بسم الله الرحمن الرحيم

Computer Architecture

- 2 -

الوحدات الوظيفية للحاسوب و ربطها

Computer Function and Interconnection

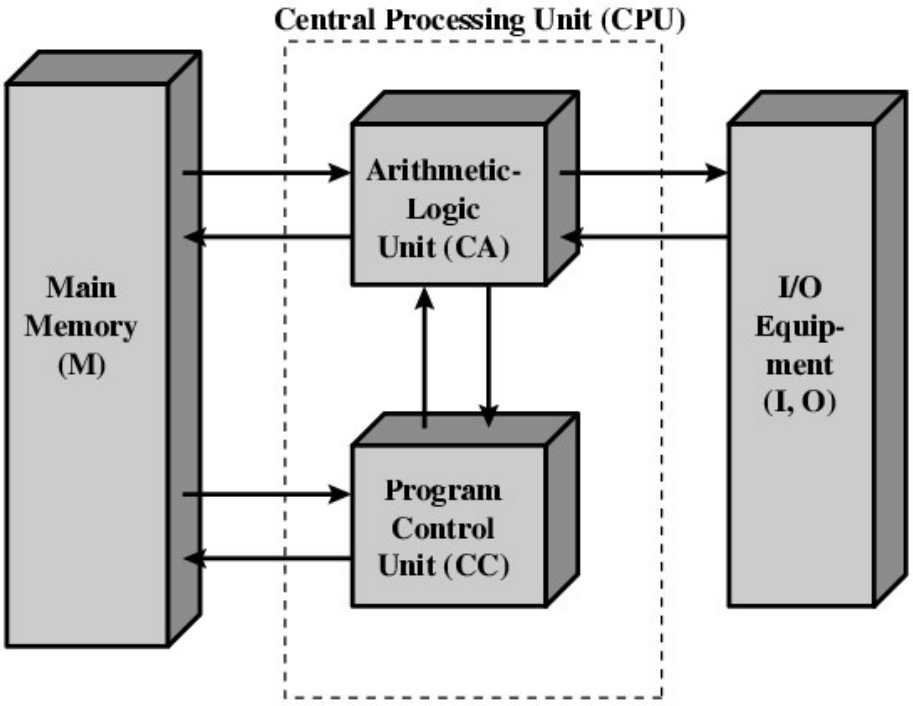
تمهيد

كما نعلم فإن جهاز الحاسوب يتكون من عدد من الوحدات الوظيفية (Functional Units): وحدة معالجة مركزية (Central Processing Unit (CPU))، ذاكرة (Memory)، وحدات إدخال و إخراج (I/O Units). و ترتبط هذه الوحدات مع بعضها البعض بطريقة معينة لأداء الوظيفة الأساسية لجهاز الحاسوب، و هي تنفيذ البرامج. لفهم جهاز الحاسوب يجب أن نبدأ أولاً بوصف وحداته الأساسية، و ذلك بتحديد وظيفة كل وحدة منها على حدة و نوع البيانات و إشارات التحكم التي تتبادلها مع الوحدات الأخرى، ثم توضيح طريقة ربط الوحدات مع بعضها البعض.

الوحدات الوظيفية للحاسوب

جميع الحواسيب المعاصرة تقريباً تعتمد ما يسمى بالـ von Neumann Architecture الذي طوره John von Neumann عام 1952 و أطلق على ذلك الحاسوب تسمية الـ IAS.،

بدأ von Neumann مع مجموعة من زملائه بتصميم أول حاسوب يستخدم أسلوب البرنامج المخزون عام 1946، في معهد برنستون للدراسات المتقدمة (the Princeton Institute for Advanced Studies) و أطلق على ذلك الحاسوب تسمية الـ IAS. اكتمل العمل في حاسوب الـ IAS عام 1952، ليصبح هو النموذج الأساسي لجميع الحواسيب التي تلته، حتى اليوم. الشكل التالي يوضح البناء العام لحاسوب الـ IAS، الذي يطلق عليه أيضاً معمار فون نويمان (von Neumann Architecture).



