

Chapitre 0 : Rappel sur la Logique combinatoire et séquentielle

Rappel : Conversion Decimal-Binaire-Hex

**Table 0-1: Base 16
Number System**

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Example 0-4

Represent binary 10011110101 in hex.

Solution:

First the number is grouped into sets of 4 bits: 1001 1111 0101.

Then each group of 4 bits is replaced with its hex equivalent:

1001 1111 0101
9 F 5

Therefore, $10011110101_2 = 9F5$ hexadecimal.

Example 0-5

Convert hex 29B to binary.

Solution:

2 9 B
29B = 0010 1001 1011

Dropping the leading zeros gives 1010011011.

Example 0-6(a) Convert 45_{10} to hex.

$\frac{32}{1}$	$\frac{16}{0}$	$\frac{8}{1}$	$\frac{4}{1}$	$\frac{2}{0}$	$\frac{1}{1}$	First, convert to binary. $32 + 8 + 4 + 1 = 45$
----------------	----------------	---------------	---------------	---------------	---------------	--

$$45_{10} = 0010\ 1101_2 = 2D\ \text{hex}$$

(b) Convert 629_{10} to hex.

$\frac{512}{1}$	$\frac{256}{0}$	$\frac{128}{0}$	$\frac{64}{1}$	$\frac{32}{1}$	$\frac{16}{1}$	$\frac{8}{0}$	$\frac{4}{1}$	$\frac{2}{0}$	$\frac{1}{1}$
-----------------	-----------------	-----------------	----------------	----------------	----------------	---------------	---------------	---------------	---------------

$$629_{10} = (512 + 64 + 32 + 16 + 4 + 1) = 0010\ 0111\ 0101_2 = 275\ \text{hex}$$

(c) Convert 1714_{10} to hex.

$\frac{1024}{1}$	$\frac{512}{1}$	$\frac{256}{0}$	$\frac{128}{1}$	$\frac{64}{0}$	$\frac{32}{1}$	$\frac{16}{1}$	$\frac{8}{0}$	$\frac{4}{0}$	$\frac{2}{1}$	$\frac{1}{0}$
------------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	---------------	---------------	---------------	---------------

$$1714_{10} = (1024 + 512 + 128 + 32 + 16 + 2) = 0110\ 1011\ 0010_2 = 6B2\ \text{hex}$$

Example 0-7

Convert the following hexadecimal numbers to decimal.

(a) $6B2_{16} = 0110\ 1011\ 0010_2$

$\frac{1024}{1}$	$\frac{512}{1}$	$\frac{256}{0}$	$\frac{128}{1}$	$\frac{64}{0}$	$\frac{32}{1}$	$\frac{16}{1}$	$\frac{8}{0}$	$\frac{4}{0}$	$\frac{2}{1}$	$\frac{1}{0}$
------------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	---------------	---------------	---------------	---------------

$$1024 + 512 + 128 + 32 + 16 + 2 = 1714_{10}$$

(b) $9F2D_{16} = 1001\ 1111\ 0010\ 1101_2$

$\frac{32768}{1}$	$\frac{16384}{0}$	$\frac{8192}{0}$	$\frac{4096}{1}$	$\frac{2048}{1}$	$\frac{1024}{1}$	$\frac{512}{1}$	$\frac{256}{1}$	$\frac{128}{0}$	$\frac{64}{0}$	$\frac{32}{1}$	$\frac{16}{0}$	$\frac{8}{1}$	$\frac{4}{1}$	$\frac{2}{1}$	$\frac{1}{0}$
-------------------	-------------------	------------------	------------------	------------------	------------------	-----------------	-----------------	-----------------	----------------	----------------	----------------	---------------	---------------	---------------	---------------

$$32768 + 4096 + 2048 + 1024 + 512 + 256 + 32 + 8 + 4 + 1 = 40,749_{10}$$

Addition of binary and hex numbers**Table 0-3: Binary Addition**

A + B	Carry	Sum
0 + 0	0	0
0 + 1	0	1
1 + 0	0	1
1 + 1	1	0

Addition de deux nombres binaires

Example 0-8

Add the following binary numbers. Check against their decimal equivalents.

Solution:

	<i>Binary</i>	<i>Decimal</i>
	1101	13
+	<u>1001</u>	<u>9</u>
	10110	22

Complément à 2 d'un nombre binaire

Example 0-9

Take the 2's complement of 10011101.

Solution:

	10011101	binary number
	01100010	1's complement
+	<u> 1</u>	
	01100011	2's complement

Addition de deux nombres hexadécimaux

Example 0-10

Perform hex addition: 23D9 + 94BE.

