

وحدة التحكم المبرمجة صغيراً

1.15 مفاهيم أساسية

التعليمات الصغيرة

وحدة التحكم المبرمجة صغيراً

وحدة تحكم ويلكس Wilkes

الميزات والمساوئ

2.15 تنالي التعليمات الصغيرة

اعتبارات تصميمية

تقنيات التنالي

توليد العناوين

تنالي التعليمات الصغيرة في النظام LSI-11

3.15 تنفيذ التعليمات الصغيرة

تصنيف التعليمات الصغيرة

ترميز التعليمات الصغيرة

تنفيذ التعليمات الصغيرة في LSI-11

تنفيذ التعليمات الصغيرة في IBM 3033

4.15 النظام TI-8800

صيغة التعليمات الصغيرة

وحدة التنالي الصغيرة

وحدة الحساب والمنطق ذات السجلات

5-15 تطبيقات البرمجة الصغيرة

6.15 قراءات يُنصح بها

7-15 مسائل



572

سُردت حالة البرمجة الصغيرة في العدد الصادر في شباط 1964 من مجلة Datamation. ولم يكن آنذاك النظام المبرمج صغيراً واسع الاستخدام. ولخصت إحدى المقالات وجهة النظر التي كانت سائدة في ذلك الوقت [HiLL64] بالقول "إن مستقبل البرمجة الصغيرة ضبابي نوعاً ما". ولم يُظهر أي مصنع ذي شأن اهتماماً في هذه التقنية، على الرغم من فحصهم لها.

تغيرت هذه الحالة تغيراً جذرياً خلال بضعة أشهر؛ إذ أعلنت الشركة IBM عن ظهور السلسلة 360/360 System في شهر نيسان، وكانت جميع نماذج تلك السلسلة مبرمجة صغيراً، باستثناء النماذج الضخمة منها. وعلى الرغم من أن السلسلة 360 سبقت توفر ذاكرات القراءة فقط ROM المبنية على أنصاف النواقل، إلا أن مزايا البرمجة الصغيرة دفعت الشركة IBM إلى إجراء هذه الخطوة. وأصبحت البرمجة الصغيرة، منذ ذلك الحين، وسيلة شائعة الاستخدام في العديد من التطبيقات، وأحد تلك التطبيقات هو تنجيز وحدة التحكم في المعالج. يعرض هذا الفصل ذلك التطبيق.

1.15 مفاهيم أساسية Basic Concepts

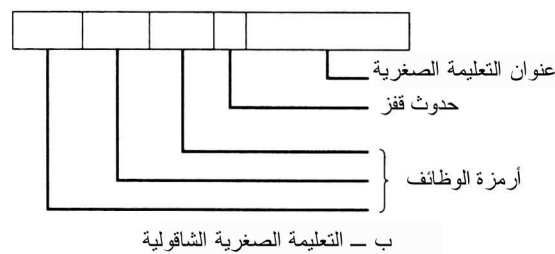
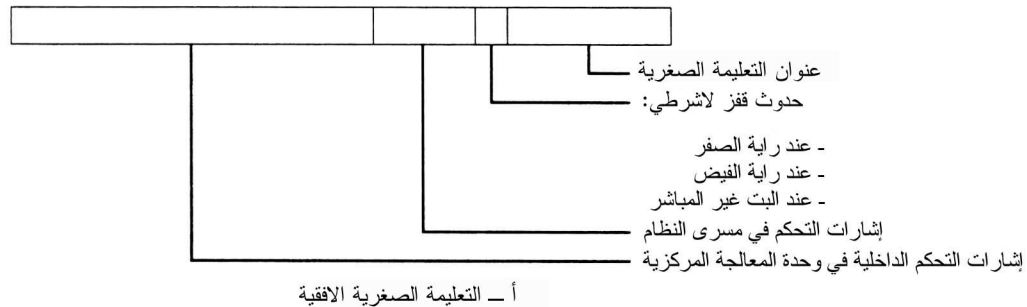
التعليمات الصغيرة Microinstructions

تبدو وحدة التحكم تجهيزة بسيطة إلى حد ما، ولكن تنجيزها بوصل العناصر المنطقية الأساسية توصيلاً بينياً ليس مهمة سهلة. ينبغي أن يتضمن التصميم المنطق اللازم لتحقيق تنالي التعليمات الصغيرة، وتنفيذ العمليات الصغيرة، وتفسير أرمزة التعليمات، واتخاذ القرارات اعتماداً على رايات وحدة الحساب والمنطق ومن الصعب تصميم ذلك عتادياً واختباره. أضف إلى ذلك، أن التصميم المبني عتادياً جامد نسبياً، إذ يصعب مثلاً تغيير التصميم إذا رغب المصمم في إضافة تعليمة آلة جديدة.

والحل البديل هو تنجيز وحدة تحكم مبرمجة صغيراً، وهذا حل شائع الاستخدام في المعالجات ذات مجموعة التعليمات المعقدة CISC المعاصرة. لنعد ثانيةً إلى الجدول 1-14. فبالإضافة إلى استخدام إشارات التحكم، تُوصف فيه كل عملية صغيرة بتدوين رمزي. ويبدو ذلك التدوين مشابهاً للغة برمجة إلى حد بعيد. وفي الواقع، فهو لغة برمجة تُعرف باسم لغة البرمجة الصغيرة *microprogramming language*. يصف كل سطر فيها مجموعة العمليات الصغيرة التي تحدث في كل لحظة، ويسمى تعليمة صغيرة *microinstruction*. يُسمى تنالي التعليمات الصغيرة البرنامج الصغير *microprogram* أو البرمجيات الراسخة *firmware*. وتعكس التسمية الأخيرة حقيقة أن البرنامج الصغير حلٌ وسط بين العتاديات والبرمجيات. ومن الأسهل التصميم بواسطة البرمجيات الراسخة بدلاً من العتاديات، ولكن كتابة برنامج راسخ أصعب من كتابة برنامج برمجي.

كيف يمكن استخدام مفهوم البرمجة الصغيرة لتنجيز وحدة التحكم؟ لنفترض، من أجل كل عملية صغيرة، أن مهمة وحدة التحكم هي توليد مجموعة من إشارات التحكم فقط. ويكون إذن، من أجل كل عملية صغيرة، لأي خط تحكم صادر عن وحدة التحكم إحدى القيمتين: الوصل 'ON' أو الفصل 'OFF'. ويمكن تمثيل ذلك بالطبع برقم اثنائي لكل خط تحكم. ولذا، نستطيع إنشاء كلمة تحكم يعبر كل بت فيها عن خط من تلك الخطوط. وتمثل عندئذ كل عملية صغيرة بشكل نمطي مختلف من الوحدات والأصفار في كلمة التحكم. لنفترض أننا وضعنا تنالياً لكلمات التحكم بغية تمثيل العمليات الصغيرة المنفذة في وحدة التحكم. ينبغي بعدئذ أن ندرك أن تنالي العمليات الصغيرة ليس ثابتاً، ففي بعض الأحيان، تُستخدم دورة عنونة غير مباشرة،

وفي أحيان أخرى لا تُستخدم تلك الدورة. لهذا دعنا نضع كلمات التحكم في الذاكرة، بحيث يكون لكل كلمة منها عنوان فريد. ونضيف الآن حقل عنوان إلى كل كلمة تحكم. للدلالة على موقع كلمة التحكم التالية الواجب تنفيذها عند تحقق شرط معين (مثلاً، عندما يكون بت العنوان غير المباشرة في تعليمة ما، ذات دلالة على الذاكرة، مساوياً للواحد). ونضيف أيضاً بضع بتات أخرى لتحديد الشرط الواجب فحصه.

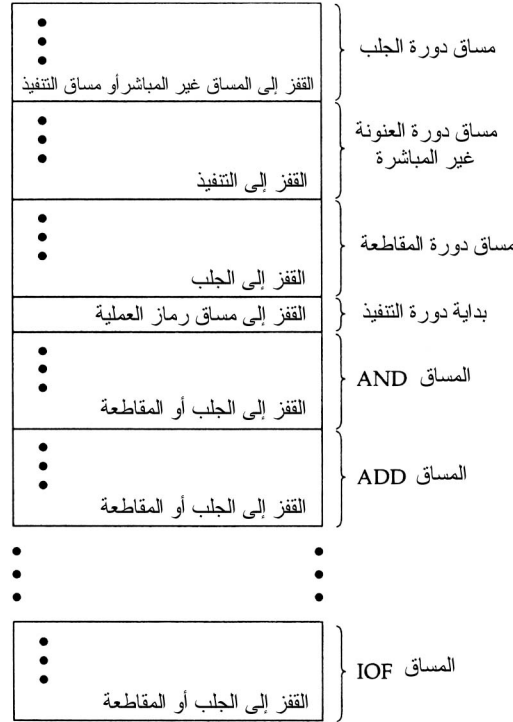


الشكل 1.15: صيغ نموذجية للتعليمات الصغيرة

يُعرف ذلك بالتعليمات الصغيرة الأفقية *horizontal microinstruction*، والتي يظهر مثال عليها في الشكل

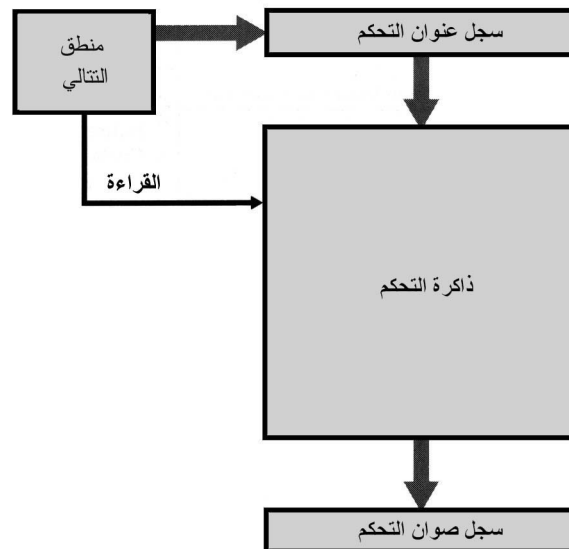
1-15-أ). وتكون صيغة التعليمة الصغيرة أو كلمة التحكم على النحو الآتي: يوجد بت لكل خط تحكم داخل المعالج، وبت لكل خط تحكم في مسرى النظام. كما يوجد حقل للشرط يشير إلى الشرط الذي سيُفحص قبل التفريع، ويوجد حقل يحوي عنوان التعليمة الصغيرة الواجب تنفيذها عند إجراء التفريع. تُفسر مثل هذه التعليمات كما يلي:

1. لتنفيذ هذه التعليمة الصغيرة، تُفعل كل خطوط التحكم المقابلة للقيمة 1 في كلمة التحكم، وتُفصل الخطوط المقابلة للقيمة 0 فيها. تؤدي إشارات التحكم الناتجة إلى إجراء عملية صغيرة واحدة أو أكثر.
 2. إذا لم يتحقق الشرط المحدد ببتات الشروط، تُنفذ التعليمة الصغيرة التالية في التالي.
 3. إذا تحقق الشرط المحدد ببتات الشروط، تُنفذ التعليمة الصغيرة المشار إليها في حقل العناوين.
- يبين (الشكل 2.15) كيف يمكن ترتيب كلمات التحكم أو التعليمات الصغيرة في ذاكرة التحكم *control memory*. ينبغي تنفيذ التعليمات الصغيرة في كل مساق *routine* تنفيذاً متتابعياً، وينتهي كل مساق بتعليمة قفز أو تفريع للدلالة على التعليمة التالية. ويوجد مساق خاص لدورة التنفيذ، يهدف فقط إلى تحديد أحد مساقات تعليمات الآلة (كالتعليمة *ADD AND* ونحوها) الواجب تنفيذها في المرحلة اللاحقة، اعتماداً على رماز العملية الحالية.



الشكل 2.15: تنظيم ذاكرة التحكم

إن ذاكرة التحكم، المبينة في (الشكل 2.15)، هي وصف موجز لعمل وحدة التحكم الكامل. فهي تعرّف تتالي العمليات الصغيرة التي ستنفذ خلال كل دورة (جلب، عنونة غير مباشرة، تنفيذ، مقاطعة)، وهي تحدّد تتالي تلك الدورات. إن لم يكن في وحدة التحكم تفاصيل أكثر مما سبق، لكان التدوين السابق أداة مفيدة لتوثيق عمل وحدة التحكم لحاسوب معين. ولكن الأمر أكثر تعقيداً من ذلك. ويمثل التدوين السابق أيضاً طريقة لتنفيذ وحدة التحكم.



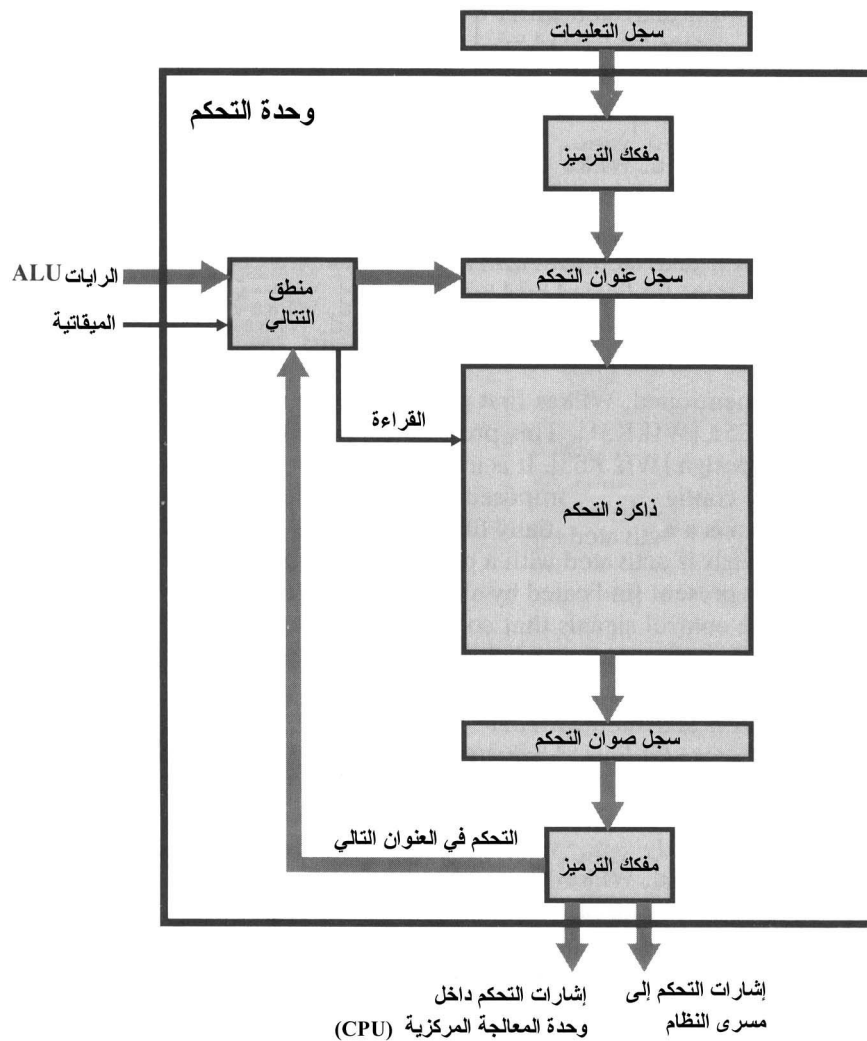
الشكل 3.15: البنينان الصفري لوحدة التحكم

وحدة التحكم المبرمجة صغرياً Microprogrammed Control Unit

تحتوي ذاكرة التحكم، المذكورة في (الشكل 2.15)، على برنامج يصف سلوك وحدة التحكم. فيمكن إذن تنجيز وحدة التحكم بتنفيذ ذلك البرنامج فقط.

يبين (الشكل 3.15) العناصر الأساسية لذلك التنجيز. تُخزَّن مجموعة التعليمات الصغيرة في ذاكرة التحكم. ويحوي سجل عنوان التحكم control address register عنوان التعليمات الصغيرة الواجب قراءتها، وتُنقل التعليمات الصغيرة، بعد قراءتها، من ذاكرة التحكم إلى سجل صنوان التحكم.

يُربط الجزء الأيسر من ذلك السجل (انظر الشكل 1.15 أ) بخطوط التحكم الصادرة عن وحدة التحكم. وتماثل عندئذ قراءة تعليمات صغيرة من ذاكرة التحكم تنفيذ تلك التعليمات. أما العنصر الثالث المبين في الشكل فهو وحدة التتالي التي تحمل سجل عنوان التحكم، وتصدر أمر القراءة.



الشكل 4.15: وحدة تحكم ويلكس المبرمجة صغيراً

لنفحص الآن هذه البنية فحوصاً مفصلاً، كما هو مبين في (الشكل 4.15). نجد مقارنة (بالشكل 4.14) أن لوحدة التحكم المداخل ذاتها (السجل IR، ورايات الوحدة ALU، والمخرجات)، والمخارج ذاتها (إشارات التحكم). تعمل وحدة التحكم على النحو الآتي:

1. لتنفيذ تعليمات ما، تصدر وحدة التتالي المنطقية أمر القراءة READ إلى ذاكرة التحكم.

