تصميم وحدة التحكم
Control Unit (1)

م. عبير ميّا م. مصعّب خباز

26/10/2022

محتوى مجاني غير مخصص للبيع التجاري

أنماط العنونة

العنونة بالسجل Register Addressing

- معامل التعليمة عبارة عن سجل.
- نتعامل مع السجلات فقط كما في التعليمة:

$$sub\ \$R_1, \$R_2, \$R_3 \Leftrightarrow \$R_1 = \$R_2 - \$R_3$$

أي نقوم بطرح محتوى السجل $\$R_3$ من محتوى السجل $\$R_2$ ونضع التابع في السجل $\$R_1$.

العنونة الفورية Immediate Addressing

- معامل التعليمة عبارة عن عدد ثابت ضمن التعليمة.
- نتعامل مع السجلات بالإضافة إلى قيم ثابتة كما في التعليمة:

$$ori\ \$R_1, \$R_2, 2 \Leftrightarrow \$R_1 = \$R_2\ or\ 2$$

أي نقوم بعملية or بين محتوى السجل $\$R_2$ والقيمة الثابتة 2 ونضع الناتج في السجل $\$R_1$.

العنونة المباشرة Direct Addressing

- معامل التعليمة عبارة عن عنوان موقع في الذاكرة.
- نتعامل مع عنوان موجود في الذاكرة كما في التعليمة: $Add_D\ \$R, [40h]$

أيّ $[40h]$ هي عبارة عن عنوان موقع في الذاكرة تتضمن القيمة التي نريد جمعها مع محتوى السجل $\$R$.

العنونة غير المباشرة Indirect Addressing

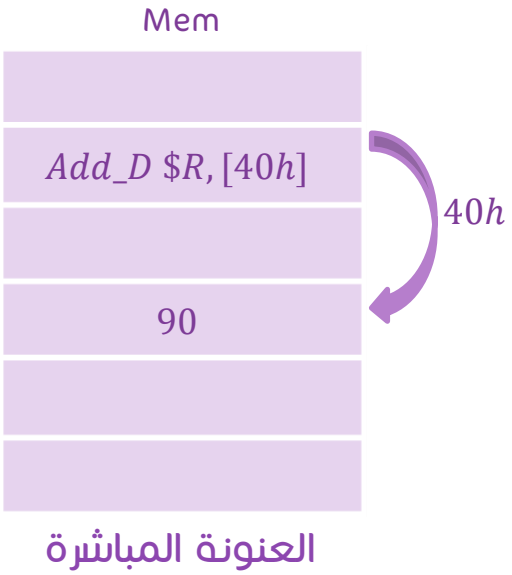
- معامل التعليمة عبارة عن عنوان لعنوان موقع آخر بالذاكرة.
- أيّ هنا نتعامل مع مؤشر يدل على العنوان الذي يحوي القيمة التي نريدها كما في التعليمة.

$$Add_ID\ \$R, @40h$$

الموقع ذو العنوان $40h$ يحوي عنوان موقع آخر في الذاكرة وذلك الموقع الآخر فيه القيمة التي نريد جمعها مع محتوى السجل $\$R$.

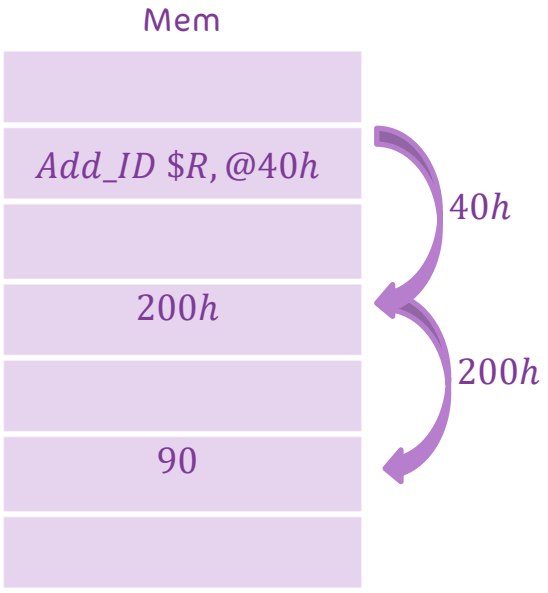
ملاحظة: $40h$ هي عبارة عن عدد ممثل بالنظام الـ$hexadecimal$ (السداسي العشري).