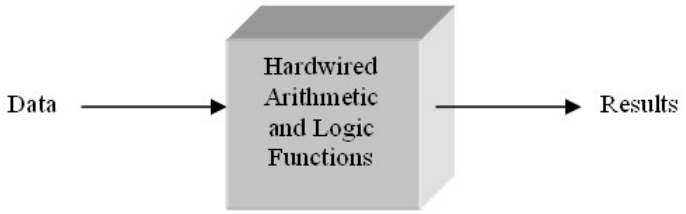
و يقوم الـ von Neumann Architecture على ثلاثة مفاهيم أساسية:

1. البيانات (data) و التعليمات (instructions) مخزنة معاً في ذاكرة قراءة/كتابة (Read/Write Memory).
2. محتويات الذاكرة يتم الوصول إليها عن طريق عناوين المواقع دون إعتبار لنوع البيانات المخزنة.
3. يتم تنفيذ البرنامج بصورة تتابعية (Sequential)، وذلك بالإنتقال من كل تعليمة إلى التعليمة التالية لها.

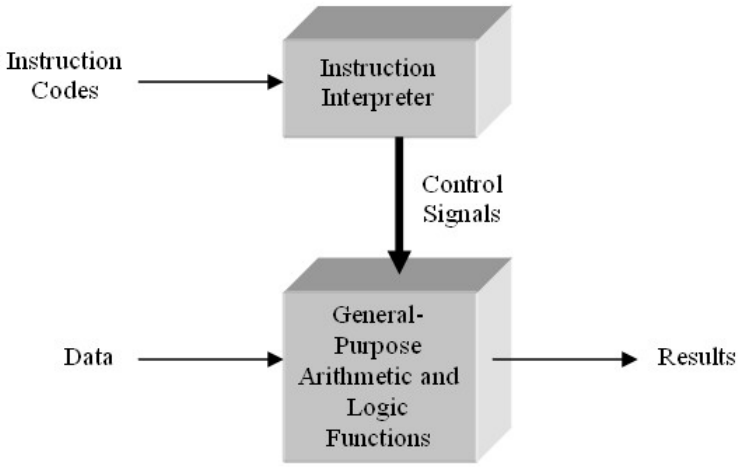
يتكون حاسوب الـ IAS من:

- ذاكرة رئيسية (Main Memory) تختزن كل من البيانات و التعليمات.
- وحدة حساب و منطق ((Arithmetic and Logic Unit (ALU) قادرة على إجراء العمليات على البيانات الممثلة في الصورة الثنائية (Binary).
- وحدة تحكم (Control Unit) تقوم بتفسير تعليمات البرنامج و تصدر إشارات التحكم اللازمة لتنفيذها.
- معدات إدخال و إخراج (Input and Output (I/O) Equipment) تقوم بتشغيلها وحدة التحكم.

للقيام بأي عملية من عمليات المعالجة (Processing) يمكننا تصميم Hardware خاص للقيام بتلك العملية. و لكن هذا الـ Hardware المتخصص لن يستطيع القيام بأي عملية معالجة أخرى خلاف تلك التي تم تصميمه من أجلها. و تعديل العملية التي يقوم بها يتطلب تعديل الـ Hardware نفسه عن طريق تغيير طريقة توصيل المكونات باستخدام مفاتيح (Switches) أو عن طريق فك و تركيب الأسلاك (Plugging and Unplugging Wires)، و هي عملية صعبة و تتطلب معرفة عميقة بالـ Hardware. و يسمى هذا الأسلوب بـ Hardwired Programming. الطريقة الأخرى هي تصميم Hardware متعدد المهام (Multifunction) يستطيع القيام بالعديد من عمليات المعالجة المختلفة، و له أطراف تحكم تستخدم في اختيار العملية المطلوبة، فلتغيير عملية المعالجة يتم هنا فقط تغيير إشارات التحكم. و هذا الأسلوب يتطلب أيضاً معرفة بالـ Hardware للتمكن من تغيير إشارات التحكم. و لكن يمكن تخطي عقبة المعرفة بتفاصيل الـ Hardware عن طريق تصميم وحدة تحكم (Control Unit) تقوم هي بتوليد إشارات التحكم المطلوبة للقيام بأي عملية من عمليات المعالجة، و يتم تحديد العملية المطلوبة بإدخال شفرة (Code) ترمز لتلك العملية. و هنا مطلوب من المستخدم فقط إدخال شفرة العملية و هذا لا يتطلب معرفة عميقة بتفاصيل الـ Hardware. و عادة ما يتم تزويد الـ Hardware بسلسلة من التعليمات تسمى برنامج (Program).



