكرة بالكلمات = 256 كلمة (على اعتبار أن الموقع الواحد يخزن كلمة)

حجم الذاكرة بالبايت $= \frac{1024}{8} =$ 128 بايت (البايت = 8 بت)

من أهم العمليات التي يمكن أجراؤها على البيانات هي عملية القراءة و الكتابة حيث تتم **عملية القراءة** كما يلي:

١. يؤخذ العنوان من وحدة المعالجة المركزية (وحدة التحكم) ويخزن في مسجل العنوان.
2. تقوم وحدة التحكم المحلية بالإشراف على عملية البحث عن الكلمة المحدد عنوانها في مسجل العنوان .

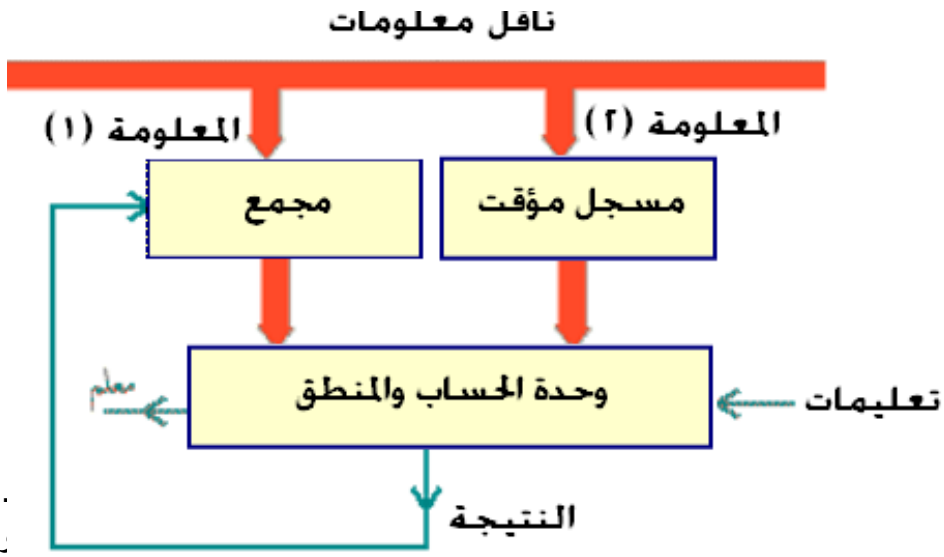
١. عند إيجاد الكلمة المعينة تحت العنوان المحدد يتم نقلها إلى مسجل الكلمة وبعدها تنقل إلى وحدة المعالجة.

أما عملية الكتابة فتتم حسب الخطوات التالية:

1. يحدد العنوان المراد تسجيل الكلمة فيه بوضع هذا العنوان في مسجل العنوان.
2. تتم عملية البحث عن المواقع بإشراف وحدة التحكم المحلية.
3. تسجل الكلمة في مسجل الكلمة وبعد تحديد الموقع تنقل من هذا المسجل إلى الموقع المحدد.

2-3 التعليمات البسيطة للعمليات الحسابية والمنطقية.

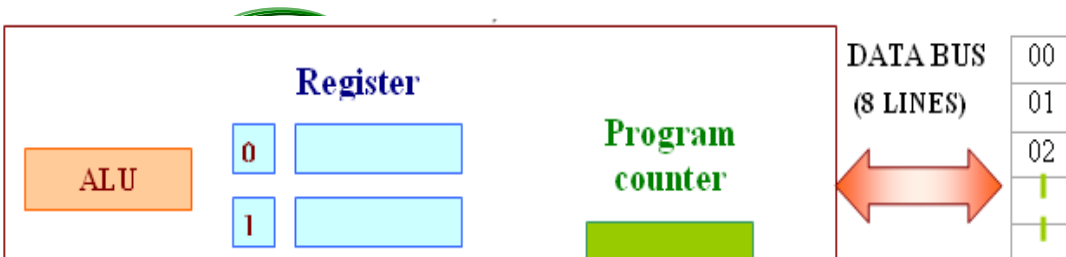
يحتوي معالج البيانات على العديد من الخلايا تسمى خانات التسجيل (registers) وهي خلايا ذاكرة داخل المعالج نفسه .كما يحتوي المعالج على وحدة تسمى بوحدة الحساب و المنطق وهي تقوم بالعمليات الحسابية. وطريقة عملها يمكن تلخيصها بهذا الشكل التالي:



تأتي التعلي
الأرقام من
الحسابية.
. ثم يأتي أحد
تتم العملية

بعد انتهاء العملية الحسابية تخزن النتيجة في المجمع وتحدد نوع النتيجة إذا كانت سالبة مثلا في المعلم (flag).

ويبين الشكل التالي هيكل لآلة افتراضية.



ويمكن تصنف التعليمات كما يلى :

1 . تعليمات نقل البيانات DATA TRANSFER INSTRUCTIONS

ومن امثلتها :-

التحميل - التخزين

" LOAD , STORE"

2. العمليات الحسابية و المنطقية ARITHMETIC& LOGIC INSTRUCTIONS

ومن امثلتها :

الجمع – الطرح – المقارنة – التحريك – النقل ....

ADD , AND ,OR , XOR , SHFT , ROTATAE

3. تعليمات التحكم "CONTROL INSTRUCTIONS"

ومن امثلتها:-

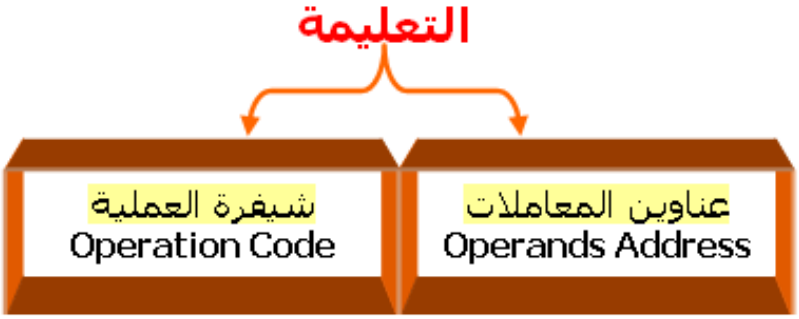
القفز او التشعب و الايقاف او الانهاء

JUMP (OR BRANCH) , HALT (OR STOP)

و تتكون التعليمات من ثلاث مجموعات (مثال من المعالج INTEL 8085)

- مجموعة تعليمات تحتل موقع واحد فى الذاكرة (1 Byte)
- مجموعة تعليمات تحتل موقعين فى الذاكرة
- مجموعة تعليمات تحتل ثلاثة مواقع فى الذاكرة

الموقع الاول دائما يحتوى على رمز العملية (The Operation code) Op – code والموقع الثانى يحتوى عناوين المعاملات (The operand)

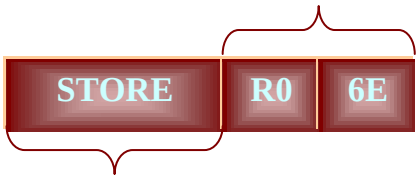


3-3 تعليمات التحميل والتخزين

LOAD , STORE"

١ - التخزين STORE

Operands-Address
عناوين المعاملات



Operation-code
شفرة العملية

هذا يعنى خزن محتويات المسجل (0) فى الذاكرة بالعنوان (6E)

ويمكن كتابة الامر على شكل ترميز كالنحو التالى:-

STORE R5, F5 ⟶ M [F5] ⟵ R5

هذا يعنى خزن محتويات المسجل (5) فى الذاكرة بالعنوان (F5)

2- التحميل

LOAD R0, AB ⟶ R0 ⟵ AB

هذا يعنى ضع محتويات AB فى السجل (0)

و اليك بعض الامثلة للتوضيح

مثال (1)

اشرح خطوات البرنامج التالى ثم اعد كتابة البرنامج مستعين بالترميز

1- الخطوة الاولى	156C
2- الخطوة الثانية	166D
3- الخطوة الثالثة	5056
4- الخطوة الرابعة	306E
5- الخطوة الخامسة	C000

الحل

1- الخطوة الاولى : حمل المحتويات الموجودة فى عنوان الذاكرة 6C فى المسجل رقم(5)

2- الخطوة الثانية : حمل المحتويات الموجودة فى عنوان الذاكرة 6D فى المسجل رقم(6)

3- الخطوة الثالثة : اجمع محتويات المسجل رقم (5) و محتويات المسجل رقم (6)

وضع النتيجة فى المسجل (0)

4- الخطوة الرابعة : خزن محتويات المسجل (0) فى عنوان الذاكرة (6E)

5- الخطوة الخامسة : توقف

و السؤال الان كيف عرفت الخطوات من الارقام والحروف ؟

هل نقوم بحفظ كل حرف وما يعنيه لا طبعا لانها بالغة الصعوبة . ولذا هناك

جدول يقوم بالترجمة وما عليك الا ان تفهم كيفية الترجمة و سنأخذ مثال يوضح ذلك بعد التعرف والاطلاع على جدول الترميز.

