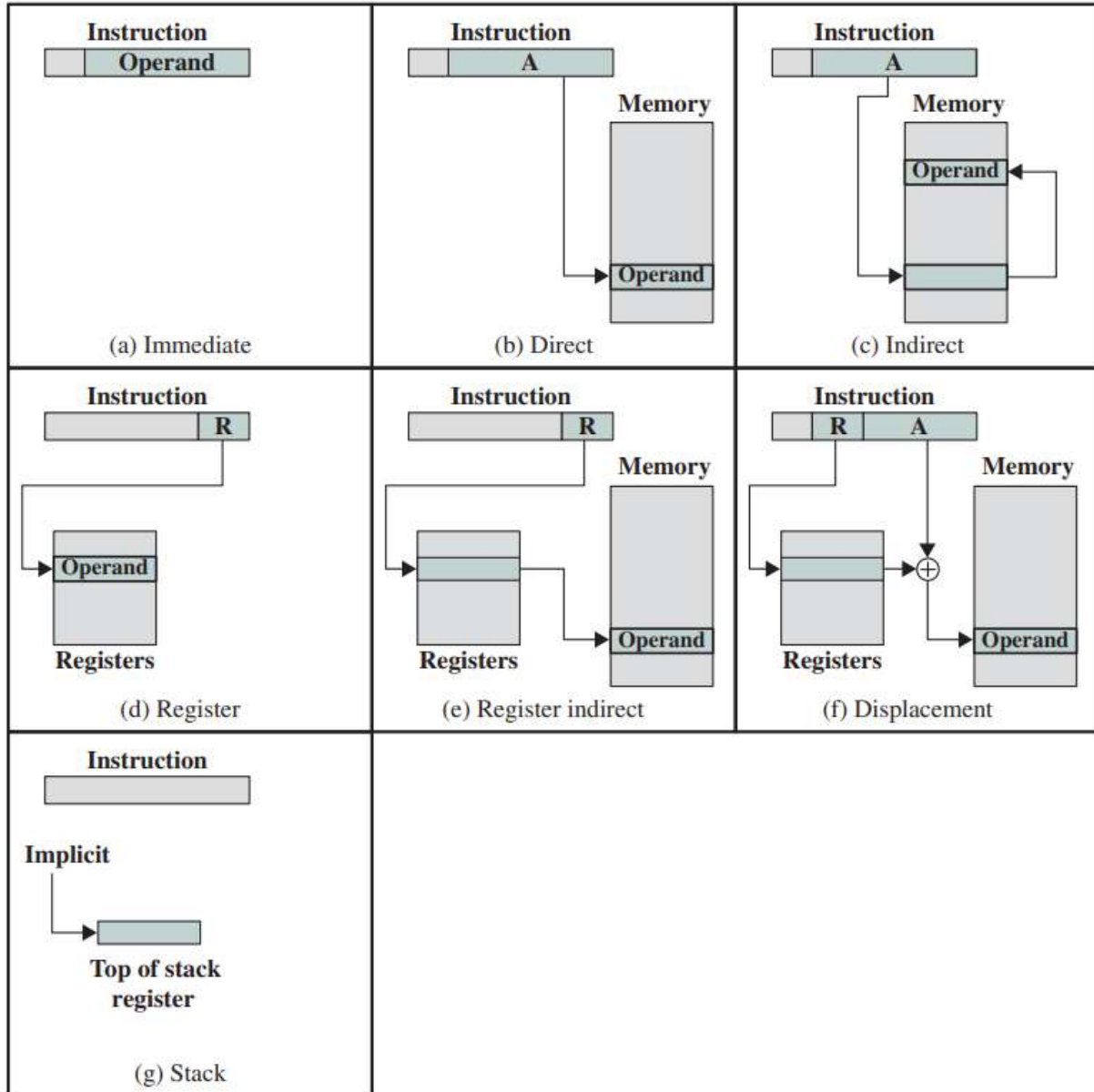


# Addressing Modes

الـ address field يعد في الـ instruction format نموذج صغيرا نسبيا. يعني نود الاشارة على مجموعة كبير من المواقع في الذاكرة الرئيسية او في الذاكرة الافتراضية. ولتحقيق هذا الهدف تم استخدام مجموعة متنوعة من addressing techniques . جميعها تشمل تنازل (trade off) بين نطاق العنوان (address range) من ناحية مرونة المعالجة (addressing flexibility) وعدد المراجع الذاكرة في التعليمات او مدى تعقيد حساب العنوان هنا سنفحص تقنيات او اوضاع العنوان الاكثر شيوعا :

- Immediate
- Direct
- Indirect
- Register
- Register indirect
- Displacement
- Stack

الصورة توضح الـ modes :



- A

تدل على محتويات حقل العنوان في التعليمات .

- R

محتويات حقل العنوان في التعليمات التي تشير الى السجل (Register).

- EA

العنوان الفعلي (effective) للموقع الذي يحتوي على المعامل المشار اليه.

- (X)

محتويات موقع الذاكرة X او السجل.

السؤال كيف يمكن للمعالج تحديد وضع العنوان الذي يتم استخدامه في تعليمات معينة. يتم اتخاذ عدة طرق. في كثير من الاحيان ستستخدم Opcodes المختلفة اوضاع المعالجة مختلفة. بالاضافة الى ذلك يمكن استخدام بت واحد او اكثر في تنسيق التعليمات mode field . نحدد قيمة mode field الذي سيتم استخدامه. وثانيا يتعلّق بتفسير عنوان (EA) . في النظام الذي لا يحتوي على ذاكرة افتراضية سيكون العنوان الفعال اما عنوان ذاكرة رئيسي او سجل. في نظام الذاكرة الافتراضية العنوان الفعال هو عنوان افتراضي او سجل. الـ mapping الفعلي لـ physical address هو احدى وظائف الـ memory management unit (MMU) . وهو يعتبر غير مرئي للمبرمج .