- في أول طريقتين للعنونة (السجل والفورية) لايوجد أي وصول للذاكرة ونقوم بجمع قيم السجلات أو القيم العددية دون الحاجة للذاكرة.
- بشكل مختصر الفرق بين العنونة المباشرة وغير المباشرة أنه في تعليمة العنونة المباشرة العدد المكتوب عبارة عن عنوان لموقع في الذاكرة يحوي القيمة المرادة أما في العنونة غير المباشرة يكون العدد المكتوب عبارة عن مؤشر (عنوان موقع في الذاكرة). يؤشر على عنوان لموقع آخر يحوي القيمة المرادة.



العنونة المباشرة

العنوان $40h$ يحوي القيمة المرادة مباشرة



العنونة غير المباشرة

العنوان $40h$ يحوي العنوان $200h$ الذي يحوي القيمة المرادة

القيمة 90 هي القيمة العددية التي سيتم جمعها مع السجل R بالحالتين.

مسألة

نريد تصميم معالج فيه المواصفات التالية:

عرض ممر المعطيات وكلمة الذاكرة وعرض التعليمة يساوي $16\ bits\ (2\ bytes)$. أيّ يقوم بعنونة 2^{12} سطر بالذاكرة (المواقع في الذاكرة 2^{12} موقع) خطوط العنونة $12\ bits$. والعنونة بالكلمة وفي مثالنا هذا يقصد بها بأنه بغض النظر عن حجم الكلمة سوف تأخذ عنوان واحد في الذاكرة أيّ (مثلاً200 201 202 ...) وفيه السجلات التالية:

1. IR سجل التعليمة $Instruction\ Register$

وهو سجل يقوم بحفظ التعليمة بعد تحميلها من الذاكرة, وعلى اختلاف المعالجات فإن IR ممكن أن يقوم بحفظ كامل التعليمة أي $16\ bit$ في هذا المثال أو أن يقوم بحفظ الـ$opcode$ فقط. ويحوي عادة إما التعليمة $instruction$ أو رمز التعليمة $opcode$ ويكون حجمه في هذه المسألة $4\ bit$ أيّ يقوم بحفظ الـ$opcode$ فقط.

2. AR سجل العنونة $Address\ Register$

وهو سجل مخصص لعنونة الذاكرة, أيّ يحوي عنوان الذاكرة لأي موقع يتم قراءته أو الكتابة عليه وهو بعرض $12\ bits$.

3. DR سجل المعطيات $Data\ Register$

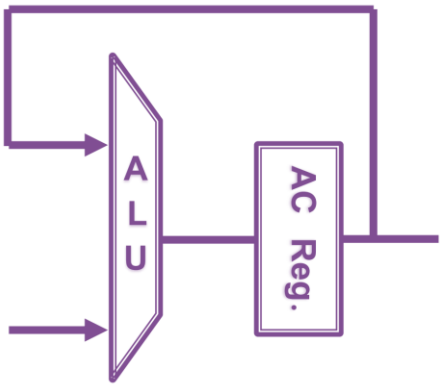
يسمى في بعض المعالجات بـ$MDR\ (Memory\ Data\ Register)$ أو $MBR\ (Memory\ Buffer\ Register)$ وهو سجل مخصص للـ$data$ في الذاكرة وهذا يعني بأن أيّ تعليمة تتعامل مع الذاكرة (قراءة معطيات من الذاكرة أو كتابتها) سوف يتم تحميلها على السجل DR أولاً, وعرضه $16\ bits$.

4. PC عداد البرنامج $Program\ Counter$

مهمته التأشير على التعليمة التالية الواجب إحضارها من الذاكرة لمعالجتها ويزيد بمقدار معين حسب نمط العنونة (وبما أن العنونة بالكلمة في مثالنا فسيكون بالشكل $PC = PC + 1$) ويكون العرض $12\ bits$.

5. AC سجل المراكم $Accumulator$

وهو أحد مداخل ALU ومخارجها أيضاً إذ يقوم مثلاً بتخزين قيمة معينة ناتجة عن جمع (عملية حسابية) عدة مرات لقيم مختلفة لذلك سُميَ بالمراكم وهو بعرض $16\ bits$.



6. R سجل المستخدم $User\ Register$

يمكن أن يوجد عدة سجلات للاستخدام ولكن في هذه المسألة يوجد سجل واحد R وهو بعرض $16\ bit$.

ملاحظات:

- تكون التعليمة مؤلفة من الحقلين: رمز التعليمة $opcode$, معامل التعليمة $operand$.

op	$Operand\ (x)$
$4\ bit$	$12\ bit$

- جميع السجلات السابقة هي سجلات قياسية العمل وموجودة في جميع المعالجات وليس فقط في هذه المسألة أيّ مثلا (سجل العنونة مخصص لعنونة الذاكرة في جميع المعالجات) وأسمائها ثابتة لا تتغير.

