الفصل السادس 6

المعالج

من البداية...حتى النهاية

Microprocessor... from start ... to end

obeikandl.com

6-1 مقدمة

سنقوم فى هذا الفصل بعملية بناء تدريجية لمعالج افتراضى يقوم بعدد محدود من العمليات الحسابية والمنطقية وله عدد محدود من الأوامر كما أن له عددا محدودا جدا من الخطوط فى مسارات البيانات والعناوين والتحكم ولذلك فإن هذا المعالج يستطيع التعامل مع كمية محدودة جدا من بايتات الذاكرة . سنبدأ عملية البناء من أقل مستوى ممكن ثم سنرتقى بها خطوة بخطوة إلى أن نصل إلى معالج متكامل ولكن بالمواصفات التى ذكرناها سابقا . من خلال عملية البناء سنتعرف على وحدة الحساب والمنطق وكيفية عملها . ولقد كان قصدنا من وضع هذا الفصل فى هذا الترتيب أن يكون القارىء قد ألم بفكرة عامة عن تركيب المعالج من الفصول السابقة ثم يجىء هذا الفصل فيؤكد هذه الفكرة ويمحصها ويضيف إليها التفاصيل الدقيقة التى قد تجيب على الكثير من الأسئلة التى تدور فى خلد أى قارىء عن كيفية تنفيذ أى معالج لأى أمر وما الذى يتحكم فى عدد أوامره وغير ذلك من الأسئلة المهمة .

إن وحدة الحساب والمنطق هى إحدى المكونات الرئيسية للمعالج ومهمتها الأساسية هى إجراء العمليات الحسابية والمنطقية الأساسية وسنبدأ فيما يلى عملية بناء هذه الوحدة ولكى نصل إلى ذلك لابد أن نتعرف أولا على كيفية تنفيذ عمليات الجمع والطرح فى النظام الثنائى .

6-2 الجمع الثنائى Binary addition

مثال 6-1

أوجد ناتج جمع الرقم A=a3a2a1a0 =1101 مع الرقم B=b3b2b1b0 =1011.

إن عملية الجمع الثنائى تتم كالتالى:

الحمل	1	1	1	1	
A		1	1	0	1
B		1	0	1	1 +
حمل ←1	1	0	0	0	
النتيجة	S3	S2	S1	S0	

كما نرى فإن عملية الجمع تتم على عدد من المراحل ، المرحلة الأولى هى جمع a0 (البت الأولى فى الرقم A) مع b0 (البت الأولى من الرقم B) فينتج من ذلك نتيجة s0=0 وحمل c0=1 إلى المرحلة التالية . فى المرحلة الثانية يتم جمع البتات الآتية :

c0 + b1 + a1

حيث c0 هى الحمل من المرحلة السابقة كما ذكرنا . نتيجة جمع المرحلة الثانية ستكون s1=0 والحمل منها سيكون c1=1 إلى المرحلة التالية . فى المرحلة الثالثة ستتم عملية الجمع التالية :

c1 + b2 + a2

وسينتج عنها s2=0 و c2=1 ثم فى مرحلة الجمع الرابعة سيتم جمع البتات التالية:

c2 + b3 + a3

وسينتج عنها s3=1 و c3=1 وبذلك تنتهى عملية الجمع ويتبقى حمل أخير للخانة الرابعة وهو c3=1 الذى سنهمله الآن . من ذلك نرى أننا فى المرحلة الأولى نجمع اثنين بت فقط (a0 + b0) وأما فى باقى المراحل فإننا نجمع ثلاث بتات (cn-1 + bn + an) حيث an هى البت رقم n فى العدد A و bn هى البت رقم n أيضا فى العدد B ، وأما cn-1 فهى الحمل الناتج من جمع المرحلة السابقة (cn-2 +bn-1+an-1) . الآن نريد تكوين دائرة منطقية تقوم بعملية جمع 2 بت وأخرى تقوم بعملية جمع 3 بتات ثم من الدائرتين نقوم بتكوين دائرة تجمع العددين A و B بأى عدد من البتات .

6-2-1 دائرة نصف المجمع Half adder circuit, HA

يقوم نصف المجمع Half Adder, HA بجمع اثنين بت a0 و b0 ويعطى فى الخرج النتيجة s0 وحمل c0 ويبين شكل (6-1) جدول الحقيقة truth table لهذه الدائرة . من جدول الحقيقة نستطيع كتابة المعادلات المنطقية التالية لكل من خرجى دائرة نصف المجمع :

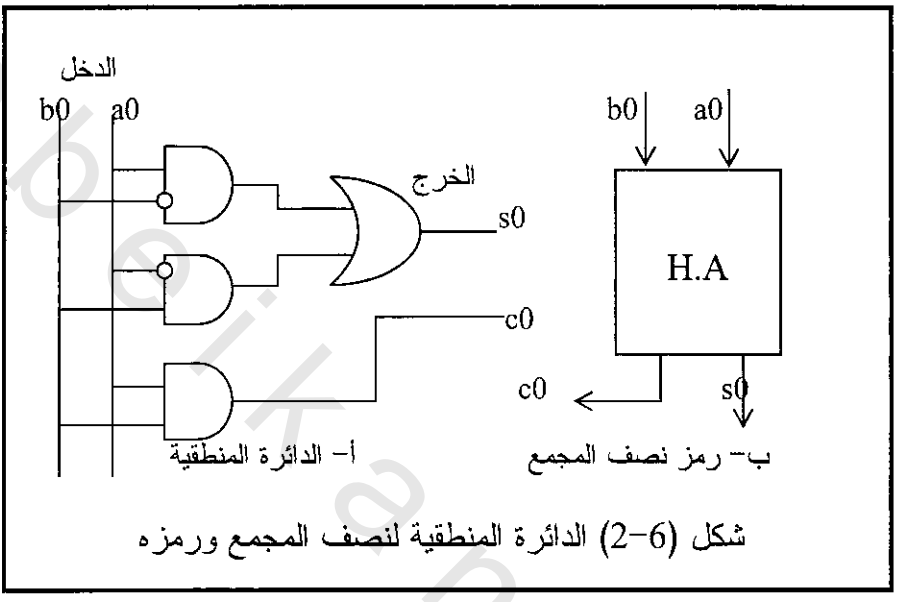
$s0 = a0\,\overline{b0} + \overline{a0}\,b0$ 1-6

$c0 = a0b0$ 2-6

الدخل		الخرج	
b0	a0	s0	c0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

شكل (6-1) جدول الحقيقة لنصف المجمع

من المعادلتين 6-1 و 6-2 نستطيع رسم دائرة منطقية لنصف المجمع كما فى الشكل (6-2أ) . انظر أيضا فى نفس الشكل إلى الرمز الذى سنستخدمه لهـذه الدائرة .



