الفصل الثالث

المسجلات
ووحدة الحساب والمنطق

Registers
and Arithmetic Logic Unit

في هذا الفصل:

1.3- مقدمة.
2.3- بنية وحدة المعالجة المركزية.
3.3- المسجلات.
4.3- تناقل البيانات بين المسجلات.
5.3- العمليات الصغرية.
6.3- تصميم وحدة الحساب والمنطق.
7.3- تمارين.

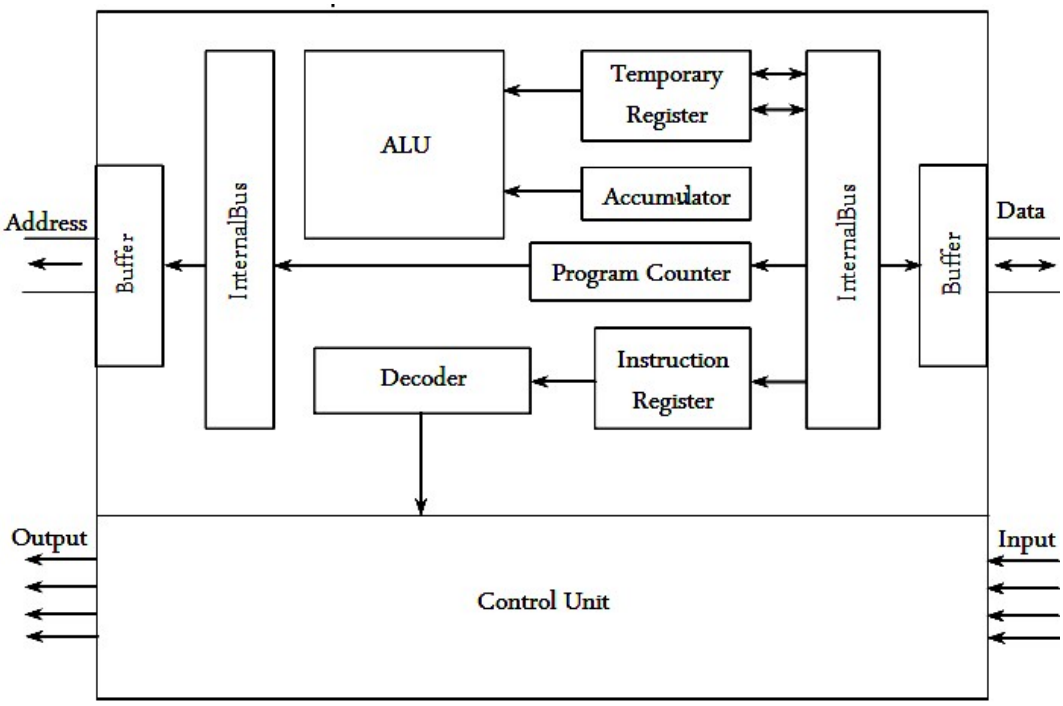
1.3– مقدمة (Introduction):

تتصل الأنظمة الرقمية داخلياً مع وحدات مادية (Hardware Modules) من أجل تنفيذ مهام معالجة البيانات. تتألف هذه الوحدات من مسجلات (Registers)، فاك شيفرة (Decoder)، عناصر حسابية (Arithmetic Elements)، وبوابات منطقية (Logic Gates)، وتتصل هذه الوحدات ببعضها من خلال ممر معطيات داخلي مشترك (Common Internal Bus).

سنناقش في هذا الفصل بنية المسجلات وكيفية وصلها مع ممرات النظـــام، بالإضـــافة إلى العمليات المنفذة على البيانات المخزنة فيها، والتي تسمى العمليات الصغرية (Microoperations).

2.3– بنية وحدة المعالجة المركزية (CPU Structure):

يبين الشكل (1.3) مخططاً توضيحياً لوحدة المعالجة المركزية.



الشكل (1.3): بنية وحدة المعالجة المركزية.

نلاحظ أن هذا المخطط مؤلف من العناصر التالية:

- ممر المعطيات (Data Bus).
- ممر العناوين (Address Bus).
- وحدة التحكم (Control Unit)، وخطوط التحكم (Control Lines).
- عوازل (Buffers).

- وحدة الحساب والمنطق (Arithmetic Logic Unit: ALU).
- ممر داخلي (Internal Bus).
- فاك شيفرة التعليمة (Instruction Decoder).
- مسجلات (Registers): مثل المسجل المؤقت (Temporary Register) والمراكم (Accumulator) وعداد البرنامج (Program Counter) ومسجل التعليمة (Instruction Register). سيأتي شرح هذه المسجلات في فقرات لاحقة.

3.3- المسجلات (Registers):

تعتبر المسجلات من أهم الدارات التتابعية، وهي مؤلفة من عدد من القلابات، وكل قلاب يمثل خانة من خانات المسجل التي تخزن فيها إحدى القيمتين المنطقيتين "0" أو "1". تشغل دراسة المسجلات أهمية كبيرة في دراسة بنية الحاسوب، ويرمز عادةً لها بأحرف كبيرة متناسبة مع وظيفة كل منها.

تتألف مسجلات النظام من نوعين أساسيين هما:

1.3.3- مسجلات الأغراض العامة (General Purpose Registers: GPR):

تستخدم هذه المسجلات من أجل التخزين المؤقت للبيانات، وتعمل على نقل هذه البيانات من وإلى نوافذ الدخل والخرج، وتتعامل هذه المسجلات أيضاً مع الذاكرة. من أهم هذه المسجلات المسجل المؤقت (Temporary Register: TR).