**Table 13.1** Basic Addressing Modes

Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	EA = R	No memory reference	Limited address space
Register indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA = A + (R)	Flexibility	Complexity
Stack	EA = top of stack	No memory reference	Limited applicability

- Immediate Addressing

ابسط شكل من اشكال العنونة هو العنونة الفورية حيث تكون قيمة المعامل موجودة في التعليمات.

Operand = A

يمكن استخدام هذا الوضع لتحديد واستخدام الثوابت (constants) او تعيين القيم الاولى للمتغيرات. وعادة ما يتم تخزين الرقم في شكل two's complement . يتم استخدام الجزء الموجود اقصى اليسار كـ sign bit . عندما يتم تحميل المعامل في سجل البيانات يتم تمديد اشارة البت (sign bit) الى اليسار والى حجم full data word . في بعض الحالات يتم تفسير القيمة الثنائية المباشرة على انها عدد صحيح غير سالب. وتتميز الـ immediate addressing في عدم الحاجة الى memory reference للحصول على operand وبالتالي توفير دورة ذاكرة واحدة. العيب هو ان حجم الرقم يقتصر على حجم address field والذي في معظم مجموعات التعليمات (instruction sets) صغيرة مقارنة بـ word length .

- Direct Addressing

أحد أشكال العنوان البسيطة هو العنوان المباشرة حيث يحتوي address field على effective address للمعامل.

$$EA = A$$

كانت هذه التقنية شائعة في الأجيال السابقة من أجهزة الكمبيوتر ولكنها ليست شائعة في contemporary architectures .

يتطلب مرجعا واحدا فقط للذاكرة ولا يتطلب أي حساب خاص.

القيد الواضح أنه يوفر limited address space .