- يمكن للمعالج تنفيذ التعليمات التسعة التالية (وهي تعليمات غير خاصة بمعالج معين بل نحن افترضناها كنوع من التجريب على المعالج الذي سنقوم بتصميمه).

1. نقل محتوى المراكم إلى السجل R (والأدق أن نقول نسخ محتوى AC للR).

$$MOV_R \Leftrightarrow R \leftarrow AC$$

2. طرح قيمة السجل R من محتوى المراكم AC وحفظها في المراكم

$$SUB_R \Leftrightarrow AC \leftarrow AC - R$$

3. تطبيق العملية المنطقية or بين محتوى المراكم والقيمة في السجل R وحفظها في المراكم AC (نمط العنونة هنا العنونة بالسجل).

$$OR_R \Leftrightarrow AC \leftarrow AC\ or\ R$$

4. تحميل القيمة الموجودة في العنوان (المعامل) x في الذاكرة ووضعها في السجل R (عنونة مباشرة $D(Direct)$).

$$LDR_D\ x \Leftrightarrow R \leftarrow M[x] \quad \text{Load from } M[x] \text{ and put it in } R$$

5. تخزين محتوى AC (المراكم) في الذاكرة عند العنوان x (العنونة مباشرة).

$$STAC_D\ x \Leftrightarrow M[x] \leftarrow AC \quad \text{store } AC \text{ in } M[x]$$

6. العنوان x يحوي قيمة هي عنوان آخر بداخله نريد تخزين قيمة السجل R. أيّ هنا نمط العنونة (غير مباشرة).

$$STR_ID\ x \Leftrightarrow M[M[x]] \leftarrow R \quad \text{store } R \text{ in } M[M[x]]$$

7. تقوم بطرح القيمة الموجودة ضمن العنوان x في الذاكرة من محتوى المراكم AC ووضع الناتج في المراكم AC (عنونة مباشرة).

$$SUB_D\ x \Leftrightarrow AC \leftarrow AC - M[x]$$

8. القفز للتعليمة الموجودة في العنوان x $JMP_D\ x \Leftrightarrow Jump\ to\ instruction(direct\ address\ x)$

9. لا تقوم بأي شيء وسنستخدمها كثيراً لاحقاً. $NOP \Leftrightarrow No\ Operation\ (Do\ Nothing)$

- ليس بالضرورة أن نجد في التعليمات ID/D لمعرفة فيما إذا كانت العنونة مباشرة أم لا ولاحتى بأسماء التعليمات فقط المهم هو التوصيف الموجود بجانب التعليمة وفهمه جيداً ومعرفة كيفية عمله وتنفيذه.
- أيّ تعليمة قفز سوف يكون التعديل على قيمة PC أياً كانت العمليات يجب أن تعدل قيمة الـPC.

1. بفرض أن القيم الابتدائية للسجلات $AC = 5, R = 1, PC = 60$ ومحتوى مواقع الذاكرة الابتدائية هي: $M[@] = @ - 10$ مثلاً $\Leftarrow M[100] = 90$

والمطلوب: ما هي قيم PC, R, AC ومواقع الذاكرة بعد تنفيذ كل من تسلسل التعليمات التالية:

التعليمات	PC	R	AC	$M[@]$
60) SUB_R $AC = 4 \Leftarrow$ طرح قيمة R من محتوى الـAC ووضع الناتج في الـAC	61	1	$AC - R \to AC$ $5 - 1 \to AC$ $4 \to AC$	= ليس هناك تعديل في الذاكرة
61) OR_R نقوم بعملية OR بين الـAC وR ونضع الناتج في الـAC	62	لن تتغير =	$AC = 4 \to 100$ $R = 1 \to 001$ *or* $AC = (101)_2 = 5$	= تنفذ هذه التعليمات على السجلات فقط
62) LDR_D 200 $R \leftarrow M[200]$ $200 - 10 = 190$ $R \leftarrow 190$	63	190	= لم يطرأ عليها أيّ تغيير	= نحن فقط نقرأ من الذاكرة ولا نعدل عليها
63) $STAC_D$ 500 $M[500] \leftarrow AC$ $M[500] \leftarrow 5$	64	=	= لم يطرأ عليها أيّ تغيير	$M[500] = 5$
64) STR_ID 350 $M[M[350]] \leftarrow R$ $M[350 - 10] \leftarrow 190$ $M[340] \leftarrow 190$	65	=	=	$M[340] = 190$
65) SUB_D 40 $AC \leftarrow AC - M[x]$ $AC \leftarrow 5 - M[40]$ $AC \leftarrow 5 - 30$	66	=	$AC = -25$	=
66) MOV_R $R \leftarrow AC$ $R \leftarrow -25$	67	-25	=	=