2.3.3- مسجلات الوظائف الخاصة (Special Function Registers: SFR):

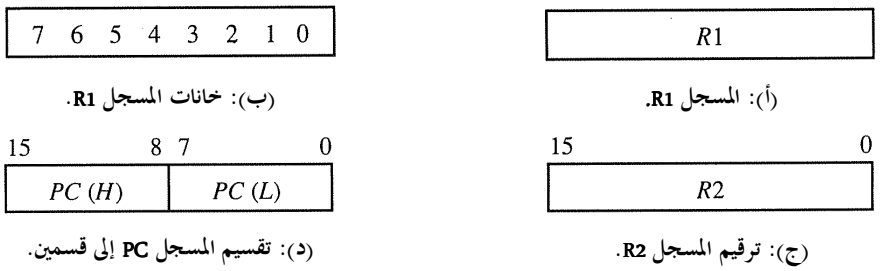
عبارة عن مسجلات تستخدم من قبل المعالج والأجزاء المحيطية للتحكم بالعمليات المطلوبة التي يقوم بها المعالج، ومن أهم هذه المسجلات نذكر التالي:

- عداد البرنامج (Program Counter:PC): يحتوي على عنوان التعليمة التالية التي ستنفذ في البرنامج، وهو نفسه عداد التعليمة (Instruction Counter).
- مسجل التعليمة (Instruction Register: IR): يحتوي على التعليمة التي تنفذ حالياً.
- المراكم (Accumulator: AC): يستخدم كمسجل مساعد في العمليات الحسابية، حيث يمكن أن يحوي أحد معاملي العملية، وربما يخزن فيه نتيجة هذه العملية.
- مسجل العنوان (Address Register: AR): يحتوي على عنوان التعليمة الحالية أو عنوان المعامل (Operand).

- مسجل البيانات (Data Register: DR): يحتوي على البيانات اللازمة لتنفيذ التعليمة.

بفرض لدينا مسجل مؤلف من m خانة، فإنها ترقم من الصفر وحتى m-1، إذ يبدأ الصفر من الخانة الواقعة أقصى اليمين (البت الأقل أهمية Least Significant Bit: LSB) وتزداد الأرقام باتجاه اليسار (البت الأكثر أهمية Most Siginificant Bit: MSB).

تمثل المسجلات عادةً بصندوق يحوي داخله اسم المسجل كما يوضح ذلك الشكل (2.3-أ)، كما ويمكن تمييز خاناته كما في الشكل (2.3-ب). أيضاً، يمكن ترقيم خانات المسجل كما في الشكل (2.3-ج)، والذي يبين المسجل R2 بطول 16 بت، وقد تم ترقيم أول وآخر خانة فوق صندوقه. قد يقسم مسجل بطول 16 بت إلى قسمين: سفلي (Low: L) يضم الخانات من صفر إلى 7، وعلوي (High: H) يضم الخانات من 8 وحتى 15، كما في الشكل (2.3-د) الذي يوضح كيفية تقسيم المسجل PC بهذه الطريقة.



(أ): المسجل R1.

(ب): خانات المسجل R1.

(ج): ترقيم المسجل R2.

(د): تقسيم المسجل PC إلى قسمين.

الشكل (2.3): المخطط الصندوقي للمسجل.

4.3- تناقل البيانات بين المسجلات (Registers Data Transfer):

تعني هذه العملية نقل محتوى مسجل مصدر (Source) إلى مسجل آخر هدف (Destination) دون تغيير محتوى المسجل المصدر، ويشار إلى عملية النقل وفق الآتي:

Destination ← Source

مثال: بعد تنفيذ العملية R1 ← R2 ، سينتقل محتوى المسجل R1 إلى المسجل R2 دون أن يتغير محتوى المسجل R1.

يوضح الجدول (1.3) بعض الرموز المستخدمة عند كتابة عمليات النقل: كالحروف والأقواس والأسهم والفواصل، ووصفها.

1.4.3- انتخاب المسجل الهدف والمصدر (Source and Destination Register Selection):

عادةً، تتم عملية النقل بين مسجلين تحت شروط معينة وليس بشكل عشوائي، حيث تخضع عملية النقل لتوليد إشارات تحكم من قبل وحدة التحكم، وهي إشارات منطقية تأخذ القيمة "0" أو "1".

Symbol	Description	Examples
Letters (and numerals)	Denotes a register	*MAR*, *R2*
Parentheses ( )	Denotes a part of a register	*R2*(0–7), *R2*(*L*)
Arrow ←	Denotes transfer of information	*R2* ← *R1*
Comma ,	Separates two microoperations	*R2* ← *R1*, *R1* ← *R2*

الجدول (1.3): الرموز الأساسية في تناقل المسجلات.

1.1.4.3– انتخاب المسجل الهدف (Destination Register Selection):

نريد نقل محتويات المسجل R1 إلى R2، وذلك عندما يتم انتخاب R2. يوضح الشكل (3.3–أ) المخطط الصندوقي لهذه العملية، حيث يكون لـــ R2 مدخل تحميل (Load) يتم تفعيله بوساطة إشارة التحكم P المولدة من دارة التحكم.

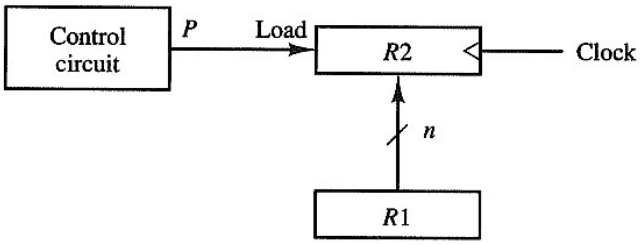
نستطيع كتابة التعبير التالي:

$$\text{If } (P = 1) \text{ Then } (R2 \leftarrow R1)$$

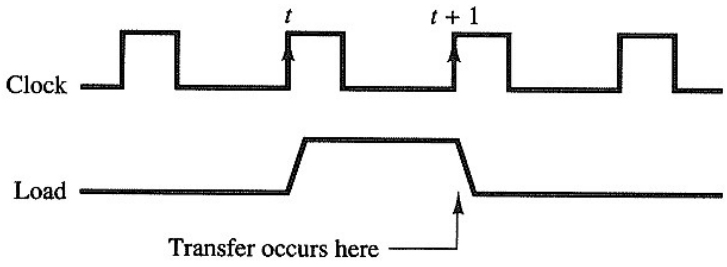
والذي يمكن كتابته أيضاً كمايلي:

$$P: R2 \leftarrow R1$$

بما أن عمليات النقل بين المسجلات هي عمليات متزامنة مع نبضات الساعة، وبالتالي حتى تتم عملية النقل لا بد من توافر إشارة التحكم المطلوبة (P=1 أي Load=1) مع نبضات الساعة كما يبين ذلك الشكل (3.3–ب).



(أ): المخطط الصندوقي.



(ب): المخطط الزمني.