Machine language

.

Op-code	operand	Description
1	**RXY**	**LOAD** the register R with the bit pattern found in the Memory cell whose address is XY. Example: 14A3 would causes the contents of the memory cell located at address A3 to be place in Register 4
2	**RXY**	**LOAD** the register r with the bit pattern XY. Example:20A3 would cause the value Aa3 to be Placed in register 0
3	**RXY**	**STORE** the bit pattern found in register r in the Memory cell whose address is XY. Example :35B1 would cause the cont extents of Register 5 to be placed in the memory cell whose Abbesses B1
4	**ORS**	**MOVE** the bit pattern found in register R to register S Example: 40 A4 would cause the contents of register A to be copicd in to register 4.
5	**RTS**	**ADD** the bit patterns in registcrs S and T as though they were twos complement representations and leave the result in register R. Example: 5726 would cause the binary values in registers 2 and 2 to be added and the sum placed In register 7.
6	**RST**	**ADD** the bit patterns in registers S and T though they represented values in floating point notation and leave the floating point result in register R Example:634E would cause the values in registers 4 and E to be added as floating point values and the result to be placed in register 3
7	**RST**	**OR** the bit patterns in registers s and t and place result in register R example : 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C
8	RST	**AND** the patterns in registers S and T and result in register R . Example：8045 would cause the result of adding the contents of registers 4 and 5 to be placed in register 0

Op-code	operand	Description
9	RST	**EXCLUSIVE** OR the bit patterns in registers S and T and place the result in register R Example: 95f3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5
A	ROX	**ROTATE** the bit pattern in register R one bit to the right X times. Each time places the bit that started at the low–order end at the high-order end Example: a403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.
B	RXY	**JUMP** to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register 0. Otherwise continue with the normal sequence of execution. (the jump is implemented by copying XY in to the program counter during the execute phase,) Example: B43C would first compare the contents of register 4 with the contents of register 0. if the two were equal the pattern 3C would be placed in the program counter so that the next instruction executed would be the one located at that memory address. Otherwise. Nothing would be done and program execution would continue in its normal sequence
C	000	**HALT** execution. Example: C000 would cause program execution to stop.

ترجمة الجدول

لغة الآلة

كل أمر آله طوله 2 بايت . الـ 4 بت الأولى تمثل رمز العملية ؛ تمثل الـ 12 بت الأخيرة حقل العمليه. القائمه التاليه تشمل الترقيم السداسى عشر مع وصف مبسط مِن الذاكرة الوسيطةِ لكل عملية . الحروف R، S، T تستعمل بدلاً من الأرقام السداسى عشر في تلك الحقول والتى تمثل سجلّ العنصر الأكثر ترتيباً الذي يغير الاعتماد على التطبيق المعيّن للأمر. إنّ الرسائل X و Y تستعمل بدلاً عَن أرقام سداسي عشر في الحقول المتغيّرة و لا تمثل سجل

كود	المعامل	وصف العملية
1	RXY	حمل المسجل R بالقيمة الموجودة فى خلية الذاكرة التي عنوانها XY. **مثال** : 14A3 يحمل محتويات خليةِ الذاكرةَ الموجودة فى العنوا ن A3 لِكي تكون في مسجل 4
2	RXY	حمل المسجل R بالقيمة XY مثال : 20A3 لتحميل القيمة A3 في المسجل (0)
3	RXY	لتخزين محتويات المسجل R في خلايا الذاكرة التي تكون في العنوان XY **مثال :** 35B1 لتخزين محتويات المسجل 5 توضع في خلايا الذاكرة التي عنوانها B1
4	ORS	نقل القيمة الموجود بالمسجل R إلى المسجل S **مثال** : 40A4 لجعل محتويات المسجل A تنقل للمسجل4
5	RST	اجمع القيم الموجوده في المسجل T، S معا وتوضع النتيجة في المسجل R على هيئة المكمل الثنائى **مثال** : 5726 لجعل القيمة الثنائية الموجودة في المسجل 2 تضاف علي القيمة الموجودة داخل المسجل 6 ووضع ناتج الجمع في المسجل 7 على صورة المكمل الثانى

كود	المعامل	وصف العملية
6	RST	اجمع القيمة الموجودة فى المسجل T,S معا وضع الناتج فى المسجل R ويضاف لقيمة النقطة العائمه **مثال**: 634E اجمع القيمة الموجوده فى المسجل 4 , E . تضاف لقيمة النقطة العائمة ويوضع الناتج في المسجل 3
7	RST	تنفيذ دالة OR على القيمة الموجودة فى المسجل S,T ثم ضع الناتج بالمسجل R **مثال** : 7CB4 ضع نتيجة الجمع المنطقى(OR) لمحتويات المسجلات 4 , B توضع فى المسجل C
8	RST	تنفيذ دالة AND على القيمة الموجودة فى المسجل S,T ثم ضع الناتج بالمسجل R **مثال** : 8045 ضع نتيجة عمليةالضرب المنطقى (AND) لمحتويات المسجلات 4 و 5 لتوضع في سجل (0)
9	RST	تنفيذ دالة XOR للفيم الموجودة في المسجل T,S وتوضع النتيجة في المسجل R **مثال** : 95F3 ضع نتيجة عملية XOR لمحتويات المسجل F و 3 وضع الناتج فى المسجل 5
A	ROX	دُورْ القيمة الثنائية الموجودة في المسجل R خانة واحده عدد X مرة الي اليمين وضعها فى R **مثال** : A403 لجعل القيمة الموجودة بالمسجل 4 تدور 3خانات من اليمين للايضاح : نفترض ان القيمة الموجودة بالمسجل 4 =01011100 بعد عملية الدوران 3 خانات يكون الناتج النهائى للمسجل 4=100010111

كود	المعامل	وصف العملية
B	RXY	أقفز إلى الأمر الموجود في خليةِ الذاكرةَ الموجودة في العنوان XY إذا كانت القيمة الموجودة بالمسجل Rمساوية للقيمة الموجودة بالمسجل غير ذلك تتابع التنفيذ يستمر بطريقة طبيعية (القفزة مطبقة بواسطة نسخ XY في عداد البرنامج لتنفيذ الامر الموجود بXY مثال: B43C أولا مقارنة محتويات المسجل 4 مع محتويات المسجل (0) واذا تساوي الاثنين المؤشر 3C يوضع في عداد البرنامج لذلك تنفذ الامر القادم الموجود داخل المسجل 3C وإذا لم يتساوى لاتقوم باى عمل على عنوان الذاكرة XY ولكن يستمر تتابع تنفيذ البرنامج
C	000	**التوقف** **المثال : C000 توقف عن تنفيذ البرنامج**

مثال

0100 0000 1111 0100

4 0 F 4

من الجدول يتم الترجمة حيث رقم (4) تعبر عن الكود و (O R X) تعبر عن المعامل

0100 0000 1111 0100

4 0 F 4

4 O R X

وعلى ذلك يكون الامر

- انقل محتوبات المسجل (F) الى المسجل رقم(4)

4-3 تعليمات القفز والتشعب

الغاية من تعليمة القفز هي تعديل طريق تنفيذ التعليمات في البرنامج. و هناك نوعان من تعليمات القفز هما :

1- القفز غير المشروط

في القفز غير المشروط لا يوجد أي شروط لحدوث القفز

2- القفز المشروط.

أما في القفز المشروط فإن الحالات الشرطية الموجودة في لحظة تنفيذ تعليمة القفز تتخذ القرار فيما إذا سيحدث القفز أم لا، ففي حال تحقق الحالات الشرطية فإنه يتم القفز، و إلا يُتابع التنفيذ بالتعليمة التي تلي تعليمة القفز في البرنامج

1) تعليمة القفز غير المشروط

و هي مشروحة في الجدول التالي:.

العملية	الصيغة	المعنى	الكلمة المختزلة
القفز إلى العنوان المحدد بواسطة operand المتحول	JMP operand	قفز غير مشروط	JMP

هناك نوعان أساسيان من القفز غير المشروط

الأول :- القفز ضمن المقطع الجزئي،

الثاني:- القفز بين المقاطع الجزئية

أي يُمَكِنُنا من القفز من أحد مقاطع الشفرة إلى مقطع آخر و إن تحقيق هذا النوع من القفز يتطلب منا تعديل محتويات كل من مقطع ال CS و مسجل مؤشر التعليمة IP ، أما القفز ضمن المقطع الجزئي فإنه يتطلب منا تعديل قيمة الIP فقط.