2. تُقرأ الكلمة التي حُدِّدَ عنوانها في سجل عنوان التحكم، وتُخزَّن في سجل صنوان التحكم.
 3. يوَلَّد محتوى سجل صنوان التحكم إشارات التحكم والمعلومات المتعلقة بالعنوان التالي لوحدة التتالي المنطقية.
 4. تحمّل وحدة التتالي المنطقية عنواناً جديداً في سجل عنوان التحكم، اعتماداً على معلومات العنوان التالي القادمة من سجل صنوان التحكم ورايات وحدة الحساب والمنطق.
- يحدث كل ما سبق في دورة ميقاتية واحدة.
- تتطلب الخطوة الأخيرة، المذكورة سابقاً، بعض الإيضاح. تحمّل وحدة التتالي المنطقية، في نهاية أي تعليمة صغيرة، عنواناً جديداً في سجل عنوان التحكم. وتُتخذ، اعتماداً على قيمة رايات وحدة الحساب والمنطق وسجل صنوان التحكم، أحد القرارات الثلاثة التالية:
- ♦ الحصول على التعليمة التالية: تُضاف القيمة 1 إلى سجل عنوان التحكم.
 - ♦ القفز إلى مساق جديد اعتماداً على تعليمة قفز صغيرة: يُحمّل حقل العناوين في سجل صنوان التحكم داخل عنوان التحكم.

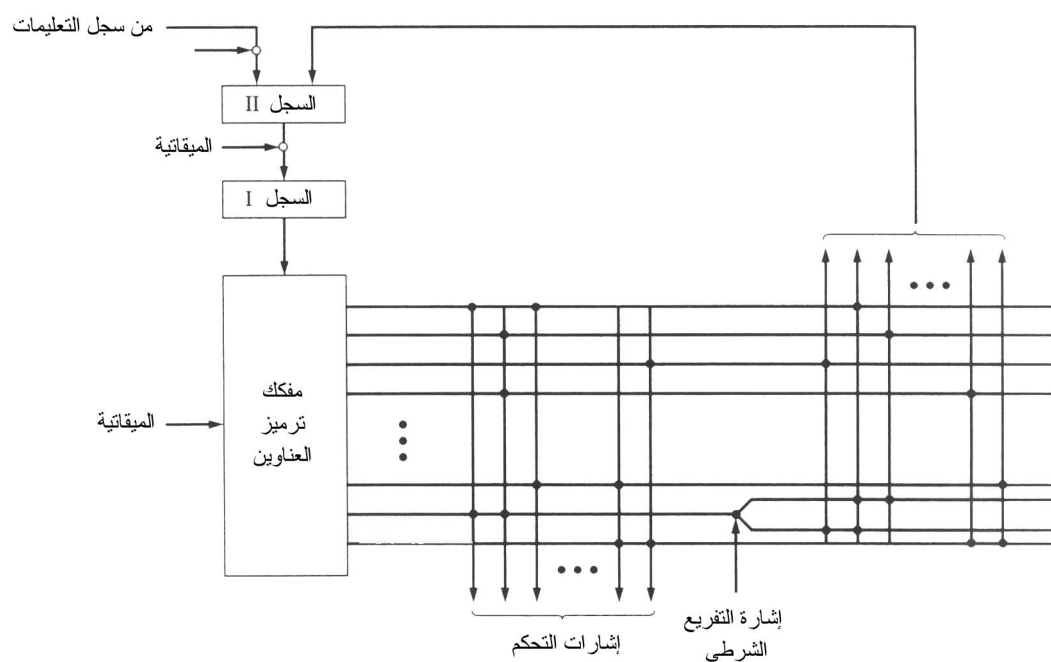
♦ القفز إلى مساق تعليمة آلة: يُحمّل سجل عنوان التحكم اعتماداً رماز العملية في سجل التعليمة IR. يبين (الشكل 4.15) مجتزأين يحملان الاسم "مفكّك الترميز decoder". يترجم المفكّك العلوي رماز العملية المخزّن في سجل التعليمة IR إلى عنوان ذاكرة تحكم. في حين لا يُستخدم المفكّك السفلي في التعليمات الصغيرة الأفقية، بل يُستخدم في التعليمات الصغيرة الشاقولية vertical mirco instruction (انظر الشكل 1.15. ب). وكما ذُكر سابقاً، يرتبط كل بت في حقل التحكم في التعليمة الصغيرة الأفقية بخط تحكم. أما في التعليمة الصغيرة الشاقولية، يُستخدم رماز لكل فعل يراد إجراؤه (مثلاً عملية النقل (PC) ← (MAR)، ويترجم المفكّك ذلك الرماز إلى إشارات تحكم فردية. تتميز التعليمات الصغيرة الشاقولية بأنها أكثر ترصاً (إذ تحتاج إلى عدد بتات أقل) مقارنة بالتعليمات الصغيرة الأفقية، ولكنها تتطلب إضافة بعض الدارات المنطقية، و من شأن ذلك زيادة التأخير الزمني.

وحدة تحكم ويلكس Wilkes control

كما ذُكر آنفاً، كان ويلكس أول من اقترح وحدة تحكم مبرمجة صغيراً في العام 1951 [wilk51]. ولقد طُوِّر ذلك المقترح ليصبح تصميمًا كامل التفاصيل [Wilk53]. ومن المفيد فحص ذلك المقترح الأساسي. يبين (الشكل 5.15) التشكيل الذي اقترحه ويلكس. يعتمد النظام على مصفوفة ثملاً جزئياً بالديودات. أثناء دورة الآلة، يُفعل سطر من المصفوفة بواسطة نبضة، ويولّد ذلك إشارات عند النقاط التي وضعت عندها الديودات، والممثلة على الشكل بنقاط غامقة. يوَلَّد الجزء الأول من السطر إشارات التحكم التي تتحكم في عمليات المعالج. أما الجزء الثاني، فيولّد عنوان السطر الواجب تفعيله في دورة الآلة القادمة. وبذا يغدو كل سطر في المصفوفة تعليمة صغيرة، ويكون مخطط المصفوفة النسقي هو ذاكرة التحكم.

يُخزَّن عنوان السطر المراد تفعيله في بداية الدورة في السجل I. يمثل ذلك العنوان دخل مفكّك الترميز، الذي يُفعل سطرًا من المصفوفة عند إرسال نبضة ميقاتية إليه. ويُنقل إلى السجل II رماز العملية المخزّن في سجل التعليمة أو الجزء الثاني من السطر المفعل تبعاً لإشارات التحكم. ويوصل السجل II بالسجل I عبر بوابة

منطقية بعد نبضة ميقائية. تفيد نبضات الميقائية المتبدلة في تفعيل سطر من المصفوفة ونقل السجل II إلى السجل I. ويُحتاج إلى ذلك الترتيب المحتوي على سجلين نظراً لأن مفكك الترميز ببساطة هو دارة تركيبية فقط، فإذا لم يُستخدم إلا سجل واحد، يصبح الخرج دخلاً خلال الدورة ذاتها، وهذا ما يؤدي إلى عمل غير مستقر. يماثل ذلك المخطط مقارنة البرمجة الصغيرة الأفقية كثيراً، والمذكورة في (الشكل 1.15. أ). ولكنه يختلف عنها بما يلي: يمكن زيادة سجل عنوان التحكم، تبعاً لما سبق، بمقدار 1 للحصول على العنوان التالي. أما في مخطط ويلكس، فالعنوان التالي محتوي في التعليمات الصغيرة. وللسماح بالتفرع، ينبغي أن يحتوي كل سطر على جزء أي عنوان، تتحكم فيهما إشارة تحكم (راية مثلاً)، كما هو موضح في الشكل.



الشكل 5.15: وحدة تحكم ويلكس المبرمجة صغرياً

قدّم ويلكس، بعد اقتراح ذلك المخطط، مثلاً على استخدام البرمجة الصغيرة في تنجيز وحدة تحكم لآلة بسيطة. ومن المفيد تكرار ذكر ذلك المثال، إذ إنه يُعدّ التصميم الأول المعروف لمعالج مُبرمج صغرياً، بالإضافة إلى شرحه للعديد من المبادئ الحديثة في البرمجة الصغيرة.

يتضمن معالج الآلة المفترضة السجلات التالية:

- المضروب به.
 - مراكم (ويمثل النصف الأقل دلالة)
 - مراكم (ويمثل النصف الأكثر دلالة)
 - سجل الإزاحة.
- إضافة إلى ذلك، هناك ثلاثة سجلات ورايتان، كل منها مرمزة على 1 بت، يمكن لوحدة التحكم فقط النفاذ إليها. وهذه السجلات هي:
- يفيد كسجل عنوان للذاكرة MAR، وكموقع تخزين مؤقت.
 - عداد البرنامج.

G. سجل مؤقت آخر يُستخدم للعدّ.

الجدول 1-15: مجموعة تعليمات الآلة لمثال ويلكس

| الوصف | الرمز |
|--|-------|
| $C(Acc) + C(n)$ to Acc_1 | $A n$ |
| $C(Acc) - C(n)$ to Acc_1 | $S n$ |
| $C(n)$ to Acc_2 | $H n$ |
| $C(Acc_2) \times Gcn$, to Acc , where $C(n) \geq 0$ | $V n$ |
| $C(Acc_1)$ to n , 0 to Acc | $T n$ |
| $C(Acc_1)$ to n | $U n$ |
| $C(Acc) \times 2^{-(n+1)}$ to Acc | $R n$ |
| $C(Acc) \times 2^{n+1}$ to Acc | $L n$ |
| IF $(CAcc) < 0$, transfer control to n ; if $C(Acc) \geq 0$, ignore (i.e., proceed serially) | $G n$ |
| Read next character on input mechanism into n | $I n$ |
| Send $C(n)$ to output mechanism | $O n$ |

المختصرات:

Acc = المراكم

Acc_1 = النصف الأكثر دلالة من المراكم.

Acc_2 = النصف الأقل دلالة من المراكم.

n = موقع التخزين n .

$C(X)$ = محتويات السجل X (حيث X هو سجل أو موقع تخزين).

يسرد الجدول 1.15 مجموعة تعليمات الآلة من أجل هذا المثال. ويحوي الجدول 2.15 مجموعة كاملة من التعليمات الصغيرة المنجزة في وحدة التحكم والتي يُعبّر عنها رمزياً. ويكفي إذن استخدام 38 تعليمة صغيرة لتعريف النظام تعريفاً كاملاً.

يقدم العمود الأول العنوان الكامل (رقم السطر) لكل تعليمة صغيرة. وتوضع لصاقات على العناوين المقابلة لأرمزة العمليات. لذا، عندما يُصادف رماز عملية تعليمة الجمع (A)، تُنفذ التعليمة الصغيرة في الموقع 5. ويعبّر العمودان 2, 3 عن الأفعال الواجب على وحدة الحساب والمنطق ووحدة التحكم إجراؤها على الترتيب. وينبغي ترجمة كل تعبير رمزي إلى مجموعة من إشارات التحكم (بتات التعليمات الصغيرة). ويتعلق العمودان 4, 5 بإعدادات الرايتين (القلابين) واستخدامهما. فيحدد العمود 4 الإشارة التي تضع القيمة 1 في الراية. وعلى سبيل المثال، يعني الرمز C_s (1) أن الراية ذات الرقم 1 تصبح مساويةً الواحد بواسطة بت الإشارة للعدد المخزن في السجل C. وإذا ضمّ العمود 5 معرفاً للراية، فإن العمودين 6, 7 يحتويان عندئذ على عناوين التعليمات الصغيرة البديلة الواجب استخدامها. وإلا، فإن العمود 6 يعين عنوان التعليمة الصغيرة التالية الواجب جلبها.

تؤلف التعليمات من 0 إلى 4 دورة الجلب. وتقدم التعليمة الصغيرة 4 رماز العملية إلى مفكك الترميز، الذي يولّد بدوره عنوان التعليمة الصغيرة المقابل لتعليمة الآلة الواجب جلبها. ويستطيع القارئ استنتاج العمل الكامل لوحدة التحكم بدراسة الجدول 2.15 دراسة متأنية.

الجدول 2.15: التعليمات الصغيرة لمثال ويلكس

تمثل الرموز A, B, C السجلات المختلفة في وحدتي سجل التحكم والحسابية وتشير العملية C to D إلى أن دارات الابتدال تربط خرج السجل C بدخل السجل D، كما تدل العملية (D+A) to C على أن سجل الخرج A مرتبط بأحد مدخلي وحدة الجمع، ويربط خرج السجل D دوماً بالمدخل الآخر لها. كما يرتبط خرج الجامع بالسجل C.

ويرمز العدد n بين علامتي الاقتباس (مثلاً 'n') إلى المصدر الذي خرج به العدد n مقدراً بواحدات الرقم الأقل دلالة.

| التعليمة الصغيرة التالية | | قلاب شرطي | | وحدة سجل التحكم | الوحدة الحسابية |
|--------------------------|----|-----------|-----|-----------------|-----------------------------|
| 0 | 1 | Set | Use | | |
| | | | | F to G and E | 0 |
| | | | | (G to '1') to F | 1 |
| | | | | Store to G | 2 |
| | | | | G to E | 3 |
| | | | | E to decoder | 4 |
| | | | | | - |
| A | 5 | | | C to D | 16 |
| S | 6 | | | C to D | 17 |
| H | 7 | | | Store to B | 0 |
| V | 8 | | | Store to A | 27 |
| T | 9 | | | C to Store | 25 |
| U | 10 | | | C to Store | 0 |
| R | 11 | | | E to G | 19 |
| L | 12 | | | E to G | 22 |
| G | 13 | | | E to G | (1)C _s 18 |
| I | 14 | | | Input to Store | 0 |
| O | 15 | | | Store to Output | 0 |
| | 16 | | | (D+Store) to C | 0 |
| | 17 | | | (D-Store) to C | 0 |
| | 18 | | 1 | | 0 1 |
| | 19 | | | D to B (R)* | (G-'1') to E 20 |
| | 20 | | | C to D | (1)E _s 21 |
| | 21 | | 1 | D to C (R) | 11 0 |
| | 22 | | | D to C(L)† | (G - '1') to E 23 |
| | 23 | | | B to D | (1)E _s 24 |
| | 24 | | 1 | D to B (L) | 12 0 |
| | 25 | | | '0' to B | 26 |
| | 26 | | | B to C | 0 |
| | 27 | | | '0' to C | '18' to E 28 |
| | 28 | | | B to D | E to G (1)B ₁ 29 |
| | 29 | | | D to B (R) | (G - '1') to E 30 |
| | 30 | | 1 | C to D (R) | (2) E _s 31 32 |
| | 31 | | 2 | D to C | 28 33 |
| | 32 | | 2 | (D + A) to C | 28 33 |
| | 33 | | | B to D | (1)B ₁ 34 |
| | 34 | | | D to B (R) | 35 |
| | 35 | | 1 | C to D (R) | 36 37 |
| | 36 | | | D to C | 0 |
| | 37 | | | (D - A) to C | 0 |

*: الإزاحة يمينا: تُرتَّب دارات الابتدال في الوحدة الحسابية بحيث يوضع الرقم الأقل دلالة من السجل C في المكان الأكثر دلالة من السجل B أثناء العمليات الصغيرة المقابلة للإزاحة يمينا. ويكرر الرقم الأكثر دلالة للسجل C (خانة الإشارة) لتصحيح الأعداد السالبة.

†: الإزاحة يساراً: تُرتَّب دارات الابتدال على نحو مماثل لتمرير الرقم الأكثر دلالة من السجل B إلى المكان الأقل دلالة في السجل C أثناء العمليات الصغيرة المقابلة للإزاحة يساراً.

الميزات والمساوئ Advantages and Disadvantages

تُعد الميزة الأساسية لاستخدام البرمجة الصغيرة لتنفيذ وحدة تحكم هي تبسيط تصميم وحدة التحكم. ولذا يكون التنفيذ أقل كلفة وأقل عرضة للأخطاء وينبغي أن تتضمن وحدة التحكم المبنية عتادياً دارات منطقية معقدة للتتالي عبر العمليات الصغيرة العديدة في دورة التعليم. في حين أن مفككات الترميز ووحدة التتالي المنطقية لوحدة تحكم مبرمجة صغيراً هي أجزاء منطقية شديدة البساطة.

تعاني الوحدة المبرمجة صغيراً من كونها أبطأ مقارنة بوحدة مبنية عتادياً ذات تقانة مماثلة. وعلى الرغم من ذلك، تُعد البرمجة الصغيرة المهيمنة على تنفيذ وحدات التحكم في المعالجات CISC الحديثة بسبب سهولة تنجيزها. أما المعالجات RISC، ذات صيغة التعليمات الأبسط، فهي تستخدم عادة وحدات تحكم مبنية عتادياً. نعرض الآن مقارنة البرمجة الصغيرة عرضاً أشد تفصيلاً.