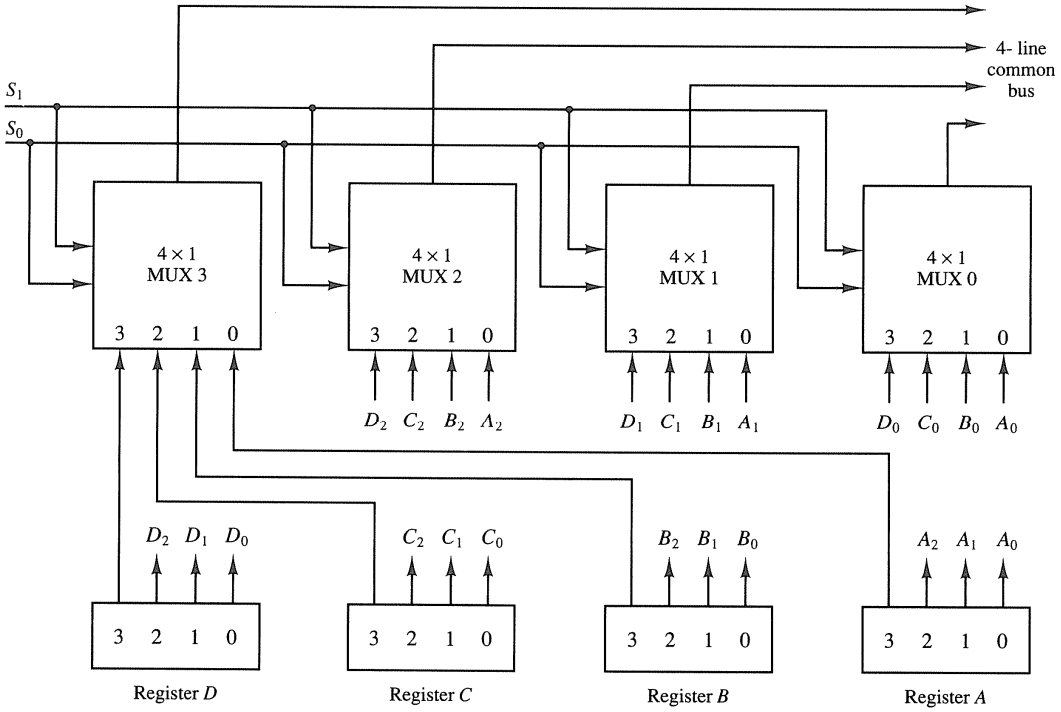
الشكل (3.3): عملية النقل من R1 إلى R2 عندما P=1.

تتفعل الإشارة P عند الحافة الصاعدة لنبضة الساعة t، وعند الحافة الصاعدة التالية t+1 يكون المدخل Load ما زال مفعلاً وتحدث عملية النقل.

2.1.4.3- انتخاب المسجل المصدر (Source Register Selection):

بما أن جميع المسجلات تتصل ببعضها بممر مشترك، ولكي لا تتداخل البيانات خلال عملية انتقالها من المسجل المصدر إلى المسجل الهدف، كان لا بد من وجود آلية تعمل على اختيار المسجل المصدر من بين مسجلات النظام، ويتم ذلك بطريقتين هما:

الطريقة الأولى: تتلخص باستخدام نواخب (Multiplexers) لتحديد المسجل المصدر حسب القيمة المنطقية الموجودة على خطوط الانتخاب كما هو موضح في الشكل (4.3).



الشكل (4.3): ممر النظام من أجل أربعة مسجلات، وكل مسجل يحوي 4 خانات.

يتم في الشكل (4.3) اختيار مسجل واحد كمصدر من أربعة مسجلات هي A (A0، A1، A2، A3)، و B (B0، B1، B2، B3)، و C (C0، C1، C2، C3)، و D (D0، D1، D2، D3) حسب القيمة المطبقة على خطي الانتخاب (S0، S1) وفق الجدول (2.3).

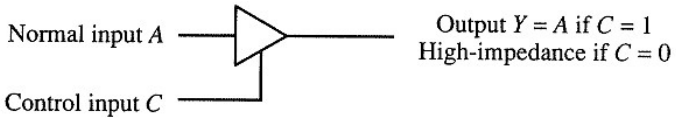
الطريقة الثانية: تكون باستخدام عوازل ثلاثية الحالة[19] (Three State Buffers)، والشكل (5.3)

[19] عازل ثلاثي الحالة: هو دارة رقمية يكون لمخارجها ثلاث حالات بدلاً من حالتين، وهذه الحالات هي إما "0" أو "1" أو ممانعة عالية.

يوضح شكل هذا العازل.

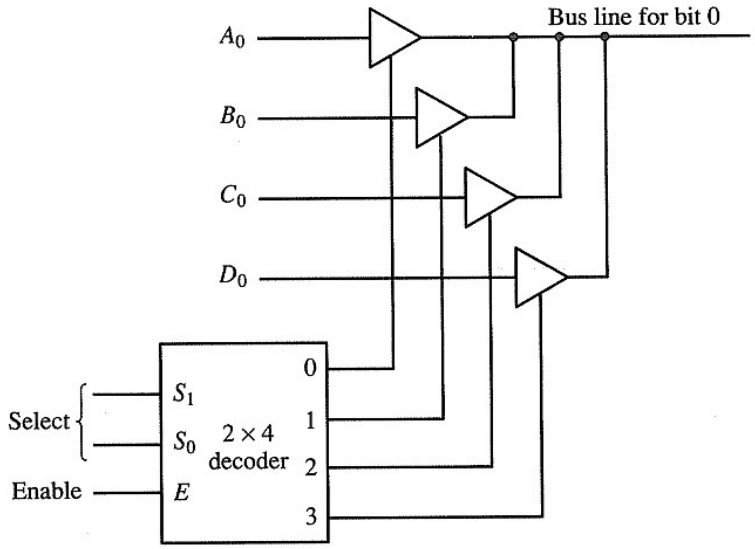
S_1	S_0	Register selected
0	0	A
0	1	B
1	0	C
1	1	D

الجدول (2.3): الجدول الوظيفي لدارة الشكل (4.3).



الشكل (5.3): عازل ثلاثي الحالة.

يقوم هذا العازل بنقل الدخل الاعتيادي A إلى الخرج Y في حال كانت القيمة المطبقة على خط التحكم مفعلة (Control Input C=1)، أما عندما يكون (Control Input C=0) فإن الخرج في حالة ممانعة عالية (High Impedance). تستخدم هذه العوازل لانتخاب المسجل المصدر، حيث يوضح الشكل (6.3) كيفية استخدام هذه البوابات لانتخاب بت واحد من بين البتات الأقل أهمية الموجودة في المسجلات الأربعة A، B، C، D.