إن المتحولات المسموحة لتعليمة القفز غير المشروط هي :

Operand		
Short_Label	متحول اللافتة القصيرة	للقفز ضمن المقطع الجزئي
Near_Label	متحول اللافتة القريبة	
Memptr16	مؤشر ذاكري 16 بت	
Regptr16	مؤشر مسجلي 16 بت	
Far_Labrl	اللافتة البعيدة	للقفز بين المقاطع الجزئية
Memptr32	مؤشر ذاكري 32 بت	

القفز ضمن المقطع الجزئي

أ) إن متحولات اللافتة القصيرة و اللافتة القريبة تحدد القفز النسبي لعنوان تعليمة القفز نفسها فمثلاً في تعليمة القفز باللافتة القصيرة يتم تشفير العدد ذي 8 بت كمتحول فوري لتحديد الإزاحة (Disp) ذات الإشارة التي تشير إلى التعليمة التالية التي سيتم تنفيذها من حجرة تعليمة القفز، و عندما تنفذ تعليمة القفز يعاد شحن الIP بقيمة جديدة موضحة كما يلي:

قيمة IP الجديدة = [(قيمة IP + طول شفرة تعليمة القفز) + مقدار الإزاحة ذات الإشارة بعد تمديدها بجعل متحول 8 بتات بالشكل 16 بت]

إن القيمة الجديدة لـ IP مع قيمة CS الحالية تعطي العنوان الفيزيائي للتعليمة التالية التي ستجلب و تنفذ.

مثال (1)

ليكن لدينا

```
IP = 0112h
JMP disp  ;  disp = 0F2h
```

إن عنوان تعليمة القفز (موجود تحت العنوان المخزن في IP) ، إذن سيتم القفز إلى التعليمة ذات العنوان التالي:

العنوان المنطقي address = IP + 2 + disp (بعد تمديد إشارتها) = 0112 + 2 + FFF2 = 0106h (أهملنا خانة الحمل)

بما أن العنوان الناتج أصغر من عنوان تعليمة القفز فهذا يعني أننا قفز إلى تعليمة تسبق تعليمة القفز أي القفز إلى الـوراء 0112 > 0106 .

مثال (2)

IP = 0112h
JMP 04
Address = 0112 + 2 + 0004 = 0118h

نلاحظ أن 0112 < 0118 فهذا يعني أن القفز نحو الأمام.
و للحصول على العنوان الفيزيائي يجب إضافة مقدار الCS لقيمة address .

ملاحظة: بما أن متحول اللافتة القصيرة ذا 8 بت فهو يسمح بالقفز في المجال من 126- إلى 129+ و سبب ذلك أنه إذا أضفنا طول شفرة تعليمة القفز و هو 2 بايت إلى المجال التالي من -128 إلى +127 سنحصل على المجال السابق. أما متحول اللافتة القريبة فهو متحول فوري ذو 16 بت و لذلك يسمح بالقفز ضمن مجال يساوي 32KB نحو الخلف أو نحو الأمام من عنوان تعليمة القفز.

مثال (3)

JMP label

هذا يعني القفز إلى نقطة في البرنامج مقابلة للمتحول label حيث تتم إضافة هذا المتحول (الإزاحة 16 بت) إلى قيمة ال IP و القيمة الجديدة لـ IP و القيمة الحالية في CS تعطي العنوان الفيزيائي للتعليمة التي ستنفذ .

ب) يمكن تحديد القفز إلى عنوان بشكل غير مباشر بواسطة محتويات حجرة ذاكرة أو محتويات مسجل أي باستخدام متحول مؤشر ذاكري 16 بت أو متحول مؤشر مسجلي 16 بت و هنا أيضاً يتم القفز ضمن مجال $\pm 32KB$.

مثال(4)

JMP BX

في هذه التعليمة يُستعمل مضمون المسجل BX من أجل الإزاحة و هذا يعني أن قيمة BX يتم تحميلها في IP ثم يحسب العنوان الفيزيائي للتعليمة التي سيتم القفز إليها باستعمال المحتويات الحالية لـ CS و القيمة الجديدة لـ IP .

بفرض أن :

BX = 0200h
CS = 0100h

العنوان الفيزيائي للتعليمة التي سيتم القفز إليها

PA = (CS x 10h) + BX = 01000 + 0200 = 01200h

ملاحظة : يمكن استخدام مختلف أنواع أنظمة العنونة لتحديد المتحول المستعمل كمؤشر ذاكري فمثلاً JMP [SI] ففي هذه التعليمة تستعمل محتويات SI كعنوان حجرة الذاكرة التي تحتوي على العنوان الفعال، هذا العنوان يتم تحميله في IP و الذي يُستعمل مع محتويات CS الحالية لحساب العنوان الفيزيائي للتعليمة التي سيتم القفز إليها و عادة في هذه الحالة تستخدم المسجلات التالية:DI, SI, BX .

القفز بين المقاطع الجزئية أو القفز خارج المقطع الجزئية

أ) تَستعمِل اللافتة البعيدة متحولاً فورياً ذا 32 بت لتحديد القفز إلى عنوان ما. حيث يتم تحميل الـ 16 بت الأولى من هذا المتحول في IP و تكون هي العنوان الفعال نسبة لمحتويات المسجل CS أما الـ

16 بت الثانية فيتم تحميلها في المسجل CS و التي تحدد مقطع الشفرة الجديد.

مثال(5)

JMP farlabel

حيث farlabel هو متحول بـ 32بت (الكلمة الأول تشحن في IP و الكلمة الثانية تشحن في الـ CS)

ب) إن الطريقة غير المباشرة لتحديد العنوان الفعال و عنوان مقطع الشفرة من أجل القفز بين المقاطع الجزئية هي باستعمال متحول مؤشر ذاكري بـ 32 بت. و في هذه الحالة فإن أربع بايتات من الذاكرة متتابعة اعتباراً من العنوان المحدد تحتوي على العنوان الفعال و عنوان مقطع الشفرة الجديد على الترتيب. و هنا أيضاً يمكن استخدام أي نوع من أنواع أنظمة العنونة المختلفة،

مثال(6)

JMP farseg [DI]

ففي هذه التعليمة تُستعمل محتويات DS, DI لحساب عنوان حجرة الذاكرة التي تتضمن الكلمة الأولى للمؤشر الذي يُعرِّف الحجرة التي سيتم القفز إليها، فإذا كان :

DI = 0200h
DS = 0100h

} **العنوان الفيزيائي للمؤشر هو**

PA = DS x 10h + DI = 01000 + 0200 =01200h

و لتكن محتويات هذه الحجرة و الحجرات التي تليها كما هو واضح في الشكل التالي:

Address (h)	Content	قيمة IP الجديدة هي IP = 3010h
01200	10	قيمة CS الجديدة هي CS = 0400h
01201	30	إذن العنوان الفيزيائي للتعليمة التي سيتم القفز إليها هو:
01202	00	PA = CS x 10h + IP = 07010h
01203	04	

2) تعليمة القفز المشروط
و هي مشروحة في الجدول التالي:

العملية	الصيغة	المعنى	الكلمة المختزلة
إذا تحقق الشرط cc فإنه يتم القفز إلى العنوان المحدد بواسطة المتحول و إلا فيتم تنفيذ التعليمة التالية لتعليمة القفز	Jccمتحول	قفز مشروط	Jcc

هناك 18 من تعليمات القفز المشروط و هي مشروحة في الجدول التالي:

المعنى	الكلمة المختزلة
القفز إذا كان CF = 1	JC
القفز إذا كان CF = 0	JNC
القفز إذا كان OF = 1	JO
القفز إذا كان OF = 0	JNO
القفز إذا كان SF = 1	JS
القفز إذا كان SF = 0	JNS
القفز إذا كان CX = 0000	JCXZ
القفز في حالة التساوي/أو إذا كان الناتج يساوي الصفر	JE/JZ
القفز إذا كان أكبر أو يساوي/القفز إذا لم يكن أصغر	JGE/JNL
القفز إذا كان فوق/القفز إذا لم يكن تحت أو يساوي	JA/JNBE
القفز إذا كان فوق أو يساوي/القفز إذا لم يكن تحت	JAE/JNB
القفز إذا كان تحت/القفز إذا لم يكن فوق أو يساوي	JB/JNAE
القفز إذا كان تحت أو يساوي/القفز إذا لم يكن فوق	JBE/JNA
القفز إذا كان أكبر/القفز إذا لم يكن أصغر أو يساوي	JG/JNLE
القفز إذا كان أصغر أو يساوي/القفز إذا لم يكن أكبر	JLE/JNG
القفز إذا لم يكن يساوي/القفز إذا كان الناتج يساوي قيمة غير صفرية	JNE/JNZ
القفز إذا كانت خانة Parity غير موجودة/القفز إذا كان PF = 0	JNB/JBO
القفز في حالة وجود خانة Parity/القفز إذا كان PF = 1	JP/JPE

ملاحظة:

للتمييز بين الأعداد ذات الإشارة و الأعداد بدون إشارة فإن هناك اسمين مختلفين يبدو أنهما نفس الشيء في تعليمات القفز و هما فوق (A) و تحت (B) من أجل مقارنة الأعداد بدون إشارة، و أصغر (L) و أكبر (G) من أجل مقارنة الأعداد ذات الإشارة. فمثلاً العدد ABCDh هو فوق العدد 1234h إذا اعتبرناهما عددين بدون إشارة. أما إذا اعتبرناهما بإشارة فإن ABCDh هو عدد سالب و 1234h هو عدد موجب و لذلك ABCDh هو أصغر من 1234h.

