CHAPTER 1

Register Transfer & Microoperation

كلية حاسبات

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Chapter 1 content

By Eng. Emad Mahdy

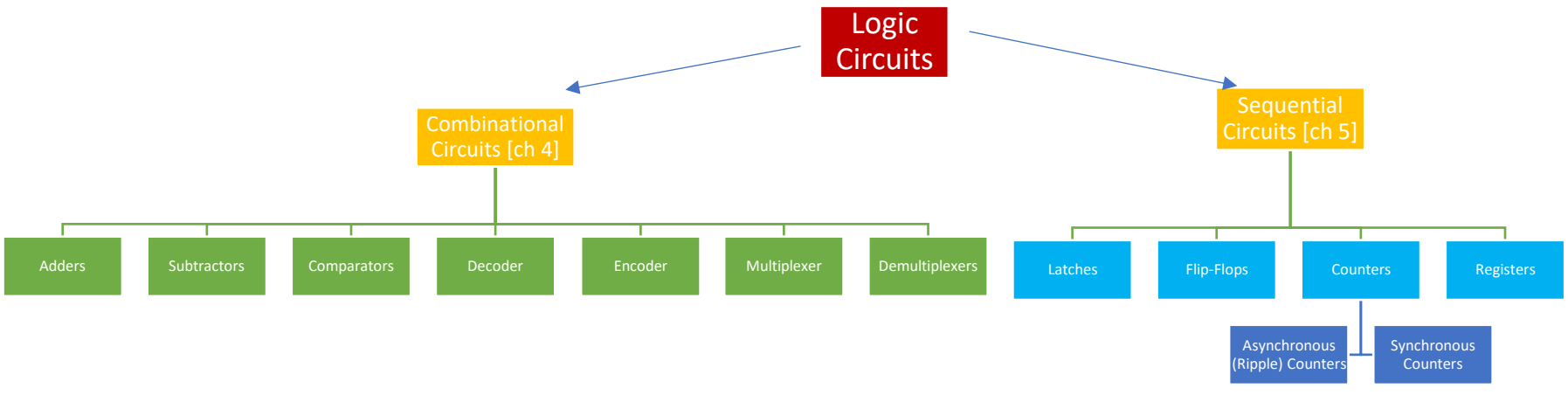
WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

1.1 Register Transfer Language

Simple digital systems الانظمة الرقمية البسيطة



- Combinational and sequential circuits can be used to create simple digital systems.
- يمكن استخدام دوائر ال Combinational و sequential (و التي درسناها سابقا في مادة التصميم المنطقي الرقمي) لإنشاء أنظمة رقمية بسيطة
- These are the low-level building blocks of a digital computer.
- هذه الدوائر هي اللبنات الأساسية للحاسب.
- Simple digital systems are frequently characterized in terms of: عادة ما يتم تمييز هذة الأنظمة الرقمية البسيطة
  a. The registers they contain. السجلات التي تحتويها
  b. The operations that they perform. العمليات التي تقوم بها (مثل الجمع – الطرح -الخ

Microoperation

Microoperation: is an elementary operation performed on the information stored in one or more registers.

عملية دقيقة: هي عملية أولية يتم إجراؤها على المعلومات المخزنة في واحد أو أكثر من السجلات.

Examples of Microoperations

- Shift
- Load
- Clear
- Increment

Register Transfer Language (RTL): The symbolic notation used to describe the microoperation transfers among registers.

(RTL): اللغة المستخدمة لوصف عمليات نقل العمليات الدقيقة بين السجلات.

Internal organization of a computer التركيب الداخلي للحاسب

(a) Set of registers and their functions
مجموعة السجلات ووظائفها

(b) Microoperations set
مجموعة العمليات الدقيقة

(c) Control signals that initiate the sequence of microoperations (to perform the functions)
إشارات التحكم التي تبدأ تسلسل العمليات الدقيقة (لتنفيذ الوظائف)

By Eng. Emad Mahdy
WhatsApp: +20 11 00 18 4676
https://www.youtube.com/@eng.emadmahdy
https://si-manual.com

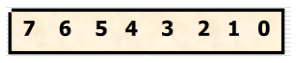
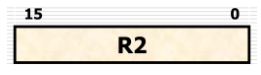
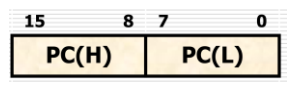
1-2 Register Transfer

Registers are designated by capital letters, sometimes followed by numbers. يتم ترميز السجلات بأحرف كبيرة ، متبوعة أحيانًا بأرقام.

- A
- R13
- IR

Often the names indicate function: غالبًا ما تشير أسماء السجلات إلى وظيفتها

- MAR → Memory Address Register
- PC → Program Counter
- IR → Instruction Register

الطرق المختلفة ل رسم ال registers	
1. Register	
2. Individual bits	
3. Numbering of bits	
4. Subfields	

Register Transfer

Register Transfer: Copying the contents of one register to another.