شكل (6-2) الدائرة المنطقية لنصف المجمع ورمزه

6-2-2 دائرة المجمع الكامل Full adder, FA

دائرة المجمع الكامل تكون قادرة على جمع ثلاثة بتـات (cn-1, bn, an) وينـتج منها المجموع sn والحمل للمرحلة القادمة cn . جدول الحقيقـة لهـذه الـدائرة موضح فى شكل (6-3) . من جدول الحقيقة نستطيع كتابة معادلات الخرج كما يلى :

$$Sn=an\ \overline{bn}\ \overline{cn-1} + \overline{an}\ bn\ \overline{cn-1} + \overline{an}\ \overline{bn}\ cn\text{-}1 + an\ bn\ cn\text{-}1 \qquad 3\text{-}6$$

$$Cn=an\ bn\ \overline{cn-1} + an\ \overline{bn}\ cn\text{-}1 + \overline{an}\ bn\ cn\text{-}1 + an\ bn\ cn\text{-}1 \qquad 4\text{-}6$$

من المعادلتين 6-3 و 6-4 نستطيع استنتاج الدائرة المنطقية للمجمع الكامل كما فى شكل (6-4) . من ذلك نرى أنه لجمع أى رقمين A و B فإننـا سـنحتاج لنصف مجمع لجمع البت رقم 0 فى كل من الرقمين ثم سنحتاج مجمعـا كـاملا لجمع كل بت فى الرقم الأول مع ما يناظرها فى الرقم الثانى مع الحمل النـاتج من عملية الجمع السابقة . فمثلا لو أن الرقمين A و B كل منهما يتكون من 4 بتات فإننا سنحتاج إلى نصف مجمع وثلاثة مجمعات كاملة لإتمام عملية جمـع

الرقمين ونفس الكلام يمكن تطبيقه على عملية جمع أى رقمين حيث كل منهما مكون من أى عدد من البتات . شكل (6-5) يبين الدائرة المستخدمة لجمع رقمين كل منهما مكون من أربعة بتات كمثال على ذلك . وترى فى نفس الشكل الرمز العام المستخدم للمجمع .

الدخل			الخرج	
cn-1	bn	an	sn	cn
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

شكل (6-3) جدول الحقيقة للمجمع الكامل

6-3 الطرح الثنائى
Binary subtraction

لإجراء عمليات الطرح الثنائى فإنه عادة ما نلجأ إلى تحويل عملية الطرح إلى عملية جمع وبعد ذلك يمكن استخدام المجمع الذى سبق شرحه لتنفيذ عملية الطرح . لتحويل عملية الطرح إلى عملية جمع ننظر إلى المثال التالى :

مثال 6-2

افترض أن لدينا الرقم A = 1101 فإن المعكوس أو المتمم الأحادى one's complement لهذا الرقم هو A = 0010 وتم ذلك عن طريق قلب كل 1 إلى 0 وكل 0 إلى 1 فى الرقم الأصلى . الآن ماذا يحدث لو جمعنا العدد الأصلى زائد متممه الأحادى زائد واحد كما يلى :

$$\begin{array}{r} A = 1101 \\ \overline{A} = 0010 \\ \hline 1 + \\ \hline 1\ 0000 \end{array}$$

الحمل ←

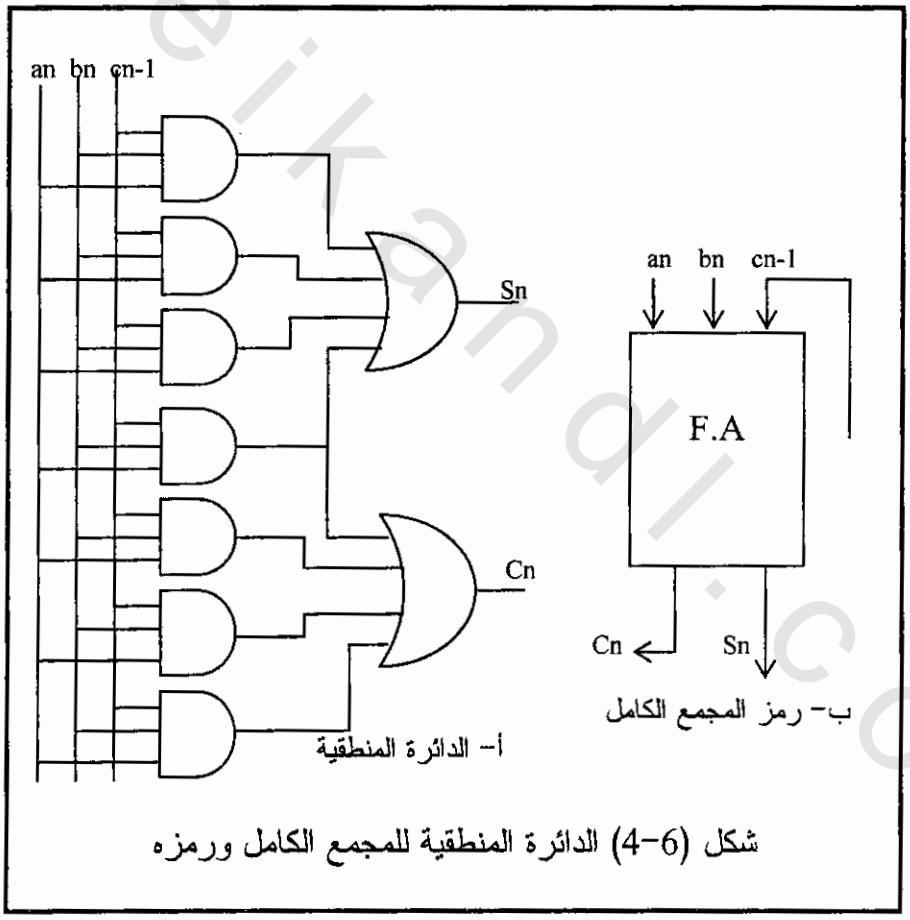
إن النتيجة كما رأينا ستكون دائما صفرا مع حمل واحد ، ولذلك فإنه بإهمال هذا الحمل يمكننا كتابة العلاقة التالية :

$$A + \overline{A} + 1 = 0$$

ومنها يمكن كتابة الرقم A على الصورة التالية :

$$-A = \overline{A} + 1 \qquad 5-6$$

وعلى ذلك فإنه من المعادلة (6-5) يمكننا أن نرى أن أى عملية طرح يمكن تحويلها إلى عملية جمع عن طريق استبدال المطروح بمتمه الثنائى (المتمم الأحادى + 1) . كمثال على ذلك انظر إلى عمليات الطرح التالية وكيف حولناها إلى عمليات جمع :