5-3 طرق العنونة

طرق عنونة الذاكرة Addressing Methods :

يستخدم حقل العنوان في التعليمة لتحديد عنوان موقع الذاكرة أو أحد مسجلات وحدة المعالجة المركزية بهدف الحصول على المعامل المطلوب أجراء العملية عليه.

تسمى الطريقة التي يتم فيها الحصول على المعامل بطريقة العنونة. والعنوان الذي يظهر مباشرة في التعليمة يسمى بالعنوان المبين Stated Address وعنوان موقع الذاكرة الذي يحتوي على المعامل يسمى بالعنوان الفعلي. Effective Address

قبل أن نستعرض طرق العنونة يجب أن نتذكر ما يلي:

- أن ذاكرة الحاسب تخزن كل من التعليمات و البيانات.
- لعنونة التعليمة يستخدم مسجل خاص في وحدة التحكم يسمى بعداد البرنامج Program Counter ولعنونة البيانات يستخدم غالباً مسجل آخر يسمى بعداد البيانات.**Data Co**unter

تستخدم في الحاسب طرق عنونة متعدد

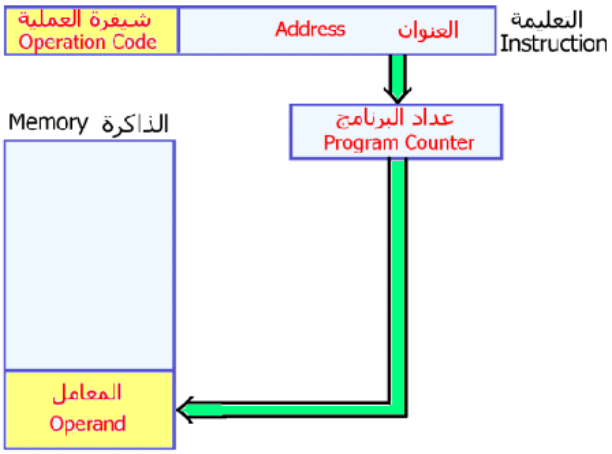
1- 5-3 العنونة الفورية: Immediate Addressing

عند استخدام هذه الطريقة فإن التعليمة تحتوي على المعامل كجزء منها. أي أن حقل العنوانيحتوي على المعامل نفسه. وفي الحواسيب الصغيرة والميكروية ، أي عند تخزين التعليمةفي أكثر من موقع ذاكرة ، فإنه يمكن الحصول على المعامل بواسطة قراءة محتوى موقعالذاكرة الذي يلي الموقع الذي تخزن فيه التعليمة.

من امثلة هذا النوع من العنونة عملية تحميل المركم فوريا LOAD A وكود العملية لهذا الامر هو (96H) وفيها يحمل المركم بالقيمة (00F0H) والتى توجد فى البايت الثانى للامر.

٣ ٥ ٢ العنونة المباشرة: Direct Addressing

وهي أكثر طرق العنونة انتشاراً ويتم فيها الحصول على العنوان الفعلي من العنوان المبين في التعليمة مباشرة. أي أن حقل العنوان يحتوي على عنوان موقع الذاكرة الذي يخزن المعامل.
الشكل يبين طريقة الحصول على المعامل في حالة العنونة المباشرة



طريقة الحصول على المعامل في العنونة المباشرة

من الأمثلة على هذه الطريقة Load Register LDR r : وتبعاً لهذه التعليمة تقرأ محتويات موقع الذاكرة المعنون بواسطة التعليمة ويجلب ليحمل في المسجل r
ومن مساوئ العنونة المباشرة أن عدد مواقع الذاكرة المعنونة محدود ،

ويمكن حل هذه المشكلة بعدة طرق منها:
1- زيادة قياس حقل العناوين وبالتالي قياس التعليمة، إلا أن هذا يتطلب زيادة قياس موقع الذاكرة.
2- تخصيص أكثر من موقع ذاكرة لتخزين التعليمة,وتستخدم هذه الطريقة في الحواسب الميكروية.
3- استخدام طرق عنونة أخرى.

٣ ـ ٥ ـ ٣ العنونة غير المباشرة : Indirect Addressing

العنوان الفعلي في هذه الطريقة هو محتوى موقع الذاكرة المعنون بواسطة التعليمة. أي أن موقع الذاكرة المعنون بواسطة العنوان المبين في التعليمة يحتوي على عنوان موقع الذاكرة الذي يخزن فيه المعامل

للحصول على المعامل لابد من الرجوع إلى الذاكرة مرتين

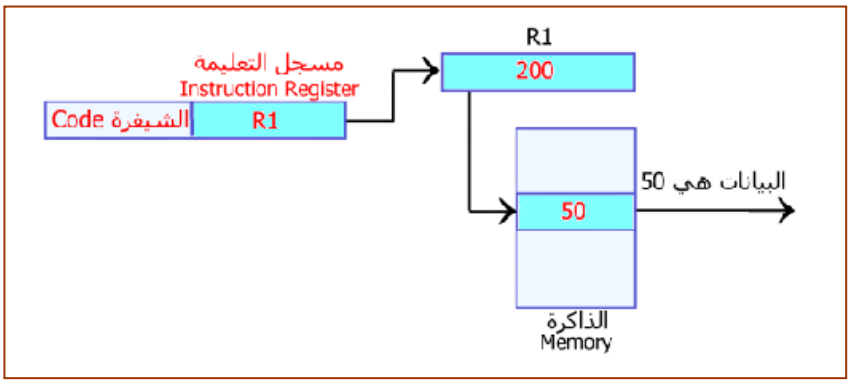
في الأولى: تقرأ محتويات موقع الذاكرة المعنون بواسطة العنوان المبين في التعليمة وتجلب إلى وحدة المعالجة المركزية (إلى عداد البيانات)

في الثانية : تقرأ محتويات موقع الذاكرة المعنون بواسطة عداد البيانات بهدف الحصول على المعامل

- لتميز نوع العنونة: هل هي مباشرة أم غير مباشرة . تستخدم عادة بت خاص، فإذا كان محتواه 1 تكون طريقة العنونة المستخدمة غير مباشرة وإذا كان محتواه تستخدم طريقة العنونة المباشرة.

توجد أنواع أخرى من طرق العنونة غير المباشرة . فمثلايمكن أن تشير التعليمة أن زوج من

مسجلات وحدة المعالجة المركزية يجب أن يستخدملعنونة الذاكرة للحصول على المعامل.

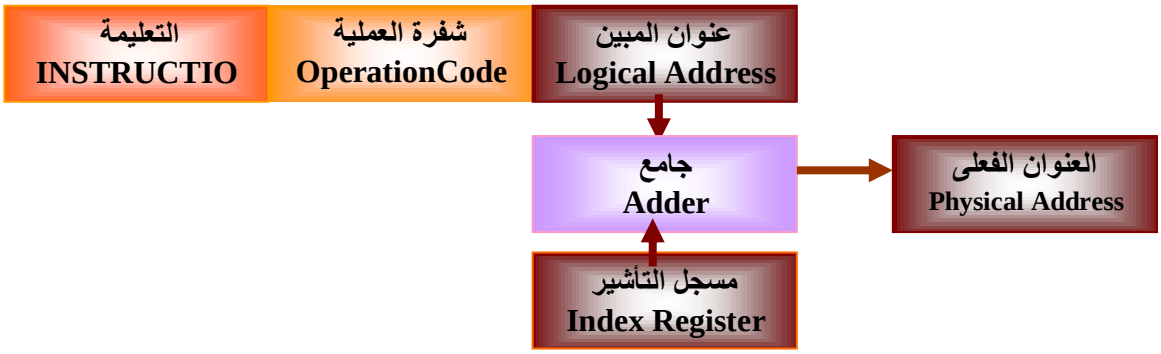


من مساوئ العنونة غير المباشرة ضرورة الرجوع إلى الذاكرة مرتين مما يؤدي إلى تدني السرعة.