Register Transfer هي نسخ محتويات سجل إلى سجل آخر.

A register transfer is indicated as: R2 ← R1

In this case the contents of register R1 are copied (loaded) into register R2.

في هذه الحالة، يتم نسخ محتويات السجل R1 إلى السجل R2.

A simultaneous transfer of all bits from the source R1 to the destination register R2, during one clock pulse.

يتم النقل المتزامن لجميع البتات من السجل المصدر R1 إلى السجل الوجهة R2 خلال نبضة ساعة واحدة.

Note that this is non-destructive, i.e., the contents of R1 are not altered by copying (loading) them to R2.

لاحظ أن هذا النقل غير مدمر، أي أن محتويات R1 لا تتغير عند نسخها إلى R2.

Example:

A register transfer such as

R3 ← R5

Implies that the digital system has: يعني أن النظام الرقمي يحتوي على

a. The data lines from the source register (R5) to the destination register (R3)

خطوط البيانات من السجل المصدر (R5) إلى السجل الوجهة (R3)

b. Parallel load in the destination register (R3)

تحميل متوازي في السجل الوجهة (R3)

c. Control lines to perform the action

خطوط التحكم لتنفيذ العملية

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Control Function

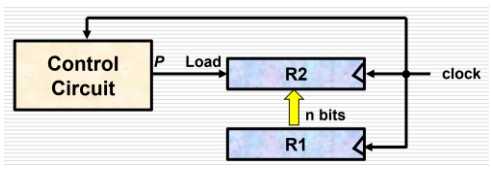
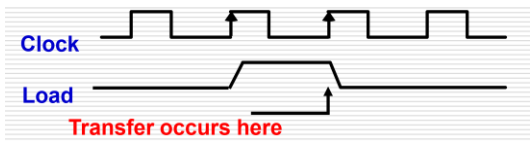
- Often actions need to only occur if a certain condition is true (like an "if" statement in a programming language).
  - غالبًا ما يجب أن تحدث الإجراءات فقط إذا كانت هناك حالة معينة صحيحة (مثل جملة "if" في لغة البرمجة).
- In digital systems, this is often done via a control signal, called a control function.
  - في الأنظمة الرقمية، يتم تنفيذ ذلك غالبًا عبر إشارة تحكم تسمى وظيفة التحكم.
- If the signal is 1, the action takes place.
  - إذا كانت الإشارة 1، يتم تنفيذ الإجراء.
- This is represented as: يتم تمثيل ذلك كالتالي

$$P: R2 \leftarrow R1$$

Which means "if P = 1, then load the contents of register R1 into register R2"

مما يعني إذا كانت P = 1 ، فقم بنسخ محتويات السجل R1 إلى السجل R2

```
if (P = 1) then (R2 ← R1)
```

Implementation of controlled transfer P: R2 ← R1	
Notation	P: R2 ← R1
Block diagram	
Timing diagram	

- The same clock controls: يتحكم نفس مولد النبضات في
  - circuits that generate the control function دوائر التحكم
  - source register السجل المصدر
  - destination register السجل الوجهة

- Registers are assumed to use positive-edge-triggered flip-flops.

يُفترض أن السجلات تستخدم فليب-فوب التي تعمل بحافة النبضة الإيجابية

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Simultaneous operations التعليمات المتزامنة

If two or more operations are to occur simultaneously, they are separated with commas.
إذا كان من المقرر أن تتم عمليتان أو أكثر في وقت واحد، يتم فصلها ب.

Example

P: R3 ← R5, MAR ← IR

Here, if the control function P = 1

a. load the contents of R5 into R3
b. at the same time (clock), load the contents of register IR into register MAR

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Basic symbols for Register Transfers

Symbols		Description	Examples
Capital letters & numerals	A R_1	Register سجل	MAR, R2
Parentheses ()	()	part of a register جزء من السجل	R2(0-7) R2(L)
Arrow ←	←	transfer of information نسخ سجل الي اخر	R2 ← R1
Colon :	:	termination of control function إنهاء وظيفة التحكم	P:
Comma ,	,	Separates two microoperations الفصل بين عمليتين دقيقتين	A ← B, B ← A

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

4.3 Bus and Memory Transfer

Connecting registers توصيل السجلات

الطريقة الثانية لتوصيل السجلات (الطريقة المستخدمة في الواقع)	الطريقة الأولي لتوصيل السجلات(غير عملية)
▪ Instead, take a different approach: بدلاً من ذلك، يتم اتباع نهج مختلف a. Have one centralized set of circuits for data transfer – the bus استخدام مجموعة مركزية واحدة من الدوائر لنقل البيانات – الحافلة (bus) b. Have control circuits to select which register is the source, and which is the destination. استخدام دوائر التحكم لتحديد السجل المصدر والسجل الوجهة.	▪ In a digital system with many registers, it is impractical to have data and control lines to directly allow each register to be loaded with the contents of every possible other registers. في نظام رقمي به العديد من السجلات ، من غير العملي توصيل كل السجلات مع بعضها بشكل مباشر. ▪ To completely connect n registers → n(n-1) lines cost $O(n^2)$ and this is not a realistic approach to use in a large digital system. لتوصيل n سجلات بالكامل → فان عدد الخطوط هي n(n-1) وتكلفتها $O(n^2)$، وهذا ليس نهجًا واقعيًا للاستخدام في نظام رقمي كبير.