2.15 تتالي التعليمات الصغيرة Microinstruction sequencing

إن المهتمين الأساسيين لوحدة تحكم مبرمجة صغيراً هما:

- تتالي التعليمات الصغيرة: ويهدف إلى الحصول على التعليمات الصغيرة التالية من ذاكرة التحكم.
 - تنفيذ التعليمات الصغيرة: ويهدف إلى توليد إشارات التحكم اللازمة لتنفيذ التعليمات الصغيرة.
- ينبغي، عند تصميم وحدة تحكم، أخذ هاتين المهمتين معاً في الحسبان، بسبب تأثيرهما على صيغة التعليمات الصغيرة، وتوقيت وحدة التحكم. نركز في هذا المقطع على مهمة التتالي، ولن نتطرق إلى موضوع التوقيت والصيغة إلا بالقدر اليسير، إذ سيتناول المقطع التالي هذين الموضوعين بالتفاصيل.

اعتبارات تصميمية Design Considerations

يدخل في تصميم تقنية تتالي التعليمات الصغيرة اعتباران وهما: حجم التعليمات الصغيرة وزمن توليد العنوان. إن الاعتبار الأول جلي، إذ تقلص كلفة ذاكرة التحكم بتقلص حجم التعليمات. أما الاعتبار الثاني فيعبر ببساطة عن الرغبة في تنفيذ التعليمات الصغيرة بأسرع قدر ممكن.

يقع عنوان التعليمات الصغيرة التالية الواجب تنفيذها، عند تنفيذ البرنامج الصغير، في أحد الأصناف التالية:

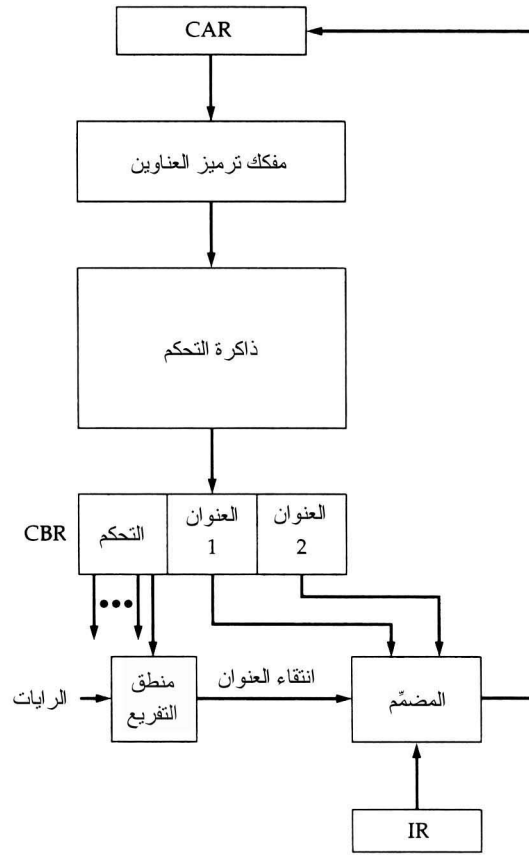
• يمكن أن يكون محدداً بسجل التعليمات

• يمكن أن يكون العنوان التالي متابعياً

• يمكن أن يكون عنوان تفرع.

يحدث النوع الأول مرةً كل دورة تعليمية، بعد جلب التعليمات مباشرة. أما النوع الثاني، فهو الأكثر شيوعاً في معظم التصميمات. ولكن لا يمكن استمثال التصميم من أجل النفاذ المتتابع فقط. فالتفريعات الشرطية واللاشرطية هي جزء ضروري من البرنامج الصغير. أضف إلى ذلك تسعى تتاليات التعليمات الصغيرة لأن

تكون قصيرة. وتصل نسبة تعليمات التفريع إلى $\frac{1}{3}$ أو $\frac{1}{4}$ من إجمالي التعليمات الصغيرة [SIEW 82]. ولذا، فمن المهم تصميم تقنيات مترابطة وفعالة زمنياً لتحقيق تعليمات التفريع الصغيرة.



الشكل 6.15: منطق التحكم في التفريع، بوجود حقلي عناوين.

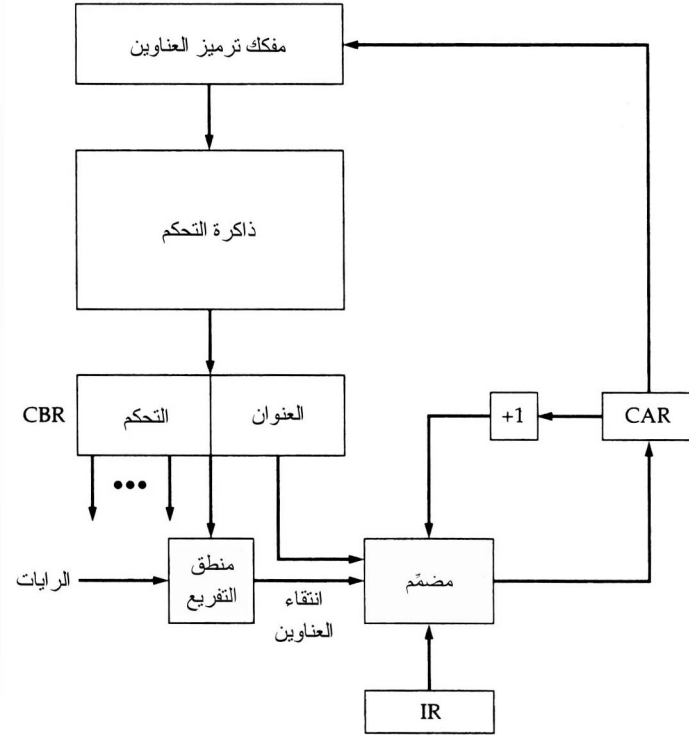
تقنيات التتالي Sequencing Techniques

ينبغي توليد عنوان ذاكرة التحكم اعتماداً على التعليمات الصغيرة الحالية ورايات الشروط ومحتويات سجل التعليمات. ولقد استخدم تنوع واسع من هذه التقنيات، يمكن تصنيفها في ثلاثة أصناف عامة، كما هو موضح في الأشكال 6.15 و 7.15 و 8.15. تعتمد هذه الأصناف على صيغة معلومات العنوان في التعليمات الصغيرة:

- المؤلف من حقلي عناوين
- المؤلف من حقل عناوين وحيد
- الصيغة المتغيرة.

تنص المقاربة الأبسط على توفير حقلي عناوين في كل تعليمات صغيرة. ويقترح (الشكل 6.15) طريقة استخدام هذه المعلومات. يفيد المضمّم multiplexor كواجهة لحقلي العناوين وسجل التعليمات. وهو ينقل رماز العملية أو أحد حقلي العناوين إلى سجل عنوان التحكم CAR تبعاً لدخل انتقاء العناوين. يُفك بعدئذٍ ترميز السجل CAR لتوليد عنوان التعليمات الصغيرة التالية. يقدّم مجتزأ التفريع المنطقية إشارات انتقاء العناوين،

ويعتمد ذلك ١ تترأ في مداخله على رايات وحدة التحكم وجزء التحكم في التعليم الصغرية.

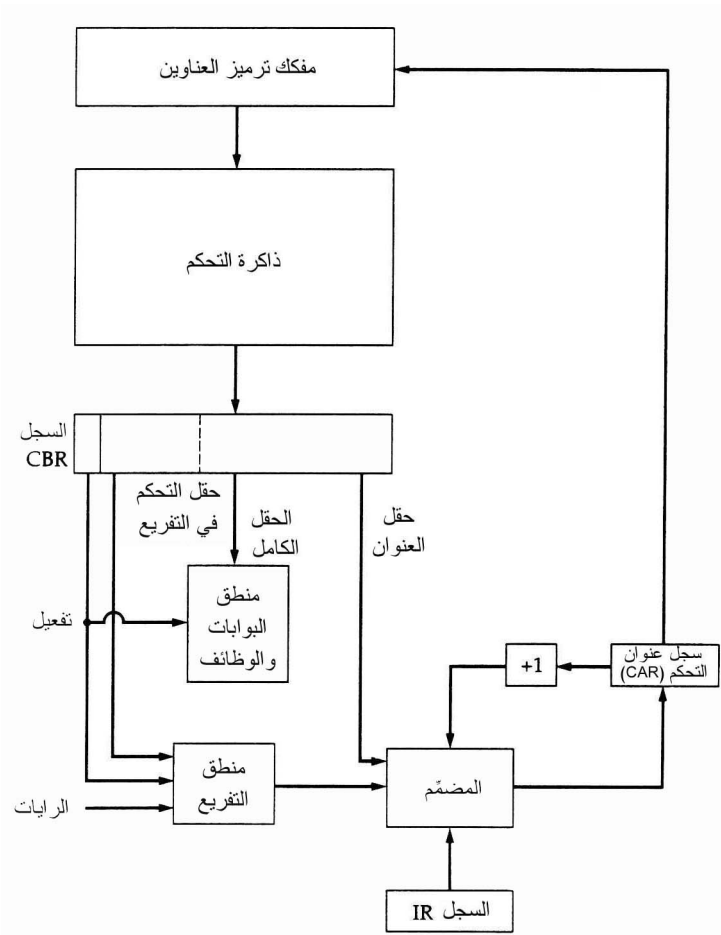


الشكل 7.15: منطق التحكم في التفريع، حقل عنوان وحيد.

وعلى الرغم من بساطة المقارنة ذات العنوانين، إلا أنها تتطلب المزيد من البتات في التعليم الصغرية مقارنةً بمقاربات أخرى. ويمكن تحقيق بعض الوفر بإضافة المزيد من المنطق. وتقوم مقارنة شائعة أخرى باستخدام حقل عنوان وحيد (الشكل 7.15). وتكون خيارات العنوان التالي عندئذٍ كما يلي:

- حقل العناوين
- رماز سجل التعليمات
- العنوان التتابعي التالي

تحدّد إشارات انتقاء العنوان الخيار المنتقى. تقلص هذه المقارنة عدد حقول العنوان، ولكننا نلاحظ أن حقل العنوان لن يُستخدم غالباً. ولذا، تعاني هذه المقارنة من نقص الفعالية في مخطط ترميز التعليمات الصغرية. وتنص مقارنة أخرى على تقديم صيغتي تعليمات صغرية مختلفتين كلياً (الشكل 8.15). يحدّد بت واحد الصيغة المستخدمة منها. ففي إحدى الصيغتين تُستخدم البتات الباقية لتفعيل إشارات التحكم. أما في الصيغة الأخرى، فتفيد بعض البتات في قيادة مجتزأ التحكم في التفريع، وتقدّم البتات الأخرى العنوان المطلوب.



الشكل 8.15: منطق التحكم في التفريع - صيغة متغيرة

ويكون العنوان التالي، في الصيغة الأولى، هو العنوان التالي تتابعياً أو العنوان المستخلص من سجل التعليمات. في حين يُوصّف في الصيغة الثانية التفريع الشرطي أو اللاشرطي. تعاني هذه المقاربة من مشكلة استهلاك دورة كاملة لكل تعليمة تفريع صغيرة، في حين أن توليد العناوين في بقية المقاربات يحدث كجزء من الدورة ذاتها، مثله كمثل إشارات التحكم وهذا ما يقلّص عدد مرات النفاذ إلى ذاكرة التحكم. تتسم المقاربات المذكورة سابقاً بعموميتها، ويؤدي تنجيز إحداها، في أغلب الأحيان، إلى تقنيات مختلفة قليلاً أو إلى مزيج من هذه التقنيات.

توليد العناوين Address Generation

لقد عُرضت مشكلة التالي من وجهة نظر الصيغ والمتطلبات المنطقية العامة. ونعرض هنا هذه المشكلة من وجهة نظر أخرى تتعلق بطريقة حساب أو استنتاج العنوان التالي.

يسرد الجدول 3.15 تقنيات توليد العناوين المتنوعة. يمكن تقسيم تلك التقنيات إلى تقنيات صريحة explicit، يُتاح فيها العنوان صراحةً في التعليمة الصغيرة، وتقنيات مضمرة implicit يتطلب توليد العنوان فيها المزيد من المنطق.

لقد عُرضت إلى الآن تقنيات صريحة. ففي مقارنة حقل العناوين، أُتيح عنوانان في كل تعليمة صغيرة. وعند استخدام صيغة حقل عنوان وحيد أو صيغة متغيرة، يمكن تنجيز عدة تعليمات تفريع. تعتمد تعليمة التفريع

الشرطي على أنواع المعلومات التالية:

الجدول 3.15: تقنيات توليد العناوين في التعليمات الصغيرة

| التقنيات الصريحة | التقنيات المضمرة |
|------------------|---------------------|
| حقلاً عنونة | تقابل |
| تفرع غير مشروط | إضافة |
| تفرع مشروط | تحكم مترسب residual |

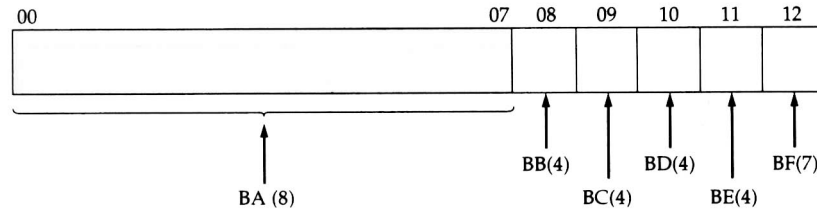
• رايات وحدة الحساب والمنطق

• جزء من رماز العملية، أو حقول العنونة في تعليمات الآلة

• أجزاء من سجل مختار مثل بت الإشارة

• بتات الحالة في وحدة التحكم

وتُستخدم أيضاً عدة تقنيات مضمرة استخداماً شائعاً. ويُحتاج إلى إحدى هذه التقنيات، وهي التقابل mapping، في جميع التصميمات افتراضاً. إذ ينبغي إيجاد تقابل بين جزء من رماز العملية لتعليمات آلة، وعنوان تعليمات صغيرة. ويحدث ذلك مرةً واحدة فقط في كل دورة تعليمات. وهناك تقنية مضمرة شائعة أخرى تتطلب جمع أو إضافة جزأي عنوان لتشكيل عنوان كامل. ولقد اعتمدت هذه المقاربة في عائلة الحواسيب IBM S/360 [Tuck67]، فاستُخدمت في العديد من النماذج S/360. ونأخذ هنا المعالج IBM 3033 مثالاً على ذلك.



الشكل 9.15: سجل عنوان التحكم في النظام IBM 3033

يتألف سجل عنوان التحكم في المعالج IBM3033 من 13 بتاً، كما هو موضح في (الشكل 9.15). يُميز في ذلك السجل جزآن للعناوين. لا تتغير عادةً العناوين ذات المرتبة العليا (00-07) من دورة تعليمات صغيرة إلى أخرى. إذ تُنسخ تلك البتات مباشرةً، عند تنفيذ تعليمات صغيرة ما، من حقل ثماني البتات في التعليمات الصغيرة (BA) إلى البتات الثمانية العليا في سجل عنوان التحكم. يعرف ذلك كتلة مؤلفة من 32 تعليمات صغيرة في ذاكرة التحكم. أما البتات الخمسة الباقية في سجل عنوان التحكم فهي تعين عنوان التعليمات الصغيرة التالية المراد جلبها. يُحدّد كل بت منها بحقل ذي 4 بتات (وهناك استثناء واحد يكون فيه هذا الحقل مؤلفاً من 7 بتات) في التعليمات الصغيرة الحالية. ويحدّد الحقل شرط وضع البت المقابل على القيمة 1. فمثلاً، قد يأخذ بت في سجل عنوان التحكم القيمة 1 أو 0 تبعاً لحدوث حمل في عملية وحدة الحساب والمنطق الأخيرة.

تسمى المقاربة الأخيرة المعروضة في الجدول 3.15 التحكم المترسب residual control، لأنها تستخدم عنوان تعليمات صغيرة مخزناً سابقاً في موقع مؤقت في وحدة التحكم. فعلى سبيل المثال، تُزوّد بعض مجموعات التعليمات الصغيرة بإمكانات مساقات فرعية، ويُستخدم عندئذ سجل داخلي أو مكس سجلات لتخزين

عناوين العودة. ونعرض فيما يلي مثلاً على ذلك، وهو المعالج LSI-11.

تتالي التعليمات الصغيرة في المعالج LSI-11

يُعدّ المعالج LSI-11 نسخة الحاسوب الصغري للحاسوب الرئيس PDP-11، بحيث وضعت المكونات الأساسية للنظام على بطاقة واحدة. يُنجزّ المعالج LSI-11 باستخدام وحدة تحكم مبرمجة صغرياً [SEBE76]. يستخدم المعالج LSI-11 تعليمات صغرية مرمزة على 22 بت، وذاكرة تحكم سعتها 2K كلمة مرمزة على 22 بت. يُحدّد عنوان التعليمات الصغيرة التالية بإحدى الطرق التالية:

- **العنوان التالي** تتابعياً **Next sequential address**: بغياب تعليمات أخرى، يُزاد سجل عنوان التحكم في وحدة التحكم بمقدار 1.

- **تقابل رماز العملية Opcode mapping**: في بداية كل دورة تعليمية، يُحدّد عنوان التعليمات الصغيرة التالية برماز العملية.

- **إمكانات المسافات الفرعية Subroutine facility**: وتُشرح لاحقاً.

- **اختبار المقاطعات Interrupt testing**: تتطلب بعض التعليمات الصغيرة اختبار المقاطعات. فإذا حدثت مقاطعة ما، فإن ذلك سيحدّد عنوان التعليمات الصغيرة التالية.