4-5-3 العنونة المؤشرة: Indexed Addressing

حسب طريقة العنونة المؤشرة، للحصول على العنوان الفعلي يجب جمع العنوان المبين في التعليمة مع محتوى مسجل خاص يسمى بمسجل التأشيرIndex Register

ويبين الشكل 5-7 طريقة الحصول على العنوان الفعلي



عند التعامل مع الجداول المخزونة في مواقع متتالية في الذاكرة يمكن زيادة أو تنقيص محتوى مسجل التأشير بمقدار 1 للحصول على العنوان الفعلي، وتسمى هذه الطريقة بالتأشير الذاتيAuto Indexing

ومن مساوئ استخدام العنونة المؤشرة ضرورة إجراء عملية الجمع للحصول على العنوان الفعلي.

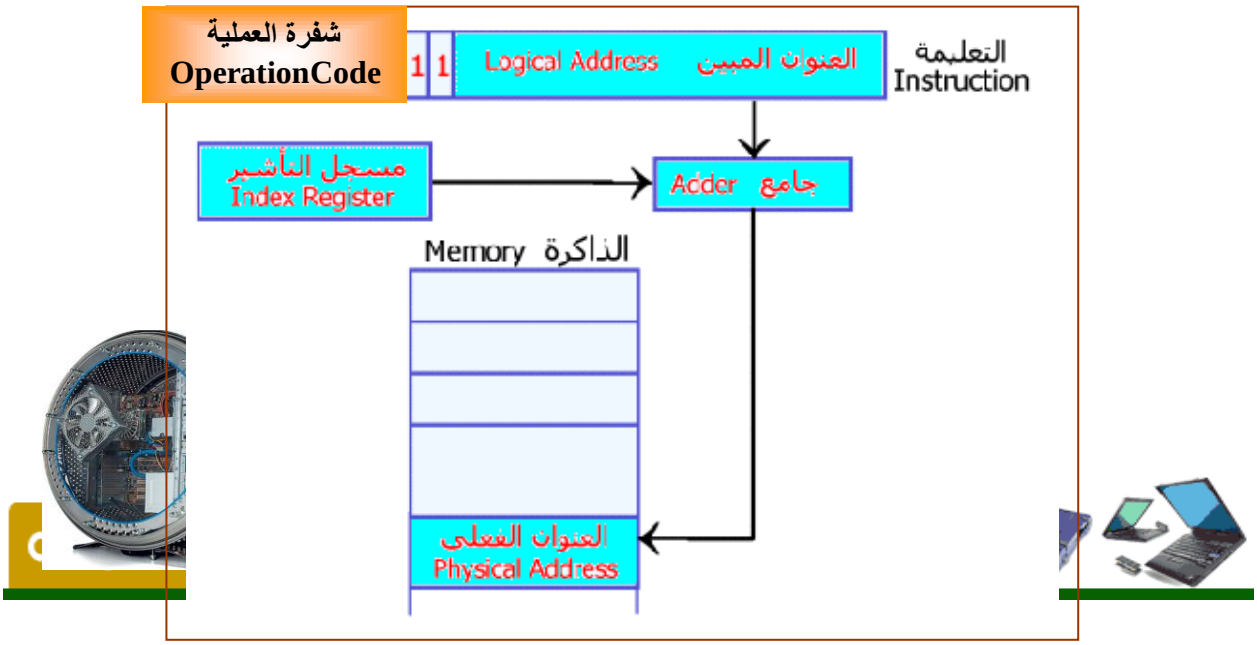
ومن مزايا هذه الطريقة بالمقارنة مع العنونة غير المباشرة , الرجوع إلى الذاكرة مرة واحدة بدلاً من مرتين.

في كثير من الحالات تستخدم طريقتا العنونة غير المباشرة والمؤشرة معاً، وفي هذه الحالة يجب أن تحتوي التعليمة على بت أضافي يدل على نوع العنونة المستخدمة.ويبين الجدول التالي طريقة العنونة المستخدمة بالاعتماد على محتوى الخانتين الإضافيتين المستخدمين لتحديد طريقة العنونة.

شيفرة العملية	خانة 1	خانة 2	العنوان

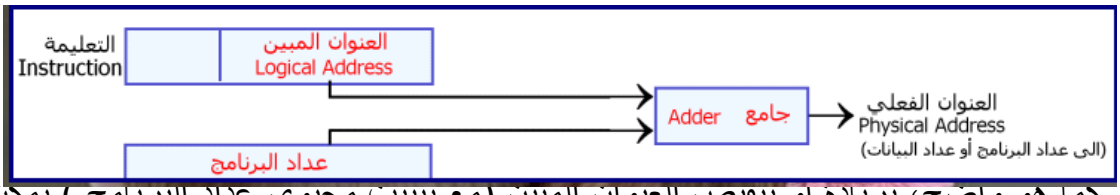
طريقة العنونة	الخانة 2	الخانة 1
عنونة مباشرة	0	0
عنونة غير مباشرة	0	1
عنونة مؤشرة	1	0
عنونة غير مباشرة - مؤشرة	1	1

ويبين الشكل طريقة الحصول على العنوان الفعلي فيالعنونة غير المباشرة_المؤشرة.



5-5-3 العنونة النسبية: Relative Addressing

سنقوم بتسمية محتوى عداد البرنامج بالعنوان القاعدي .Base Address للحصول على العنوان الفعلي في هذه الطريقة يجب جمع العنوان المبين في التعليمة مع العنوان القاعدي.
ويبين الشكل 5-9 طريقة الحصول على العنوان الفعلي في العنونة النسبية.



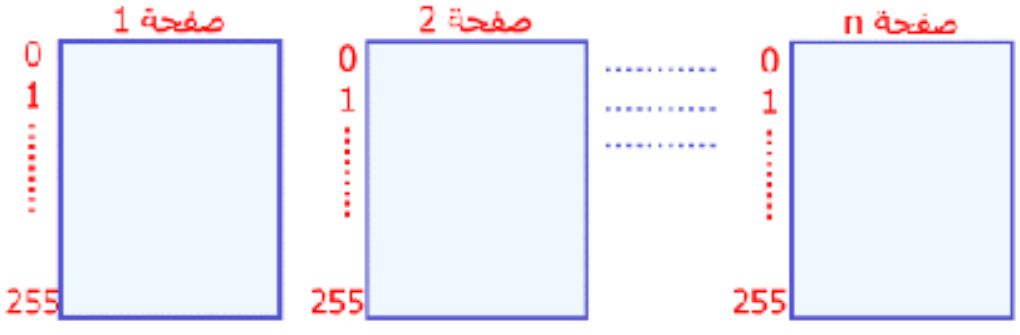
كما هو واضح، بزيادة أو تنقيص العنوان المبين (مع تثبيت محتوى عداد البرنامج) يمكن التفرع إلى الأمام أو الخلف. لهذا السبب يسمى العنوان المبين في التعليمة بقيمة الإزاحة Displacement وعادة توضع إشارة لقيمة الإزاحة.

6-5-3 العنونة الصفحية Addressing Page

كما هو معروف فإن قياس موقع الذاكرة محدود. فمثلاً يبلغ قياس موقع الذاكرة في أغلب الحواسب الميكروية بايت واحد فقط. هذا الوضع يؤدي إلى تحديد قياس حقل العناوين في التعليمة

من جهة أخرى، إذا كان قياس حقل العناوين في التعليمة بايت واحد، فإن عدد مواقع الذاكرة الممكن عنونتها لا يزيد على 256 موقعاً، وواضح أن هذه السعة لا تسد الاحتياجات المطلوبة. وإحدى طرق حل هذه المشكلة هي زيادة سعة الذاكرة حسب الحاجة و تقسيم الذاكرة إلى صفحات Pages تحتوي كل منها على عدد متساوي من المواقع. وقياس الصفحة (عدد المواقع التي تحتويها) عادة يساوي عدد المواقع التي يمكن عنونتها بواسطة حقل العنوان في التعليمة

ويبين الشكل ذاكرة مقسمة إلى n صفحة (قياس كل صفحة 256 موقعاً).



يتألف العنوان الفعلي من جزأين: الجزء الأول يحدد الصفحة ويحدد الجزء الثاني عنوان الموقع داخل الصفحة. أي أن العنوان الفعلي هو عنوان موقع الذاكرة بالنسبة للصفحة الحالية

بمعني آخر فإنه يجب جمع عنوان أول موقع في الصفحة الحالية مع العنوان المبين للحصول على العنوان الفعلي

من مزايا العنونة الصفحية أمكانية زيادة سعة الذاكرة رغم القيود على قياس العنوان.