- Indirect Addressing

مع العنوان المباشرة يمكن طول الـ address field عادة أقل من word length من ما يعطي limiting the address range . أحد الحلول هو جعل address field يشير إلى عنوان الـ word الموجودة في الذاكرة التي بدورها تحتوي على عنوان كامل للمعامل.

$$EA = (A)$$

كما تم تعريفه سابقا يجب على تفسير الأقواس على أنها محتويات ذات معنى. الميزة الواضحة لهذا الأسلوب هي أنه بالنسبة لطول الـ (word) N تتوفر الآن مساحة قدرها  $2^n$  . العيب هو أن تنفيذ التعليمات يتطلب مرجعين للذاكرة لجلب المعامل : أحدهما للحصول على address الثاني الحصول على القيمة. على الرقم من عدد الكلمات التي يمكن معالجتها يساوي الآن  $2^{nN}$  إلا أن عدد العناوين الفعالة المختلفة التي يمكن الرجوع إليها في أي وقت يقتصر على  $2^k$  حيث أن K هي address field .

عادة القيد هذا ليس مرهقا ويمكن أن يكون أحد asset .

في virtual memory environment يمكن تقييد كافة العناوين الفعالة إلى صفحة 0 في أي عملية.

القيد الوحيد هو أن حجم الصفحة يجب أن يكون أكبر من أو يساوي  $2^k$  . عندما تكون العملية نشطة ستكون هناك إشارات متكررة إلى صفحة 0 مما يجعلها تبقى في الذاكرة الحقيقية.

- Register Addressing

الـ Register addressing تشبه direct addressing . والفرق الوحيد بينهم أن الـ address field يشير إلى register بدلا من الـ main memory address :

$$EA = R$$

للتوضيح إذا كانت محتويات register address field هي 5 فإن R5 هو العنوان المقصود وقيمة المعامل موجودة في R5 . المزايا عنوان التسجيل في أنه 1 – لا يلزم سوى حقل عنوان صغير في التعليمات 2 – لا يلزم وجود مراجع ذاكرة تستغرق وقتا طويلا.

العيب أن المساحة محدودة للغاية. بسبب العدد المحدود للغاية من السجلات (مقارنة بمواقع الذاكرة الرئيسية) فإن استخداماتها بهذه الطريقة يكون منطقيا فقط إذا تم استخدامها بكفاءة .

إذا تم احضار كل معامل إلى السجل من الذاكرة الرئيسية وتشغيله مرة واحدة ثم اعادته إلى الذاكرة الرئيسية فسيتم إضافة خطوة وسيطة مهدرة .

- Register Indirect Addressing

الـ register addressing تشبه direct addressing فان الـ register indirect addressing تشبه الـ indirect addressing . في كلتا الحالتين الاختلاف الوحيد هو اذا كان حقل العنوان يشير الى موقع الذاكرة او السجل.

$$EA = (R)$$

- Displacement Addressing

السؤال الرئيسي الذي يطرح نفسه بالنسبة لوضع معالجة اسلوب Displacement هو مدى الازاحة المستخدمة بناء على استخدام احجام الازاحة المختلفة يمكن اتخاذ قرار بشأن الاحجام التي سيتم دعمها يعد اختيار احجام حقل الازاحة امرا مهما لانها تؤثر بشكل مباشر مع طول التعليمات.

$$\text{displacement addressing : } EA = A + (R)$$

. ثلاثة استخدامات الاكثر شيوعا لـ displacement addressing :

### 1. Relative addressing

الـ Relative addressing هو تقنية معالجة التعليمات ومناطق البيانات من خلال تحديد موقعها بالنسبة لـ Program Counter PC او الى بعض الـ symbolic location . هذا النوع من العنونة يكون دائما بالبايت وليس بالبت او الكلمات او التعليمات بتالي فان التعبير  $+4$  \* يحدد عنوان اكبر بـ 4 بايت من القيمة الحالية لـ PC . بالنسبة لـ IBM\* .

