

## دورة تنفيذ التعليمات **Instruction Execution Cycle**

The execution of a single machine instruction can be divided into a sequence of individual operations called the instruction execution cycle. Before executing, a program is loaded into memory. The instruction pointer contains the address of the next instruction. The instruction queue holds a group of instructions about to be executed. Executing a machine instruction requires three basic steps: fetch, decode, and execute. Two more steps are required when the instruction uses a memory operand: fetch operand and store output operand.

يمكن تقسيم تنفيذ تعليمات آلة واحدة إلى سلسلة من العمليات الفردية تسمى دورة تنفيذ التعليمات. قبل التنفيذ ، يتم تحميل البرنامج في الذاكرة. يحتوي مؤشر التعليمات على عنوان التعليمات التالية. تحتوي قائمة انتظار التعليمات على مجموعة من التعليمات على وشك التنفيذ. يتطلب تنفيذ تعليمات الآلة ثلاث خطوات أساسية: الجلب وفك التشفير والتنفيذ. مطلوب خطوتان أخريان عندما تستخدم التعليمات معاملاً ذاكرة: إحضار المعامل وتخزين معامل الإخراج.

:: يتم وصف كل خطوة على النحو التالي Each of the steps is described as follows

- **Fetch:** The control unit fetches the next instruction from the instruction queue and increments the instruction pointer (IP). The IP is also known as the program counter .
- الجلب: تقوم وحدة التحكم بإحضار التعليمات التالية من قائمة انتظار التعليمات وتزيد من مؤشر التعليمات (IP). يُعرف IP أيضاً باسم عداد البرنامج.
- **Decode:** The control unit decodes the instruction's function to determine what the instruction will do. The instruction's input operands are passed to the ALU, and signals are sent to the ALU indicating the operation to be performed .
- فك الشفرة: تقوم وحدة التحكم بفك تشفير وظيفة التعليمات لتحديد ما ستفعله التعليمات. يتم تمرير معاملات إدخال التعليمات إلى ALU ، ويتم إرسال الإشارات إلى ALU للإشارة إلى العملية التي سيتم تنفيذها.

- **Fetch operands:** If the instruction uses an input operand located in memory, the control unit uses a read operation to retrieve the operand and copy it into internal registers. Internal registers are not visible to user programs.  
• معاملات الجلب: إذا كانت التعليمات تستخدم معامل إدخال موجوداً في الذاكرة ، فإن وحدة التحكم تستخدم عملية قراءة لاسترداد المعامل ونسخه في السجلات الداخلية. السجلات الداخلية غير مرئية لبرامج المستخدم.
- **Execute:** The ALU executes the instruction using the named registers and internal registers as operands and sends the output to named registers and/or memory. The ALU updates status flags providing information about the processor state.  
• تنفيذ: ينفذ ALU التعليمات باستخدام المسجلات المسماة والسجلات الداخلية كمعاملين ويرسل الإخراج إلى السجلات و / أو الذاكرة المسماة. تقوم ALU بتحديث أعلام الحالة التي توفر معلومات حول حالة المعالج.
- **Store output operand:** If the output operand is in memory, the control unit uses a write operation to store the data  
• مخزن معامل الإخراج: إذا كان معامل الإخراج في الذاكرة ، فإن وحدة التحكم تستخدم عملية كتابة لتخزين البيانات..

The sequence of steps can be expressed neatly in pseudocode:

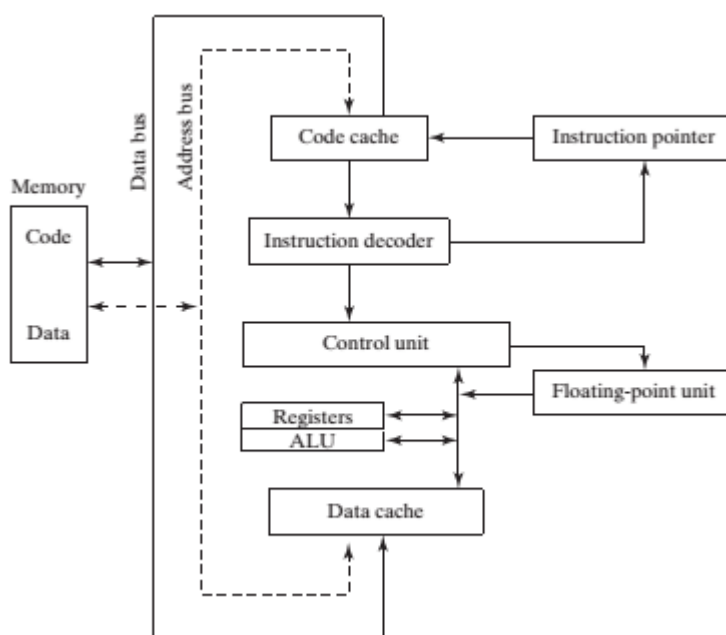
```
loop
  fetch next instruction
  advance the instruction pointer (IP)
  decode the instruction
  if memory operand needed, read value from memory
  execute the instruction
  if result is memory operand, write result to memory
  continue loop
```

يمكن التعبير عن تسلسل الخطوات بدقة في الكود الكاذب:

حلقة  
إحضار التعليمات التالية  
تقدم مؤشر التعليمات (*IP*)  
فك التعليمات  
إذا كانت هناك حاجة لمعامل الذاكرة ، اقرأ القيمة من الذاكرة  
تنفيذ التعليمات  
إذا كانت النتيجة هي معامل الذاكرة ، فاكتب النتيجة إلى الذاكرة  
تواصل الحلقة

A block diagram showing data flow within a typical CPU is shown in Fig 4-1.

يظهر مخطط كتلة يوضح تدفق البيانات داخل وحدة المعالجة المركزية النموذجية في الشكل 4-1.



The diagram helps to show relationships between components that interact during the instruction execution cycle. In order to read program instructions from memory, an address is placed on the address bus. Next, the memory controller places the requested code on the data bus, making the code available inside the code cache. The instruction pointer's value determines which instruction will be executed next. The instruction is analyzed by the instruction decoder, causing the appropriate digital signals to be sent to the control unit, which coordinates the ALU and floating-point unit.

يساعد الرسم التخطيطي في إظهار العلاقات بين المكونات التي تتفاعل أثناء دورة تنفيذ التعليمات. لقراءة تعليمات البرنامج من الذاكرة ، يتم وضع عنوان على ناقل العنوان. بعد ذلك ، تضع وحدة التحكم في الذاكرة

الكود المطلوب على ناقل البيانات ، مما يجعل الكود متاحًا داخل ذاكرة التخزين المؤقت للرمز. تحدد قيمة مؤشر التعليمات التي سيتم تنفيذها بعد ذلك. يتم تحليل التعليمات بواسطة وحدة فك ترميز التعليمات ، مما يؤدي إلى إرسال الإشارات الرقمية المناسبة إلى وحدة التحكم ، والتي تنسق وحدة ALU ووحدة النقطة العائمة.

Although the control bus is not shown in this figure, it carries signals that use the system clock to coordinate the transfer of data between the different CPU components.

على الرغم من عدم ظهور ناقل التحكم في هذا الشكل ، إلا أنه يحمل إشارات تستخدم ساعة النظام لتنسيق نقل البيانات بين مكونات وحدة المعالجة المركزية المختلفة.

### تعليمات وضع العنوان **Instructions Address mode**

Any instruction issued by the processor must carry at least two types of information. These are the operation to be performed, encoded in what is called the **op-code field**, and the address information of the operand on which the operation is to be performed, encoded in what is called the **address field**.