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

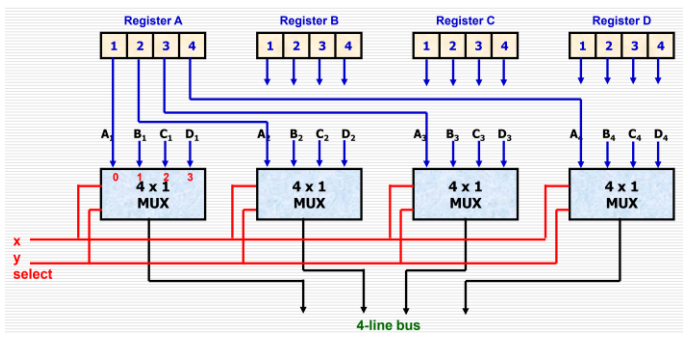
Bus and Bus Transfer

A bus is a path (of a group of wires) over which information is transferred, from any of several sources to any of several destinations.

الناقل هو مسار (مجموعة من الأسلاك) يتم خلاله نقل المعلومات، من أي من مكان إلى اخر.

Transfer From a register to bus:
BUS ← R

Using Multiplexer



N → number of registers

n → number of bits in each register

S_n → number of selection inputs for each Multiplexer

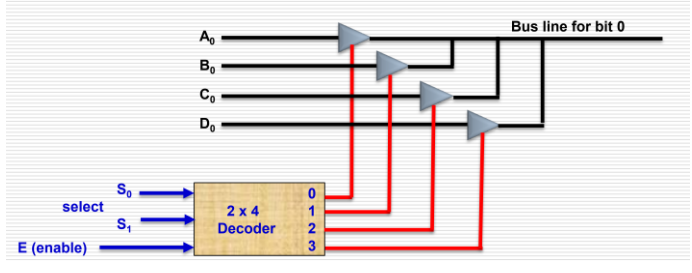
$$S_n = log_2[N]$$

Size of each Multiplexer = N×1

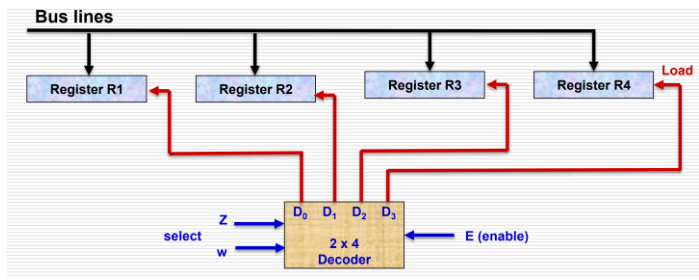
Number of Multiplexer = n

x	y	Register selected
0	0	A
0	1	B
1	0	C
1	1	D

Using Three-State Bus Buffers



Transfer From a bus to a destination register:
R ← BUS



By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Bus Transfer in RTL

Depending on whether the bus is to be mentioned explicitly or not, register transfer can be indicated as either implicit or explicit.

بناءً على ما إذا كان سيتم ذكر الناقل (bus) بشكل صريح أم لا، يمكن الإشارة إلى نقل السجل إما بشكل ضمني أو صريح.

The bus is implicit	The is explicitly indicated (former case)
$R2 \leftarrow R1$ دي الطريقة المستخدمة دائما	$BUS \leftarrow R1, R2 \leftarrow BUS$ الطريقة الرسمية (غير مستخدمة)

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

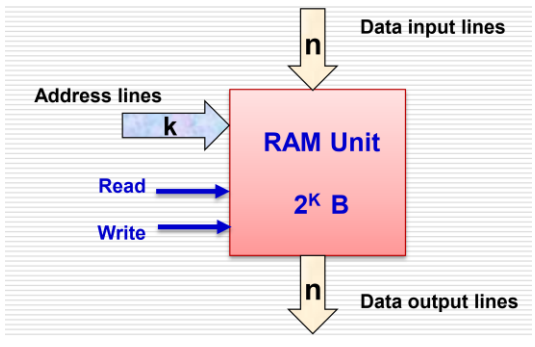
Memory (RAM)

- Memory (RAM) can be thought as a sequential circuit containing some number of registers.
- يمكن اعتبار الذاكرة (RAM) كدائرة تسلسلية تحتوي على عدد من السجلات.
- These registers hold the words of memory.
- هذه السجلات تحتفظ بكلمات الذاكرة.
- Each of the r registers is indicated by an address.
- يتم الإشارة إلى كل واحد من السجلات r بعنوان.
- These addresses range from 0 to r-1
- تتراوح هذه العناوين من 0 إلى r-1.
- Each register (word) can hold n bits of data.
- يمكن لكل سجل (كلمة) أن يحتفظ بـ n بت من البيانات.