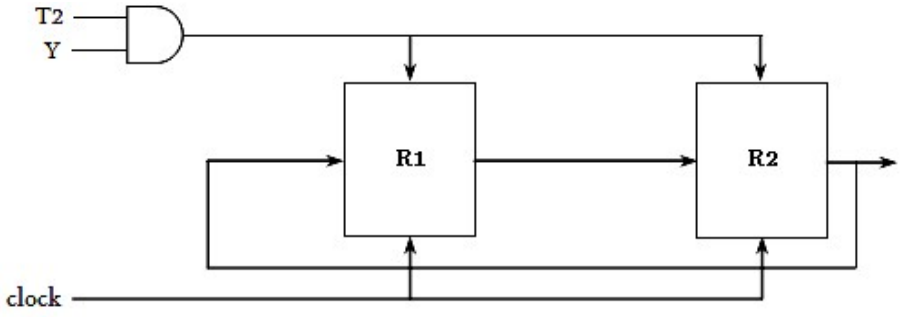
الشكل (6.3): أحد خطوط الممر من أجل العوازل ثلاثية الحالة.

ملاحظة: تكافئ الدارة الموجودة بالشكل (6.3) لعمل MUX (4×1) الموجود في الشكل (4.3).

مثال: ارسم المخطط الصندوقي الذي يصف عملية النقل التالية:

YT2: R1 ← R2, R2 ← R1

الحل: يوضح الشكل (7.3) المخطط الصندوقي المطلوب.



الشكل (7.3): المخطط الصندوقي المطلوب.

2.4.3- تناقل البيانات بين المسجلات والذاكرة (Memory and Register Data Transfer):

يقصد بهذه العملية كتابة محتوى أحد المسجلات في موقع ذاكري أو قراءة موقع ذاكـــري وتخزينه في أحد المسجلات، ويعبر عن هاتين العملتين بالشكل التالي:

عملية الكتابة: $M[AR] \leftarrow R1$

عملية القراءة: $R1 \leftarrow M[AR]$

حيث يشير الحرف M إلى موقع ذاكري، ويمثل AR عنوان ذلك الموقع الذي ستتم القراءة منه أو الكتابة فيه.

مثال: ما هو الخطأ الموجود في العبارة التالية:

$$XT: AR \leftarrow PC, AR \leftarrow 0$$

الحل: عملية النقل غير صحيحة، لأن المسجل الهدف هو نفسه في العمليتين، إذ لا يمكن نقل قيمة PC إلى المسجل AR، وتصفير AR في نفس الوقت.

5.3- العمليات الصغرية (Microoperations):

تنجز هذه العمليات على البيانات المخزنة داخل المسجلات، وتصنف إلى أربعة أقسام رئيسة وفق الآتي:

1- عمليات نقل البيانات بين المسجلات، وقد تم شرحها سابقاً.

2- العمليات الحسابية الصغرية.

3- العمليات المنطقية الصغرية.

4- عمليات الإزاحة.

بشكل عام، يمر تصميم وحدة ما بمرحلتين هما:

1- مرحلة التطبيق البرمجي (Software Implementation): تتمثل بالمعادلات وكتابة الجداول وغيرها.

2- مرحلة التطبيق المادي (Hardware Implementation): تتمثل بتصميم الدارات الالكترونية بالاعتماد على العناصر المنطقية لتنفيذ المعادلات والجداول.

1.5.3- العمليات الحسابية الصغرية (Arithmetic Microoperations):

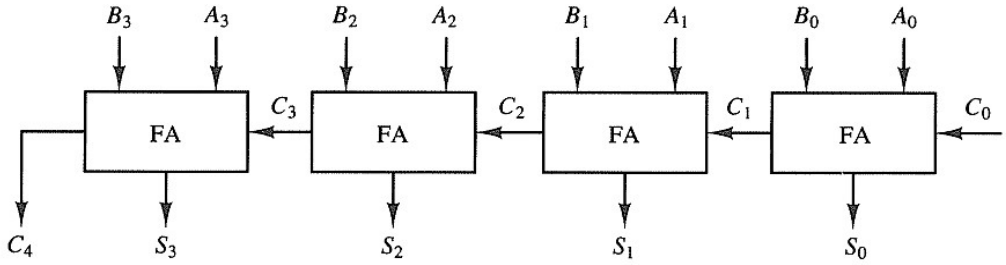
تتمثل العمليات الحسابية الأساسية في عمليتي الجمع والطرح، ويضاف إليهما عمليتي الزيادة والنقصان بمقدار واحد.

1.1.5.3- عملية الجمع الصغرية (Addition Microoperation):

يرمز لعملية الجمع بالرمز +، وتمثل وفق الآتي:

$$R3 \leftarrow R1 + R2$$

أي ستتم إضافة محتوى المسجل R1 إلى المسجل R2، وتخزن نتيجة الجمع في المسجل R3. لتنفيذ عملية الجمع هذه يلزمنا ثلاثة مسجلات، والدارة التي تقوم بعملية الجمع هي الجامع الكامل (Full-Adder: FA). يبين الشكل (8.3) المسجلين A، B، حيث يتألف كل منهما من أربع خانات، ونتيجة الجمع تخزن في المسجل S، مع مراعاة الحمل الناتج (C).



الشكل (8.3): دارة جامع كامل بأربع خانات.

2.1.5.3- عملية الطرح الصغرية (Subtraction Microoperation):

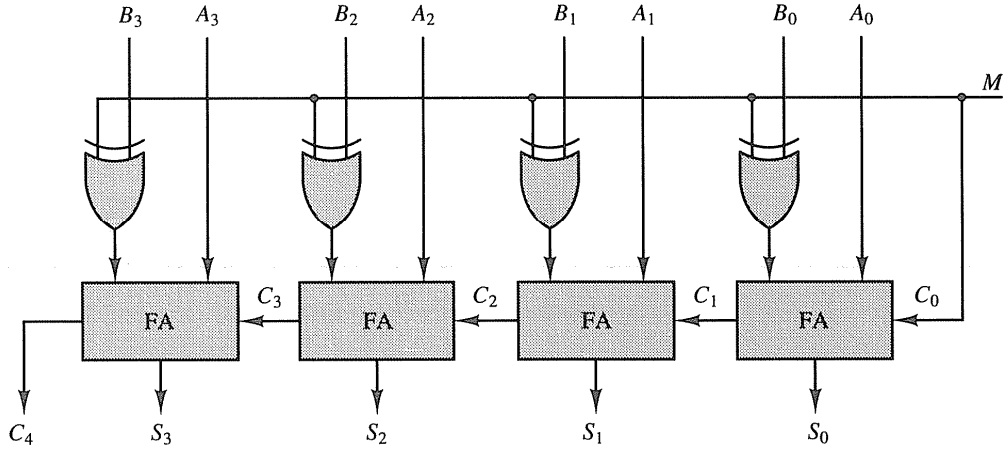
من أجل زيادة التكامل في الأنظمة الحاسوبية، يتم تحويل عملية الطرح إلى عملية جمع مـــع المتمم الثاني كمايلي:

$$R3 \leftarrow R1 - R2$$

$$R3 \leftarrow R1 + \overline{R2} + 1$$

حيث: $\overline{R2}$ هو المتمم الأول للمسجل R2، و $\overline{R2}+1$ هو المتمم الثاني له، أي سيتم إضافة محتوى المسجل R1 إلى المتتم الثاني للمسجل R2، وتخزين نتيجة الجمع في المسجل R3.

بعد أن حولنا عملية الطرح إلى عملية جمع، يمكن تصميم دارة واحدة تقوم بعمليتي الجمع والطرح تسمى دارة جامع/طارح، وهي موضحة في الشكل (9.3)، حيث لدينا المسجلان A، B كل منهما مؤلف من أربع خاتات، ونتيجة الجمع تخزن في المسجل S، مع مراعاة الحمل الناتج.



الشكل (9.3): دارة جامع/طارح بأربع خانات.

عندما تكون إشارة التحكم بعملية الاختيار بين الجمع والطرح "Mode" (M=0) فالدارة هي دارة جمع، لأن $B \oplus 0=B$، وبالتالي سيكون على أحد مداخل الجامع القيمة A، وعلى المدخل الآخر القيمة B. أما عندما تكون (M=1) فالدارة هي دارة طرح، لأن $B \oplus 1=\overline{B}$، وبالتالي سيكون على أحد مداخل الدارة القيمة A، وعلى المدخل الآخر القيمة $\overline{B}$ ، وستدخل قيمة M كحمل، أي سيتم الجمع وفق التالي: $A+\overline{B}+1$.

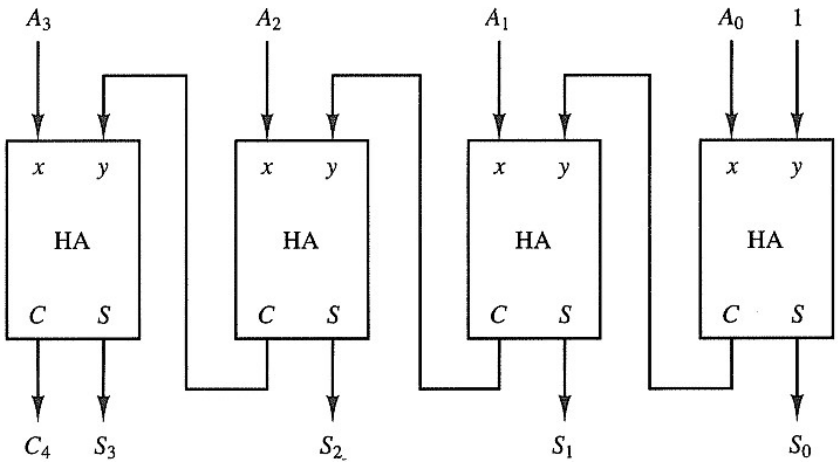
3.1.5.3– عملية الزيادة الصغرية (Increment Microoperation):

تتمثل عملية الزيادة بجمع القيمة "1" إلى محتوى المسجل المراد زيادة قيمته. الدارة التي تقوم بعملية الزيادة هي دارة نصف الجامع (Half-Adder) كما يبين الشكل (10.3).

نفس الأمر ينطبق على عملية النقصان بعد تحويلها إلى عملية زيادة على المتتم الثاني.

4.1.5.3– الدارة الحسابية (Arithmetic Circuit):

الدارة الحسابية التي تقوم بالعمليات الحسابية كافة موضحة في الشكل (11.3)، إذ نلاحظ أنه عندما تكون القيمة المنطقية المطبقة على خطوط الانتخاب (S0، S1) هي (00) فإن القيمة B (B0، B1، B2، B3) تطبق على الدخل Y (Y0، Y1، Y2، Y3) وتجمع مع القيمة A إذا كانت قيمة الحمل (C_{in}=0)، وإلا ستتم عملية جمع لـ ($A + B + C_{in}$) أي جمع مع حمل.



الشكل (10.3): دارة الزيادة بأربع خانات.

أما عندما تكون قيمة (S0، S1) هي (01) فإنه سيطبق على الدخل Y القيمة $\overline{B}$ ، وفي حال كانت (C_{in}=1) فإن الخرج سيكون (A + $\overline{B}$ +1)، وهي تمثل عملية الطرح أي (A - B). وهكذا تتم باقي العمليات الحسابية كما هو موضح في الجدول (3.3).

Select			Input	Output	
S_1	S_0	C_{in}	Y	$D = A + Y + C_{in}$	Microoperation
0	0	0	B	$D = A + B$	Add
0	0	1	B	$D = A + B + 1$	Add with carry
0	1	0	$\overline{B}$	$D = A + \overline{B}$	Subtract with borrow
0	1	1	$\overline{B}$	$D = A + \overline{B} + 1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A + 1$	Increment A
1	1	0	1	$D = A - 1$	Decrement A
1	1	1	1	$D = A$	Transfer A

الجدول (3.3): الجدول الوظيفي للدارة الحسابية.