Instructions can be classified based **on the number of operands** as:

يجب أن تتضمن أي تعليمات صادرة عن المعالج نوعين على الأقل من المعلومات. هذه هي العملية المراد تنفيذها ، المشفرة فيما يسمى حقل كود المرجع ، ومعلومات عنوان المعامل الذي سيتم تنفيذ العملية عليه ، مشفرة فيما يسمى حقل العنوان. يمكن تصنيف التعليمات بناءً على عدد المعاملات على النحو التالي:

- Three - address instruction: takes the form operation addr-1, addr-2 ,addr-3, each of addr-1, addr-2, and addr-3 refers to a register or to a memory location.  
Add R1,R2,R3 It also indicates that the values to be added are those stored in registers R1 and R2 that the results should be stored in register R3.
- ثلاثة - تعليمات العنوان: تأخذ شكل العملية addr-1 ، addr-2 ، addr-3 ، كل من addr-1 ، addr-2 ، و addr-3 يشير إلى سجل أو إلى موقع ذاكرة. أضف R1 و R2 و R3 كما يشير إلى أن

القيم المراد إضافتها هي تلك المخزنة في السجلات R1 و R2 التي يجب تخزين النتائج في السجل R3.

- Two - address instruction: takes the form operation add-1, add-2 . In this form, each of add-1 and add-2 refers to a register or to a memory location , ADD R1,R2, adds the contents of register R1 to the contents of register R2 and stores the results in register R2. The original contents of register R2 are lost due to this operation while the original contents of register R1 remain intact .

• اثنان - تعليمات العنوان: تأخذ شكل العملية add-1, add-2. في هذا النموذج ، تشير كل من add-1 و add-2 إلى سجل أو إلى موقع ذاكرة ، ADD R1 ، R2 ، وتضيف محتويات السجل R1 إلى محتويات السجل R2 وتخزن النتائج في السجل R2. تم فقد المحتويات الأصلية للسجل R2 بسبب هذه العملية بينما تظل المحتويات الأصلية للسجل R1 كما هي.

- One - address instruction: takes the form operation add1. Add1 refers to a register or to a memory location, ADD R1 In this case the instruction implicitly refers to a register, called the Accumulator Racc, such that the contents of the accumulator is added to the contents of the register R1 and the results are stored back into the accumulator Racc.
- تعليمات عنوان واحد: تأخذ شكل العملية add1. يشير Add1 إلى سجل أو إلى موقع ذاكرة ، ADD R1 في هذه الحالة تشير التعليمات ضمناً إلى سجل يسمى Accumulator Racc ، بحيث تتم إضافة محتويات المجمع إلى محتويات السجل R1 ويتم تخزين النتائج العودة إلى المجمع Racc.

- One and half - address instruction: take the form operation add1, add2, one refers to a register and other refers to memory and vice versa. The instruction ADD B, R1. In this case, the instruction adds the contents of register R1 to the contents of memory location B and stores the result in register R1.
- تعليمات عنوان واحد ونصف: خذ عملية الشكل add1, add2 ، يشير أحدهما إلى سجل والآخر يشير إلى الذاكرة والعكس صحيح. التعليمات ADD B, R1. في هذه الحالة ، تضيف التعليمات محتويات السجل R1 إلى محتويات موقع الذاكرة B وتخزن النتيجة في السجل R1.

- Zero - address instruction: instructions that use stack operation.
- تعليمات العنوان الصفري: التعليمات التي تستخدم عملية المكس.

The different ways in which operands can be addressed are called the addressing modes. Addressing modes (described in table 1) differ **in the way the address information of operands** is specified which include:

تسمى الطرق المختلفة التي يمكن من خلالها معالجة المعاملات بأوضاع العنوان. تختلف أوضاع المعالجة (الموضحة في الجدول 1) في الطريقة التي يتم بها تحديد معلومات عنوان المعاملات والتي تشمل:

- immediate mode: the value of the operand is (immediately) available in the instruction itself, it is customary to prefix the operand by the special character (#).  
The use of immediate addressing leads to poor programming practice because a change in the value of an operand requires a change in every instruction that uses the immediate value of such an operand .