Assume the RAM contains $r = 2^k$ words.

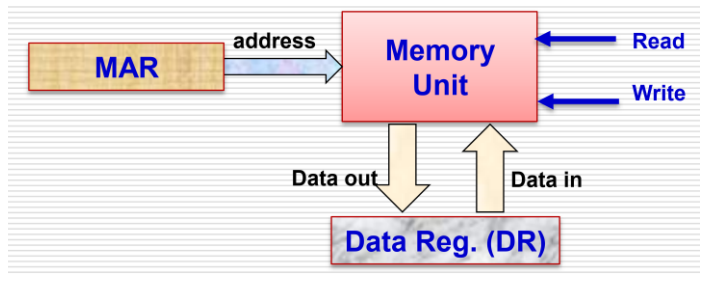
It needs the following:

- n→ data input lines
- n→ data output lines
- k→ address lines
- Read control lines.
- Write control line



Memory Transfer

- the memory is viewed at the register level as a device, M.
  - يتم النظر إلى الذاكرة على مستوى السجل كجهاز يسمى M.
- Since it contains multiple locations, we must specify which address in memory we will be using, and this is done by indexing memory references.
  - بما أنها تحتوي على مواقع متعددة، يجب علينا تحديد العنوان الذي سنستخدمه في الذاكرة، ويتم ذلك عن طريق فهرسة الإشارات إلى الذاكرة.
- Memory is usually accessed in computer systems by putting the desired address in a special register, the Memory Address Register (MAR, or AR)
  - عادةً ما يتم الوصول إلى الذاكرة في أنظمة الحاسوب عن طريق وضع العنوان المطلوب في سجل خاص يسمى سجل عنوان الذاكرة (MAR أو AR).
- When memory is accessed, the contents of the MAR get sent to the memory unit's address lines.
  - عند الوصول إلى الذاكرة، يتم إرسال محتويات سجل عنوان الذاكرة (MAR) إلى خطوط العناوين في وحدة الذاكرة.



	Memory Read	Memory Write
RTL notation:	R1 ← M[MAR] read a value from a location in memory and load it into a register قراءة قيمة من موقع في الذاكرة وتحميلها إلى سجل	M[MAR] ← R1 write a value from a register to a location in memory كتابة قيمة من سجل إلى موقع في الذاكرة
This causes the following to occur:	a. The contents of the MAR get sent to the memory address lines. يتم إرسال محتويات سجل عنوان الذاكرة (MAR) إلى خطوط عناوين الذاكرة. b. A Read = 1 gets sent to the memory unit يتم إرسال إشارة قراءة = 1 إلى وحدة الذاكرة. c. The contents of the specified address are put on the memory's output data lines. يتم وضع محتويات العنوان المحدد على خطوط بيانات الخرج الخاصة بالذاكرة. d. These get sent over the bus to be loaded into register R1 يتم إرسال هذه البيانات عبر الحافلة ليتم تحميلها إلى السجل R1.	a. The contents of the MAR get sent to the memory address lines. يتم إرسال محتويات سجل عنوان الذاكرة (MAR) إلى خطوط عناوين الذاكرة. b. A Write = 1 gets sent to the memory unit يتم إرسال إشارة كتابة = 1 إلى وحدة الذاكرة. c. The values in register R1 get sent over the bus to the data input lines of the memory. يتم إرسال القيم الموجودة في السجل R1 عبر الحافلة إلى خطوط إدخال البيانات الخاصة بالذاكرة. d. The values get loaded into the specified address in the memory يتم تحميل القيم في العنوان المحدد داخل الذاكرة.

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Summary of Register Transfer Microoperations

Transfer Microoperation	Description
A ← B	Transfer content of reg. B into reg. A نقل محتويات السجل B إلى السجلA
AR ← DR(AD)	Transfer content of AD portion of reg. DR into reg. AR نقل محتويات الجزء AD من السجل DR إلى السجلAR
A ← constant	Transfer a binary constant into reg. A نقل قيمة ثابتة إلى السجلA
ABUS ← R1 , R2 ← ABUS	Transfer content of R1 into bus A and, at the same time, transfer content of bus A into R2 نقل محتويات السجل R1 إلى الحافلةA ، وفي نفس الوقت نقل محتويات الحافلة A إلى السجلR2
AR	Address register سجل العناوين
DR	Data register سجل البيانات
M[R]	Memory word specified by reg. R كلمة الذاكرة المحددة بواسطة السجلR
M	Equivalent to M[AR] يعادلM[AR]
DR ← M	Memory read operation: transfers content of memory word specified by AR into DR عملية قراءة الذاكرة: تنقل محتويات كلمة الذاكرة المحددة بواسطة AR إلىDR
M ← DR	Memory write operation: transfers content of DR into memory word specified by AR عملية كتابة الذاكرة: تنقل محتويات DR إلى كلمة الذاكرة المحددة بواسطةAR