- **التفرع Branch**: تُستخدم تعليمات صغرية للتفرع الشرطي واللاشرطي.

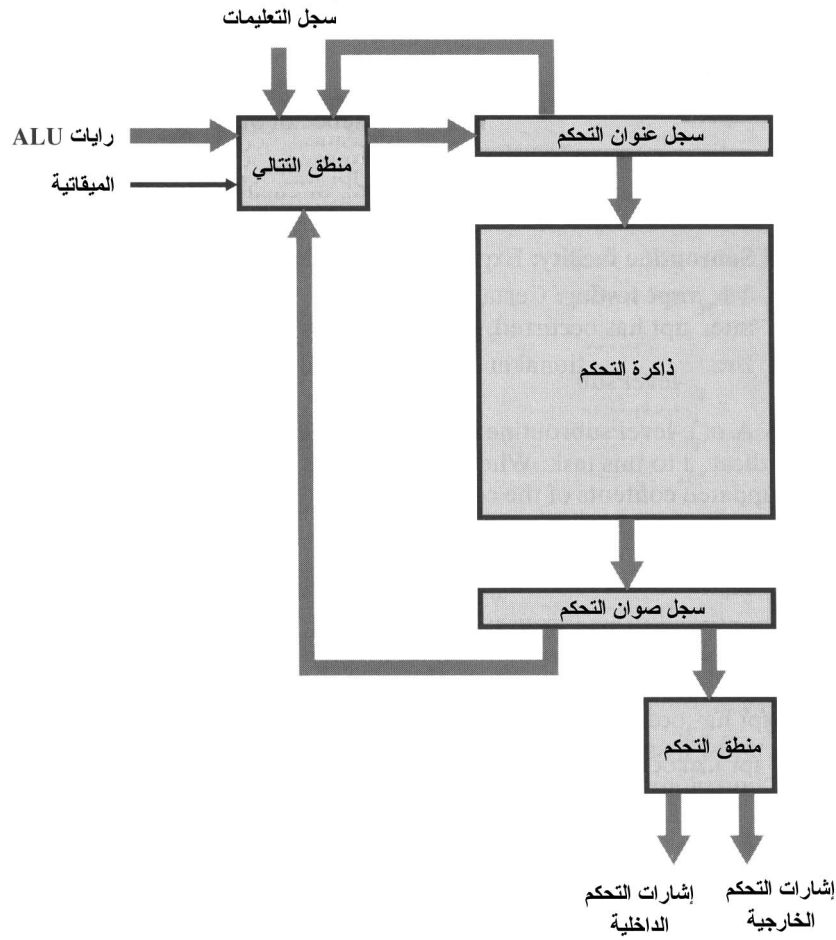
يتوفر في هذا الحاسوب إمكانات مساق فرعي بمستوى واحد. ويُكرّس في كل تعليمات صغرية واحد بت واحد لهذه المهمة. فعندما يوضع في البت القيمة 1، يُحمّل سجل عودة، مرمّز على 11 بت، بمحتويات سجل عنوان التحكم المحددة. تسبّب تعليمات صغرية لاحقة، وهي تعليمات العودة، تحميل سجل عنوان التحكم بمحتويات سجل العودة.

تُعدّ العودة شكلاً من تعليمات التفرع اللاشرطي. ويسبّب شكل آخر للتفرع اللاشرطي تحميل سجل عنوان التحكم بمحتويات حقل ذي 11 بت في التعليمات الصغيرة. تستخدم تعليمات التفرع الشرطي رماز اختبار ذي 4 بتات ضمن التعليمات الصغيرة. ويعيّن ذلك الرماز رايات وحدة الحساب والمنطق الواجب اختبارها لاتخاذ قرار التفرع. فإذا لم يكن الشرط محققاً، يُنتقى العنوان التالي تتابعياً. وإذا تحقق الشرط، تُحمّل البتات الثمانية ذات المرتبة الدنيا في سجل عنوان التحكم، بمحتويات ثمانية بتات من التعليمات الصغيرة. وهذا ما يسمح بالتفرع ضمن صفحة ذاكرة طولها 256 كلمة.

يتضمن إذن الحاسوب LSI-11 إمكانات تتالي عناوين فعالة في وحدة التحكم. ويسمح ذلك للمبرمج الصغري بمرونة كبيرة، ويسهّل مهمة البرمجة الصغيرة. ومن جهة أخرى، تتطلب هذه المقارنة المزيد من المنطق في وحدة التحكم، مقارنةً بالمقاربات ذات الإمكانات الأبسط.

3.15 تنفيذ التعليمات الصغيرة Microinstruction Execution

إن دورة التعليمات الصغيرة هي الحدث الأساسي في المعالج المبرمج صغرياً. تتألف كل دورة من جزأين: الجلب والتنفيذ، يجري في الجلب توليد عنوان التعليمات الصغيرة، ولقد عُرض ذلك في المقطع السابق. أما هذا المقطع فيتناول تنفيذ التعليمات الصغيرة.



الشكل 10.15: تنظيم وحدة التحكم

لنتذكر هنا أن أثر تنفيذ تعليمة صغيرة هو توليد إشارات التحكم. تتحكم بعض هذه الإشارات في نقاط داخلية للمعالج. أما الإشارات الباقية فهي تتجه نحو مسرى التحكم الخارجي أو أي واجهة ترابط خارجية أخرى. ويتحدد إذن عنوان التعليمة الصغيرة التالية كوظيفة عارضة تؤديها وحدة التحكم. يقترح الوصف السابق تنظيم وحدة التحكم على النحو المذكور في (الشكل 10.15). ويُعد هذا التنظيم نسخة منقّحة (للكل 4.15) وهذا ما سيركز عليه هذا المقطع، يجب أن يكون الآن عمل معظم المتزات الأساسية في ذلك المخطط واضحاً. يحتوي مجتزأ التتالي المنطقي على المنطق اللازم لأداء الوظائف المناقشة في المقطع السابق. فهو يولّد عنوان التعليمة الصغيرة التالية باستخدام سجل التعليمات ورايات وحدة الحساب والمنطق وسجل عنوان التحكم (لإجراء الزيادة) وسجل صوان التحكم مداخله. يُقدّم المدخل الأخير العنوان الفعلي أو بتات التحكم أو الاثنين معاً هذه المتزات بميقاتية تحدّد توقيت دورة التعليمة الصغيرة. يولّد مجتزأ التحكم المنطقي إشارات التحكم كتابع لبعض البتات في التعليمة الصغيرة. ولا يخفى أن صيغة التعليمة الصغيرة ومحتواها يحدّدان تعقيد مجتزأ التحكم المنطقي.

تصنيف التعليمات الصغيرة A Taxonomy of Microinstructions

يمكن تصنيف التعليمات الصغيرة وفق طرق متعددة، نذكر منها ما يشيع استخدامه في الأدبيات، وهي:

- التعليمات الأفقية / الشاقولية vertical/horizontal.
- التعليمات المخزومة / غير المخزومة Packed/unpacked.
- البرمجة الصغرية العتادية / البرمجة Hard/soft microprogramming.
- الترميز المباشر / غير المباشر Direct/indirect encoding.

تتعلق كل الطرق السابقة بصيغة التعليمات الصغرية. ولم يُستخدم أي مصطلح من المصطلحات السابقة استخداماً متسقاً ودقيقاً في جميع الأدبيات. ولكن يفيد فحص أزواج الصفات السابقة في إيضاح البدائل الممكنة في تصميم التعليمات الصغرية. ننظر في الفقرات اللاحقة في موضوع التصميم الجوهرى، والمتعلق بجميع هذه المميزات، ثم نعرض المفاهيم المرتبطة بكل زوج من المميزات.

يُؤد كل بت في التعليمات الصغرية، في مقترح ويلكس الأساسي [WILK51]، مباشرة إشارة تحكم أو بتاً في العنوان التالي. ولقد رأينا في المقطع السابق إمكانية اعتماد مخططات تنالي أعقد، واستخدام عدد بتات أقل في التعليمات الصغرية. يتطلب ذلك مجتزأ تنال منطقي أعقد. ونجد نوعاً مشابهاً للحلول الوسطى هذه في جزء التعليمات الصغرية المتعلقة بإشارات التحكم. إذ يمكن، بترميز معلومات التحكم وفكها لاحقاً عند توليد إشارات التحكم، توفير عدد البتات في كلمة التحكم.

كيف يمكن إجراء ذلك الترميز؟ افترض للإجابة على ذلك أن هناك K إشارة تحكم داخلية وخارجية مختلفة، ينبغي على وحدة التحكم قيادتها. ويُكرس، وفق مخطط ويلكس، K بت في التعليمات الصغرية لهذا الغرض وهذا ما يسمح بوجود 2^K تركيبة مختلفة لإشارات التحكم أثناء أي دورة تعليمية. ويمكن الحصول على حل أفضل بملاحظة عدم استخدام جميع التراكيب الممكنة. نقدّم فيما يلي بعض الأمثلة على ذلك:

- لا يمكن تفعيل ربط مصدرين بالوجهة ذاتها في آن واحد (أي C_2 و C_8 في الشكل 5.14).
- لا يمكن لسجل أن يكون مصدراً ووجهة معاً (أي C_5 و C_{12} في الشكل 5.14).
- لا يمكن تقديم أكثر من نمط pattern واحد من إشارات التحكم إلى وحدة الحساب والمنطق في آن واحد.

• لا يمكن تقديم أكثر من نمط واحد من إشارات التحكم إلى مسرى التحكم الخارجى في آن واحد. ولذا، يمكن تحديد جميع تراكيب إشارات التحكم المسموح بها من أجل معالج معين، وهذا ما يجعل عددها $2^K > Q$. ترمز تلك التراكيب باستخدام عدد من البتات قدره: $\log_2 Q$ ، بحيث $\log_2 Q < K$. ويكون ذلك الشكل الأقصر للترميز الذي يحافظ على جميع تراكيب إشارات التحكم المسموح بها. وفي الواقع، لا تُستخدم تلك الصيغة من الترميز لسببين:

- تصعب برمجة مخطط ويلكس اللامرّز بحذافيره. وتناقش هذه النقطة لاحقاً في هذا الفصل.
- يتطلب تحقيق ذلك مجتزأ تحكم منطقي معقد، ولذا فهو قد يكون بطيئاً.
- يُستعاض عن ذلك ببعض الحلول الوسطى، وهي نوعان:
- يُستخدم عدد أكبر تماماً من عدد البتات اللازم لترميز جميع التراكيب الممكنة.
- يتعذر ترميز بعض التراكيب، التي تكون مسموحة من الناحية المادية.

يؤدي ذلك النوع الأخير من الحلول الوسطى إلى تقليص عدد البتات. نستخدم في نهاية المطاف أكثر من $\log_2 Q$ بت. سنناقش في الفقرة التالية بعض تقنيات الترميز المحددة، في حين يتطرق الجزء الباقي في هذا المقطع

إلى تأثيرات الترميز والطرق المختلفة المستخدمة لوصفه.

ونرى، اعتماداً على ما سبق، أن لجزء إشارات التحكم في صيغة التعليمات الصغيرة طيفاً ممتداً. ففي إحدى نهايتي الطيف، يكون لكل إشارة تحكم بت واحد، وفي النهاية الأخرى، تُستخدم صيغة عالية الترميز. يظهر الجدول 4.15 أن المميزات الأخرى لوحدة تحكم مبرمجة صغيراً تقع أيضاً ضمن طيف ذي امتداد يتغير تبعاً لدرجة الترميز المعتمدة.

يبدو الزوج الثاني من المميزات، المذكورة في الجدول، واضحاً إلى حد ما. إذ يتطلب تطبيق مخطط ويلكس بكل حذفه العدد الأكبر من البتات. ويتضح إذن أن هذه النهاية تمثل المرأى الأشد تفصيلاً للعتاد، إذ تكون كل إشارة تحكم قابلة للتحكم من المبرمج الصغيري على نحو إفرادي. ويُجرى الترميز بحيث يتألف مع الوظائف أو الموارد، وهذا ما يسمح للمبرمج الصغيري برؤية المعالج بمستوى أعلى، أقل تفصيلاً. أضف إلى ذلك، إن الترميز يُصمم لتسهيل البرمجة الصغيرة. ولا تخفى صعوبة الإدراك الكامل لإشارات التحكم وتنظيم استخدامها. وكما ذكر آنفاً، يفيد الترميز، فيما يفيد، في تجنب استخدام بعض التراكيب غير المسموح بها.

عرضت الفقرة السابقة تصميم التعليمات الصغيرة من وجهة نظر المبرمج الصغيري. ولكن يمكن رؤية درجة الترميز اللازمة من حيث تأثيراتها العتادية. ففي حالة الصيغة غير المرمزة على الإطلاق، قد يُحتاج إلى القليل من منطق فك الترميز، وقد لا يُحتاج إليه نهائياً، إذ يرتبط كل بت بإشارة تحكم واحدة. أما عند استخدام المزيد من مخططات الترميز المتراصة والائتلافية، فيُحتاج إلى منطق أعقد فأعقد. ويؤثر ذلك بدوره على الأداء، بسبب الحاجة إلى وقت أكبر لانتشار الإشارات عبر البوابات في مجتزأ التحكم المنطقي الأكثر تعقيداً. ولذا، يتطلب تنفيذ التعليمات الصغيرة المرمزة مدة أطول من التعليمات غير المرمزة.

تقع جميع المميزات المذكورة في الجدول 4-15 ضمن طيف من استراتيجيات التصميم. وفي الحالة العامة، عندما يقع تصميم ما إلى يسار الطيف، فهذا يعني أنه يهدف إلى استمثال أداء وحدة التحكم. أما التصميمات الواقعة إلى اليمين، فهي تهتم في استمثال عملية البرمجة الصغيرة. وفي الواقعة، تشبه مجموعات التعليمات الصغيرة الواقعة قرب النهاية اليمنى للطيف مجموعات تعليمات الآلة إلى حد بعيد. ومثال ذلك تصميم الحاسوب LSI-11، المذكور لاحقاً في هذا المقطع. وعندما نهدف إلى تنجز وحدة تحكم فحسب، يقترب التصميم من نهاية الطيف اليسرى. ويقع الحاسوب IBM 3033، المذكور آنفاً، ضمن هذا الصنف. تسمح بقية النظم، كما سنرى لاحقاً، لمستخدمين متنوعين بإنشاء برامج صغيرة مختلفة باستخدام إمكانات التعليمات الصغيرة ذاتها. ومن المحتمل، في مثل هذه الحالات، أن يقع التصميم قرب نهاية الطيف اليمنى.

الجدول 4.15: طيف التعليمات الصغيرة

| المميزات | |
|--|------------------------------|
| تعليمات غير مرمزة | عالية الترميز |
| بتات عديدة | بتات قليلة |
| مرأى مفصّل للعتاديات | مرأى متراس للعتاديات |
| صعبة البرمجة | سهلة البرمجة |
| استثمار كامل التساير | عدم الاستثمار الكامل للتساير |
| الحاجة إلى قليل من المنطق أو عدم الحاجة إلى ذلك مطلقاً | الحاجة إلى منطق معقد |
| تنفيذ سريع | تنفيذ بطيء |
| أداء مستمثل | برمجة مستمثلة |

المصطلحات

| | |
|----------------|------------|
| تعليمات مخزومة | غير مخزومة |
| تعليمات أفقية | شاقولية |
| برمجة عتادية | برمجة |

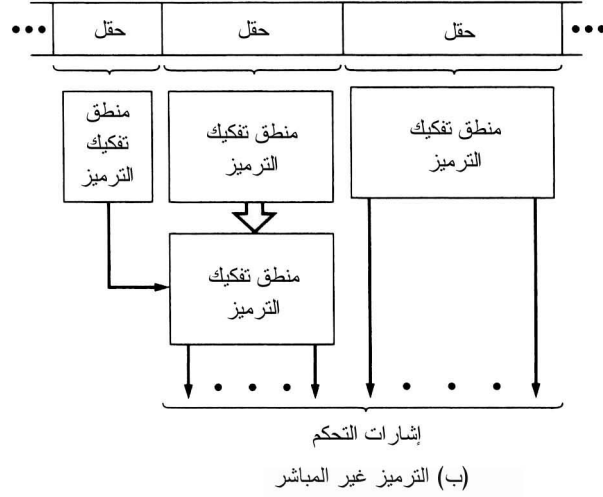
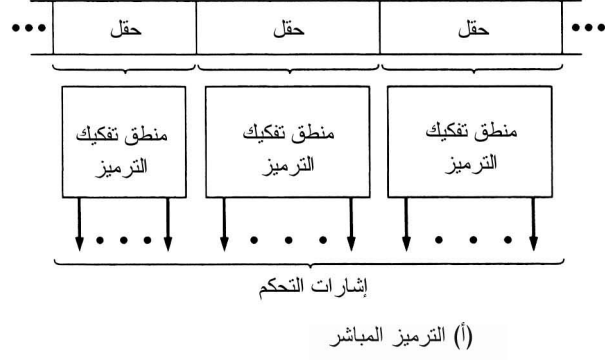
يمكن الآن التعامل مع بعض المصطلحات المذكورة آنفاً. يشير الجدول 4.15 إلى ثلاثة أزواج من المصطلحات المتعلقة بطيف التعليمات الصغيرة. ويصف جميع هذه الأزواج أساساً الأمر ذاته، ولكن كلاً منها يركز في مميزات تصميم مختلفة.