عند استخدام العنونة الصفحية توجد عدة مشاكل أهمها: ماذا يحدث إذا خرجت قيمة العنوان الذي يحدد الموقع في الصفحة الحالية عن مجال القيم المسموح بها، أي إذا وصلت قيمة هذا العنوان إلى 111....1؟
ولحل هذه المشكلة يمكن زيادة هذا العنوان بمقدار 1 للحصول على 00....0

وفيض يساوي 1 واستخدام أحدى الطرق التالية :-

1. إهمال الفيض وبهذا فإنه تتم عنونة الموقع الأول من الصفحة الحالية.
2. جمع الفيض مع الجزء الثاني في عداد البرنامج والذي يحدد عنوان الصفحة و بهذا نكون قد انتقلنا إلى الموقع الأول في الصفحة التالية.

تطبيقات

تنفيذ برنامج بسيط بالعنونة الفورية :

مثال (1)

أذكر بإيجاز الإجراءات اللازمة لتنفيذ برنامج بسيط لجمع الرقم 29 والعدد 8 و العدد 45 بواسطة الحاسب الدقيق

الحل

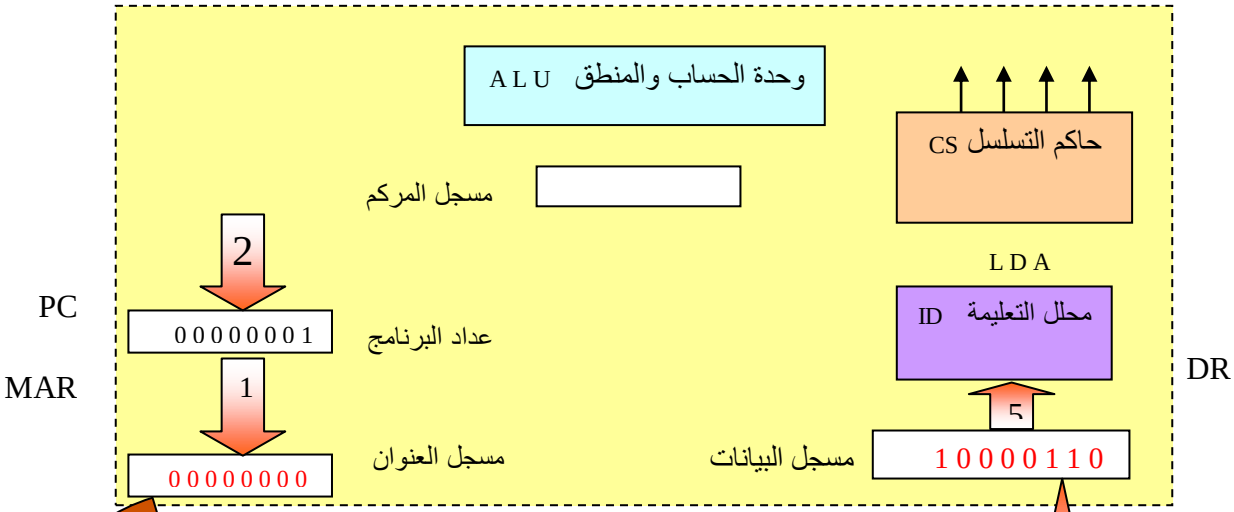
1- إحضار تعليمة LDA .
2- تنفيذ تعليمة LDA 29 .
3- إحضار تعليمة ADD الأولى (التعليمة الثانية)
4- تنفيذ التعليمة الثانية الإضافة الأولى ADD 8.
5- إحضار التعليمة الثالثة (الإضافة الثانية)
6- تنفيذ التعليمة الثالثة (الاضافة الثانية) A D D 45 .
7- إحضار وتنفيذ تعليمة التوقف H L T .

مثال (2)

نفذ برنامج بالعنونة الفورية لجمع الأعداد العشرية الآتية (29 , 8) .

الحل

أولا :- إحضار تعليمة LDA



عنوان ثنائى	محتويات ثنائية	محتويات رمزية وعشرية
00000000	10000110	LDA
00000001	00011101	29
00000010	10001011	ADD
00000011	00001000	8
00000110	00111110	HLT

خطوات احضار التعليمة الاولى بالعنونة الفورية

1- يضع عداد البرنامج عنوان التعليمة المراد تنفيذها $2_{(00000000)}$ فى مسجل العنوان .

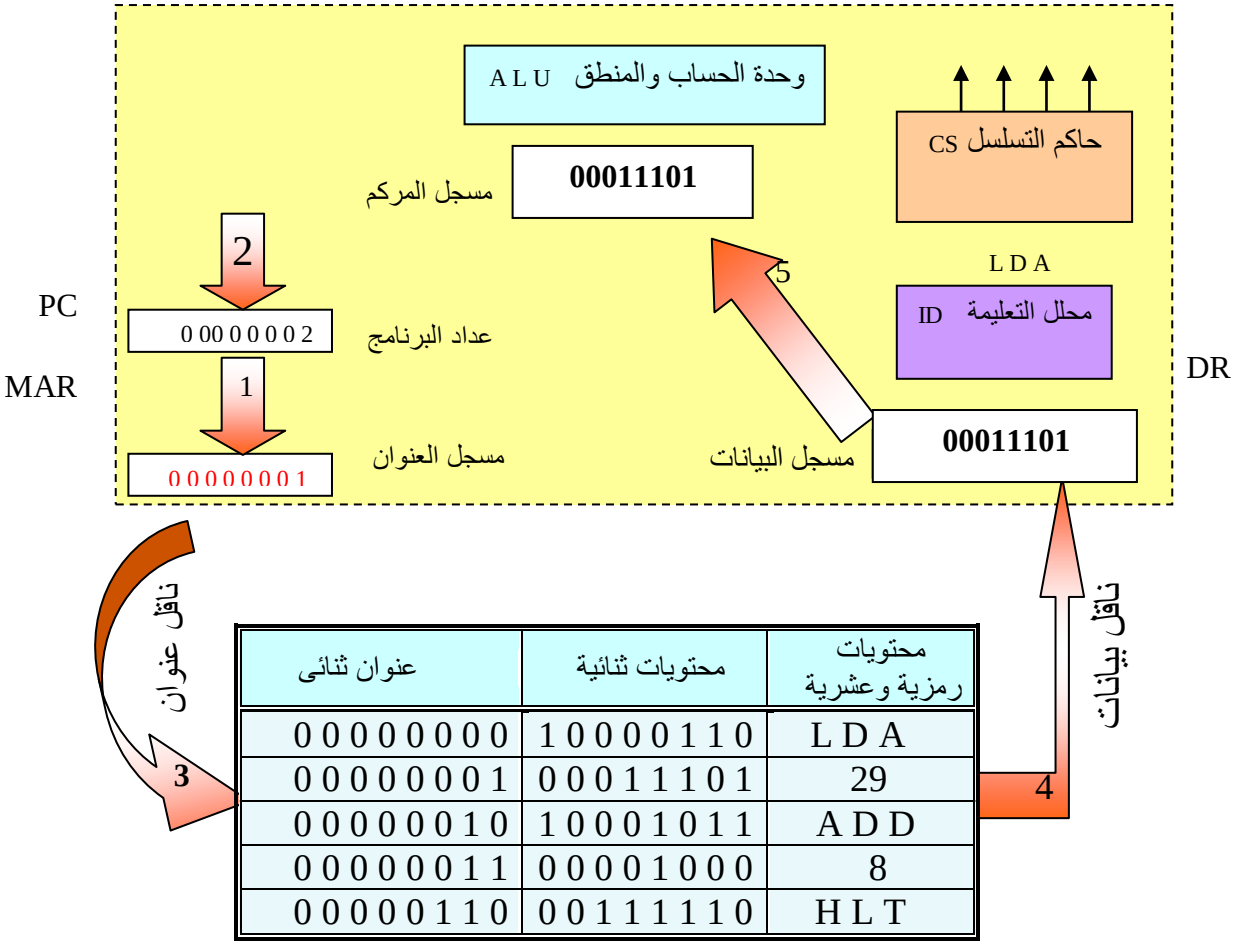
2- يزيد العداد فى العد بمقدار (1)

3- يضع مسجل العنوان العنوان $2_{(00000000)}$ على ناقل العنوان ويحدد موضعة الصحيح فى الذاكرة .

4- تنتقل المحتويات الثنائية للموضع المحدد إلى مسجل البيانات.

5- تنتقل شفرة العملية من مسجل البيانات إلى محلل التعليمة فيتم تحليلها وتفسيرها ثم يخبر حاكم التسلسل لإنتاج النبضات اللازمة لتنفيذ العملية التى تنص عليها هذه التعليمة .