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

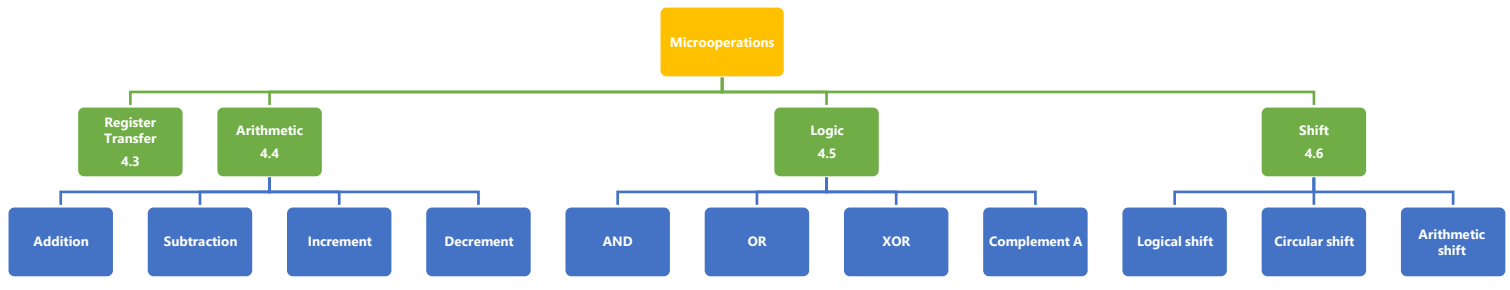
https://si-manual.com

1.4 Arithmetic Microoperations

Microoperations

Computer system microoperations are of four types:

- Register transfer microoperations.
- Arithmetic microoperations
- Logic microoperations
- Shift microoperations



The basic arithmetic microoperations are:

- Addition
- Subtraction
- Increment
- Decrement

The additional arithmetic microoperations are:

- Add with carry.
- Subtract with borrow.
- Transfer/Load
- etc. ...

Summary of Typical Arithmetic Microoperations

Microoperation	Description
$R3 \leftarrow R1 + R2$	Contents of R1 plus R2 transferred to R3
$R3 \leftarrow R1 - R2$	Contents of R1 minus R2 transferred to R3
$R1 \leftarrow \overline{R2}$	Complement the contents of R2
$R2 \leftarrow \overline{R2} + 1$	2's complement the contents of R2 (negate)
$R3 \leftarrow R1 + \overline{R2} + 1$	subtraction
$R1 \leftarrow R1 + 1$	Increment
$R1 \leftarrow R1 - 1$	Decrement

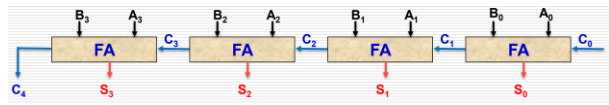
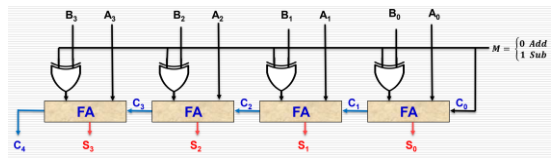
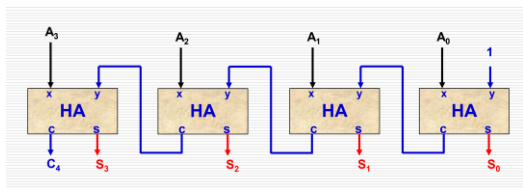
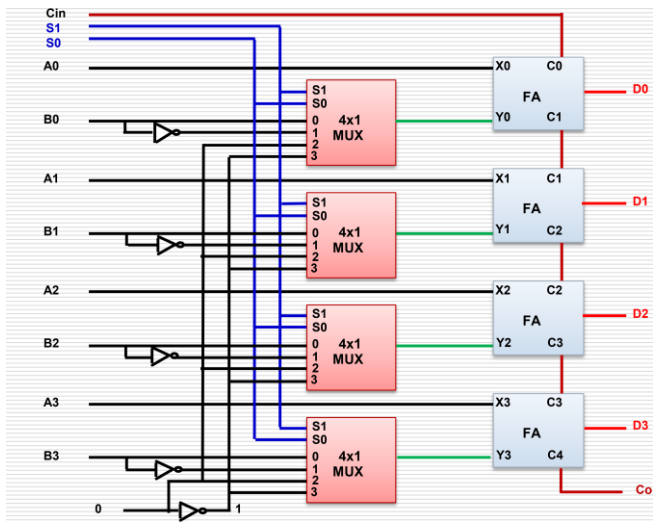
By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Arithmetic Circuit