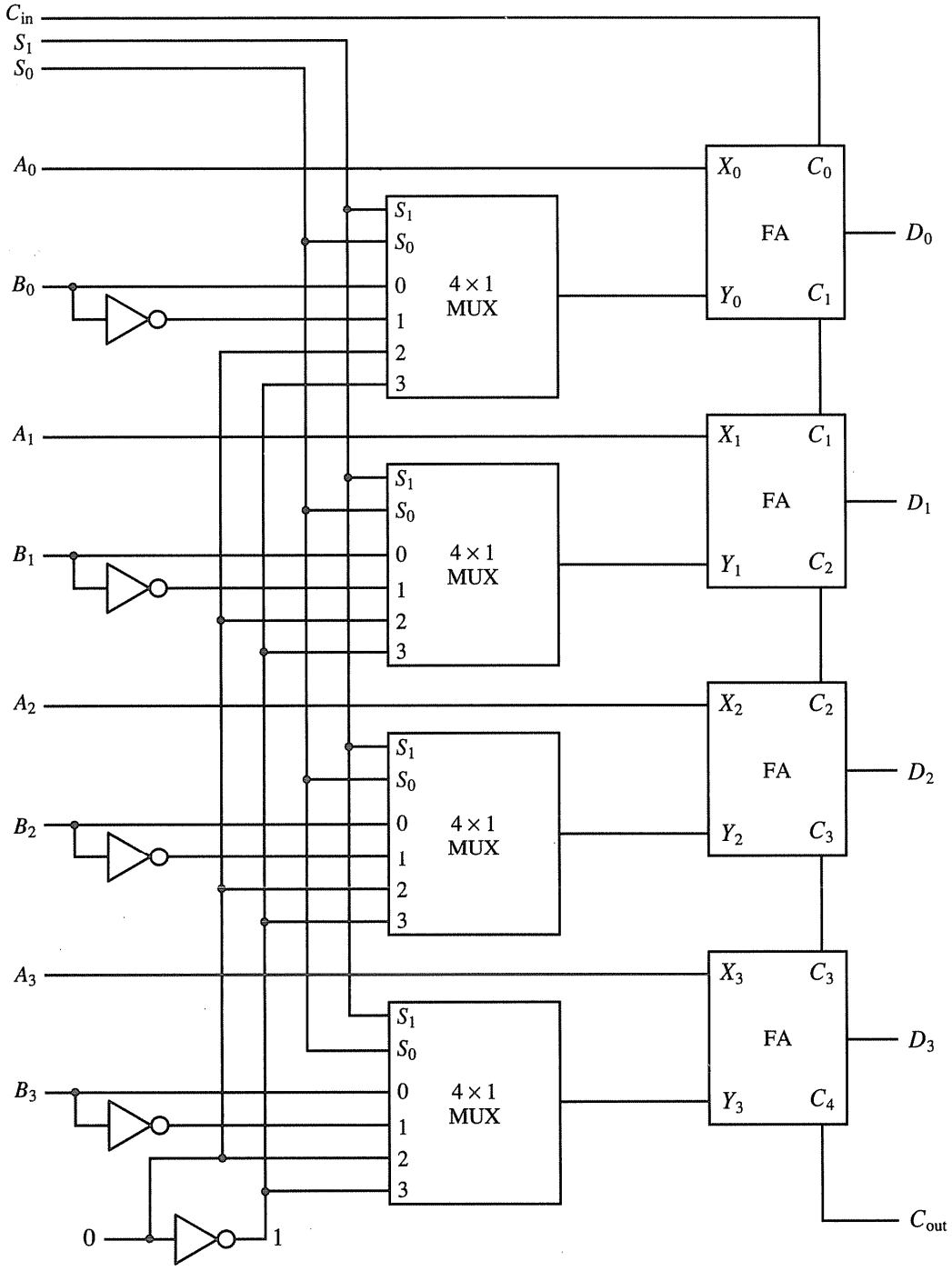
2.5.3– العمليات المنطقية الصغرية (Logic Microoperations):

تطبق العمليات المنطقية على سلسلة البتات المخزنة داخل المسجل، ومن أشهر العمليات المنطقية عمليات AND، OR، XOR، NOT. لتطبيق هذه العمليات على مستوى البنية المادية فإن ذلك يتطلب البوابات المنطقية الأساسية بالإضافة إلى ناخب يتم من خلاله اختيار نوع العملية المنطقية حسب القيمة المطبقة على خطوط الانتخاب، كما هو موضح في الشكل (12.3).

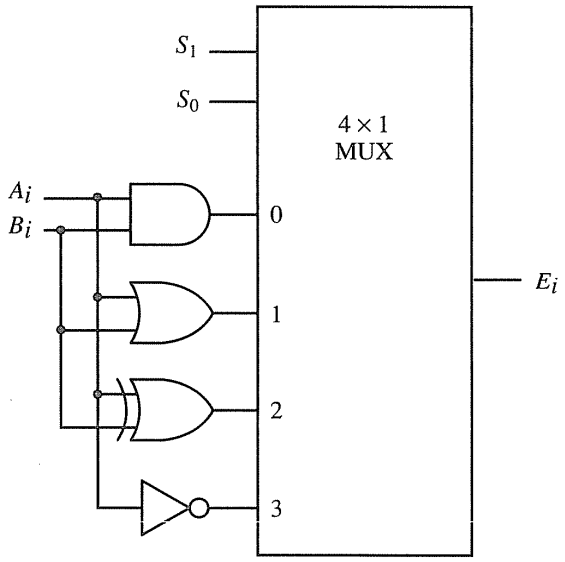
تسمح هذه الدارة بتنفيذ العمليات المنطقية الأربع AND، OR، XOR، NOT وفق الجدول (4.3).

3.5.3- عمليات الإزاحة الصغرية (Shift Microoperations):

تستخدم عمليات الإزاحة من أجل النقل التسلسلي للبيانات، وتتم هذه العمليات على سلسلة البتات المخزنة في المسجل. يوجد ثلاثة أنواع رئيسة للإزاحة.



الشكل (11.3): الدارة الحسابية.



الشكل (12.3): الدارة المنطقية.

S_1	S_0	Output	Operation
0	0	$E = A \wedge B$	AND
0	1	$E = A \vee B$	OR
1	0	$E = A \oplus B$	XOR
1	1	$E = \overline{A}$	Complement

الجدول (4.3): الجدول الوظيفي للدارة المنطقية.

1.3.5.3- الإزاحة المنطقية (Logic Shift):

لها نوعان هما:

1- إزاحة نحو اليمين: يرمز لها بالرمز shr، وفيها يتم وضع صفر في الخانة الأكثر أهمية (MSB) وضياع أو إهمال للخانة الأقل أهمية (LSB).

2- إزاحة نحو اليسار: يرمز لها بالرمز shl، وفيها يتم وضع صفر في الخانة الأقل أهمية (LSB) وضياع أو إهمال للخانة الأكثر أهمية (MSB).

مثال: مسجل يحوي القيمة المنطقية 1001، ما قيمة المسجل بعد تنفيذ shr ومن ثم shl، معتمداً في كل مرة على قيمته الابتدائية؟

الحل:

- بعد تنفيذ shr تصبح قيمة المسجل هي 0100.