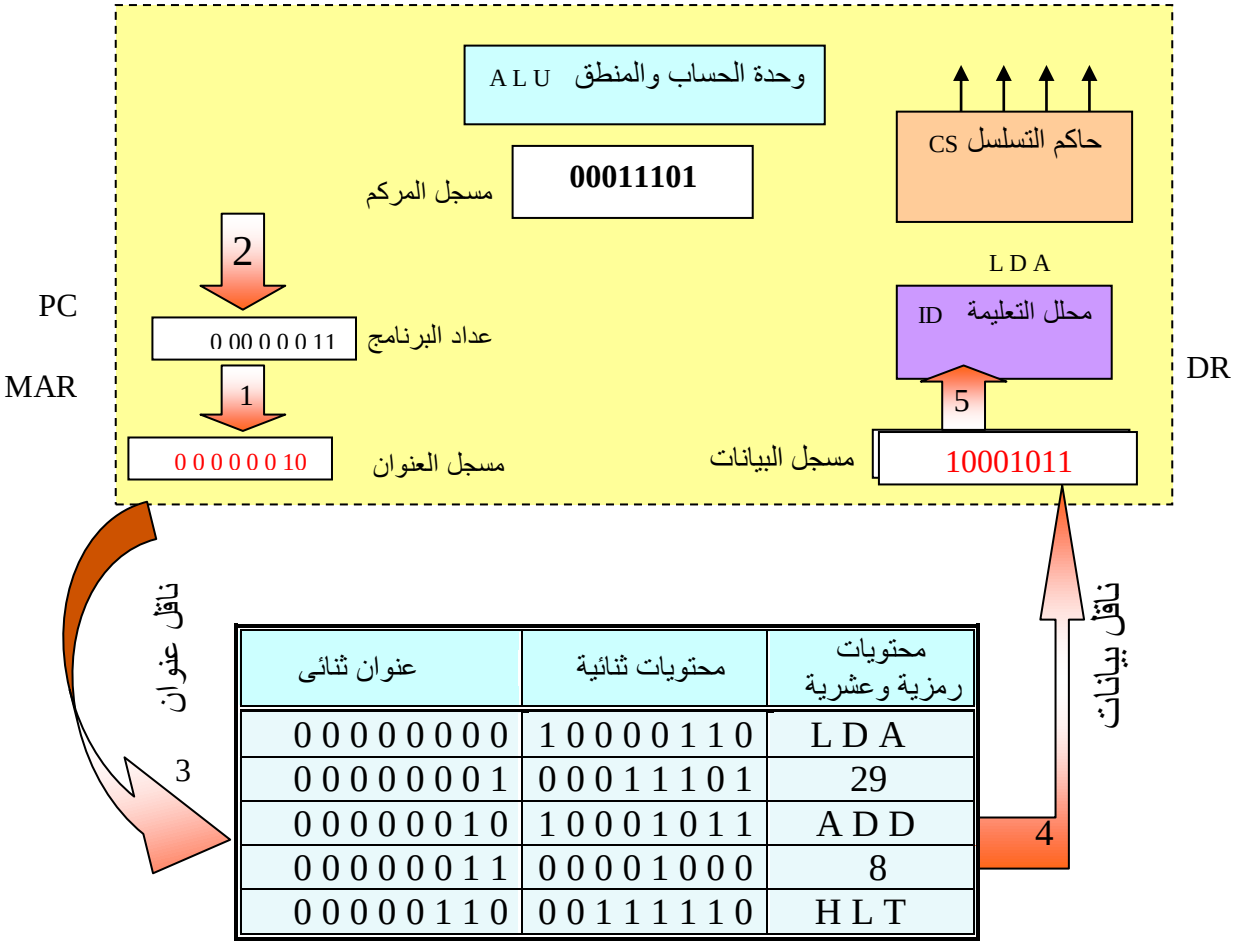
ثانيا :- تنفيذ تعليمة حمل المركم L D A 29



عنوان ثنائى	محتويات ثنائية	محتويات رمزية وعشرية
0 0 0 0 0 0 0 0	1 0 0 0 0 1 1 0	L D A
0 0 0 0 0 0 0 1	0 0 0 1 1 1 0 1	29
0 0 0 0 0 0 1 0	1 0 0 0 1 0 1 1	A D D
0 0 0 0 0 0 1 1	0 0 0 0 1 0 0 0	8
0 0 0 0 0 1 1 0	0 0 1 1 1 1 1 0	H L T

1- يضع عداد البرنامج عنوان التعليمة المراد تنفيذها $_2(00000001)$ فى مسجل العنوان .
2- يزيد العداد فى العد بمقدار (1) .
3- يضع مسجل العنوان $_2(00000001)$ على ناقل العناوين ويحدد موضعه الصحيح بالذاكرة .
4- تنتقل المحتويات الثنائية للموضع المحدد (المعامل) إلى مسجل البيانات.
5- ينتقل المعامل الأول $_2(00011101)$ من مسجل البيانات إلى المركم .

ثالثاً :- إحضار تعليمة ADD



عنوان ثنائى	محتويات ثنائية	محتويات رمزية وعشرية
00000000	10000110	LDA
00000001	00011101	29
00000010	10001011	ADD
00000011	00001000	8
00000110	00111110	HLT

خطوات إحضار التعليمة الثانية بالعنونة الفورية

1- يضع عداد البرنامج عنوان التعليمة المراد تنفيذها $_2$ (00000010) فى مسجل العنوان .

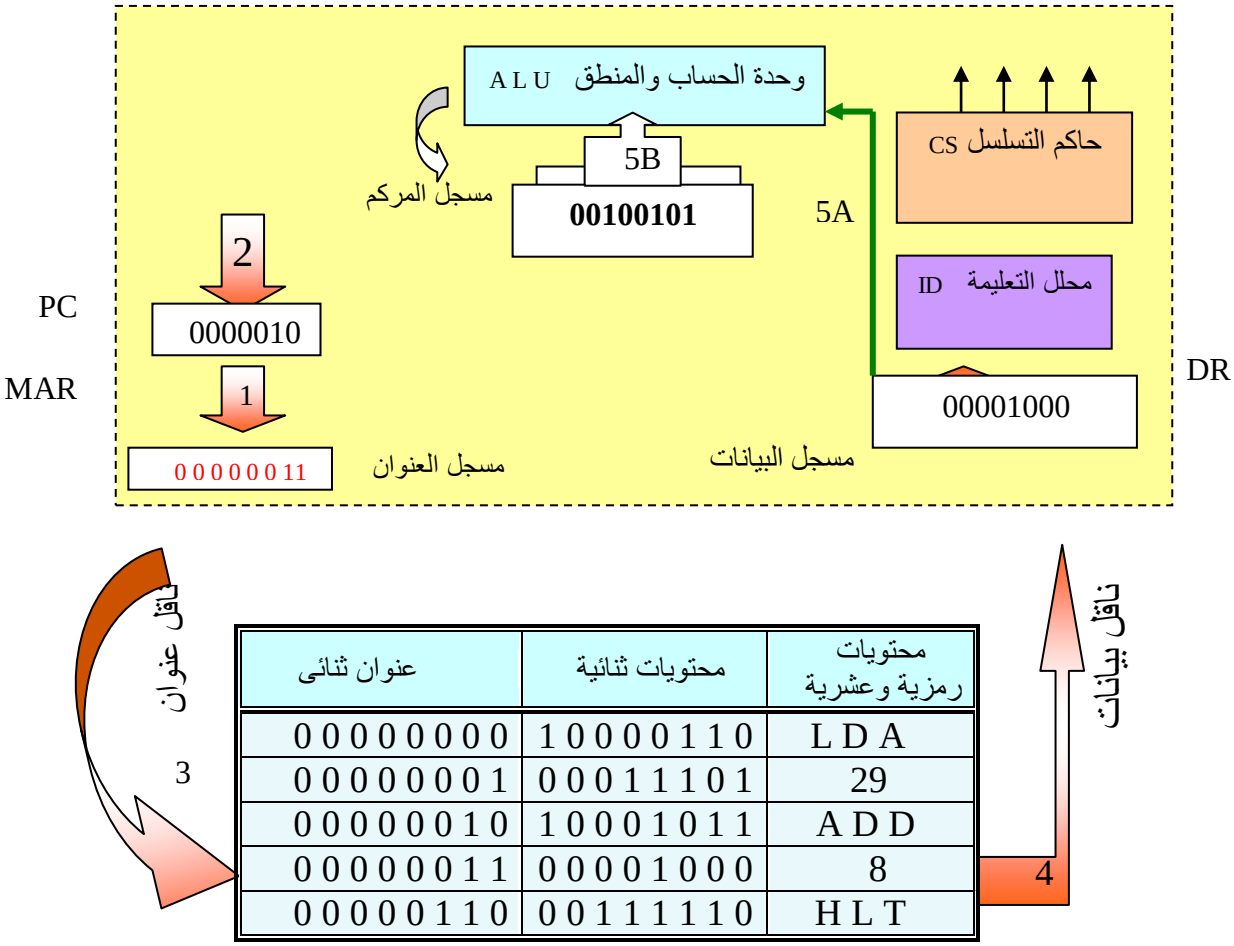
2- يزيد العداد فى العد بمقدار (1)

3- يضع مسجل العنوان العنوان $_2$ (00000010) على ناقل العنوان ويحدد موضعة الصحيح فى الذاكرة .