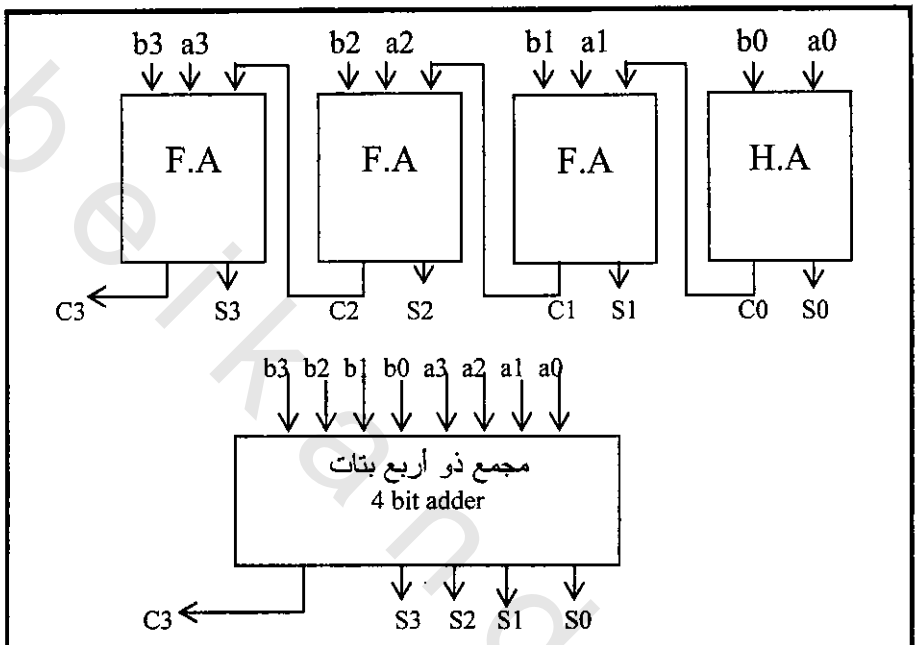


شكل (6-4) الدائرة المنطقية للمجمع الكامل ورمزه

$$A - B = A + \overline{B} + 1 \qquad 6-6$$

$$B - C = B + \overline{C} + 1 \qquad 7-6$$

وبذلك نستطيع القول أنه يمكننا استخدام دائرة المجمع التى سبق شرحها فى تنفيذ عمليات الطرح أيضا بعد إجراء بعض التعديلات الطفيفة عليها . شكل (6-6) يبين دائرة مجمع وقد تم عليها هذا التعديل لتقوم بعمليات الجمع أو الطرح عن طريق بعض خطوط التحكم التى يمكن بواسطتها اختيار إما عملية الجمع أو عملية الطرح .



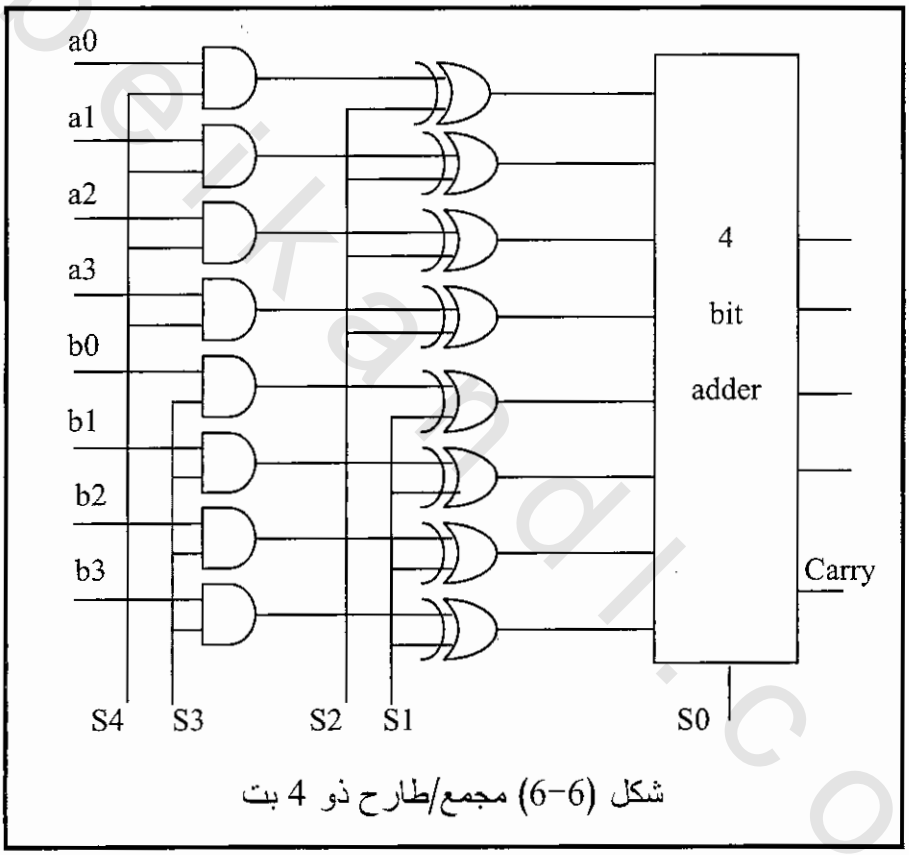
شكل (6-5) التركيب والرمز العام لمجمع ذو 4 بتات

كما نعلم فإنه من خواص بوابة XOR أن لها دخلان عندما يكون أحدهما يساوى واحدا فإن الخرج يكون معكوس الدخل الآخر ، وعندما يكون أحد هذين الدخلين يساوى صفرا فإن الخرج يكون مثل الدخل الآخر ، أى أن بوابة XOR يمكن استخدام أحد دخليها للتحكم فى جعل الخرج يساوى إما الدخل الآخر أو معكوسه. يمكن أن نرى ذلك بوضوح فى شكل (6-6) بحيث أنه إذا كان خط التحكم S1=1 فإن الرقم B سيدخل للمجمع معكوس . أما إذا كان S1=0 فإن الرقم B سيدخل إلى المجمع بقيمته الحقيقية . نفس الشىء تم تطبيقه على العدد A باستخدام خط التحكم S2 .

من شكل (6-6) أيضا يمكننا استنتاج وظيفة خطى التحكم S4, S3 بأن كل منهما بمثابة مفتاح (ON/OFF) للرقم الذى يعمل معه . فالخط S3 سيسمح بمرور العدد B أو صفر بدلا منه ، والخط S4 سيسمح بمرور الرقم A أو صفر بدلا منه .

إنه باستخدام خطوط التحكم S4, S3, S2, S1 يمكن استخدام المجمع الموجود فى شكل (6-6) فى أكثر من وظيفة منها الجمع والطرح .

إفترض مثلا أن S1=0, S0=0 S4=1, S3=1, S2=0, فى هذه الحالة بما أن S4, S3 كل منهما يساوى واحدا فإن العددين A و B سيعبران من بوابات ال AND بنفس قيمهما وبما أن S2, S1 كل منهما يساوى صفرا فإن العددين A و B سيعبران من بوابات ال XOR بنفس قيمهما الحقيقية ومن ذلك نرى أن المجمع سيكون دخلاه المباشران هما العددان A و B وصفر من خط التحكم S0 لذلك سيقوم المجمع بجمعها .



شكل (6-6) مجمع/طارح ذو 4 بت

افترض الآن الوضع التالى S4=1, S3=1, S2=0, S1=1, S0=1 من شكل (6-6) نرى أنه طالما أن S4, S3 كل منهما يساوى واحدا فإن العددان B و A سيعبران من بوابات ال AND ، وبما أن S2=0 فإن العدد A سيعبر من بوابات ال XOR بقيمته الحقيقية وطالما أن S1=1 فإن العدد B سيعبر من بوابات ال XOR معكوسا وبذلك نستطيع القول أن المجمع سيقوم بعملية الجمع التالية :

$A + \overline{B} + 1$

حيث الواحد هو قيمة خط التحكم S0 . كما علمنا من قبل فإن $-B=\overline{B}+1$ لذلك فإننا نستطيع القول بأن المجمع فى هذه الحالة يقوم بعملية طرح العدد B من العدد A ، أى (A - B) .