- بعد تنفيذ shl تصبح قيمة المسجل هي 0010.

2.3.5.3- الإزاحة الدورانية (Rotate Shift or Circular Shift):

فيها ينتقل البت المزاح من LSB إلى MSB أو العكس حسب اتجاه الإزاحة دون ضياع. ولها نوعان هما:

1- إزاحة نحو اليمين: يرمز لهـــا بالرمز Ror أو Cir، فيها تنتقل LSB إلى MSB وتزاح باقي البتات تسلسلياً.

2- إزاحة نحو اليسار: يرمز لها بالرمز Rol أو Cil، فيها تنتقل MSB إلى LSB وتزاح باقي البتـــات تسلسلياً.

مثال: مسجل يحوي القيمة المنطقية 1001، ما قيمة المسجل بعد تنفيذ Ror ومن ثم Rol، معتمداً في كل مرة على قيمته الابتدائية؟

الحل:

- بعد تنفيذ Ror تصبح قيمة المسجل هي 1100.
- بعد تنفيذ Rol تصبح قيمة المسجل هي 0011.

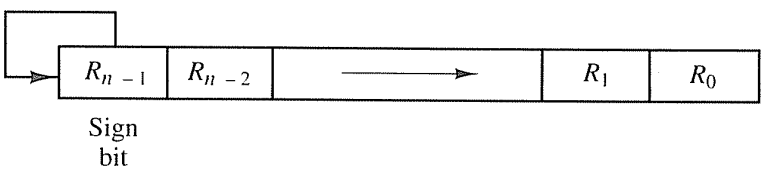
3.3.5.3- الإزاحة الحسابية (Arithmetic Shift):

تتم هذه الإزاحة على الأعداد المؤشرة الممثلة بالمتمم الثاني، حيث تمثل خانة MSB إشـــارة العدد، فعندما (MSB=0) يكون العدد موجب، أما عندما (MSB=1) فإن العدد سالب.

ملاحظة: يجب أن يبقى البت المخصص للإشارة (Sign Bit) على حاله حتى لا تتغير إشارة العـــدد المزاح.

للإزاحة الحسابية نوعين هما:

1- الإزاحة الحسابية نحو اليمين: يرمز لهـــا بالرمز ashr، حيث يتم المحافظة على الخانة Rn-1 كما هي، وتنقل نفسها إلى الخانة Rn-2، وتزاح باقي الخانات تسلسلياً نحو اليمين وتهمل الخانة R0 وفق الشكل (13.3).



الشكل (13.3): الإزاحة الحسابية نحو اليمين.

2- الإزاحة الحسابية نحو اليسار: يرمز لهـــا بالرمز ashl، حيث يدخل صفر إلى الخانة R0، وتزاح باقي الخانات تسلسلياً نحو اليسار، وبالتالي تتوضع الخانة Rn-2 مكان الخانـــة Rn-1. وفي حـــال اختلاف قيمة Rn-1 عن Rn-2 ستتغير إشارة العدد، وتسمى هذه الحالة بالطفحان (Overflow).

يستخدم علم V_S لتحديد حالة الطفحان، وتحسب قيمته من العلاقة التالية:

$$V_S = R_{n-2} \oplus R_{n-1}$$

إذا كان (V_S=0) عندها لا يوجد طفحان وتعتبر الإزاحة صحيحة. أما إذا كـــان (V_S=1) عندها يوجد طفحان وخانة الإشارة تغيرت بعد الإزاحة ويجب تغييرها.

يوضح الجدول (5.3) عمليات الإزاحة الصغرية.

Symbolic designation	Description
$R \leftarrow$shl R	Shift-left register R
$R \leftarrow$shr R	Shift-right register R
$R \leftarrow$cil R	Circular shift-left register R
$R \leftarrow$cir R	Circular shift-right register R
$R \leftarrow$ashl R	Arithmetic shift-left R
$R \leftarrow$ashr R	Arithmetic shift-right R

الجدول (5.3): عمليات الازاحة الصغرية.

الدارة التي تقوم بعمليات الإزاحة موضحة في الشكل (14.3)، وذلك باعتبار أن الدخل I_R مخصص لخانات الإزاحة اليمينية، والدخل I_L مخصص لخانات الإزاحة اليسارية. تقوم هذه الـــدارة بإزاحة محتوى المسجل A نحو اليمين أو اليسار حسب القيمة المطبقة على خط الانتخاب S كما هو موضح في الجدول (6.3).

Function table

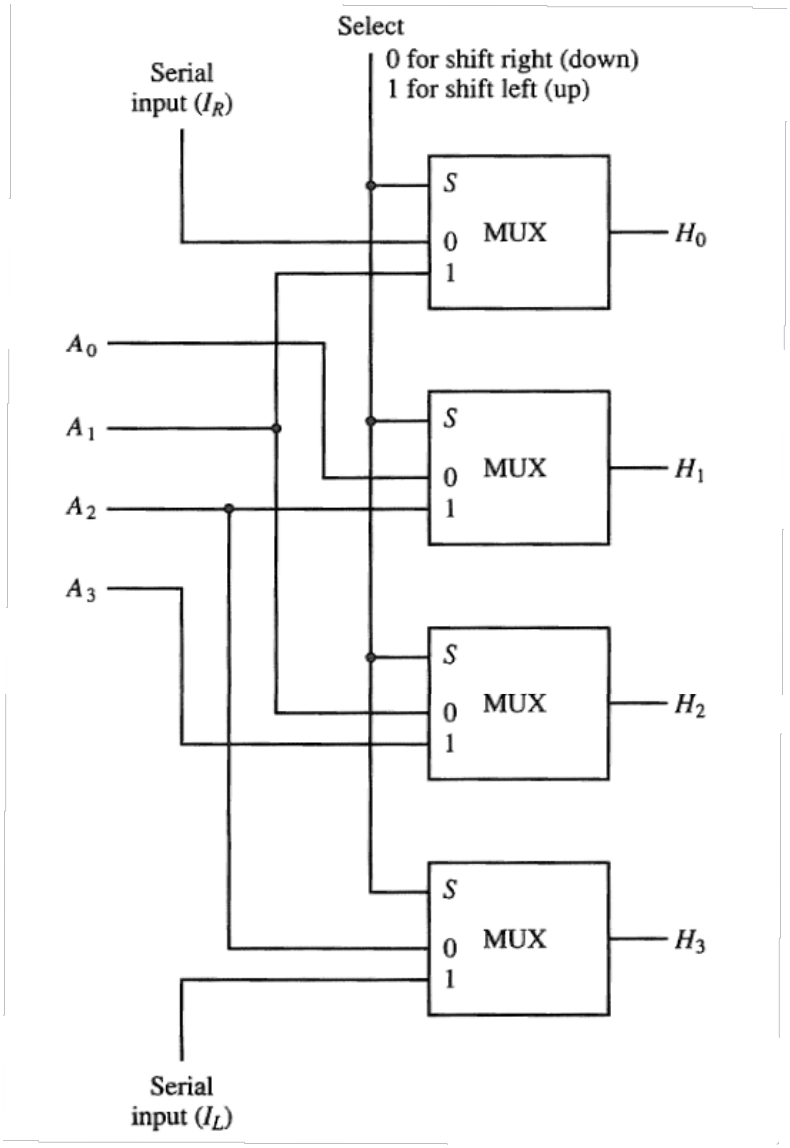
Select	Output			
S	H_0	H_1	H_2	H_3
0	I_R	A_0	A_1	A_2
1	A_1	A_2	A_3	I_L