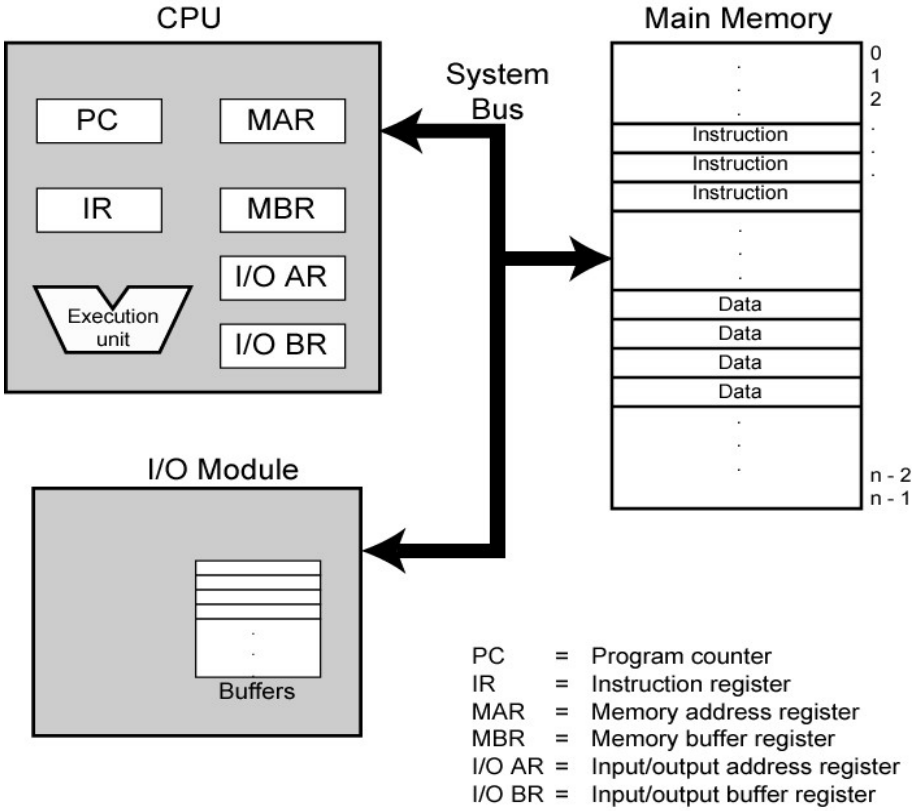
(a) Programming in Hardware



(b) Programming in Software

لاستكمال الـ Hardware الموضح أعلاه و تكوين حاسوب يجب إضافة المزيد من الأجزاء الضرورية. فمثلاً نحتاج لذاكرة (Memory) تستخدم في تخزين البرنامج (Program) و البيانات (Data)، و أحياناً نتائج المعالجة (Results). كما نحتاج لوحدة إدخال (Input Unit) تستخدم لإدخال البرنامج و البيانات إلى الذاكرة، و لوحدة إخراج (Output Unit) لعرض أو طباعة نتائج المعالجة (Results).

الشكل التالي يوضح الوحدات الوظيفية للحاسوب و طريقة ارتباطها مع بعضها البعض عن طريق ناقل النظام (System Bus):