شكل (6-6) به خمسة خطوط تحكم هى S4, S3, S2, S1, S0 كل منها يمكن أن يكون واحدا أو صفرا لذلك فإن هناك 32 (2^5) شفرة أو كود يمكن أن تكون عليها هذه الخطوط ولكل شفرة من هذه الشفرات سيكون هناك خرج معين لدائرة المجمع/الطارح الموضحة فى شكل (6-6) . شكل (6-7) يبين جميع هذه الشفرات والخرج الناتج عن كل منها . إننا هنا لن نقوم بمراجعة جميع الحالات الموجودة فى شكل (6-7) لمعرفة كيف يكون الخرج فى كل حالة ومطابقة ذلك على الدائرة الموجودة فى شكل (6-6) ولكننا سنترك هذه العملية للقارىء اكتفاء بالمثالين اللذين شرحناهما سابقا أحدهما للجمع والآخر للطرح .

من الملاحظات المهمة على الخرج أن هناك الكثير من الحالات قد يكون فيها الخرج غير مهم مثل الحالات التى يكون فيها الخرج يساوى 0 أو 1- أو 2- وغير ذلك من الحالات ، ويلاحظ أيضا وجود الكثير من الحالات التى يكون الخرج فيها مكررا مثل الحالة التى يكون فيها الخرج يساوى B قد كررت ثلاث مرات والحالة التى يكون فيها الخرج يساوى 0 كررت أيضا أكثر من مرة. كيف سنتخلص من هذا التكرار وهذه الحالات التى نعتبرها غير مهمة ؟ إن هذا ما سنراه فى الأجزاء القادمة .

6-4 وحدة الحساب والمنطق
Arithmatic and Logic Unit, ALU

من العمليات التى يقوم بها المعالج دائما بجانب العمليات الحسابية ، العمليات المنطقية أيضا . لذلك فإننا سنحاول فى هذا الجزء وعن طريق إضافة بعض خطوط التحكم أن نجعل الدائرة الموجودة فى شكل (6-6) قادرة على تنفيذ العمليات المنطقية أيضا بجانب العمليات الحسابية (الجمع والطرح) . إن العمليات المنطقية التى سنحاول إضافتها إلى وحدة المجمع/الطارح التى سبق شرحها هى عمليات XOR, OR, AND وهذا كمثال فقط حيث بالطبع يمكن إضافة المزيد .

إن هناك أكثر من طريقة لتطوير دائرة المجمع/الطارح التى سبق شرحها لتستطيع تنفيذ العمليات المنطقية وسنعرض هنا أبسط هذه الطرق . شكل (6-8) يبين وحدة حساب ومنطق تستطيع تنفيذ عمليات الجمع , والطرح XOR, OR, AND . لاحظ فى هذا الشكل أن الدخلين B و A كل منهما مكون من n من البتات ولكن لتبسيط الرسم تم رسمه كخط واحد فقط . شكل (6-8) مكون من

أربعة صناديق كل منها يمثل عملية من عمليات وحـدة الحســاب والمنطــق . الصندوق الأول خاص بعمليتى الجمع والطرح ومحتوياته هى الدائرة الموجودة فى شكل (6-6) .

s4	s3	s2	s1	s0	الخـــــرج
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	-1
0	0	0	1	1	0
0	0	1	0	0	-1
0	0	1	0	1	0
0	0	1	1	0	-2
0	0	1	1	1	-1
0	1	0	0	0	B
0	1	0	0	1	B+1
0	1	0	1	0	-B-1=B
0	1	0	1	1	-B
0	1	1	0	0	B-1
0	1	1	0	1	B
0	1	1	1	0	-B-2
0	1	1	1	1	-B-1=B
1	0	0	0	0	A
1	0	0	0	1	A+1
1	0	0	1	0	A-1
1	0	0	1	1	A
1	0	1	0	0	-A-1=A
1	0	1	0	1	-A
1	0	1	1	0	-A-2
1	0	1	1	1	-A-1=A
1	1	0	0	0	A+B
1	1	0	0	1	A+B+1
1	1	0	1	0	A-B-1
1	1	0	1	1	A-B
1	1	1	0	0	B-A-1
1	1	1	0	1	B-A
1	1	1	1	0	-A-B-2
1	1	1	1	1	-A-B-1

شكل (6-7) جميع الحالات الممكنة لخرج المجمع/الطارح فى شكل (6-6)

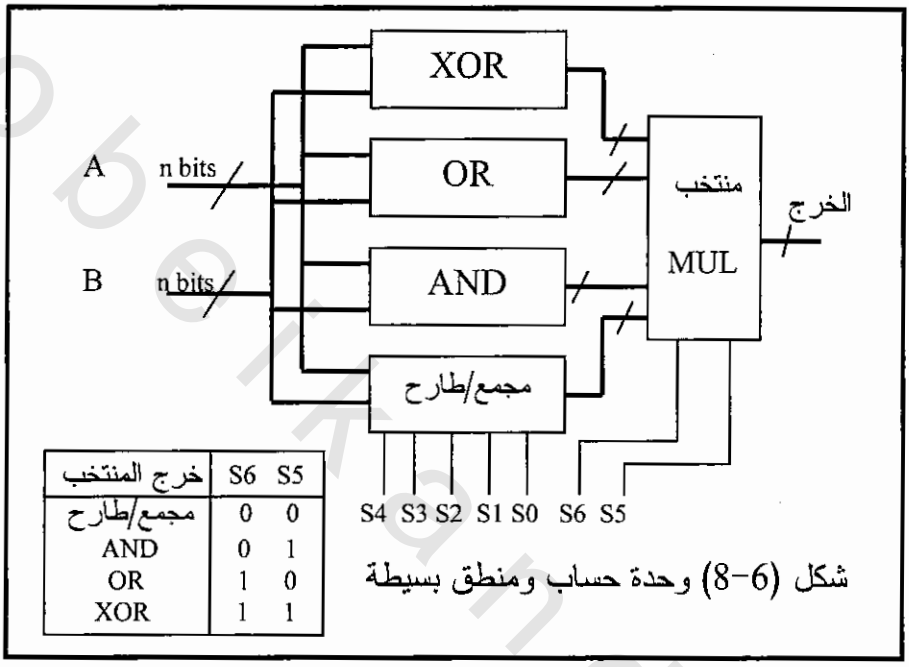
الصندوق الثانى خاص بعملية AND ويحتوى عددا n من بوابات AND ثنائية الدخل حيث أحد هذين الدخلين هو بت من بتات الدخل A والدخل الآخر هو البت المناظرة من الدخل B . الصندوق الثالث خاص بعملية OR ومحتوياته هى عدد n من بوابات OR ثنائية المدخل ، وأما الصندوق الرابع فخاص بعملية XOR ومحتوياته هى عدد n من بوابات XOR .