Circuit	Block Diagram
Binary Adder	
Binary Adder-Subtractor	
Binary Incrementor	
Complete Arithmetic Circuit	 $D = A + Y + C_{in}$

S1	S0	Cin	X	Y	Output	Microoperation
0	0	0	A	B	D = A + B	Add
0	0	1	A	B	D = A + B + 1	Add with carry
0	1	0	A	B'	D = A + B'	Subtract with borrow
0	1	1	A	B'	D = A + B'+ 1	Subtract
1	0	0	A	0	D = A	Transfer A
1	0	1	A	0	D = A + 1	Increment A
1	1	0	A	1	D = A - 1	Decrement A
1	1	1	A	1	D = A	Transfer A

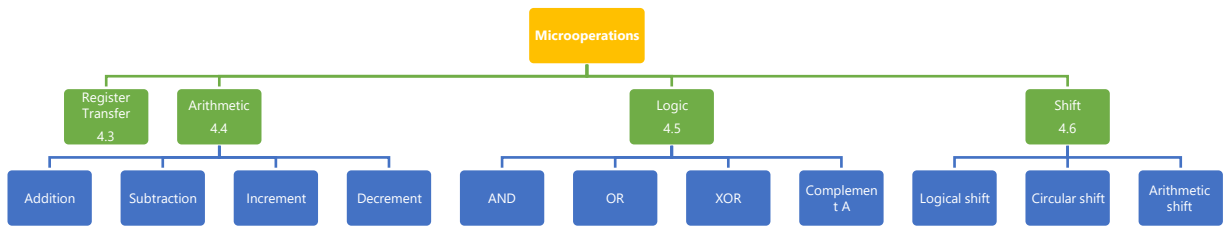
By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

1.5 Logic Microoperations



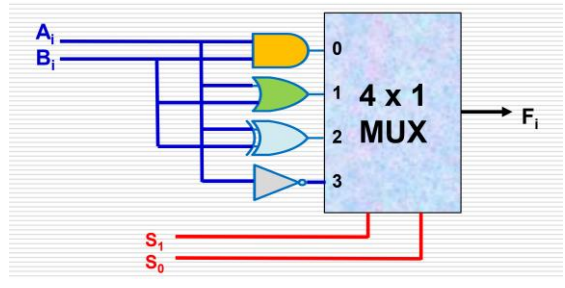
- Logic microoperations are bit-wise operations, i.e., they work on the individual bits of data
- useful for bit manipulations on binary data
- useful for making logical decisions based on the bit value.

There are, in principle, 16 different logic functions that can be defined over two binary input variables
However, most systems only implement four of these:

- **AND** ($\wedge$)
- **OR** ($\vee$)
- **XOR** ($\oplus$)
- **Complement/NOT**($'$)

The others can be created from a combination of these.

Hardware Implementation of Logic Microoperations



s_1	s_0	F	Microoperation
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	Complement A

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Applications of logic microoperations

Operation		Example
Selective-set	$A \leftarrow A + B$	نحول الاماكن باللون الاحمر الي 1 1 1 0 0 A_t 1 0 1 0 B 1 1 1 0 A_{t+1} $(A \leftarrow A + B)$
Selective-clear	$A \leftarrow A \cdot B$	نحول الاماكن باللون الاحمر الي 0 1 1 0 0 A_t 0 1 0 1 B 0 1 0 0 A_{t+1} $(A \leftarrow A \cdot B)$
Selective-complement	$A \leftarrow A \oplus B$	نحول كل واحد الي صفر و كل صفر الي واحد (في الاماكن اللي باللون الاحمر فقط) 1 1 0 0 A_t 1 0 1 0 B 0 1 1 0 A_{t+1} $(A \leftarrow A \oplus B)$
Insert	$A \leftarrow A \cdot B + C$	نضع ارقام في اماكن معينة و نترك باقي الاماكن كما هي ❖ Example ❖ Suppose you wanted to introduce 1010 into the low order four bits of A: 1101 1000 1011 0001 A (Original) 1101 1000 1011 1010 A (Desired) 1101 1000 1011 0001 A (Original) 1111 1111 1111 0000 Mask (AND) 1101 1000 1011 0000 A (Intermediate) 0000 0000 0000 1010 Added bits (OR) 1101 1000 1011 1010 A (Desired)