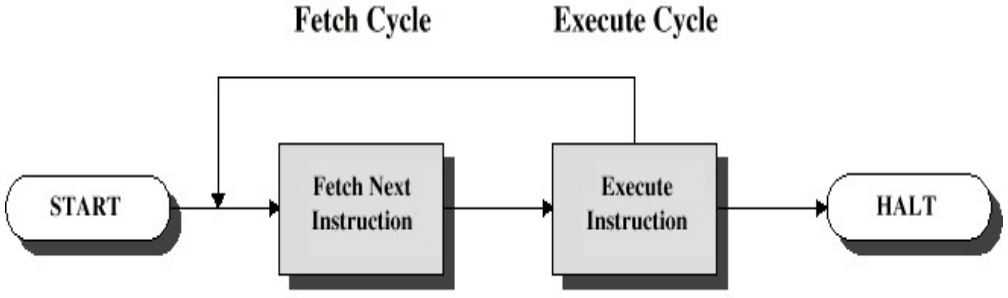


تتبادل وحدة المعالجة المركزية (CPU) البيانات مع الذاكرة عبر المسجلين MAR و MBR. حيث يتم وضع عنوان موقع الذاكرة المطلوب الوصول إليه في المسجل MAR، و يتم وضع البيانات المتبادلة مع الذاكرة في المسجل MBR. و بنفس الأسلوب يتم التعامل مع وحدات الإدخال و الإخراج (I/O Modules) عبر المسجلين I/O AR و I/O BR، حيث يوضع عنوان الوحدة المطلوب الوصول إليها في المسجل I/O AR و البيانات المطلوب تبادلها مع تلك الوحدة في المسجل I/O BR. تكون وحدة الإدخال و الإخراج (I/O Module) مسؤولة عن جهاز أو أكثر من أجهزة الإدخال و الإخراج (I/O Devices)، فتعمل كوسيط ما بين الجهاز و المعالج و تعالج مشكلة الإختلاف الكبير في السرعة ما بين المعالج و أجهزة الإدخال و الإخراج. فعندما يرغب المعالج مثلاً بإرسال بيانات لجهاز إخراج معين (Output Device) فأنه يقوم بإرسالها إلى وحدة الإدخال و الإخراج (I/O Module) المسؤولة عن ذلك الجهاز ليتم اختزان تلك البيانات في المسجلات (Buffers) الموجودة بالوحدة لحين إرسالها إلى الجهاز المعني. لاحظ أن المعالج هنا لا يتعامل مباشرة مع جهاز الإدخال أو الإخراج و لا يحتاج بالتالي لمعرفة طبيعته و خصائصه و طريقة التعامل معه و يترك كل ذلك لوحدة الإدخال و الإخراج المعنية.

طريقة عمل الحاسوب (Computer Function)

المهمة الأساسية لجهاز الحاسوب هي تنفيذ برنامج (Program) ما، يتكون من مجموعة من التعليمات (Instructions) المختزنة في الذاكرة. حيث يقوم الحاسوب بتنفيذ تلك التعليمات حسب ترتيب ورودها بالبرنامج. و تتم معالجة التعليمات على مرحلتين: مرحلة الإلتقاط (Fetch) و مرحلة التنفيذ (Execution). و يطلق على المرحلتين معاً تسمية دورة الإلتقاط/التنفيذ (Fetch/Execute Cycle) أو دورة التعليمة (Instruction Cycle). في مرحلة الإلتقاط يتم قراءة التعليمة من الذاكرة، و يكون عنوان التعليمة التي وصل إليها الدور في التنفيذ موجوداً في مسجل يطلق عليه عدّاد البرنامج (Program Counter (PC))، الذي يتم تحديثه بعد نهاية تنفيذ كل تعليمة بحيث يحتوي على عنوان التعليمة التي تليها. بعد إلتقاط التعليمة توضع في مسجل التعليمة (Instruction Register (IR)) تمهيداً للمرحلة التالية و هي مرحلة التنفيذ. في مرحلة التنفيذ تقوم وحدة التنفيذ (Execution Unit) بفك شفرة التعليمة (Decoding) ثم تنفيذ التعليمة عن طريق توليد تسلسل إشارات التحكم اللازمة لإتمام العملية. بعد إنتهاء الدورة يتم الإنتقال للتعليمة التالية ... و هكذا. و يتم تكرار دورة الإلتقاط/التنفيذ حتى يصادف المعالج تعليمة توقف (Halt) أو يحدث خطأ غير قابل للعلاج (Irrecoverable Error) أثناء المعالجة.

الشكل التالي يوضح دورة التعليمة (Instruction Cycle):



و تنفيذ التعليمة الواحدة عادة ما يتطلب عدة خطوات. مثلاً التعليمة ADD B,A، التي تعني "اجمع محتوى موقع الذاكرة B إلى محتوى موقع الذاكرة A و خزن النتيجة في الموقع A"، يتم تنفيذها كالتالي:

1. إلتقاط التعليمة ADD B,A
2. قراءة محتوى موقع الذاكرة B و وضعه في أحد مسجلات المعالج
3. قراءة محتوى موقع الذاكرة A و وضعه في مسجل آخر من مسجلات المعالج
4. جمع القيمتين و تخزين النتيجة في أحد مسجلات المعالج
5. نقل النتيجة من المعالج و تخزينها في موقع الذاكرة A

لاحظ أن التعليمة هنا هي ADD أما ما يلي التعليمة، أي A و B، فيطلق عليها الـ Operands.
لاحظ في المثال السابق أن تنفيذ التعليمة قد يتطلب أكثر من عملية رجوع للذاكرة، و أحياناً قد يتطلب التنفيذ إجراء عملية I/O.