الخرج النهائى لوحدة الحساب والمنطق يتم اختياره من بين خروج الأربعة صناديق السابقة عن طريق منتخب multiplexer له خطى تحكم S6, S5 تتم بهما عملية اختيار أى واحد من الصناديق سيتم توصيل خرجه إلى خرج المنتخب وبالتالى إلى خرج وحدة الحساب والمنطق ، فمثلا إذا كان S6=1, S5=1 فإن خرج صندوق XOR يوصل إلى خرج المنتخب وفى هذه الحالة فإن وحدة الحساب والمنطق ستقوم بتنفيذ عملية XOR على الدخلين A, B . شكل (6-8) يبين أيضا جدول الحقيقة لهذا المنتخب لجميع حالات الخطين S6, S5 . لاحظ أن عملية تخصيص أى شفرة على الخطين S6, S5 ستخرج XOR وأيها ستخرج AND وغير ذلك من العمليات الموضحة فى شكل (6-8) تتوقف على المصمم إذ هو الذى يحدد هذه الشفرات . عندما يكون الخطان S6=0, S5=0 فإن خرج وحدة الحساب والمنطق سيكون هو خرج الصندوق المجمع/الطارح وأما العملية التى ستنفذ فستكون إما عملية جمع أو طرح أو غير ذلك على حسب الشفرة الموجودة على باقى خطوط التحكم S0 إلى S4 .

الدائرة الموجودة فى شكل (6-8) تحتوى على سبعة خطوط تحكم S0 إلى S6 وعلى ذلك فإن هذه الدائرة مفروض أن تكون قادرة على إجراء 128 (2^7) عملية مختلفة بناء على جدول حقيقة يمكن وضعه مماثلا للجدول المبين فى شكل (6-7) . ولكن كما رأينا فى شكل (6-7) فإن معظم هذه العمليات (128) إما أنها ستكون عمليات غير مهمة أى غير ذات فائدة أو أنها ستكون عمليات مكررة . لذلك فإننا سنقوم بتصفية هذه 128 عملية إلى 13 عملية فقط من العمليات المهمة والغير مكررة وسنهمل باقى العمليات . لاحظ أن اختيار 13 عملية تعتبر من عمل المصمم حيث هو الذى يختار عدد العمليات المهمة وأى العمليات تهمل ؟ وأيها يؤخذ فى الإعتبار ؟ ولذلك فإن اختيارنا 13 عملية هنا ليس إلا مجرد مثال فقط ولا توجد أى ضرورة لاختيار الرقم 13 بالذات .

بما أننا سنختار 13 عملية فقط وسنهمل الباقى فإنه من البديهى أن يكون هناك 4 خطوط تحكم فقط كافية لتشفير هذه العمليات حيث $2^4 = 16$ (أكبر من 13) . لذلك فإننا سنحتاج هنا إلى دائرة تكون مهمتها هى تحويل الشفرات الرباعية إلى شفرات سباعية تتناسب مع خطوط التحكم الموجودة فى وحدة الحساب والمنطق المبينة فى شكل (6-8) ، هذه الدائرة سنسميها محول شفرات وعادة ما تكون

هذه الدائرة عبارة عن ROM مكونة من 16 بايت ولها أربعة خطـوط عنـاوين تعطى عليها الشفرة الرباعية فتخرج محتويات البايت المقابلـة وهـى الشـفرة السباعية التى من المفروض أن تكون مخزنة سلفا .



خرج المنتخب	S6	S5
مجمع/طارح	0	0
AND	0	1
OR	1	0
XOR	1	1

شكل (6-8) وحدة حساب ومنطق بسيطة

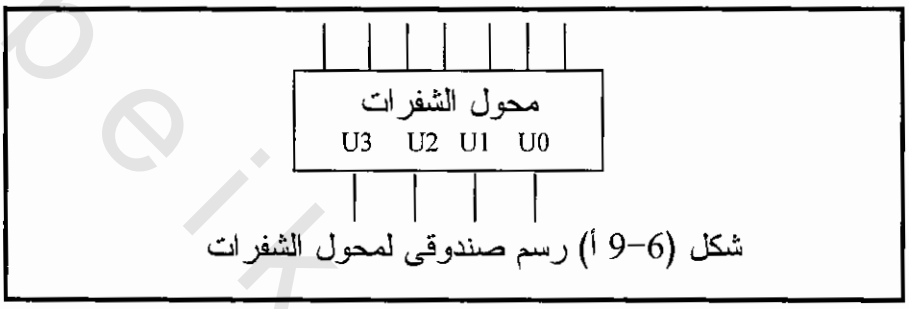
شكل (6-9) يبين رسما صندوقيا وجدول الحقيقة لمحول الشفرات . شكل (6-10) يبين الشكل النهائى لوحدة الحساب والمنطق بعد إضافة محول الشفرات إليهـا . لإجراء عملية الجمع مثلا يجب أن نضع الشفرة الرباعية 1 1 1 0 على خطـوط التحكم U3 U2 U1 U0 فيقوم محول الشفرات بإعطاء الشفرة السباعية المنـاظرة والتى هى الشفرة السباعية لعملية الجمع (0 0 1 1 0 0 0) كما فى جدول الحقيقـة فى شكل (6-9) .

6-5 مسجل التراكم
Accumulator register

إن إضافة مسجل التراكم Accumulator إلى وحدة الحساب والمنطق هو الخطوة التالية لتطوير هذه الوحدة وإعدادها لتكون جزءا من أجزاء المعالج . شكل (6-11) يوضح هذا التطوير ، من هذا الشكل نلاحظ أن مسجل التراكم يعتبـر مخزنـا لتسجيل الناتج من وحدة الحساب والمنطق حيث أنه موصل مباشرة على خرجهـا ،

obeikandl.com

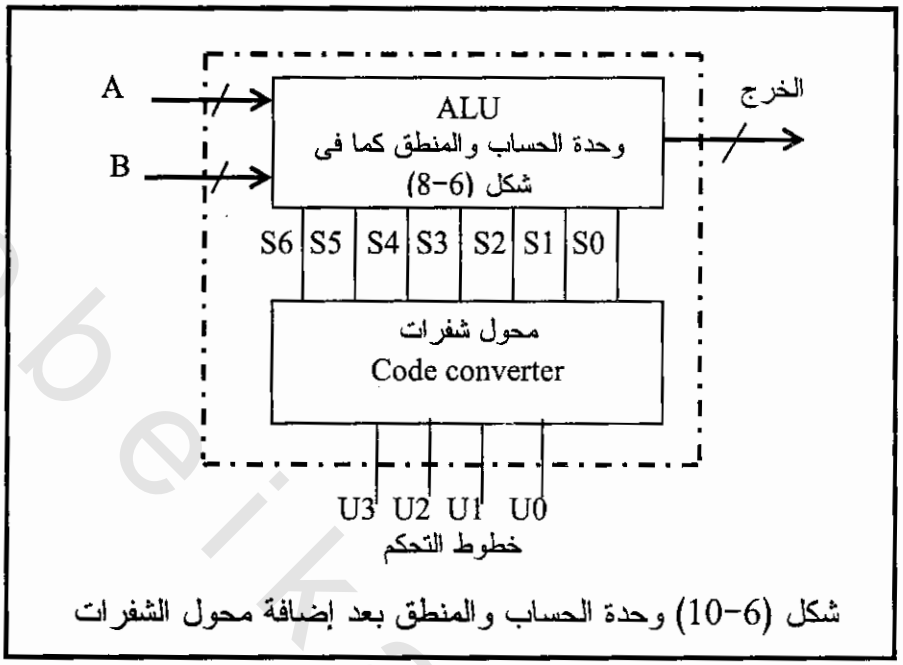
ولذلك فإننا نتوقع أن ناتج أى عملية حسابية أو منطقية سيذهب إلى مسجل التراكم . نلاحظ أيضا من شكل (6-11) أن هناك نوعا من أنواع التغذية المرتدة حيث قد تم توصيل خرج مسجل التراكم على أحد دخلى وحدة الحساب والمنطق (الدخل A) ولم يبق سوى دخل واحد فقط لوحدة الحساب والمنطق (الدخل B) . نلاحظ أيضا أن مسجل التراكم مثله مثل أى مسجل حيث لن ينتقل دخله إلى خرجه إلا بعد إعطاء نبضة تزامن clock ، لذلك كان لابد من توصيل clock إلى مسجل التراكم .