تتعلق درجة التحزيم Packing بدرجة التطابق بين مهمة تحكم معينة وبتات التعليمات الصغيرة المحددة. وعندما تصبح البتات مخزومة أكثر، فإن العدد المعطى للبتات يحوي المزيد من المعلومات وهكذا فإن التحزيم يفيد ضمناً بالترميز. أما المصطلحات "تعليمات أفقية". "وتعليمات شاقولية" فيتعلقان بالعرض النسبي للتعليمات الصغيرة. ويقترح المرجع [SIEW 82] قاعدة تجريبية، تنص على أن للتعليمات الصغيرة الشاقولية طولاً يتراوح بين 10 و 40 بت أما التعليمات الصغيرة الأفقية فلها طول يتراوح بين 40 و 100 بت. ويدل المصطلحان "برمجة صغيرة عتادية" و "برمجة صغيرة برمجة" على مدى الالتصاق بإشارات التحكم وبمخطط العتاديات النسقي. ففي الحالة العامة، تكون البرامج الصغيرة العتادية ثابتة ومخزنة في ذاكرة قراءة فقط. أما البرامج الصغيرة البرمجية فهي أقدر على التغيير وفقاً لرغبة المبرمج الصغري.

ويشير زوج المصطلحات الآخر، المذكور في بداية هذه الفقرة، إلى الترميز المباشر وغير المباشر، وهذا ما تعرضه الفقرة التالية.

ترميز التعليمات الصغيرة Microinstruction Encoding

لا تُصمَّم، عملياً، وحدات التحكم المبرمجة صغرياً باستخدام صيغة تعليمات صغري أفقية كلياً، أي غير مرمزة على الإطلاق. يُستخدم على الأقل نوع من الترميز لتقليص عرض ذاكرة التحكم، ولتبسيط مهمة البرمجة الصغيرة. تُوضح التقنية الأساسية للترميز في (الشكل 11.15. أ)، إذ تُنظَّم التعليمات الصغيرة كمجموعة حقول. يتضمن كل حقل رمزاً يفعل إشارة تحكم واحدة أو أكثر عند فك ترميزه.



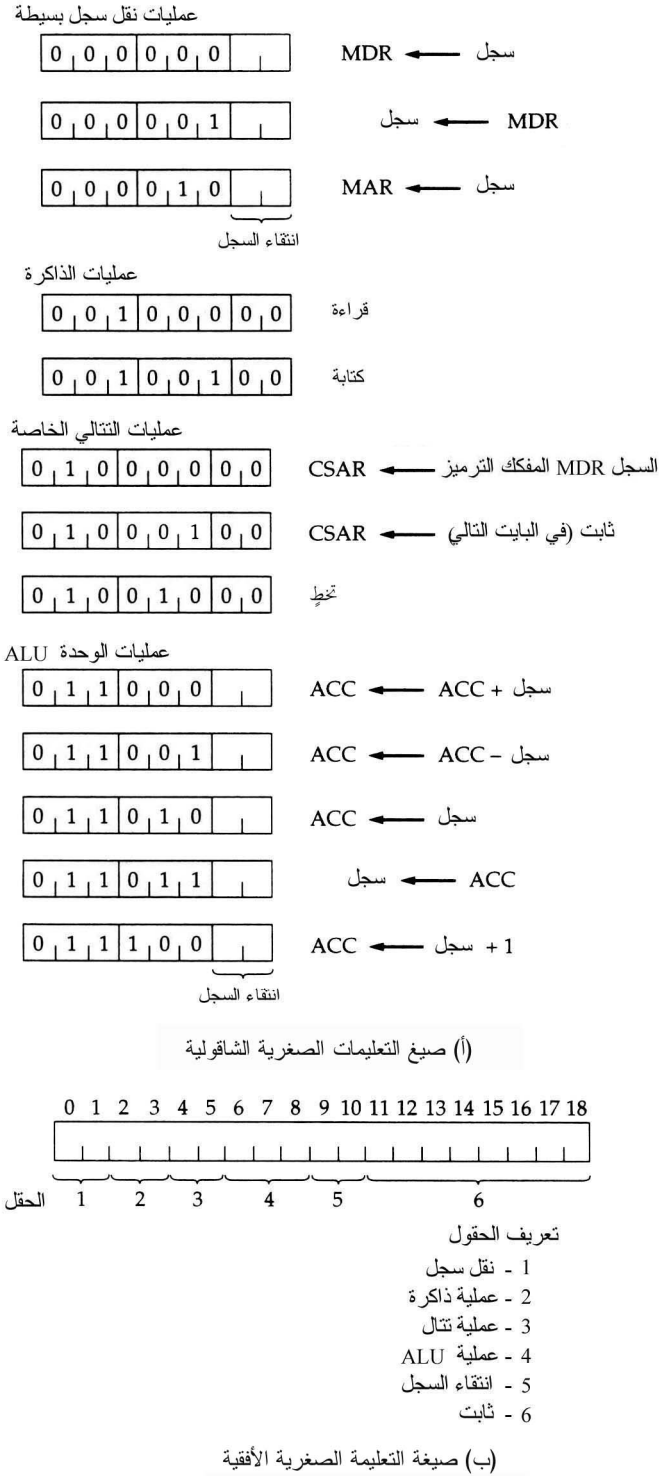
الشكل 11.15: ترميز التعليمات الصغيرة

لنفحص الآن نتائج هذا المخطط النسقي. يفك، عند تنفيذ تعليمة صغيرة ترميز كل حقل وتُولد إشارات التحكم. ولذا، يُحدّد بواسطة N حقل عدد من الأفعال الآتية، مقداره N ، يفعل كل منها إشارة تحكم واحدة أو أكثر. وفي الحالة العامة، ولكن ليس دائماً، نرغب في تصميم الصيغة بحيث لا تُفعل أي إشارة تحكم بأكثر من حقل واحد. وتتضح إذن ضرورة توفر إمكانية لتفعيل أي إشارة تحكم بحقل واحد على الأقل.

لنفحص الآن الحقل الإفرادي. يمكن لحقل مؤلف من L بت أن يحوي 2^L فك رماز، يرمز كل منها ليقابل نمطاً معيناً من إشارات التحكم. ولما كان من غير الممكن ظهور أكثر من رماز واحد في الحقل في لحظة ما، هذه الأرمزة تكون ذات استبعاد متبادل *mutually exclusive*. وتكون لذلك الأفعال التي تولدها ذات استبعاد متبادل.

يمكن الآن وصف تصميم صيغة التعليمات الصغيرة المرمزة بكلمات بسيطة:

- ينبغي تنظيم الصيغة بحقول مستقلة، أي يحدّد كل حقل مجموعة أفعال (أي أشكال نمطية لإشارات التحكم) بحيث يمكن للأفعال المقابلة لحقول مختلفة أن تحدث في آن واحد.
- ينبغي تعريف كل حقل بحيث تكون الأفعال البديلة الممكن تعيينها بذلك الحقل ذات استبعاد متبادل. وبكلمات أخرى، يمكن حدوث فعل واحد فقط من الأفعال المحددة بذلك الحقل في لحظة معينة.



الشكل 12.15: صيغ بديلة للتعليمات الصغيرة لآلة بسيطة

يمكن اعتماد إحدى مقاربتين لتنظيم التعليمات الصغيرة المرمزة في حقول: مقارنة وظيفية ومقاربة موارد. تعرّف طريقة الترميز الوظيفي الوظائف في الآلة، وتشير إلى الحقول وفق نوع الوظائف. فمثلاً، إذا كان بالإمكان استخدام مصادر متنوعة لنقل المعطيات إلى المراكز، يمكن عندئذ تخصيص حقل لهذا الغرض، فيعيّن كل رماز مصدراً مختلفاً. أما ترميز الموارد، فهو ينظر إلى الآلة كمجموعة موارد مستقلة، ويخصّص حقلاً لكل منها (كالدخل/الخروج، الذاكرة، وحدة الحساب والمنطق).

يمكن أن يتسم الترميز بأنه مباشر أو غير مباشر (الشكل 11.15. ب). ففي الترميز غير المباشر، يُستخدم حقل واحد لتحديد تفسير حقل آخر. ونفحص، كمثال على ذلك، وحدة حساب ومنطق قادرة على إجراء 8 عمليات حسابية مختلفة أو 8 عمليات إزاحة مختلفة. يُستخدم حقل ذو بت واحد للدلالة على عملية الإزاحة أو العملية الحسابية، ويشير حقل ذو 3 بتات إلى نوع العملية المنفّذة. تتطلب هذه التقنية عموماً مستويي فك ترميز، وهذا من شأنه زيادة مدة تأخير الانتشار propagation delay.

يُعد المثال المذكور في (الشكل 12.15) مثلاً بسيطاً على هذه المفاهيم. فهو يفترض أن لمعالج مراكماً وحيداً وعدة سجلات داخلية، تعدد البرنامج وسجل تخزين مؤقت لدخل وحدة حساب ومنطق. ونجد في (الشكل 12.15. أ). صيغة تعليمات صغيرة شاقولية إلى درجة كبيرة. تشير البتات الثلاثة الأولى إلى نوع العملية. أما البتات الثلاثة التالية فهي ترمز العملية، ويتتقي البتان الأخيران سجلاً داخلياً. يظهر (الشكل 12.15. ب). مقارنة أكثر أفقية، على الرغم من استخدامها الترميز. تظهر في هذه الحالة وظائف متباينة في حقول مختلفة.

تنفيذ التعليمات الصغيرة في المعالج LSI - 11

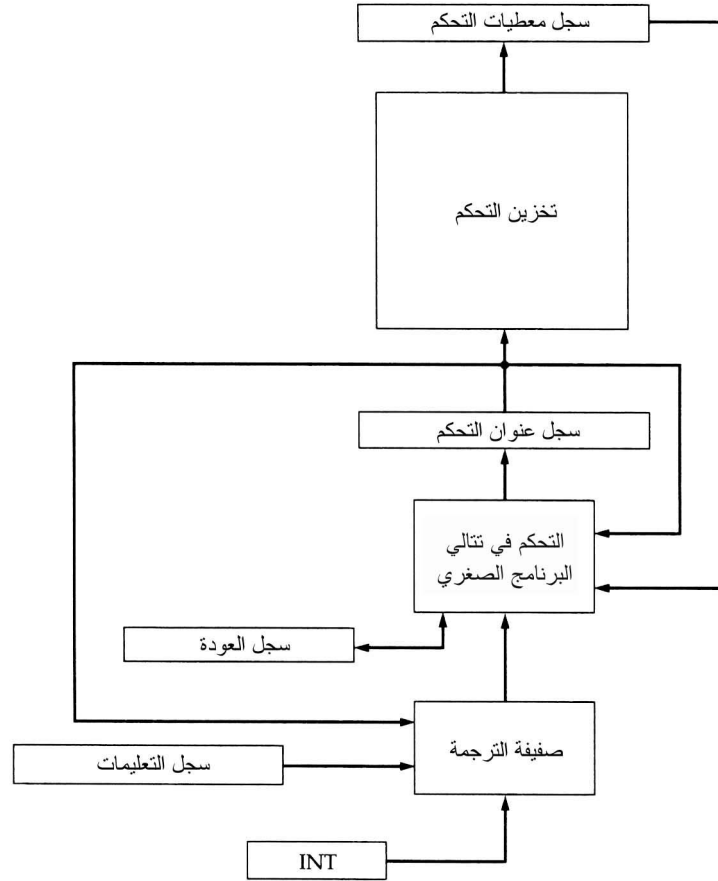
يُعدّ المعالج LSI-11 [SEBE 76] مثلاً جيداً على مقارنة التعليمات الصغيرة الشاقولية. ننظر أولاً إلى تنظيم وحدة التحكم، ثم إلى صيغة التعليمات الصغيرة.

تنظيم وحدة التحكم في المعالج LSI-11

إن المعالج LSI-11 هو العضو الأول في عائلة الحواسيب PDP-11، المقدم تجارياً على أنه معالج وحيد البطاقة single-board processor. تتضمن البطاقة 3 رقاقات من شركة LSI، ومسرى داخلياً يُعرف باسم مسرى التعليمات الصغيرة microinstruction bus MIB، وبعض دارات المنطقية الإضافية لمواجهة الترابط.

يوضح (الشكل 13.15)، ببساطة، تنظيم المعالج LSI-11. الرقاقات الثلاثة هي رقاقات المعطيات وتخزين التحكم. تحوي رقاقة المعطيات وحدة حساب ومنطق ذات 8 بتات، و26 سجلاً ذا 8 بتات، ومواقع تخزين لرايات الشروط المتعددة. يُستخدم 16 سجلاً من هذه السجلات في تنجيز السجلات العامة الاستخدام في الحاسوب PDP-11، وعددها الكلي 8 سجلات، لكل منها 16 بت. أما السجلات الأخرى فتتضمن كلمة حالة البرنامج، وسجل عنوان الذاكرة MAR، وسجل صوان الذاكرة. ولما كانت وحدة الحساب والمنطق تتعامل مع 8 بتات في آن واحد، فإن العمليات الحسابية في الحاسوب PDP-11، والتي تجري على كلمات ذات 16 بت، تتطلب مرورين عبر وحدة الحساب والمنطق. ويتحكم في ذلك برنامج صغير.

تحتوي رقاقة (أو رقاقات) تخزين التحكم ذاكرة تحكم بعرض 22 بت. تحتوي رقاقة التخزين المنطق اللازم لتحقيق التالي وتنفيذ التعليمات الصغيرة. وهي تضم سجل عنوان التحكم وسجل معطيات التحكم ونسخة من سجل تعليمات الآلة.



الشكل 14.15: تنظيم وحدة التحكم في الحاسوب Lsi-11

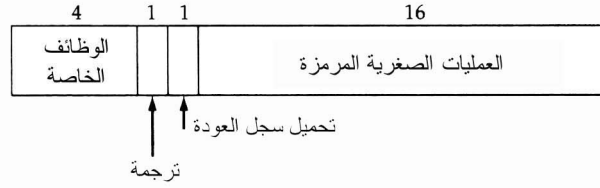
صيغة التعليمات الصغرية في الحاسوب LSI - 11

تُستخدم في المعالج LSI-11 صيغة تعليمات شاقولية إلى أبعد درجة، يبلغ عرض كل منها 22 بتاً. وتشبه مجموعة التعليمات الصغرية مجموعة تعليمات الآلة في الحاسوب PDP-11 شهاً كبيراً. ويهدف التصميم هنا إلى استمثال أداء وحدة التحكم باستخدام تعليمات صغرية مبرمجة بسهولة وشاقولية. يُدرج الجدول 5.15 بعض التعليمات الصغرية للمعالج Lsi-11.

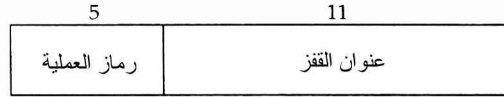
الجدول 5.15: بعض تعليمات الحاسوب Lsi-11 الصغرية

| العمليات الحسابية | إزاحة كلمة (بايت) بميناً مع حمل (أو بدونه) إتمام كلمة (بايت) العمليات العامة نقص كلمة (بايت) قفز قفز شرطي وضع الرايات على القيمة 1 أو 0 تحميل الجزء الأدنى من السجل نقص كلمة نقلاً شرطياً (بايت) عمليات الدخل/الخروج إدخال كلمة (بايت) |
|--|--|
| جمع كلمة (بايت أو رقم) فحص كلمة (بايت أو رقم) زيادة كلمة بمقدار 1 (بايت) زيادة كلمة بمقدار 2 (بايت) نفي كلمة (بايت) زيادة (انقاص شرطية لبايت) جمع شرطي للكلمة (بايت) طرح كلمة (بايت) مع الحمل طرح كلمة (بايت) مقارنة كلمة (بايت، رقم) | |

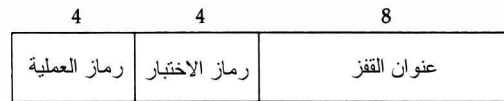
| | |
|---------------------------------|---|
| طرح كلمة (بايت) مع الحمل | إدخال كلمة الحالة (بايت) |
| انقاص كلمة (بايت) بمقدار 1 | قراءة |
| العمليات المنطقية | كتابة |
| عملية and مع كلمة (بايت أو رقم) | قراءة (كتابة) وزيادة الكلمة بمقدار 1 (بايت) |
| فحص كلمة (بايت) | قراءة (كتابة) وزيادة الكلمة بمقدار 2 (بايت) |
| عملية OR مع كلمة (بايت) | قراءة (كتابة) إشعار |
| عملية XOR مع كلمة (بايت) | إخراج كلمة (بايت أو كلمة حالة) |
| كلمة محو بت (بايت) | |



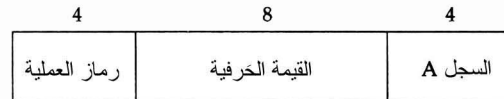
(أ) صيغة التعليمات الصغيرة الكاملة للحاسوب LSI-11



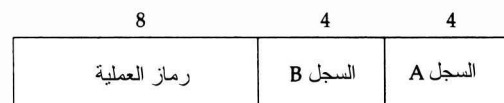
صيغة التعليمات الصغيرة للقفز اللاشرطي



صيغة التعليمات الصغيرة للقفز الشرطي



صيغة التعليمات الصغيرة للقيمة الحرفية



صيغة التعليمات الصغيرة للسجل

(ب) صيغة الجزء المرمز في تعليمات الحاسوب LSI-11 الصغيرة

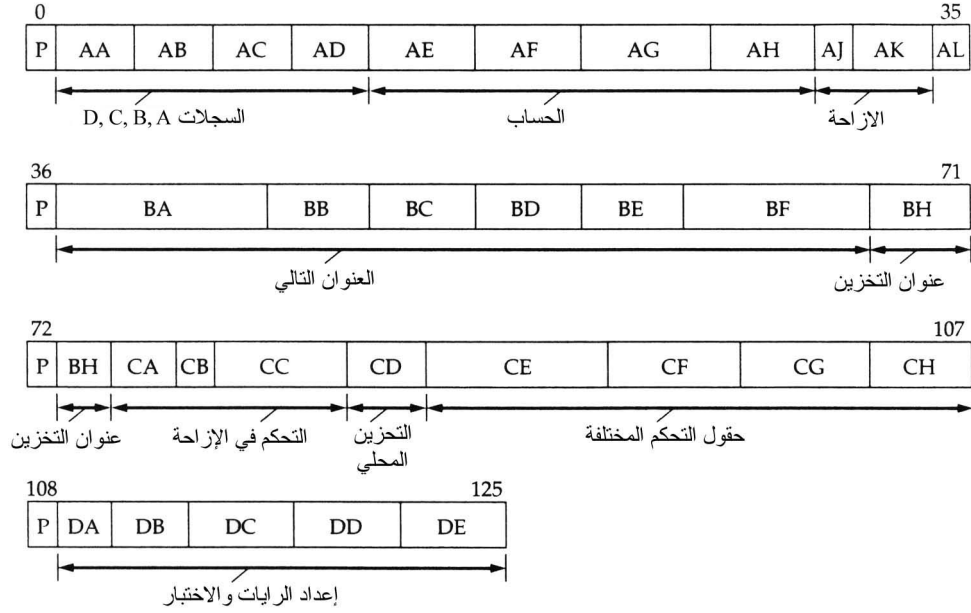
الشكل 15.15: صيغة تعليمات المعالج LSI-11 الصغيرة.