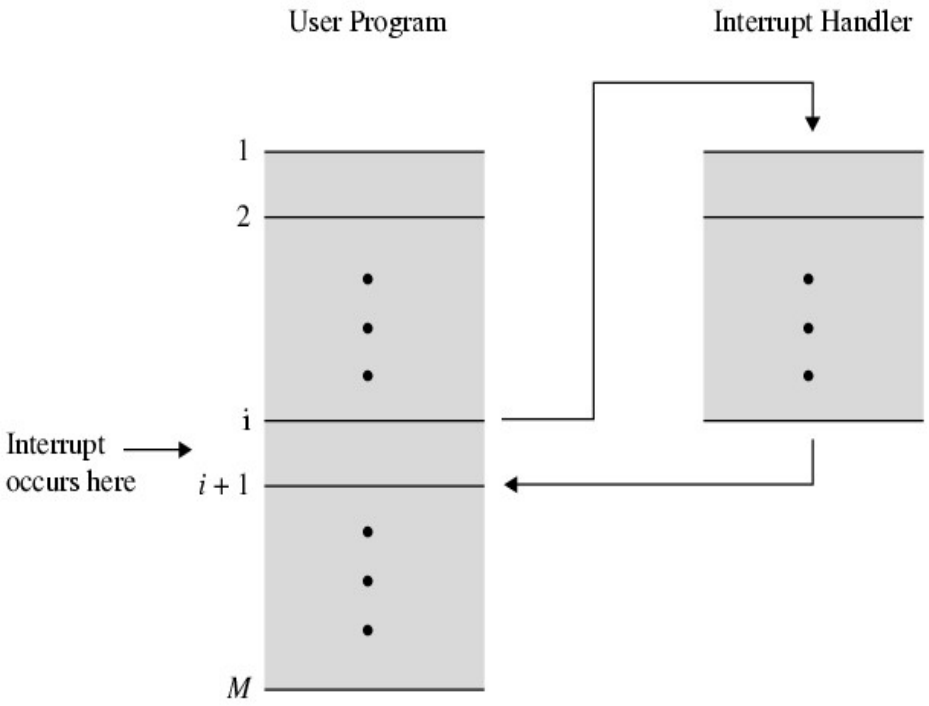
المقاطعات (Interrupts)

عادة ما توفر جميع أجهزة الحاسوب ميكانيكية (Mechanism) تستطيع عن طريقها الوحدات الوظيفية الأخرى (مثل الـ I/O و الذاكرة) مقاطعة المعالج أثناء عملية المعالجة. و الجدول التالي يوضح أنواع المقاطعات (Interrupts) الشائعة:

Program	تحدث نتيجة بعض الظروف التي تطرأ أثناء تنفيذ تعليمات البرنامج، مثل الـ Arithmetic overflow، القسمة على صفر، محاولة تنفيذ تعليمة غير قانونية، أو الرجوع لموقع ذاكرة خارج نطاق العناوين المتاحة للمستخدم.
Timer	يتم توليدها بواسطة مؤقت (Timer) موجود داخل المعالج كإشارة لإنقضاء فترة زمنية محددة، مما يسمح لنظام التشغيل بالقيام بعمليات معينة بعد مرور فترات زمنية محددة.
I/O	يتم توليدها بواسطة وحدات الإدخال/الإخراج (I/O Modules) كإشارة لإكتمال عملية I/O معينة بنجاح أو كإشارة لحدوث خطأ أثناء العملية.
Hardware failure	تنشأ نتيجة حدوث عطب في الـ Hardware كحدوث مشكلة في التزويد بالقدرة (Power supply) أو مشكلة في لوحة المفاتيح أو مروحة تبريد المعالج.