$$EA = PC + \text{Address field value}$$

$$PC = PC + \text{Relative value}$$

### 2. Base-register addressing

الـ Base register addressing يتم استخدام وضع معالجة السجل الاساسي لتنفيذ التحكم بين الـ segment . في هذا الوضع يتم الحصول على العنوان الفعال عن طريق اضافة قيمة base register الى قيمة حقل العنوان :

$$EA = \text{Base register} + \text{Address field value}$$

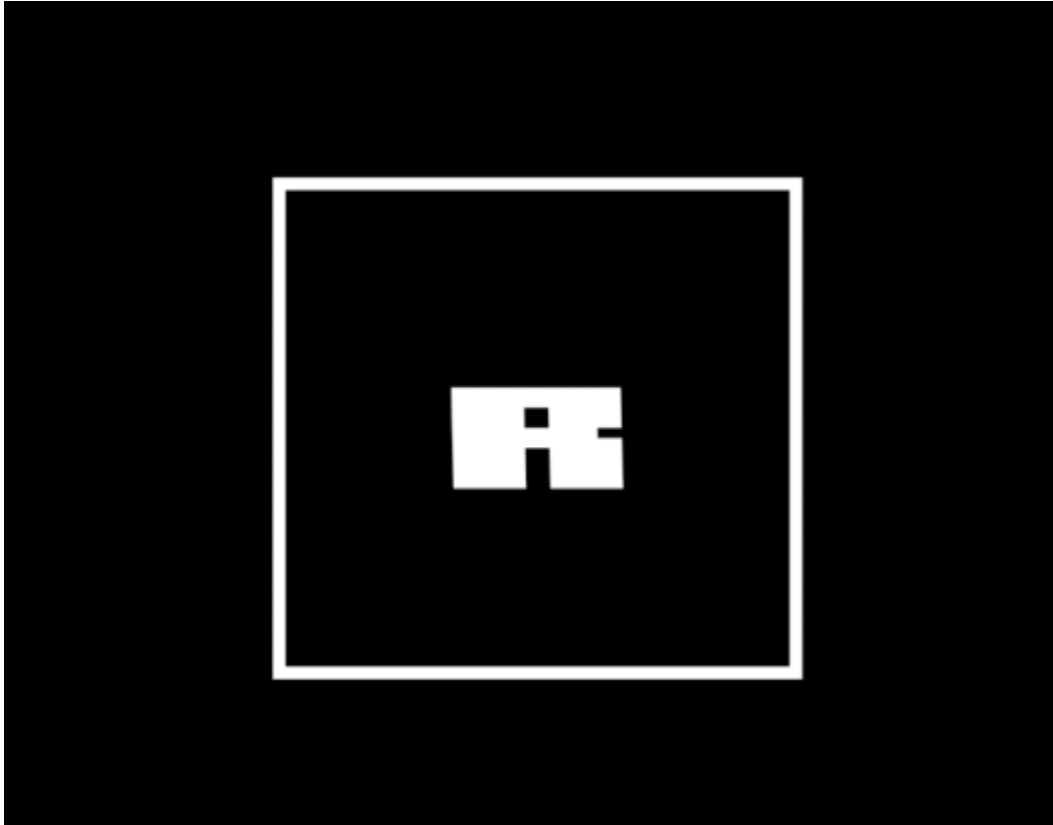
$$PC = \text{Base register} + \text{Relative value}$$

### 3. Indexing

يعد طريقة لتوليد عنوان فعال يعدل العنوان المحدد الوارد في التعليمات من خلال محتويات سجل الفهرس . عادة ما يكون التعديل عبارة عن اضافة محتويات الى سجل الفهرس الى العنوان المحدد.

يعد وضع عنونة الفهرس مفيد جدا عندما تصل التعليمات الموجودة في البرنامج الى مصفوفة او نطاقات مشابهه او كبيرة من عناوين الذاكرة. بالملخص تعني تحديد العنوان النهائي للبيانات عن طريق اضافة ازاحة الى العنوان الاساسي وفي كثير من الاحيان يتم تخزينها ككتل كاملة في الذاكرة .

**AhmadAlFareed**



Twitter : [https://twitter.com/dr\\_retkit](https://twitter.com/dr_retkit)

YouTube : <https://www.youtube.com/@retkit1823>