شكل (6-9 أ) رسم صندوقى لمحول الشفرات

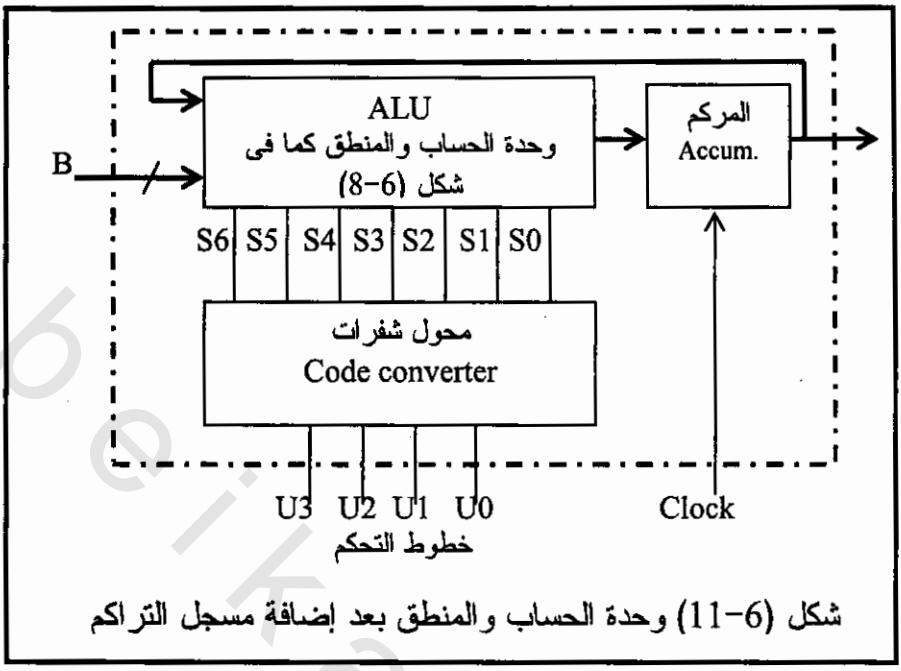
U3	U2	U1	U0	العملية	s6	s5	s4	s3	s2	s1	s0
0	0	0	0	A	0	0	1	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	1
0	0	1	0	A	0	0	1	0	1	0	0
0	0	1	1	B	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	A+1	0	0	1	0	0	0	1
0	1	1	0	A-1	0	0	1	0	0	1	0
0	1	1	1	A+B	0	0	1	1	0	0	0
1	0	0	0	A-B	0	0	1	1	0	1	1
1	0	0	1	A∧B	1	0	d	d	d	d	d
1	0	1	0	A∨B	0	1	d	d	d	d	d
1	0	1	1	A+B	1	1	d	d	d	d	d
1	1	0	0	-1	0	0	0	0	1	1	1
1	1	0	1	هذه الحالات الثلاث							
1	1	1	0	غير مستخدمة الآن							
1	1	1	1	وسوف نستخدمها مستقبلا							
* d تعنى do not care أى لا يهم أن يكون هذا الخط صفر أو واحد .											

شكل (6-9ب) جدول الحقيقة لمحول الشفرات



شكل (6-10) وحدة الحساب والمنطق بعد إضافة محول الشفرات

بعد هذا التعديل بإضافة مسجل التراكم إلى وحدة الحساب والمنطق لابد وأن يطرأ تعديل مناظر على ال 13 عملية التى تم تخصيصها لهذه الوحدة للقيام بها والتى سبق أن حددناها سابقا بعد إضافة محول الشفرات فى شكل (6-9) . شكل (6-12) يبين جدولا لهذه العمليات وقد تم إعطاء شفرة حرفية mnomenic أو شفرة أسمبلى لكل عملية بجانب شفرتها الثنائية (الشفرة الرباعية) أو التى سنسميها شفرات الماكينة machine codes . لنأخذ على ذلك المثال التالى : افترض أن خطوط التحكم U3 U2 U1 U0 تساوى 0101 فإنه من شكل (6-9) ستكون العملية (الأمر) التى ستنفذ هى A+1 ، أى أن الدخل A لوحدة الحساب والمنطق الذى هو خرج مسجل التراكم سيزداد بمقدار واحد . بافتراض أن خرج مسجل التراكم كان صفرا فى البداية فإن ال ALU ستجمع واحد زائد صفر وتكون النتيجة واحدا يوضع على خرج ال ALU . مع أول نبضة تزامن فإن هذا الواحد ينتقل إلى خرج مسجل التراكم حيث سيكون موجودا أيضا على الدخل A لل ALU . ومع بقاء نفس الشفرة على الخطوط U0 إلى U3 فإن هذا الواحد سيصبح اثنين على خرج ال ALU . بإعطاء نبضة التزامن الثانية ستنتقل الاثنين إلى خرج مسجل التراكم والدخل A لل ALU وهكذا تستمر العملية . أى أن هذا الأمر يزيد واحدا على محتويات مسجل التراكم مع كل نبضة تزامن كما لو كان عدادا ولذلك فقد تم اختصاره إلى INC أى زد بمقدار واحد Increment .



شكل (6-11) وحدة الحساب والمنطق بعد إضافة مسجل التراكم

كما نلاحظ من شكل (6-11) فإن خرج مسجل التراكم متصل دائما بالدخل A لوحدة الحساب والمنطق لذلك فإنه دائما يطلق عليه إسم المسجل A ، وبما أن النتائج دائما تتراكم فيه كما رأينا فى المثال السابق فقد أطلق عليه أيضا اسم مسجل التراكم أو المركم Accumulator . لنأخذ مثالا آخر أكثر تعقيدا لكى نفهم كيفية عمل الدائرة الموجودة فى شكل (6-11) . افترض أن المطلوب هو ضرب العددين 3×6 . كما نرى من شكل (6-12) فإنه ليس هناك أى أمر يقوم بعملية الضرب ، لذلك فإننا سننفذ الضرب عن طريق الجمع المتكرر ، أى أننا سنجمع العدد 6 مع نفسه ثلاث مرات . لذلك فإننا سنضع العدد 6 على الدخل B أولا ثم سنضع الشفرة 0011 على خطوط التحكم U0 إلى U3 وهذه الشفرة من شأنها أن تنقل الموجود على الدخل B إلى خرج مسجل التراكم مع أول نبضة تزامن . أى أنه بعد أول نبضة تزامن سيكون العدد 6 موجودا على خرج مسجل التراكم وبالتالى أيضا على الدخل الآخر (الدخل A) لوحدة الحساب والمنطق . الآن سنضع الشفرة 0111 على خطوط التحكم U0 إلى U3 وهذه الشفرة كما فى شكل (6-12) من شأنها أن تجمع محتويات الدخل B مع محتويات الدخل A لل ALU . وعلى ذلك فإنه بإعطاء نبضة تزامن ثانية ستكون محتويات مسجل التراكم تساوى12 وهى حاصل جمع 6 الموجودة على الدخل B و 6 الموجودة على خرج مسجل التراكم بعد النبضة الأولى . لاحظ أن الدخل A لل ALU بعد