يبين (الشكل 15.15) صيغة التعليمات الصغيرة في المعالج LSI-11، والمُرَمَّزة على 22 بت. تتحكم البتات الأربعة الأعلى في الوظائف الخاصة لبطاقة المعالج. ويسمح بت الترجمة لصيغة الترجمة بفرض المقاطعات المطلوبة. يُستخدم بت تحميل load سجل العودة في نهاية المساق الفرعي للسماح لعنوان التعليمات الصغيرة التالية بأن يُحمَّل من سجل العودة.

تُستخدم البتات المتبقية، وعددها 16، في العمليات الصغيرة البالغة الترميز. تشبه الصيغة المذكورة تعليمات الآلة كثيراً، وهي ذات رماز عملية متغير الطول، بعامل واحد أو أكثر.

تنفيذ التعليمات الصغرية في المعالج IBM 3033

تبلغ سعة ذاكرة التحكم القياسية في الحاسوب IBM 3033 4K كلمة. يحتوي النصف الأول منها (من العنوان 0000 إلى 07FF) على 108 تعليمة صغرية، في حين يُستخدم الجزء الباقي (من العنوان 0800 إلى 0FFF) لتخزين تعليمات صغرية مرمزة على 126 بت. يبين (الشكل 16.15) هذه الصيغة. وعلى الرغم من أن تلك الصيغة أفقية إلى حد ما، إلا إنها تستخدم الترميز استخداماً كبيراً. يلخص الجدول 6.15 الحقول الأساسية في تلك الصيغة.



الشكل 16.15: صيغة التعليمات الصغرية في المعالج IBM 3033

تستخدم وحدة الحساب والمنطق في عملها المداخل القادمة من 4 سجلات مخصصة، غير مرئية للمستخدم، وهي A و B و C و D. تحتوي صيغة التعليمات الصغرية على حقول لتحديد مصدر تحميل هذه السجلات من السجلات المرئية للمستخدم، وإجراء عملية ما في وحدة الحساب والمنطق، وتخزين النتيجة في أحد السجلات المرئية للمستخدم. كما تحتوي الصيغة أيضاً على حقول لتحميل المعطيات في السجلات والذاكرة وتخزينها. نوقشت آلية التتالي في المعالج IBM 3033 في المقطع 2.15.

4.15 المعالج TI-8800

تُعدّ بطاقة تطوير البرمجيات SDB TI-8800، المصنّعة من شركة Texas Instruments، بطاقة حاسوب ذات 32 بتاً، قابلة للبرمجة صغرياً. يتضمن النظام مكان تخزين التحكم، وقابلاً للكتابة، منجزاً بواسطة ذاكرة نفاذ عشوائي RAM عوضاً عن ذاكرة قراءة فقط ROM. لا يحقق مثل هذا النظام سرعة نظام مبرمج صغرياً يخزن التحكم في ذاكرة قراءة فقط ROM، ولا يوازيه في الكثافة. ولكنه يفيد في تطوير النماذج الأولية وفي الأغراض التعليمية.

الجدول 6.15: حقول التحكم في التعليمات الصغرية للمعالج IBM 3033

حقول التحكم في وحدة الحساب والمنطق

| | |
|--------|---|
| AA (3) | تحميل السجل A من أحد سجلات المعطيات. |
| AB (3) | تحميل السجل B من أحد سجلات المعطيات. |
| AC (3) | تحميل السجل C من أحد سجلات المعطيات. |
| AD (3) | تحميل السجل D من أحد سجلات المعطيات. |
| AE (4) | تسيير البتات المحددة من السجل A إلى وحدة الحساب والمنطق. |
| AF (4) | تسيير البتات المحددة من السجل B إلى وحدة الحساب والمنطق. |
| AG (5) | تحديد العملية الحسابية المنفذة على المدخل A في وحدة الحساب والمنطق. |
| AH (4) | تحديد العملية الحسابية المنفذة على المدخل B في وحدة الحساب والمنطق. |
| AJ (1) | تحديد الدخل B أو D المرتبط بوحدة الحساب والمنطق من الجانب B. |
| AK (4) | تسيير الخرج الحسابي إلى سجل الإزاحة. |
| AB (1) | تفعيل سجل الإزاحة. |
| CC (5) | تحديد الوظائف المنطقية ووظائف الحمل. |
| CE (7) | تحديد مقدار الإزاحة. |
| CA (3) | تحميل السجل F. |

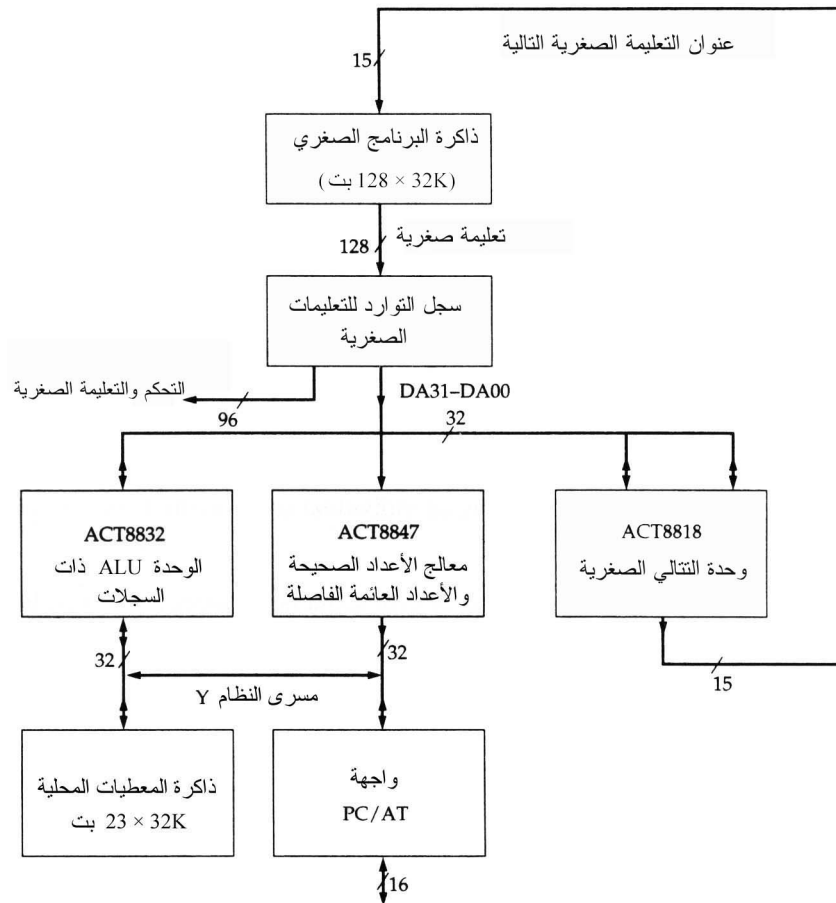
حقول التفريع والتتالي

| | |
|--------|--|
| AL (1) | إنهاء العملية وإجراء التفريع. |
| BA (8) | وضع بتات سجل عنوان التحكم ذات المرتبة العليا (07-00) على القيمة 1. |
| BB (4) | تحديد الشرط اللازم لوضع البت 8 في سجل عنوان التحكم على القيمة 1. |
| BC (4) | تحديد الشرط اللازم لوضع القيمة 1 في البت 9 من سجل عنوان التحكم. |
| BD (4) | تحديد الشرط اللازم لوضع القيمة 1 في البت 10 من سجل عنوان التحكم. |
| BE (4) | تحديد الشرط اللازم لوضع القيمة 1 في البت 11 من سجل عنوان التحكم. |
| BF (4) | تحديد الشرط اللازم لوضع القيمة 1 في البت 12 من سجل عنوان التحكم. |

يتألف النظام SDB Ti800 من المكونات التالية (الشكل 17.15):

- ذاكرة البرنامج الصغري.
- وحدة التتالي الصغرية.
- وحدة الحساب والمنطق ذات سجلات 32 بت.
- معالج عائم الفاصلة، ومعالج أعداد صحيحة.
- ذاكرة معطيات محلية.

يربط مسريان المكونات الداخلية للنظام. يقدم المسرى DA المعطيات الصادرة من حقل معطيات التعليمات الصغرية إلى وحدة الحساب والمنطق، أو معالج الأعداد العائمة الفاصلة أو وحدة التتالي الصغرية. وفي الحالة الأخيرة، تتألف المعطيات من عنوان يُستخدم في تعليمة التفريع. يمكن استخدام المسرى أيضاً لنقل معطيات وحدة الحساب والمنطق، أو وحدة التتالي الصغرية إلى بقية المكونات. يربط مسرى النظام Y وحدة الحساب والمنطق ومعالج الأعداد العائمة الفاصلة بالذاكرة المحلية و تنزآت الخارجية عبر واجهة الحاسوب الشخصي PC. يمكن تطبيق البطاقة في حاسوب شخصي مضيف، متوافق مع الحواسيب IBM، ويُعد الحاسوب المضيف المنصة الملائمة لجميع البرنامج الصغري وتنقيحه.



الشكل 17.5: المخطط الصندوقي للنظام TI8800.

صيغة التعليمات الصغيرة Microinstruction Format

تتضمن صيغة التعليمات الصغيرة للبطاقة SDB-TI 8800 عدداً من البتات مقداره 128 بت، وهي موزعة على 30 حقلاً وظيفياً كما هو مبين في الجدول 7.15. يحوي كل حقْل بتاً واحداً أو أكثر، وتُجمع البتات في خمس فئات أساسية:

- تحكم في البطاقة
- رقاقة معالج الأعداد الصحيحة والأعداد العائمة الفاصلة (8847)

الجدول 7.15: صيغة التعليمات الصغيرة للنظام TI-8800

| رقم الحقل | عدد البتات | الوصف |
|---|------------|---|
| التحكم في البطاقة | | |
| 1 | 5 | انتقاء دخل راية الشرط |
| 2 | 1 | تفعيل / إلغاء تفعيل إشارة طلب الدخل/الخروج |
| 3 | 2 | تفعيل / إلغاء تفعيل عمليات قراءة/كتابة ذاكرة المعطيات المحلية |
| 4 | 1 | تحميل الحالة / عدم تحميل الحالة |
| 5 | 2 | تحديد وحدة قيادة المسرى Y |
| 6 | 2 | تحديد وحدة قيادة المسرى DA |
| رقاقة معالجة الأعداد الصحيحة والعائمة الفاصلة 8847 | | |
| 7 | 1 | التحكم في السجل C: إرسال ميقاتية/ عدم إرسالها |

| | | |
|--|----|----|
| انتقاء البتات الأكثر دلالة أو الأقل دلالة للمسرى Y | 1 | 8 |
| مصدر معطيات السجل C: وحدة الحساب والمنطق/ المضمّم | 1 | 9 |
| انتقاء النمط IEEE أو FAST لوحدة الحساب والمنطق والمضمّم | 4 | 10 |
| انتقاء مصادر عوامل المعطيات: السجلات RA أو RB أو P أو السجل 5 أو C | 8 | 11 |
| التحكم في السجل RB: إرسال ميقانية/ عدم إرسالها | 1 | 12 |
| التحكم في السجل RA: إرسال ميقانية/ عدم إرسالها | 1 | 13 |
| تأكيد مصدر المعطيات | 2 | 14 |
| تأهيل/إلغاء تأهيل سجلات التوارد | 2 | 15 |
| وظيفة وحدة الحساب والمنطق 8847 | 11 | 16 |
| وحدة الحساب والمنطق | | |
| تفعيل إخراج المعطيات/ إلغاء التفعيل إلى السجل المنتقى: النصف الأكثر دلالة أو الأقل دلالة | 2 | 17 |
| انتقاء مصدر معطيات ملف السجلات: المسرى DA أو DB أو خرج المضمّم ALU-Y □ | 2 | 18 |
| مسرى النظام Y | | |
| مغيّر تعليمية الإزاحة | 3 | 19 |
| إجبار/ عدم إجبار الحمل في الدخل | 1 | 20 |
| تحديد نمط تشكيل وحدة الحساب والمنطق: 8 أو 16 أو 32 | 2 | 21 |
| تحديد مدخل المضمّم 5: ملف السجلات، المسرى DB، السجل HQ | 2 | 22 |
| انتقاء مدخل المضمّم R: ملف السجلات، المسرى DA | 1 | 23 |
| انتقاء السجل في الملف C للكتابة. | 6 | 24 |
| انتقاء السجل في الملف B للقراءة | 6 | 25 |
| انتقاء السجل في الملف A للكتابة. | 6 | 26 |
| وظيفة وحدة الحساب والمنطق. | 8 | 27 |
| حقل التالي الصغرية 8818 | | |
| إشارة دخل التحكم إلى الرقاقة 8818 | 12 | 28 |
| حقل المعطيات لذاكرة التخزين القابلة للكتابة WCS | | |
| البتات الأكثر دلالة في حقل المعطيات WCS | 16 | 29 |
| البتات الأقل دلالة في حقل المعطيات WCS | 16 | 30 |

• وحدة الحساب والمنطق ذات السجلات (8832)

• وحدة التالي الصغرية (8812)

• حقل المعطيات لتخزين قابل للكتابة WCS.

يُربط حقل بالمعطيات التخزين القابل للكتابة WCS، وطوله 32 بت، بالمسرى DA الذي يقدم المعطيات إلى وحدة الحساب والمنطق ومعالج الأعداد العائمة الفاصلة ووحدة التالي الصغرية، كما هو مبين في (الشكل 17.15).

أما البتات الأخرى في التعليمية الصغرية، وعددها 96 بت (الحقول من 1 إلى 27)، فهي إشارات تحكم تُربط مباشرةً بالنسيقة المناسبة. ولم تُظهر الروابط الأخرى على (الشكل 17.15) حفاظاً على بساطته. تتعلق الحقول الستة الأولى بالعمليات المرتبطة بالتحكم في البطاقة، ولا تتعلق بالتحكم في المكونات إفرادياً. تتضمن عمليات التحكم ما يلي:

• انتقاء رايات الشروط للتحكم في وحدة التالي. يشير البت الأول في الحقل 1 إلى القيمة المراد وضعها في

راية الشرط (القيمة 0 أو 1)، وتشير البتات الأربعة الباقية إلى الراية الواجب وضعها على الواحد.

- إرسال طلب دخل/خرج إلى الحاسوب PC/AT.
- تفعيل عمليات القراءة/الكتابة في ذاكرة المعطيات المحلية.
- تحديد الوحدة التي تقود مسرى النظام Y ويتبقى أحد المكونات الأربعة المرتبطة بالمسرى (الشكل 17.15).
أما البتات الأخيرة، وعددها 32 بت، فهي تحوي المعلومات الخاصة بتعليمية صغيرة معينة.
تُناقش الحقول الباقية في التعليمية الصغيرة مناقشة أفضل عند عرض التجهيزة التي تتحكم بها تحليل الحقول.
نناقش في بقية هذا المقطع وحدة التتالي الصغيرة ووحدة الحساب والمنطق ذات السجلات، وتنحطى وحدة معالجة الأعداد العائمة الفاصلة، لأنها لا تزودنا بمفاهيم جديدة.