الجدول (6.3): الجدول الوظيفي لدارة الازاحة.

6.3- تصميم وحدة الحساب والمنطق (Arithmetic and Logic Unit Design):

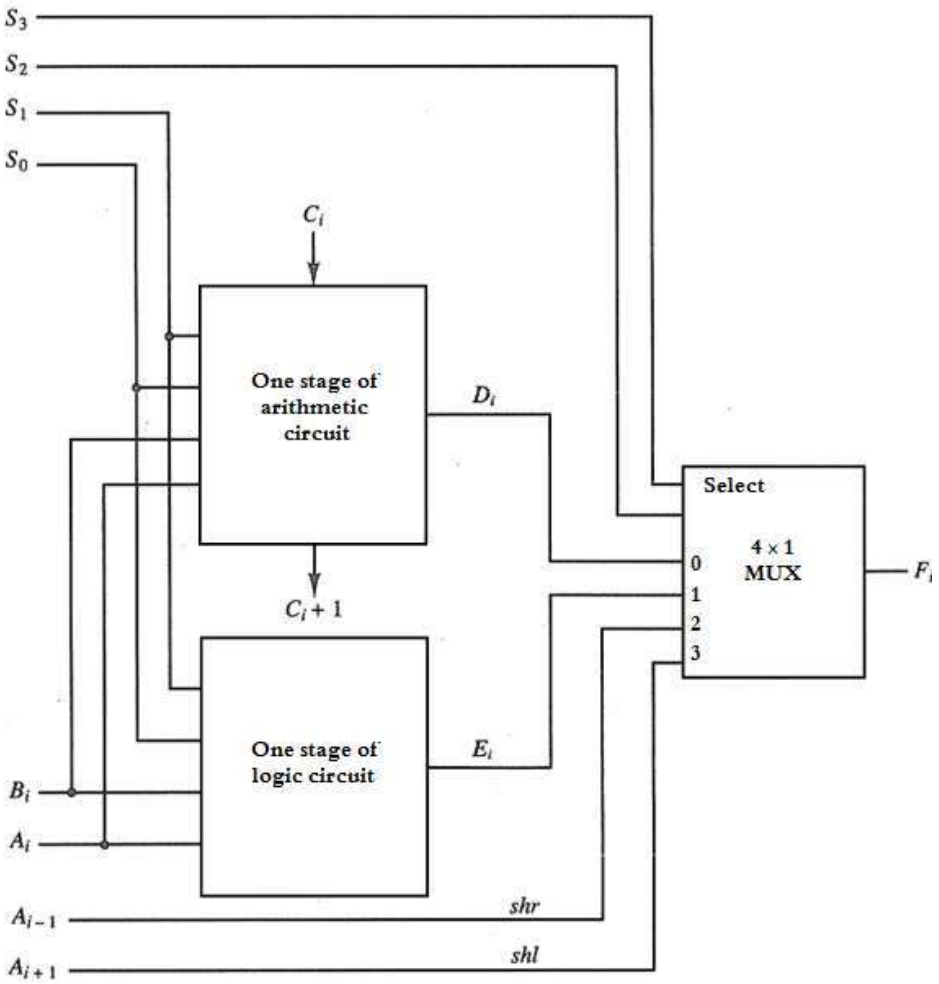
بعد أن درسنا جميع العمليات الحسابية والمنطقية والإزاحة الصغرية، لا بد من تصميم دارة

متكاملة تقوم بجميع هذه العمليات وتستقبل دخلها من المسجلات وتخزن النتيجة أيضاً في أحد المسجلات، تسمى هذه الدارة بوحدة الحساب والمنطق (ALU)، حيث تضم وحدة منفصلة للقيام بالعمليات الحسابية، وأخرى للعمليات المنطقية، بالإضافة إلى عمليات الإزاحة كما هو موضح في الشكل (15.3).



الشكل (14.3): دارة الازاحة بأربعة بتات.

تحتوي هذه الدارة على أربعة خطوط انتخاب. يقوم S2 و S3 باختيار نوع العملية (حسابية أو منطقية أو إزاحة)، بينما يستخدم S0 و S1 لاختيار نوع العملية ضمن الوحدة الحسابية أو المنطقية ذاتها. ويتم العمل حسب الجدول (7.3).



الشكل (15.3): وحدة الحساب والمنطق.

Operation select						
S_3	S_2	S_1	S_0	C_{in}	Operation	Function
0	0	0	0	0	$F = A$	Transfer A
0	0	0	0	1	$F = A + 1$	Increment A
0	0	0	1	0	$F = A + B$	Addition
0	0	0	1	1	$F = A + B + 1$	Add with carry
0	0	1	0	0	$F = A + \overline{B}$	Subtract with borrow
0	0	1	0	1	$F = A + \overline{B} + 1$	Subtraction
0	0	1	1	0	$F = A - 1$	Decrement A
0	0	1	1	1	$F = A$	Transfer A
0	1	0	0	×	$F = A \wedge B$	AND
0	1	0	1	×	$F = A \vee B$	OR
0	1	1	0	×	$F = A \oplus B$	XOR
0	1	1	1	×	$F = \overline{A}$	Complement A
1	0	×	×	×	$F =$ shr A	Shift right A into F
1	1	×	×	×	$F =$ shl A	Shift left A into F

الجدول (7.3): الجدول الوظيفي لوحدة الحساب والمنطق.