النبضة الثانية سيصبح 12 أيضا . لو احتفظنا بنفس الشفرة 0111 على خطوط التحكم U0 إلى U3 وأعطينا نبضة تزامن أخرى فإن خرج مسجل التراكم سيصبح 18 وهى حاصل جمع 6 الموجودة على الدخل B و 12 الموجودة على خرج مسجل التراكم بعد النبضة الثانية . بذلك نكون قد أنهينا عملية ضرب العددين 3×6 بعد ثلاث نبضات تزامن وبعد أن وضعنا الشفرات التالية على خطوط التحكم U0 إلى U3 :

	U3	U2	U1	U0
أول نبضة تزامن	0	0	1	1
ثانى نبضة تزامن	0	1	1	1
ثالث نبضة تزامن	0	1	1	1

من المثال السابق يمكننا أن نقول أن أى عملية مركبة يمكن تبسيطها إلى مجموعة من الشفرات التى توضع على خطوط التحكم U0 إلى U3 واحدة بعد الأخرى بحيث تنفذ العملية المناظرة لكل شفرة مع نبضة تزامن . الآن ألا يمكننا أن نسمى كل شفرة من هذه الشفرات بالأمر Instruction ، ومجموعة الشفرات أو الأوامر المطلوبة لتنفيذ أى عملية مركبة ألا نسميها برنامج program ، ألا يمكن أن نسمى الدائرة التى لها قائمة الأوامر الموجودة فى شكل (6-12) بالمعالج . نعم إن الدائرة التى حصلنا عليها فى شكل (6-11) تعتبر صورة مبسطة جدا لمعالج افتراضى محدود الإمكانيات إذا ما قورن بأى معالج حقيقى . إن المعالج الإفتراضى hypothetical الذى وصلنا إليه حتى الآن يستطيع فقط إجراء العمليات الحسابية والمنطقية ، السؤال الآن هل نستطيع تطوير الوحدة السابقة لكى تستطيع التعامل مع ذاكرة ووحدات إدخال أو إخراج كما فى المعالج العادى ؟ إن هذا ما سنراه فى الجزء التالى :

6-6 إضافة ذاكرة للمعالج الإفتراضى

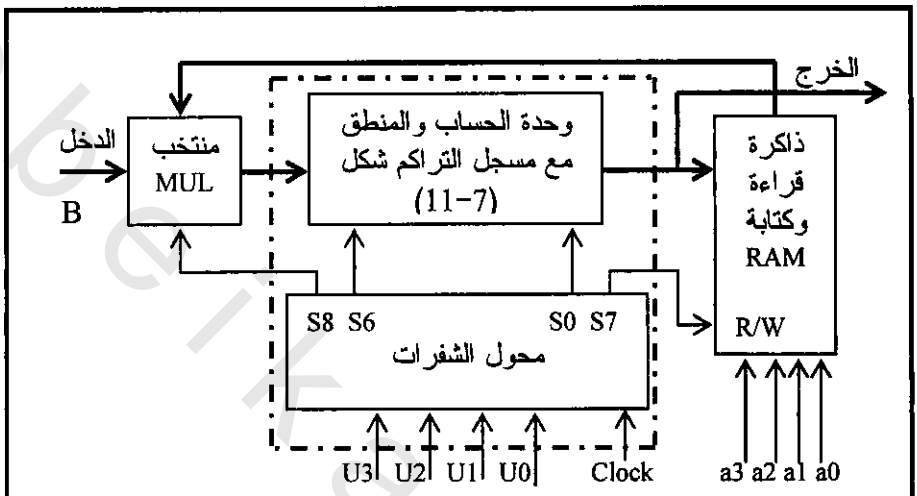
بالنظر إلى الدائرة الموجودة فى شكل (6-13) نجد أن هناك منتخبا تمت إضافته فى الدخل وهذا المنتخب له دخلان وخط تحكم واحد S8 تمت إضافته على خرج محول الشفرات لهذا الغرض . أحد دخلى هذا المنتخب هو خرج الذاكرة والدخل الآخر هو الدخل B حيث يتم اختيار أحدهما ليعبر إلى خرج المنتخب ومنه كدخل إلى وحدة الحساب والمنطق ثم إلى مسجل التراكم عن طريق خط التحكم S8 . وعلى ذلك فعندما يكون S8=0 فإن خرج الذاكرة يكون موصلا إلى وحدة الحساب والمنطق ، وأما إذا كان S8=1 فإن الدخل B يوصل إلى وحدة الحساب والمنطق .

U3 U2 U1 U0	اختصار العملية	العمليـــــــة	الحمل Carry
0 0 0 0	NOP	لاتعمل شيء	no
0 0 0 1	SP1	A = +1	no
0 0 1 0	CMA	اعكس A	no
0 0 1 1	LDA	حمل A بالدخل B	no
0 1 0 0	CLA	A = 0	no
0 1 0 1	INC	A+1 → A	yes
0 1 1 0	DEC	A-1 → A	yes
0 1 1 1	ADD	A+B → A	yes
1 0 0 0	SUB	A-B → B	yes
1 0 0 1	AND	A∧B → A	no
1 0 1 0	OR	A∨B → A	no
1 0 1 1	XOR	A⊕B → A	no
1 1 0 0	SM1	-1 → A	no
1 1 0 1			
1 1 1 0			
1 1 1 1			
* A مقصود بها مسجل التراكم و B يقصد بها الدخل B لوحدة الحساب والمنطق			

شكل (6-12) قائمة بالعمليات التى تنفذها وحدة الحساب والمنطق التى معها مركم كما فى شكل (6-11)

المشكلة الآن أننا نريد توصيل أحد هذين الدخلين إلـى مسـجل التـراكم فـأى الشفرات نضعها على خطوط التحكم U0 إلى U3 ؟ ليس هناك الا شفرة واحـدة فقط وهى الشفرة 0011 التى تضع دخل وحدة الحساب والمنطق فـى مسـجل التراكم وهى شفرة الأمر LDA وكما رأينا فإن دخل وحدة الحساب والمنطق إما أن يكون من الذاكرة أو من B . لذلك فإنه لتمييز كل من الدخلين من الآخر فإننا سنخصص الشفرة 0011 التى هى الأمر LDA لتحميل المركم بمحتويات دخـل وحدة الحساب والمنطق إذا كان هذا الدخل قادما من الذاكرة أى S8=0 . ولذلك فإنه فى هذه الحالة(S8=0) لابد من تحديد العنوان الذى سيتم التعامل معه علـى خطوط عنوان الذاكرة a0 إلى a3 . أما إذا كان الـدخل قادمـا مـن B فإننـا سنستخدم له شفرة من الشفرات الغير مستخدمة والمتبقيـة مـن ال 16 شـفرة

الموجودة فى قائمة الأوامر ولتكن الشفرة 1101 والتى سنعطيها الاختصار INP أى الدخل من خارج المعالج مثل بوابة إدخال مثلا . لاحظ أنه مع الأمر INP فإن خط التحكم S8 يكون واحدا فى هذه الحالة وستكون الشفرة المقابلة على خرج محول الشفرات S0 إلى S8 هى 00001000 .