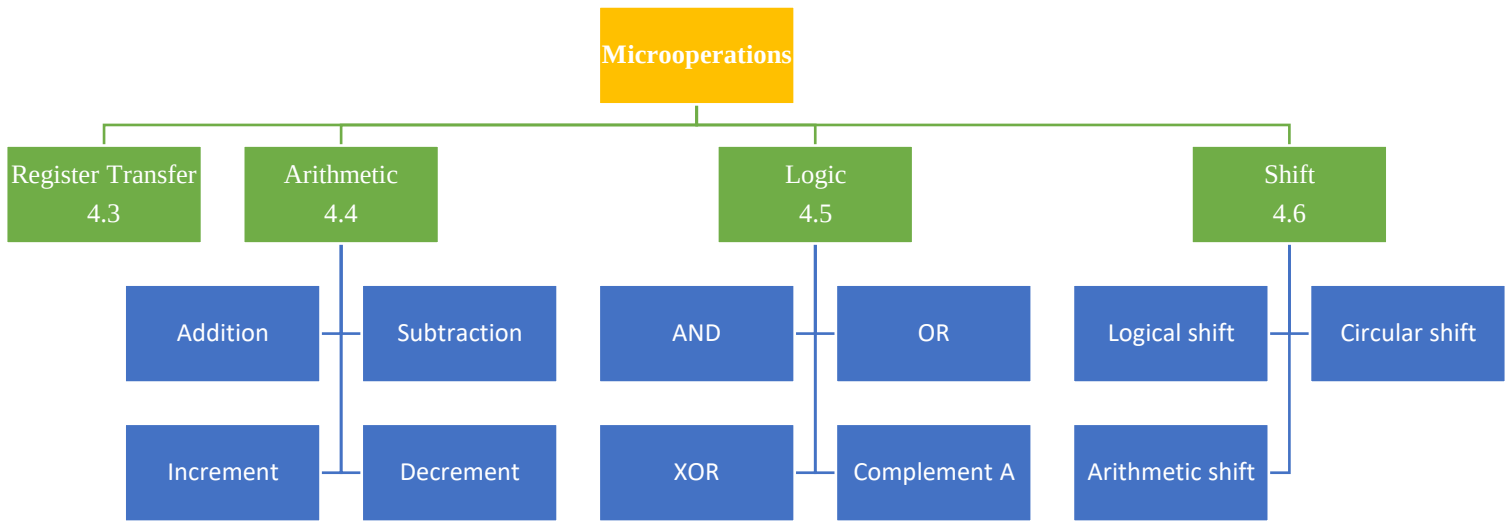
By Eng. Emad Mahdy

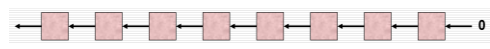
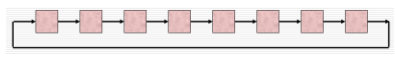
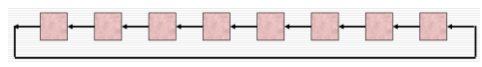
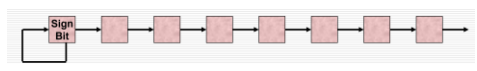
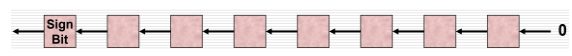
WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

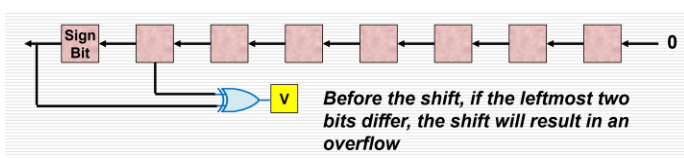
1.6 Shift Microoperation



Type of shift	Right	Left
Logical shift	 shr → logical shift right Example: ▪ R2 ← shr R2	 shl → logical shift left. Example: ▪ R3 ← shl R3
Circular shift	 cir → circular shift right Example: ▪ R2 ← cir R2	 cil → circular shift left. Example: ▪ R3 ← cil R3
Arithmetic shift	 ashr → arithmetic shift right An arithmetic right shift divides a signed number by two Example: ▪ R2 ← ashr R2	 ashl → arithmetic shift left. An arithmetic left shift multiplies a signed number by two Example: ▪ R3 ← ashl R3

The main distinction of an arithmetic shift is that it must keep the sign of the number the same as it performs the multiplication or division

Note: A left arithmetic shift operation must be checked for the overflow



Shift Type	Symbolic designation	Description
Logical Shift	R ← shl R	Shift-left register R
	R ← shr R	Shift-right register R
Circular Shift	R ← cil R	Circular shift-left register R
	R ← cir R	Circular shift-right register R
Arithmetic Shift	R ← ashl R	Arithmetic shift-left R
	R ← ashr R	Arithmetic shift-right R

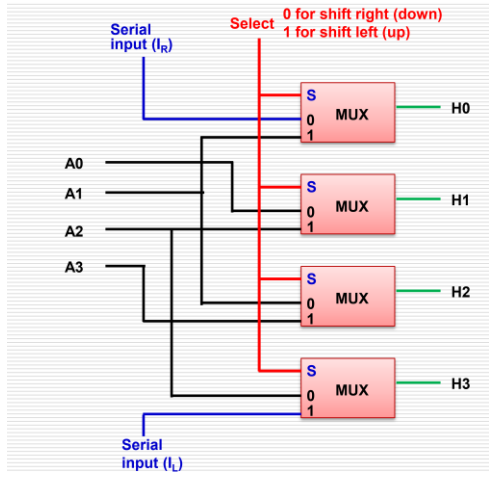
By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