67) $JMP_D\ 79$ قفز عنونة مباشرة $M[79] = 79 - 10 = 69$	69	=	=	=
68)OR_R لن تتنفذ لأننا قفزنا سابقاً للسطر 69	=	=	=	=
69) NOP هي تعليمة فقط لتضيّع الوقت وسنتعرف استخداماتها لاحقاً.	70 قمنا بزيادته ليؤشر على التعليمة التالية	=	=	=

ملاحظات

- يقصد بالإشارة = أنه لا يوجد أيّ تعديل على القيم هنا.
- في التعليمات التي ستنفذ فقط على السجلات مثل $(AND_R, OR_R, SUB_R, ADD_R)$ لن يكون هناك تغيير يطرأ على الذاكرة لأننا لم نتعامل معها أصلاً.
- في التعليمات المنطقية مثل (OR وAND وغيرها) يجب ألا ننسى تحويل الأعداد الموجودة في التعليمة إلى النظام الثنائي مثل السطر الذي يحوي التعليمة 61.
- قمنا بزيادة الـPC عند كل تعليمة ليؤشر على التعليمة التي تليها.
- نطبق تعليمة JMP على الـPC فقط, مثال:

$$JMP\ 79\ (\text{عنونة فورية}) \Rightarrow PC = 79$$

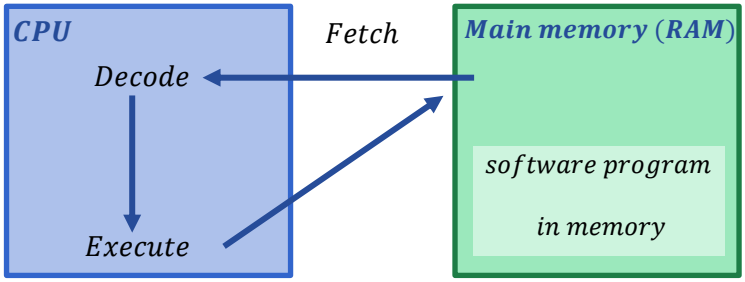
$$JMP\ D\ 79\ (\text{عنونة مباشرة}) \Rightarrow PC = M[79] = 79 - 10 = 69$$

$$JMP\ ID\ 79(\text{عنونة غير مباشرة}) \Rightarrow PC = M[M[79]] = M[69]$$

$$PC = 69 - 10 = 59$$

- ولا ننسى أن الـPC يدلنا على التعليمة التي ستنفذ بعدها أيّ وفي مثالنا السابق سننتقل للسطر 69 لتنفيذه ⇐ السطر 68 لن ينفذ.

2. اكتب مراحل تنفيذ التعليمات التسعة السابقة بلغة نقل السجلات $RTL\ (Register\ Transfer\ Language)$



- Fetch إخراج التعليمة من الذاكرة.
- Decode تحليل التعليمة ومعرفة إشارات التحكم وقراءة معاملات التعليمة.
- Execute تنفيذ التعليمة.

مرحلة جلب التعليمة Fetch وزيادة العداد PC, يتضمن الجلب وقراءة التعليمة من الذاكرة وحتى حفظها في سجل التعليمة IR ويمر بثلاث مراحل:

$$F_1: AR \leftarrow PC$$

$$F_2: DR \leftarrow M[AR]; PC \leftarrow PC + 1$$

$$F_3: IR \leftarrow DR(OP); AR \leftarrow DR(x)$$

تذكر

- أي عنوان ذاكرة يجب التعامل معه عن طريق AR.
- أي قيمة نقوم بجلبها من الذاكرة أو كتابتها بالذاكرة يجب أن تمر على DR.

ملاحظات

- (;) يدل على حدوث أكثر من عملية صغرية على التوازي في نفس التعليمة الصغرية.
- جميع التعليمات تمر بهذه المراحل الثلاثة لجلبها من الذاكرة.
- لقد وضعنا تعليمة زيادة الـ PC في F_2 لأنه لا يمكن وضعها في F_1 إذ لايمكن جلب القيمة وتخزينها وزيادتها في نفس الوقت. ويمكننا وضعها في F_3 ولكن وضعناها في F_2 للتوازي في تنفيذ التعليمات وعدم هدر الوقت.

النهاية

"Practice isn't the thing you do once you're good. It's the thing you do that makes you good."

— Malcolm Gladwell