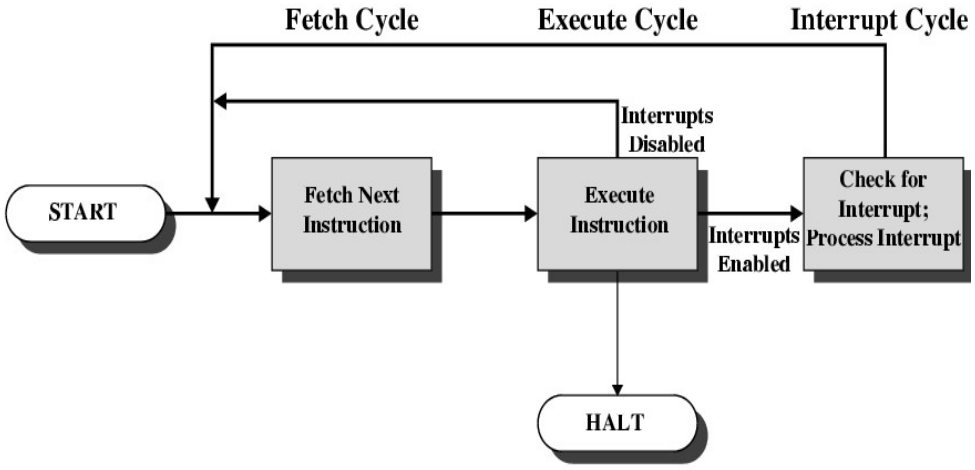
الهدف الأساسي من استخدام المقاطعات هو تحسين كفاءة المعالجة و حل مشكلة الإختلاف الكبير في السرعة ما بين المعالج و معظم الأجهزة الخارجية (External Devices) التي قد يحتاج للتعامل معها. على سبيل المثال، إذا كان المعالج يقوم بإرسال بيانات إلى الطابعة باستخدام دورة التعليمة (Instruction Cycle)، التي قمنا بتوضيحها في ما سبق، فأنه سيضطر للتوقف بعد كل تعليمة كتابة (Write) حتى فراغ الطابعة من تنفيذها. و فترة التوقف هذه قد تعادل عدة مئات و ربما آلاف من دورات التعليمة العادية التي لا تتضمن تعاملاً مع الطابعة، مما يعني ضياعاً كبيراً لزمن المعالج الثمين. حل هذه المشكلة يكون في عدم تعامل المعالج مع الطابعة بصورة مباشرة و ترك هذه المهمة لوحدة إدخال/إخراج (I/O Module). حيث يكتفي المعالج بإصدار تعليمة الكتابة للـ I/O Module ثم يعود للقيام بعمليات المعالجة الأخرى دون توقف أو

انتظار، و يتولى الـ I/O Module متابعة تفاصيل العملية مع الطابعة. و بعد إنتهاء العملية يقوم الـ I/O Module بإرسال إشارة مقاطعة للمعالج تفيد باكتمال عملية الكتابة بنجاح (أو فشلها نتيجة لخطأ ما). عند استلام المعالج لإشارة المقاطعة يقوم بتعليق عمليات المعالجة التي يقوم بها بصورة مؤقتة و ينتقل لتنفيذ مجموعة من التعليمات تسمى Interrupt Handler أو Interrupt Service Routine (ISR) يقوم خلالها بتقديم الخدمة المطلوبة للطابعة، مثل إظهار رسالة للمستخدم تفيد، على سبيل المثال، بنفاذ الورق من الطابعة. بعدها يستأنف المعالج عمليات المعالجة العادية من حيث توقف. كما هو موضح بالشكل التالي:



فخلال الفترة ما بين قيام المعالج بإصدار تعليمة الكتابة و حتى إنتهاء العملية لم يضطر المعالج للإنتظار بل استمر في إجراء عمليات معالجة أخرى لحين حاجة الطابعة لتدخله في نهاية العملية.

الشكل التالي يوضح دورة التعليمة (Instruction Cycle) بعد إضافة دورة مقاطعة (Interrupt Cycle) إليها:



في دورة المقاطعة (Interrupt Cycle) يقوم المعالج بعد الفراغ من تنفيذ أي تعليمة بالتحقق من ورود أي إشارات مقاطعة. في حالة عدم وجود مقاطعات يقوم المعالج بالتقاط التعليمة التالية. أما في حالة وجود مقاطعة فيقوم المعالج بالخطوات التالية:

1. تعليق عمليات المعالجة الحالية و تخزين كل ما يتعلق بها، مثل محتوى الـ Program Counter (PC) (الذي يمثل نقطة الرجوع بعد الفراغ من خدمة نداء المقاطعة) و كل البيانات التي تتم معالجتها.
2. وضع عنوان بداية الـ Interrupt Handler Routine في الـ PC.
3. يتم بعد ذلك تنفيذ تعليمات الـ Interrupt Handler Routine، التي تنتهي دائماً بتعليمة عودة (Return) مهمتها إعادة المعالج إلى استئناف عمليات المعالجة الطبيعية من حيث توقف.

لاحظ أن الـ Interrupt Handler Routine الذي يتم استخدامه يعتمد على مصدر إشارة المقاطعة و نوع الخدمة المطلوبة. و عادة ما تكون هذه الـ Routines جزءاً من نظام التشغيل (Operating system).

الشكل التالي يوضح مخطط الحالات لدورة التعليمة في وجود المقاطعات (Instruction Cycle State Diagram with Interrupts):