4- تنتقل المحتويات الثنائية للموضع المحدد إلى مسجل البيانات.

5- تنتقل شفرة العملية من مسجل البيانات إلى محلل التعليمة فيتم تحليلها وتفسيرها ثم يخبر حاكم التسلسل لإنتاج النبضات اللازمة لتنفيذ العملية التى تنص عليها هذه التعليمة .

رابعا :- تنفيذ تعليمة ADD



عنوان ثنائى	محتويات ثنائية	محتويات رمزية وعشرية
00000000	10000110	LDA
00000001	00011101	29
00000010	10001011	ADD
00000011	00001000	8
00000110	00111110	HLT

الإضافة الأولى ADD بالعنونة الفورية

1- يضع عداد البرنامج عنوان التعليمة المراد تنفيذها 2 (00000011) فى مسجل العنوان .

2- يزيد العداد فى العد بمقدار (1)

3- يضع مسجل العنوان العنوان 2 (00000011) على ناقل العنوان ويحدد موضعه الصحيح فى الذاكرة .

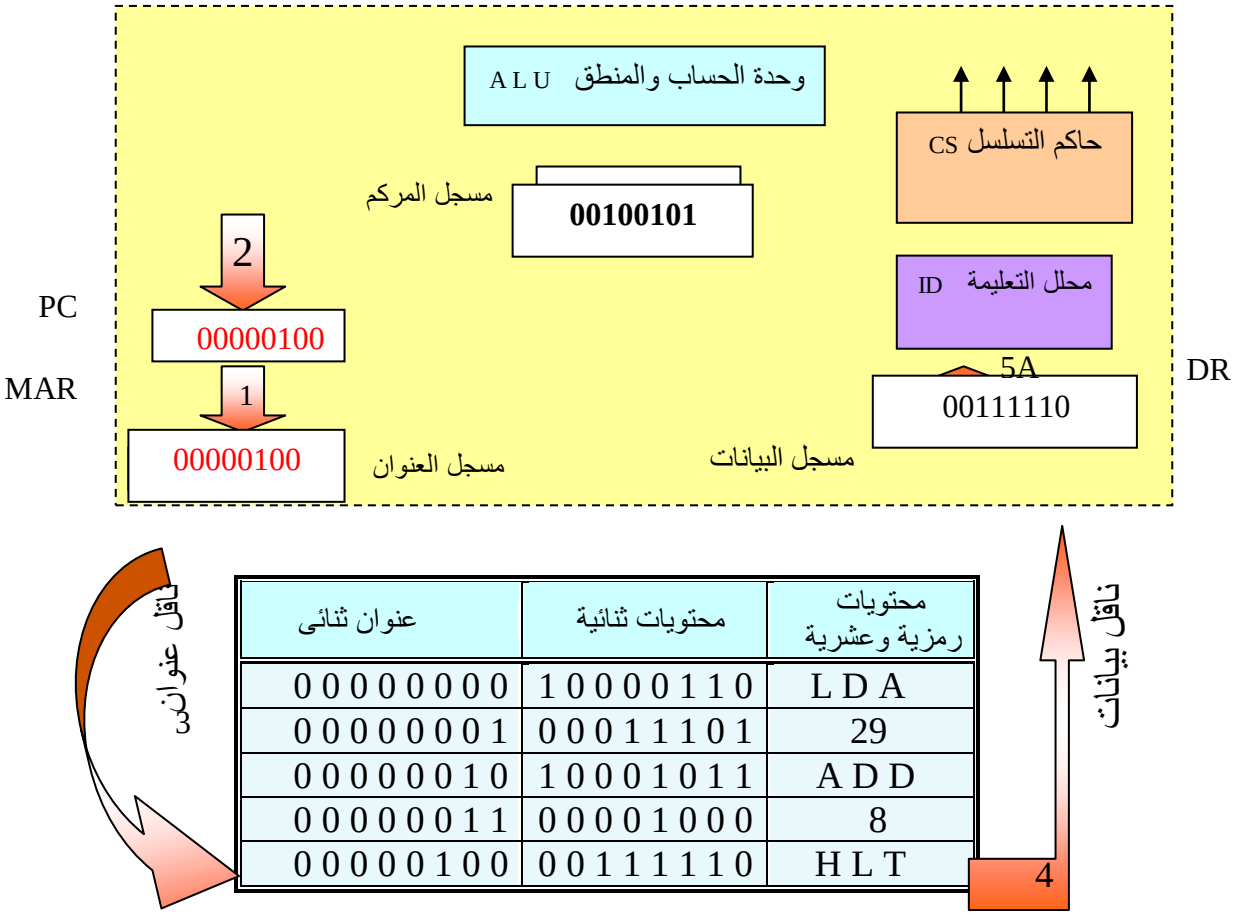
4- تنتقل المحتويات الثنائية للموضع المحدد إلى مسجل البيانات عن طريق ناقل البيانات .

5A - ينتقل المعامل الثانى من مسجل البيانات الى المدخل الأول الى وحدة الحساب والمنطق .

5B - فى نفس الوقت تنتقل محتويات المركم الى وحدة الحساب والمنطق .

5 – تقوم وحدة الحساب والمنطق بجمع المعاملين (29) + (8)
وتخزين المجموع فى المركم

خامساً: إحضار و تنفيذ تعليمة التوقف (HLT)



عنوان ثنائى	محتويات ثنائية	محتويات رمزية وعشرية
0 0 0 0 0 0 0 0	1 0 0 0 0 1 1 0	L D A
0 0 0 0 0 0 0 1	0 0 0 1 1 1 0 1	29
0 0 0 0 0 0 1 0	1 0 0 0 1 0 1 1	A D D
0 0 0 0 0 0 1 1	0 0 0 0 1 0 0 0	8
0 0 0 0 0 1 0 0	0 0 1 1 1 1 1 0	H L T

آخر تعليمة فى البرنامج هى التعليمة توقف (HLT) التى يتم احضارها من الذاكرة باستعمال نفس الاجراءات السابقة .

العنوان ينقل من عداد البرنامج خلال مسجل العنوان الى ناقل العنوان حيث يتم قراءة موضع الذاكرة $(00000100)_2$ و تحميل شفرة العملية لتعليمة توقف (HLT) فى مسجل البيانات . وبعد ذلك يتم شفرة العملية وتنفذ التعليمة بتوقف حاكم التسلسل عن إصدار نبضات ومن ثم تتوقف جميع عمليات الحاسب

وبذلك يكون البرنامج قد حقق الهدف منه بجمع $(37)_{10}$

$(8)_{10} + (29)_{10} =$

فى المركم

أسئلة الباب الثالث

1- ماهي النواقل المستعمله من طرف المعالج ؟ وضح عن طريق أسهم اتجاه المعلومات علي هذه النواقل

2 - اذكر وظيفة كل مما ياتي :
وحده الحساب والمنطق – الذاكرة – وحده الدخل – وحده الخرج – ناقل البيانات – ناقل العنوان – ناقل التحكم

3- عرف العنونة وما هى انواعها ؟

4- كيف يتم الحصول على التعليمة فى حالة العنونة النسبية ؟

5– كيف يتم الحصول على المعامل فى العنونة المباشرة مع الرسم ؟

6– عرف العنونة المفهرسة (المؤشرة) وما هى مميزاتها وعيوبها ؟

7- اذكر أنواع تعليمات القفز .

8- إختار الإجابة الصحيحة من بين الأقواس:
(ا) المرحلة التى يتم من خلالها إحضار التعليمة من الذاكرة
(تنفيذ – نقل – إحضار)
(ب) فى كل أنواع العنونة ماذا يخص البايت الأول من التعليمة
(عنوان المعامل – شفرة العملية – المعامل)
(ج) البايت التالي من التعليمة فى حالة العنونة الفورية
(شفرة العملية – المعامل – المركم)
(د) يخص البايت التالى من التعليمة فى حالة العنونة المباشرة
(مسجل العنوان – المعامل – عنوان المعامل)
(هـ) طريقة العنونة التى يستخدم فيها تعليمة أحادية البايت
(العنونة الفورية – العنونة الذاتية – العنونة الصفحية)
(و) طريقة العنونة التى تحتاج الى فترة تنفيذ اقل
(العنونة الفورية – العنونة المباشرة – الغير مباشرة)

9- "يراد تنفيذ برنامج بسيط لجمع الرقم 9 والعدد 13 بواسطة الحاسب الدقيق "
(ا) اذكر بإيجاز الإجراءات اللازمة لتنفيذ البرنامج
(ب) ارسم شكلا توضيحيا يبين خطوات إحضار التعليمة الأولى فى البرنامج السابق .