شكل (6-13) وحدة الحساب والمنطق وقد تم توصيلها على الذاكرة

بذلك نكون قد رأينا كيفية اختيار دخل وحدة الحساب والمنطق بين الذاكرة أو الدخل B القادم من خارج المعالج ، ماذا عن خرج وحدة الحساب والمنطق وكيف يمكن توصيله إلى الذاكرة ؟

إن خط التحكم S7 الذى أضيف على خرج محول الشفرات كما فى شكل (6-13) مهمته هى التحكم فى كتابة خرج مسجل التراكم فى الذاكرة إذا كان S7=1 أو إخراجه إلى خارج المعالج عندما يكون S7=0 . لاحظ أن كلا من هاتين العمليتين ليس لها أمر يقابلها فى قائمة الأوامر التى درسناها حتى الآن فى شكل (6-12) ولذلك فإننا سنستخدم الشفرتين المتبقيتين من ال 16 شفرة فى نفس الشكل لهذا الغرض ، أى أن الشفرة 1110 والتى سنختصرها STA أى خزن فى الذاكرة Store accumulator ستجعل الخط S7=1 الذى سيجعل الذاكرة فى حالة استقبال للمعلومات أى فى حالة كتابة فيها لأن الخط R/W=1 . لاحظ أن هذا الأمر سيتطلب أن نضع العنوان الذى سيتم التعامل معه داخل الذاكرة على مسار العناوين (a0 إلى a3) .

U3 U2 U1 U0	العنـــــوان a3 a2 a1 a0	الأمر	العمـليـــة	الحمل
0 0 0 0	x x x x	NOP	لاتعمل شيء	no
0 0 0 1	x x x x	SP1	A = +1	no
0 0 1 0	x x x x	CMA	اعكس A	no
0 0 1 1	a a a a	LDA	A ⟵ الذاكرة	no
0 1 0 0	x x x x	CLA	A = 0	no
0 1 0 1	x x x x	INC	A+1 ⟶ A	yes
0 1 1 0	x x x x	DEC	A-1 ⟶ A	yes
0 1 1 1	a a a a	ADD	A ⟵ A +(عنوان)	yes
1 0 0 0	a a a a	SUB	A ⟵ A -(عنوان)	yes
1 0 0 1	a a a a	AND	A ⟵ A∧(عنوان)	no
1 0 1 0	a a a a	OR	A ⟵ A∨(عنوان)	no
1 0 1 1	a a a a	XOR	A ⟵ A⊕(عنوان)	no
1 1 0 0	x x x x	SM1	-1 ⟶ A	no
1 1 0 1	x x x x	INP	B ⟶ A	no
1 1 1 0	a a a a	STA	A ⟵ الذاكرة	no
1 1 1 1		OUT	A ⟵ الخرج	no

شكل (6-14) قائمة الأوامر بعد تعديلها لتلائم عملية الإتصال بالذاكرة

أما الشفرة 1111 المتبقية فسوف نخصصها لإخراج محتويات المركم خارج المعالج ولذلك سنرمز لها بالرمز OUT أى إخراج output حيث سيكون الخط S7=0 فى هذه الحالة . شكل (6-14) يبين قائمة الأوامر بعد تعديلها لتناسب عملية التوصيل للذاكرة وإضافة الأوامر الجديدة إليها .

إننا بعد دراسة هذا الفصل يجب أن نقف وقفة تفكير فى الفرق بين هذا المعالج الإفتراضى وأى واحد من المعالجات الحقيقية التى درسناها فى الفصول السابقة. أسئلة كثيرة يمكن أن ترد هنا بعد دراسة الأشكال المختلفة والمتعددة التى رأيناها فى هذا الفصل . كيف سيكون شكل المعالج لو أننا رسمناه على أساس 8 بتات أو 16 أو حتى 32 بت ؟ وماذا لو جعلنا الشفرة الداخلة إلى محول الشفرات شفرة من 16 بت بدلا من الشفرة الرباعية ؟ وكيف سيكون شكل هذه الدائرة إذا أضفنا بعض العمليات الأخرى مثل عمليات دوران محتويات المسجل A إلى اليمين أو إلى اليسار وعمليات الإزاحة من اليمين أو اليسار وعمليات الضرب والقسمة ؟ كيف سيكون شكل هذه الدائرة لو أضفنا لهذه الوحدة بعض المسجلات عامة الأغراض التى تستخدم لأغراض التخزين وتبادل المعلومات مع المسجل

A ؟ كيف سيكون شكل الدائرة لو أن الوحدة تتعامل مع ذاكرة لها 16 أو 32 خطا من خطوط العناوين بدلا من 4 كما افترضنا ؟ كل هذه أسئلة توضح مدى بساطة فكرة عمل المعالج ولكنها فى نفس الوقت توضح مدى تعقيد تركيبه ومكوناته .

6-7 تمارين

1. دائرة نصف المجمع الموجودة فى شكل (6-2) يمكن بناؤها باستخدام بوابات XOR ، وضح ذلك ؟
2. دائرة المجمع الكامل الموجودة فى شكل (6-4) يمكن بناؤها باستخدام نصفى مجمع ، وضح ذلك ؟
3. هناك بعض الشرائح التى تعمل كمجمع ، اذكر واحدة من هذه الشرائح واشرح طريقة عملها واختبرها معمليا ؟ استعن بكتاب data book دليل الشرائح .
4. يلعب المتمم الثنائى دورا مهما فى تحويل عملية الطرح إلى عملية جمع ، وضح ذلك بالشرح ؟
5. ما هو دور وحدة الحساب والمنطق فى المعالج ؟
6. هناك بعض الشرائح التى تعمل كوحدة حساب ومنطق ، اذكر واحدة من هذه الشرائح ، واشرح طريقة عملها واختبرها معمليا ؟
7. شكل (6-9) يبين محول شفرات رباعية إلى سباعية ، ماذا يحدث لو استخدمنا محول شفرات خماسية إلى سباعية بدلا منه ؟ اكتب قائمة العمليات الجديدة بعد العمليات الاضافية ؟
8. هل مسجل التراكم الموجود فى شكل (6-11) يقوم بنفس دور مسجل التراكم فى أى معالج من حيث أنه يكون طرفا فى أى عملية حسابية أو منطقية والنتيجة تخزن فيه ؟
9. ما مقدار الذاكرة التى يستطيع المعالج الافتراضى الموجود فى شكل (6-13) أن يتعامل معها ؟ وكيف نزيد هذه الكمية ؟
10. لو أردنا إضافة مسجل أوامر للوحدة الموجودة فى شكل (6-13) فأين يكون ، وضح ذلك .