وحدة التتالي Microsequencer

إن الوظيفة الأساسية لهذه الوحدة هي توليد عنوان التعليمية الصغيرة التالية للبرنامج الصغري. يُقدّم هذا العنوان المرمز على 15 بت إلى ذاكرة البرنامج الصغيرة (الشكل 17.15).

يمكن انتقاء العنوان التالي من أحد المصادر الخمسة:

1. سجل عداد البرنامج الصغري (MPC)، المستخدم لتكرار التعليمات (إعادة استخدام العنوان ذاته) والاستمرار في تنفيذ التعليمات (زيادة العنوان بمقدار 1).
 2. المكس، الذي يقبل استدعاءات المساق الفرعي الصغري، والحلقات التكرارية والعودة من المقاطعات.
 3. المنفذان DRA و DRB، اللذان يشكّان ممرين إضافيين من العتاد الخارجي الذي يمكن أن يولد عناوين البرامج الصغرية. يُربط هذان المنفذان إلى النصف الأعلى (16 بت) والأدنى من المسرى DA على الترتيب. ويسمح ذلك لوحدة التتالي الصغيرة بالحصول على عنوان التعليمية التالية من حقل معطيات التخزين القابل للكتابة WCS في التعليمية الصغيرة الراهنة، أو من النتيجة المحسوبة في وحدة الحساب والمنطق ALU.
 4. عدادا السجلات RCA و RCB الممكن استخدامهما كمكان تخزين إضافي للعنوان.
 5. مدخل خارجي على المنفذ Y الثنائي الاتجاه، لدعم المقاطعات الخارجية.
- يُعدّ (الشكل 18.15) مخططاً صندوقياً منطقياً للرفاقاة 8818 ويتألف هذا العنصر من 1 موعات الوظيفية الأساسية التالية:

- عداد برنامج صغري ذو 16 بت (MPC) يضم سجلاً ودارة زيادة.
- عدادا سجلات RCA و RCB لعدّ الحلقات وتخزين عناوين التفرع، أو قيادة التجهيزات الخارجية.
- مكس سعته 65 كلمة عرض كل منها 16 بت، يسمح باستدعاءات المساقات الفرعية في البرنامج الصغري بالمقاطعات.
- سجل عودة من المقاطعات، وتفعيل الخرج Y لمعالجة المقاطعات على مستوى التعليمات الصغيرة.
- مضمّم الخرج Y، يمكن بواسطته انتقاء العنوان التالي من السجلات RCA, MPC, RCB، ومنفذ المسريين الخارجيين DRA و DRB والمكس.

السجلات/العدادات Registers/Counters

يمكن تحميل السجلين RCA و RCB من المسرى DA بواسطة التعليمات الصغيرة الحالية، أو خرج وحدة الحساب والمنطق. يمكن استخدام تلك القيم عدادات للتحكم في انسياب التنفيذ، ويمكن إنقاصها آلياً عند النفاذ إليها. كما يمكن استخدام تلك القيم عناوين تعليمات صغيرة تُقدّم إلى مضمّم الخرج Y. ومن الممكن التحكم في السجلين على نحو مستقل أثناء دورة التعليمات الصغيرة الوحيدة، ويُستثنى من ذلك الإنقاص الآني للسجلين معاً.

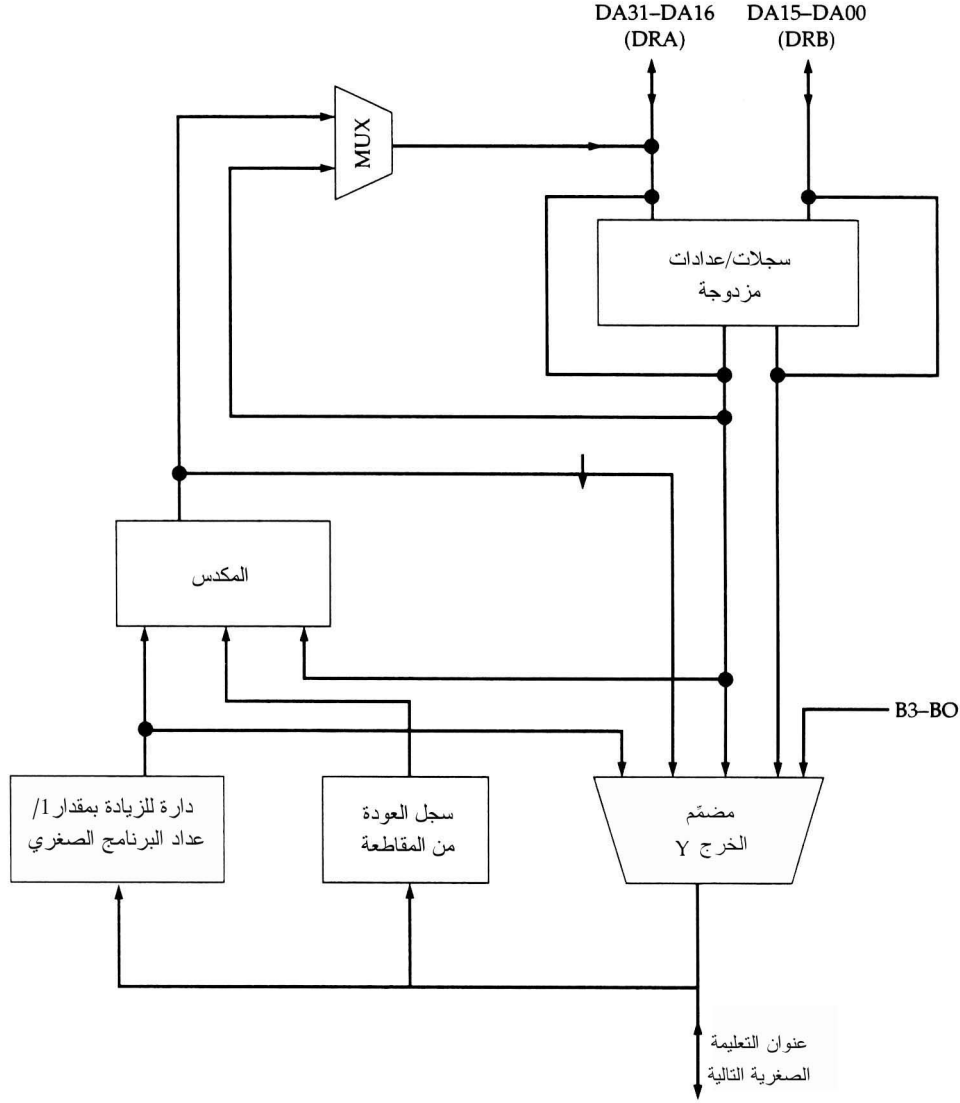
المكدس Stack

يسمح المكدس بعدة مستويات من الاستدعاءات المتداخلة أو المقاطعات، ويمكن استخدامه لدعم التفرع والحلقات. ولبيق ماثلاً في الأذهان أن هذه العملية تشير إلى وحدة التحكم فقط، ولا تشير إلى المعالج الإجمالي، وأن العناوين المستخدمة هي عناوين التعليمات الصغيرة المخزّنة في ذاكرة التحكم. يمكن إجراء 6 عمليات مختلفة في المكدس:

1. المحو، الذي يضع مؤشر المكدس إلى القيمة 0، والذي يؤدي إلى إفراغ المكدس.
2. النزع pop، والذي ينقص مؤشر المكدس.
3. الدفع push، والذي يخزّن محتويات السجل MPC، وسجل العودة من المقاطعة، أو منفذ المسرى DRA في المكدس، ويزيد مؤشر المكدس.
4. القراءة، والتي تجعل العنوان الذي يشير إليه مؤشر القراءة متاحاً عند مضمّم الخرج Y.
5. المسك Hold، ويسبّب الحفاظ على عنوان مؤشر المكدس دون تغيير.
6. تحميل مؤشر المكدس، ويؤدي إلى إدخال البتات الأقل دلالة (وعددتها 7) في المسرى DRA، إلى مؤشر المكدس.

التحكم في وحدة التتالي الصغيرة Control of Microsequencer

يتحكم في وحدة التتالي الصغيرة، أساساً، حقل ذو 12 بت في التعليمات الصغيرة الحالية، وهو الحقل 28 في الجدول 7.15. يتألف هذا الحقل من الحقول الفرعية التالية:



الشكل 18.15: وحدة التتالي الصغرية في النظام TI-8800

- **الحقل OSEL (1 بت):** انتقاء الخرج، ويحدد القيمة التي ستوضع في خرج المضمّم لتغذية المسرى DRA (المبيّن في الزاوية اليسرى العليا في الشكل 18.15). يُنتقى الخرج لربطه بالمكس أو بالسجل RCA. ويفيد عندئذ المسرى DRA كدخل إلى مضمّم الخرج Y أو السجل RCA.
- **الحقل SELDR (1 بت):** انتقاء المسرى DR. إذا كانت قيمة هذا الحقل 1، ينتقى المسرى DA الخارجي كدخل للمسريين DRA, DRB. وإذا كانت قيمته معدومة، فهو يُنتقى خرج المضمّم DRA لربطه بالمسرى DRA (والذي يتحكم فيه الحقل PSEL) ولربط محتويات السجل RCB بالمسرى DRB.
- **الحقل ZEROIN (1 بت):** ويُستخدم للدلالة على التفرع الشرطي. يعتمد عندئذ سلوك وحدة التتالي الصغرية على الشرط المنتقى في الحقل 1 في الجدول 7.15.
- **الحقل RC2-RC0 (3 بتات):** وهي بتات تحكم في السجلات، إذ تحدّد التغير في محتويات السجلين RCA و RCB. ويمكن أن تبقى قيمة أي سجل كما هي، أو تُنقص تلك القيمة، أو تُحمّل من المسريين DRA, DRB.

• **الحقل S_2-S_0 (3 بتات):** وهي بتات تحكم في المكس؛ إذ تحدّد عملية المكس الواجب تنفيذها.

• **الحقل MUX0 - MUX2 (3 بتات):** وهي بتات تحكم في الخرج. تتحكم هذه البتات، عند استخدامها مع رايات الشروط، في مضمّم الخرج Y، وهي تتحكم إذن في عنوان التعليميّة الصغرى. يستطيع المضمّم انتقاء خرج من المكس أو من السجلات DRA .DRB .MPC.

يمكن للمبرمج وضع هذه البتات إفرادياً على الواحد، ولكن لا تجري العملية على هذا النحو عادة، إذ يستخدم المبرمج بدلاً من ذلك رموزاً تذكيرية mnemonics تكافئ الأشكال النمطية للبتات المطلوبة. يسرد الجدول 8.15 الرموز التذكيرية اللازمة للحقل 28، وعددها 15. يحوّل مجمّع البرنامج الصغرى تلك الرموز إلى الأشكال النمطية للبتات.

وعلى سبيل المثال، تؤدي التعليميّة INC88181 إلى انتقاء التعليميّة الصغرى التالية في التالي إذا كان الشرط المنتقى حالياً محققاً. ونجد من الجدول 8.15 ما يلي:

$$\text{INC } 88181 = 0000 \quad 0011 \quad 1110$$

والتي يفك ترميزها مباشرة على النحو التالي:

• **الحقل OSEL = 0:** وينتقي ذلك السجل RCA كخرج لمضمّم الخرج DRA. وفي هذه الحالة، لا فائدة من هذا الانتقاء.

• **الحقل SELDR = 0:** وكما ذكر سابقاً، لا علاقة لذلك في هذه التعليميّة.

• **الحقل ZEROIN = 0:** والذي يشير، عند ضمّه إلى قيمة المضمّم، إلى عدم إجراء أي تفريع.

• **الحقل R = 0 0 0:** وهو يحافظ على القيمة الحالية للسجلين RA وRC.

• **الحقل MUX = 110:** ويختار السجل MPC عندما يتحقق الشرط، ويختار السجل DRA عند عدم تحققه.

وحدة الحساب والمنطق ذات السجلات Registered ALU

إن الرقاقة 8832 هي وحدة حساب ومنطق ذات 64 سجل، يمكن تشكيلها بحيث تعمل كأربع وحدات حساب ومنطق ذات 8 بتات، أو كوحدة حساب ومنطق ذات 16 بت، أو كوحدة حساب ومنطق ذات 32 بت. يتحكم في الرقاقة 8832 عدد من البتات، وقدره 39، تمثل الحقول من 17 إلى 27 في التعليميّة الصغرى (الجدول 7.15). تُقدّم هذه البتات إلى وحدة الحساب والمنطق كإشارات تحكم. إضافة إلى ذلك، وكما هو مبين في (الشكل 17.15)، للرقاقة 8832 ارتباطات خارجية بالمسرى DA، وله 32 بت، ومسرى النظام Y، وله 32 بت. يمكن تقديم المداخل من المسرى DA كمعطيات دخل إلى ملف السجلات، والذي يضم 64 سجلاً، وإلى النسيقة المنطقية ALU في آن واحد. وتُخرّج نتائج عمليات الإزاحة ووحدة الحساب والمنطق إلى المسرى DA أو مسرى النظام Y. ويمكن تغذية النتائج خلفياً إلى ملف السجلات الداخلي.

تسمح منافذ العناوين، وهي ذات 6 بتات، بتنفيذ عملية جلب ذات عاملين، وعملية كتابة في ملف السجلات في آن واحد. ويمكن تشكيل السجل MQ وسجل الإزاحة MQ أيضاً ليعملا على نحو مستقل بغية تنجيز عمليات إزاحة على 8 بتات، أو 16 بت، أو 32 بت.

تتحكم الحقول من 17 إلى 26 من كل تعليميّة صغرى في طريقة انسياب المعطيات في الرقاقة 8832 وفي

انسيابها بين الرقاقة والوسط الخارجي. وهذه الحقول هي:

17. *Write enable*: يعبّر هذان البتان عملية كتابة لكلمة ذات 32 بت، أو كتابة البتات الست عشرة الأكثر دلالة، أو البتات الست عشرة الأقل دلالة، أو عدم الكتابة في ملف السجلات. يُعرّف ملف الوجهة بالحقل 24.

الجدول 8.15 بتات التعليمات الصغيرة في وحدة تنال النظام TI-8800

| الوصف | القيمة | الرمز التذكيري |
|---|--------------|----------------|
| تعليلة الاستبداء | 00000000110 | RST8818 |
| تفريع إلى تعليلة DRA | 011000111000 | BRA88181 |
| تفريع إلى تعليلة DRA | 010000111110 | BRA88180 |
| تعليلة متابعة | 000000111110 | INC88181 |
| تعليلة متابعة | 001000001000 | INC88180 |
| قفز إلى المساق الفرعي في العنوان المحدد بالسجل DRA | 010000110000 | CAL88181 |
| قفز إلى المساق الفرعي في العنوان المحدد بالسجل DRA | 010000101110 | CAL88180 |
| العودة من المساق الفرعي | 000000011010 | RET8818 |
| دفع عنوان العودة من المقاطعة إلى المكس | 000000110111 | PUSH8818 |
| عودة من المقاطعة | 100000010000 | POP8818 |
| تحميل العداد DRA من المسرى DA | 000010111110 | LOADDRA |
| تحميل العداد DRB من المسرى DA | 000110111110 | LOADDRB |
| تحميل العداد DRB/DRA | 000110111100 | LOADDRA |
| B | | |
| انقاص العداد DRA والتفريع إذا كانت قيمته غير معدومة | 010001111100 | DECRDRA |
| انقاص العداد DRB والتفريع إذا كانت قيمته غير معدومة | 010101111100 | DECRDRB |

18. *Select register file data source*: إذا وجبت الكتابة في ملف السجلات، يعبّر هذان البتان المصدر: المسرى DA أو المسرى DB، أو خرج الوحدة ALU، أو مسرى النظام Y.

19. *Shift instruction modifier*: يعبّر الخيارات المتعلقة بتقدم بتات ملء النهاية، وقراءة البتات المزاحة أثناء تعليمات الإزاحة.

20. *Carry in*: يشير هذا البت إلى استخدام بت كحمل دخل في عملية تُنفذ في وحدة الحساب والمنطق.

21. *ALU configuration*: يمكن تشكيل الرقاقة 8832 للعمل كوحدة ALU ذات 32 بت، أو كوحدين لكل منهما 16 بت، أو كأربع وحدات ALU ذات 8 بتات.

22. *S input S*: يزود مضمّمان داخليان مداخل 1 بتاً المنطقي ALU، ويشار إليهما بالمضمّم R و S. ينتقي هذا الحقل الدخل المرتبط بالمضمّم S من الخيارات التالية: ملف السجلات، المسرى DB، السجل MQ. ويُعرّف سجل المصدر بالحقل 25.

23. *R input R*: ينتقي الدخل المرتبط بالمضمّم R: وهو إما ملف السجلات أو المسرى DA.

24. *Destination register*: عنوان السجل في ملف السجلات الواجب استخدامه عامل

وجهة.

25. سجل المصدر *Source register*: عنوان السجل في ملف السجلات الواجب استخدامه عامل مصدر، ويقدمه المضمّم S.