Hardware Implementation of Shift Microoperations



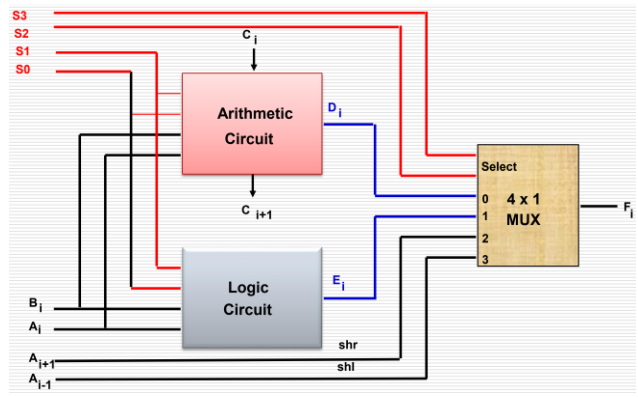
By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

1.7 Arithmetic Logic Shift Unit



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 + B'	Subtract with borrow
0	0	1	0	1	F = A + 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	X	F = A ∧ B	AND
0	1	0	1	X	F = A ∨ B	OR
0	1	1	0	X	F = A ⊕ B	XOR
0	1	1	1	X	F = A'	Complement A
1	0	X	X	X	F = shr A	Shift right A into F
1	1	X	X	X	F = shl A	Shift left A into F

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

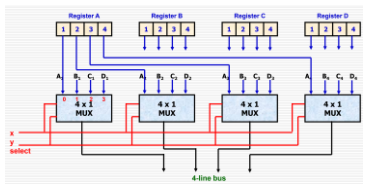
https://www.youtube.com/@eng.emadmahdy

https://si-manual.com

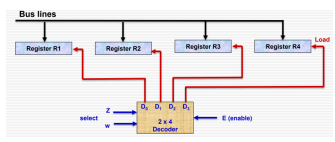
Register Transfer

Symbols		Description	Examples
Capital letters & numerals	A R_1	register	MAR, R2
Parentheses ()	()	part of a register	R2(0 – 7) R2(L)
Arrow ←	←	transfer of information	R2 ← R1
Colon :	:	termination of control function	P:
Comma ,	,	Separates two microoperations	A ← B, B ← A

Transfer From a register to bus: BUS ← R

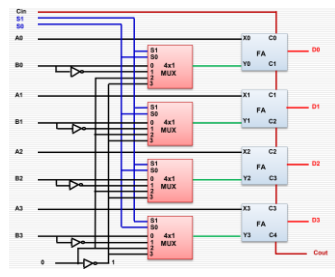


Transfer From a bus to a destination register: R ← BUS



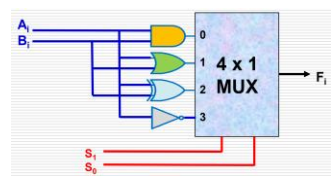
Arithmetic

Microoperation	Description
R3 ← R1 + R2	Contents of R1 plus R2 transferred to R3
R3 ← R1 – R2	Contents of R1 minus R2 transferred to R3
R1 ← $\overline{R2}$	Complement the contents of R2
R2 ← $\overline{R2}$ + 1	2's complement the contents of R2 (negate)
R3 ← R1 + $\overline{R2}$ + 1	subtraction
R1 ← R1 + 1	Increment
R1 ← R1 – 1	Decrement



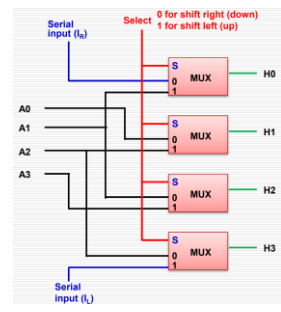
Logic

S_1	S_0	F	Microoperation
0	0	F = A∧B	AND
0	1	F = A∨B	OR
1	0	F = A⊕B	XOR
1	1	F = A′	Complement A

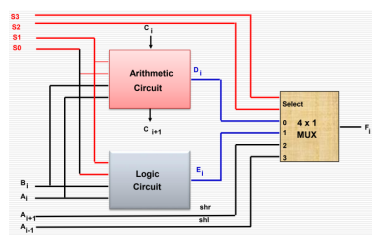


Shift

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



Arithmetic Logic Shift Unit



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 + B′	Subtract with borrow
0	0	1	0	1	F = A + 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	X	F = A ∧ B	AND
0	1	0	1	X	F = A ∨ B	OR
0	1	1	0	X	F = A ⊕ B	XOR
0	1	1	1	X	F = A′	Complement A
1	0	X	X	X	F = shr A	Shift right A into F
1	1	X	X	X	F = shl A	Shift left A into F

By Eng. Emad Mahdy

WhatsApp: +20 11 00 18 4676

https://www.youtube.com/@eng.emadmahdy

https://si-manual.com