• الوضع الفوري: قيمة المعامل متاحة (على الفور) في التعليمات نفسها ، ومن المعتاد أن تسبق المعامل بالحرف الخاص (#). يؤدي استخدام العنوان الفوري إلى ممارسة البرمجة السيئة لأن التغيير في قيمة المعامل يتطلب تغييراً في كل تعليمات تستخدم القيمة الفورية لمثل هذا المعامل.

- Direct mode: the operand of instruction represent address of the memory location that holds the operand.

• الوضع المباشر: يمثل معاملي التعليمات عنوان موقع الذاكرة الذي يحمل المعامل.

- Indirect mode: The operand of instruction ( a register or a memory location) holds the (effective) address of the operand . It is customary to include the name of the register or the memory location in parentheses.

• الوضع غير المباشر: يحمل معاملي التعليمات (سجل أو موقع ذاكرة) العنوان (الفعال) للمعامل. من المعتاد تضمين اسم السجل أو موقع الذاكرة بين قوسين.

- Index mode: the address of the operand is obtained by adding a constant to the content of a register, called the index register.

• وضع الفهرس: يتم الحصول على عنوان المعامل بإضافة ثابت إلى محتوى السجل ، يسمى سجل الفهرس.

- Relative mode: is the same as indexed addressing except that the program counter (PC) replaces the index register.

• الوضع النسبي: هو نفس العنوان المفهرسة فيما عدا أن عداد البرنامج (PC) يحل محل سجل الفهرس.

- Autoincrement mode: uses a register to hold the address of the operand. The content of the autoincrement register is first is used as the effective address of the operand and then increment the content of the autoincrement register.

- Autodecrement mode: uses a register to hold the address of the operand. The content of the autodecrement register is first decremented and the new content is used as the effective address of the operand.
- وضع Autoincrement: يستخدم السجل للاحتفاظ بعنوان المعامل. يتم استخدام محتوى سجل التزايد التلقائي أولاً كعنوان فعال للمعامل ثم زيادة محتوى سجل التزايد التلقائي.
- وضع Autodecrement: يستخدم السجل للاحتفاظ بعنوان المعامل. يتم أولاً إنقاص محتوى سجل الاستدلال التلقائي ويتم استخدام المحتوى الجديد كعنوان فعال للمعامل.

Table 1: summary of address mode

Addressing mode	Definition	Example	Operation
Immediate	Value of operand is included in the instruction	<i>load #1000, R<sub>i</sub></i>	$R_i \leftarrow 1000$
Direct (Absolute)	Address of operand is included in the instruction	<i>load 1000, R<sub>i</sub></i>	$R_i \leftarrow M[1000]$
Register indirect	Operand is in a memory location whose address is in the register specified in the instruction	<i>load (R<sub>j</sub>), R<sub>i</sub></i>	$R_i \leftarrow M[R_j]$
Memory indirect	Operand is in a memory location whose address is in the memory location specified in the instruction	<i>load (1000), R<sub>i</sub></i>	$R_i \leftarrow M[1000]$
Indexed	Address of operand is the sum of an index value and the contents of an index register	<i>load X(R<sub>ind</sub>), R<sub>i</sub></i>	$R_i \leftarrow M[R_{ind} + X]$
Relative	Address of operand is the sum of an index value and the contents of the program counter	<i>load X(PC), R<sub>i</sub></i>	$R_i \leftarrow M[PC + X]$
Autoincrement	Address of operand is in a register whose value is incremented after fetching the operand	<i>load (R<sub>auto</sub>)+, R<sub>i</sub></i>	$R_i \leftarrow M[R_{auto}]$ $R_{auto} \leftarrow R_{auto} + 1$
Autodecrement	Address of operand is in a register whose value is decremented before fetching the operand	<i>load -(R<sub>auto</sub>), R<sub>i</sub></i>	$R_{auto} \leftarrow R_{auto} - 1$ $R_i \leftarrow M[R_{auto}]$