26. سجل المصدر *Source register*: عنوان السجل في ملف السجلات الواجب استخدامه عامل مصدر، ويقدمه المضمّم R.

ويعين أخيراً الحقل 27، وله 8 بتات، الوظيفة الحسابية أو المنطقية الواجب إجراؤها في وحدة الحساب والمنطق. يسرد الجدول 9.15 العمليات المختلفة الممكن تنفيذها.

الجدول 9.15: حقل التعليمات الخاص بوحدة الحساب والمنطق ذات السجلات في الرقاقة TI-8800 (الحقل 27)

| المجموعة 1 | | الوظيفة |
|------------|------|--|
| ADD | H#01 | $R + S + C_n$ |
| SUBR | H#02 | $(NOT\ R) + S + C_n$ |
| SUBS | H#03 | $R = (NOT\ S) + C_n$ |
| INSC | H#04 | $S + C_n$ |
| INSCNS | H#05 | $(NOT\ S) + C_n$ |
| INCR | H#06 | $R + C_n$ |
| INCNR | H#07 | $(NOT\ R) + C_n$ |
| XOR | H#09 | $R\ XOR\ S$ |
| AND | H#0A | $R\ AND\ S$ |
| OR | H#0B | $R\ OR\ S$ |
| NAND | H#0C | $R\ NAND\ S$ |
| HOR | H#0D | $R\ NOR\ S$ |
| ANDNR | H#0E | $(NOT\ R)\ AND\ S$ |
| المجموعة 2 | | الوظيفة |
| SRA | H#00 | إزاحة حسابية إلى اليمين أحادية الدقة |
| SRAD | H#10 | إزاحة حسابية إلى اليمين مضاعفة الدقة |
| SRL | H#20 | إزاحة منطقية إلى اليسار أحادية الدقة |
| SRLD | H#30 | إزاحة منطقية إلى اليسار مضاعفة الدقة |
| SLA | H#40 | إزاحة حسابية إلى اليسار أحادية الدقة |
| SLAD | H#50 | إزاحة حسابية إلى اليسار مضاعفة الدقة |
| SLC | H#60 | إزاحة دائرية إلى اليسار أحادية الدقة |
| SLCD | H#70 | إزاحة دائرية إلى اليسار مضاعفة الدقة |
| SRC | H#80 | إزاحة دائرية إلى اليمين أحادية الدقة |
| SRCD | H#90 | إزاحة دائرية إلى اليمين مضاعفة الدقة |
| MQSRA | H#A0 | إزاحة السجل MQ إزاحة حسابية إلى اليمين |
| MQSRL | H#B0 | إزاحة السجل MQ إزاحة منطقية إلى اليمين |
| MQSLL | H#C0 | إزاحة السجل MQ إزاحة منطقية إلى اليسار |
| MQSLC | H#D0 | إزاحة السجل MQ إزاحة دائرية إلى اليسار |
| LOADMQ | H#E0 | تحميل السجل MQ |

| | | |
|------------|------|---------------------------------|
| PASS | H#F0 | تمرير ALU إلى Y (بدون إزاحة) |
| المجموعة 3 | | الوظيفة |
| SET1 | H#08 | وضع القيمة 1 في البت 1 |
| Set0 | H#18 | وضع القيمة 1 في البت 0 |
| TB1 | H#28 | اختبار البت 1 |
| TB0 | H#38 | اختبار البت 0 |
| ABS | H#48 | القيمة المطلقة |
| SMTc | H#58 | الاتمام إلى اثنين/ مقدار وإشارة |
| ADDI | H#68 | إضافة قيمة فورية |
| SUBI | H#78 | طرح قيمة فورية |
| BADD | H#88 | جمع بايت في R إلى S |
| BSUBS | H#98 | طرح بايت في S من R |
| BSUBR | H#A8 | طرح بايت في R من S |
| BINCS | H#B8 | زيادة البايت S بمقدار 1 |
| BINCNS | H#C8 | زيادة البايت السالب S بمقدار 1 |
| BXOR | H#D8 | عملية XOR بين R و S |
| BAND | H#E8 | عملية AND بين R و S |
| BOR | H#F8 | عملية OR بين R و S |
| المجموعة 4 | | الوظيفة |
| CRC | H#00 | مراكمة محرف الفائض الدوّار |
| SEL | H#10 | انتقاء S أو R |
| SNORM | H#20 | تسوية إلى الطول الأحادي |
| DNORM | H#30 | تسوية إلى الطول المضاعف |
| DIVRE | H#40 | إصلاح باقي القسمة |
| SDIVQF | H#50 | اصلاح ناتج القسمة ذي الإشارة |
| SMULI | H#60 | تكرار الضرب ذي الإشارة |
| SMULT | H#70 | إنهاء الضرب ذي الإشارة |
| SDIVIN | H#80 | استهلال القسمة ذات الإشارة |
| SDIVIS | H#90 | البدء بالقسمة ذات الإشارة |
| SDIVI | H#A0 | تكرار القسمة ذات الإشارة |
| UDIVIS | H#B0 | البدء بالقسمة بلا إشارة |
| UDIVI | H#C0 | تكرار القسمة بلا إشارة |
| UMULI | H#D0 | تكرار الضرب بلا إشارة |
| SDIVIT | H#E0 | إنهاء الضرب بلا إشارة |
| UDIVIT | H#F0 | إنهاء القسمة بلا إشارة |
| المجموعة 5 | | الوظيفة |
| LOADFF | H#0F | تحميل قلايات القسمة/الأعداد BCD |
| CLR | H#1F | المحو |
| DUMPF | H#5F | إخراج قلايات القسمة/الأعداد BCD |

| | | |
|--------|------|-------------------------------|
| BCDBIN | H#7F | تحويل من BCD إلى اثنائي |
| EC3BC | H#8F | تصحيح البايت Excess-3 |
| EX3C | H#9F | تصحيح الكلمة Excess-3 |
| SDIVO | H#AF | اختبار فيض القسمة ذات الإشارة |
| BINEX3 | H#DF | تحويل اثنائي إلى Excess |
| NOP32 | H#FF | بلا عمل |

نأخذ مثلاً على الترميز المستخدم لتوصيف الحقول من 17 إلى 27 تعليمة جمع محتويات السجل 1 إلى السجل 2، ووضع النتيجة في السجل 3. إن التعليمة الرمزية هي:

CONT11 [17], WELH, SELRFYF MX, [24],

R3, R2, R1, PASS + ADD

سيترجم أ مع ذلك إلى قيمة مناسبة للبتات. ويمكن وصف المكونات الإفرادية للتعليمة على النحو الآتي:

- CONT11: وهي تعليمة الالعمل الأساسية (NOP).
- يأخذ الحقل 17 القيمة WELH (تفعيل الكتابة الدنيا والعليا بحيث يمكن كتابة سجل ذي 32 بت).
- يأخذ الحقل 18 القيمة SELRFYMX لانتقاء التغذية الراجعة من خرج المضمّم ALU-Y.
- يأخذ الحقل 24 القيمة R3 كسجل وجهة.
- يأخذ الحقل 25 القيمة R2 كأحد سجلات المصدر.
- يأخذ الحقل 26 القيمة r1 كأحد سجلات المصدر.
- يتغير الحقل 27 ليحدد العملية ADD كوظيفة لوحدة الحساب والمنطق، ويأخذ سجل الإزاحة في الوحدة ALU القيمة PASS، لهذا لا يزاح خرج الوحدة ALU في سجل الإزاحة.
- يمكن الإشارة إلى عدة نقاط تتعلق بالتدوين الرمزي. ليس ضرورياً تحديد رقم الحقل من أجل الحقول المتتابة، أي إن التعليمة: CONT11[17], WELH, [18], SELRFYMX يمكن كتابتها كما يلي:

CONT11 [17], WELH, SELRFYMX

لأن القيمة SELRFYMX تقع في الحقل 18.

يجب استخدام تعليمات وحدة الحساب والمنطق، المذكورة في أ موعة 1 في الجدول 9.15، إلى جانب تعليمات أ موعة 2. ولا ينبغي استخدام تعليمات الوحدة ALU في أ موعات 3-5 إلى جانب أ موعة 2.

5.15 تطبيقات البرمجة الصغيرة Applications of Microprogramming

لقد أصبحت تطبيقات البرمجة الصغيرة، منذ ظهورها، ولا سيما منذ نهاية الستينيات، شائعة وشديدة التنوع. فقد كانت معظم الاستخدامات الحديثة، إن لم نقل جميعها، ظاهرة للعيان منذ عام 1971 [FLYN71]. وتناقش دراسات مسح لاحقة مجموعة التطبيقات ذاتها (مثل المرجع [RAU80]). تتضمن مجموعة التطبيقات الراهنة للبرمجة الصغيرة ما يلي:

- تنجيز الحواسيب

• التقليد Emulation

- دعم نظم التشغيل
- إنجاز بعض التجهيزات الخاصة الاستخدام
- دعم اللغات العالية المستوى
- التشخيص الصغري
- تطبيقات يخصصها المستخدم.

لقد كُرس هذا الفصل لمناقشة إنجاز الحواسيب، إذ تقدّم مقارنة البرمجة الصغرية تقنية منهجية لتنفيذ وحدة التحكم. وهناك تقنية تتعلق بذلك، وهي التقليد [MALL75]. ويشير التقليد إلى استخدام برنامج صغري على آلة معينة لتنفيذ برامج مكتوبة أصلاً لآلة أخرى. وتفيد التقليد أساساً في مساعدة المستخدمين على الانتقال من حاسوب إلى آخر ويقوم بذلك غالباً المصنّعون لتسهيل انتقال مستخدمي الحواسيب القديمة إلى حواسيب أحدث، للحيلولة دون انتقاهم إلى مصنع منافس. ويدهش المستخدمون في أغلب الأحيان عند اكتشاف العمر المديد لهذه الأدوات. فلقد لاحظ أحد المستخدمين [MALL83] أن بالإمكان أن نجد، حتى العام 1983 حواسيب من طراز IBM S/370 تقلد الحاسوب IBM 1401 الذي استعيض عنه منذ أكثر من عقد ونصف. وهناك استخدام آخر مفيد للبرمجة الصغرية يقع في حقل دعم نظم التشغيل. إذ يمكن استخدام البرامج الصغرية لتنفيذ البدئيات primitives التي تحل محل أجزاء مهمة من برمجيات نظام التشغيل. وتسهّل هذه التقنية مهمة تنفيذ نظام التشغيل وتحسّن أدائه.

تفيد البرمجة الصغرية كوسيلة لتنفيذ التجهيزات ذات الاستخدام الخاص *special - purpose devices* الممكن تضمينها في حاسوب مضيف. ومثال ذلك بطاقة نقل المعطيات. تضم تلك البطاقة معالجها الخاص بها. ولما كانت تلك البطاقة خاصة الاستخدام، فمن المفيد تنفيذ جزء من وظائفها في البرمجيات الراسخة بدلاً من تنفيذها في برمجيات عادية، بغية تحسين أدائها. يُعدّ دعم لغات البرمجة العالية المستوى مجالاً آخر لتطبيق تقنيات البرمجة الصغرية. إذ يمكن تنفيذ العديد من الوظائف وأنواع المعطيات مباشرة في البرمجيات الراسخة. وينتج من ذلك سهولة تصريف البرنامج لتحويله إلى لغة الآلة. وفي الواقع، تُبنى لغة الآلة لملاءمة متطلبات اللغات العالية المستوى (مثل الفورتران والكوبول والآداء).

كما يمكن استخدام البرمجة الصغرية لدعم مراقبة أخطاء النظام واكتشافها وعزلها وتصحيحها، ويُعرف ذلك باسم التشخيص الصغري *microdiagnostics*. وهو يفيد في تعزيز إمكانيات صيانة النظام على نحو ملحوظ. وتسمح هذه المقاربة للنظام بإعادة تشكيل ذاته عند اكتشاف عطل ما. فمثلاً، عندما يتعطل ضارب مرتفع السرعة، يمكن لضارب مبرمج صغرياً أن يحل مكانه.

وهناك فئة عامة من التطبيقات التي يخصصها المستخدم *user tailoring*. تُبنى بعض الآلات باستخدام مكان تخزين قابل للكتابة *writable control store* (أي ذاكرة تحكم منجزة بالذاكرة RAM بدلاً من ROM)، وهي تسمح للمستخدم بكتابة البرامج الصغرية. وتُستخدم عموماً مجموعة تعليمات صغرية شاقولية إلى حد بعيد وسهلة الاستخدام. ويتيح ذلك للمستخدم تخصيص تلك الآلة وفق التطبيق المطلوب.

6.15 قراءات ينصح بها Recommended Reading

هناك عدد من الكتب المكرّسة للبرمجة الصغيرة، وقد يكون أسهلها فهماً المرجع [LYNC93]. ويقدم المرجع [SEGE91] أساسيات البرمجة الصغيرة وتصميم النظم المبرمجة صغيراً، وذلك بتصميم معالج بسيط ذي 16 بت خطوةً بخطوة. يقدم المرجع [CART96] أيضاً المفاهيم الأساسية باستخدام آلة بسيطة. ويقدم المرجعان [PARK89] و [TI90] وصفاً مفصلاً لبطاقة تطوير البرمجيات TI-8800.

المراجع

- CART96 Carter, J. Microprocessor Architecture and Microprogramming. Upper Saddle River, NJ: Prentice Hall, 1996
- LYNC93 Lynch, M. Microprogrammed State Machine Design. Boca Raton, FL: CRC Press, 1993
- PARK89 Parker, A., and Hamblen, J. An Introduction to Microprogramming with Exercises Designed for the Texas Instruments SN74ACT8800 Software Development Board. Dallas, TX: Texas Instruments, 1989
- SEGE91 Segee, B.. and Filed, J. Microprogramming and Computer Architecture. New York: Wiley. 1991
- TI90 Texas Instruments Ins. SN74ACT880 Family Data Manual. SCSS006C, 1990

7.15 مسائل Problems

1.15. صف تنجيز تعليمة الضرب في الآلة الافتراضية التي صمّمها ويلكس. اشرح ذلك مستعيناً بمخطط انسيابي.

2.15. لنفترض مجموعة تعليمات صغيرة تتضمن تعليمة لها الشكل الرمزي التالي:

IF ($AC_0 = 1$) THEN $CAR \leftarrow (C_{0-6})$ ELSE $CAR \leftarrow (CAR) + 1$

حيث AC_0 هو بت الإشارة في المراكم، و C_{0-6} هي البتات السبعة الأولى في التعليمة الصغيرة. اكتب باستخدام هذه التعليمة الصغيرة برنامجاً صغيراً ينجز تعليمة الآلة BRM، والتي تؤدي إلى التفريع إذا كان السجل AC سالباً. وافترض أن البتات $C_1 - C_n$ في التعليمة الصغيرة تحدّد مجموعة متوازية من العمليات الصغيرة عبر عن ذلك رمزياً.

3.15. لمعالج بسيط أربع مراحل أساسية في دورة التعليمة: الجلب والعنونة غير المباشرة، والتنفيذ والمقاطعة. تشير رايتان وحيدتا البت إلى المرحلة الراهنة في التنجيز المبني عتادياً.

a. لم يحتاج إلى هاتين الرايتين؟

b. لم يحتاج إلى هاتين الرايتين في وحدة تحكم مبرمجة صغيراً؟

4.15. لنفترض وحدة التحكم المذكورة في (الشكل 7.15)، وليكن عرض ذاكرة التحكم هو 24 بت. يُقسم جزء التحكم في صيغة التعليمات الصغيرة إلى حقلين: يعيّن حقل التعليمة الصغيرة، وعرضه 13 بت، العمليات الصغيرة الواجب تنفيذها. ويعيّن حقل انتقاء العناوين الشرط الذي يؤدي إلى حدوث التفريع؛ بالاعتماد على الرايات، والبالغ عددها 8.

أ. ما هو عدد البتات في حقل انتقاء العناوين؟

ب. ما هو عدد البتات في حقل العنوان؟

ج. ما هي سعة ذاكرة التحكم؟

5.15. كيف يمكن إجراء تفريع لا شرطي في ظروف المسألة السابقة؟ وكيف يمكن تجنب التفريع (أي صف تعليمية صغيرة لا تحدّد أي تفريع شرطي أو لا شرطي).

6.15. نوّد إعطاء 8 كلمات تحكم لكل مساق تعليمية آلة. ولكل رماز عملية في أي تعليمية آلة 5 بتات، وتبلغ سعة ذاكرة التحكم 1024 كلمة. اقترح تقابلاً بين سجل التعليمات وسجل عنوان التحكم.

7.15. نوّد استخدام صيغة تعليمية صغيرة مرمّزة. بين كيف يمكن تقسيم حقل عملية صغيرة ذي 9 بت إلى حقول فرعية لتعيين 46 فعلاً مختلفاً.

8.15. يملك معالج 16 سجلاً، ووحدة حساب ومنطق ذات 16 وظيفة حسابية و 16 وظيفة منطقية وسجل إزاحة له 8 عمليات. ترتبط جميع هذه المكونات بمسرى معالج داخلي. صمّم صيغة تعليمات صغيرة لتعيين العمليات الصغيرة المتنوعة للمعالج.