Solution:

	23D9	LSD: 9 + 14 = 23	23 - 16 = 7 with a carry
+	<u>94BE</u>	1 + 13 + 11 = 25	25 - 16 = 9 with a carry
	B897	1 + 3 + 4 = 8	
		MSD: 2 + 9 = B	

Soustraction de deux nombres hexadécimaux

Example 0-11

Perform hex subtraction: 59F - 2B8.

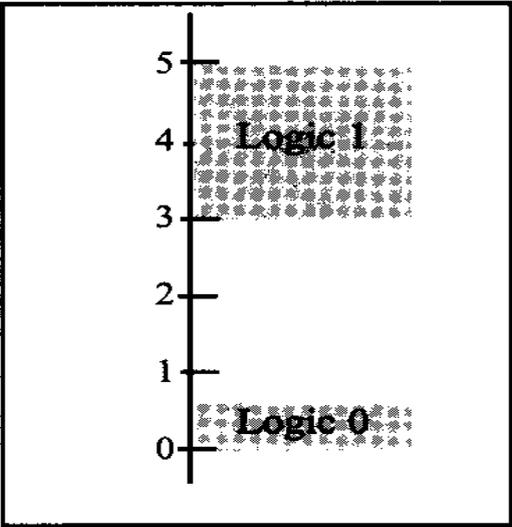
Solution:

	59F	LSD: 8 from 15 = 7
-	<u>2B8</u>	11 from 25 (9 + 16) = 14 (E)
	2E7	2 from 4 (5 - 1) = 2

Code ASCII : tout caractère possède un code dit ASCII

<i>Hex</i>	<i>Symbol</i>	<i>Hex</i>	<i>Symbol</i>
41	A	61	a
42	B	62	b
43	C	63	c
44	D	64	d
...
59	Y	79	y
5A	Z	7A	z

Niveau De tension pour représenter les nombres binaire '0' et '1'



Portes Logiques de base

Logical AND Function

Inputs	Output
X Y	X AND Y
0 0	0
0 1	0
1 0	0
1 1	1

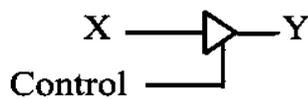


Logical OR Function

Inputs	Output
X Y	X OR Y
0 0	0
0 1	1
1 0	1
1 1	1

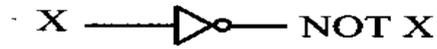


Buffer



Logical Inverter

Input	Output
X	NOT X
0	1
1	0



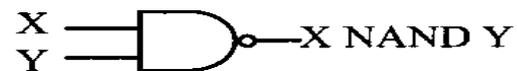
Logical XOR Function

Inputs	Output
X Y	X XOR Y
0 0	0
0 1	1
1 0	1
1 1	0



Logical NAND Function

Inputs	Output
X Y	X NAND Y
0 0	1
0 1	1
1 0	1
1 1	0



Logical NOR Function

Inputs	Output
X Y	X NOR Y
0 0	1
0 1	0
1 0	0
1 1	0



Logic design using gates

Next we will show a simple logic design to add two binary digits. If we add two binary digits there are four possible outcomes:

	<i>Carry</i>	<i>Sum</i>
$0 + 0 =$	0	0
$0 + 1 =$	0	1
$1 + 0 =$	0	1
$1 + 1 =$	1	0

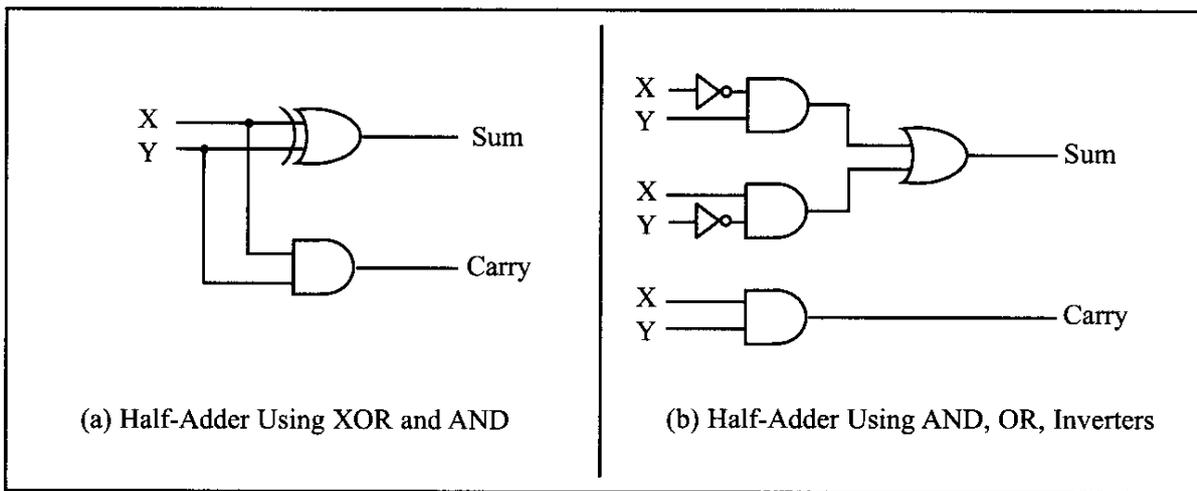


Figure 0-3. Two Implementations of a Half-Adder

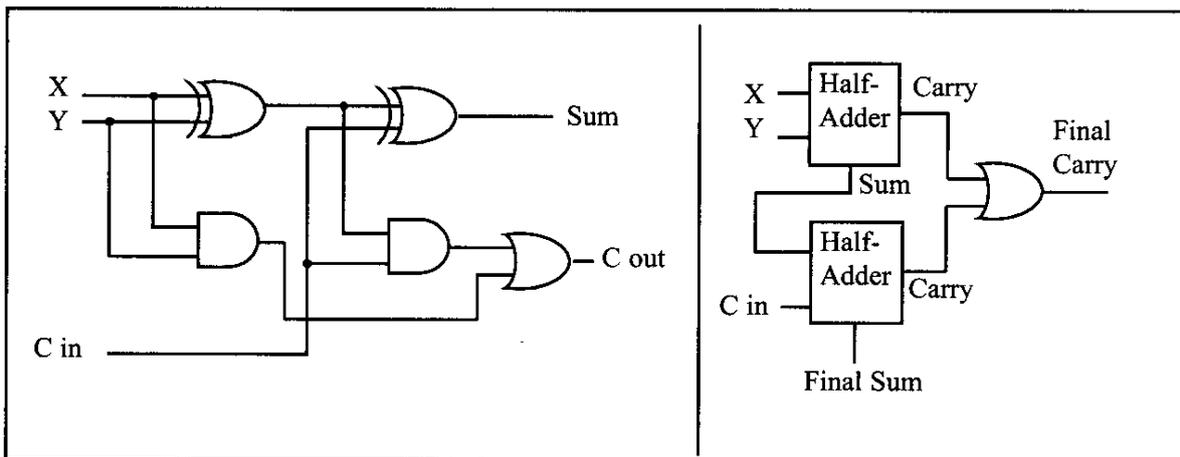


Figure 0-5. Full-Adder Built from a Half-Adder

Exemple d'un additionneur 3 bits avec 3 Full-Adder

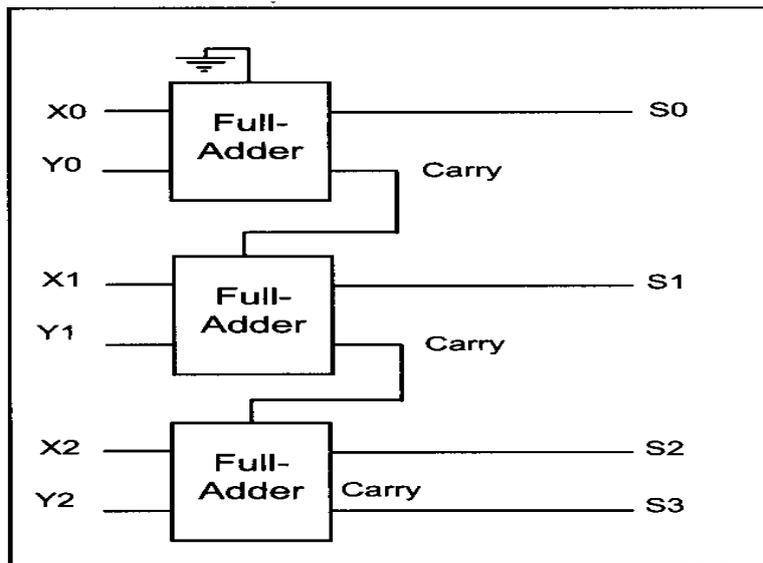


Figure 0-6. 3-Bit Adder Using Three Full-Adders

Décodeurs d'adresse et bascule D

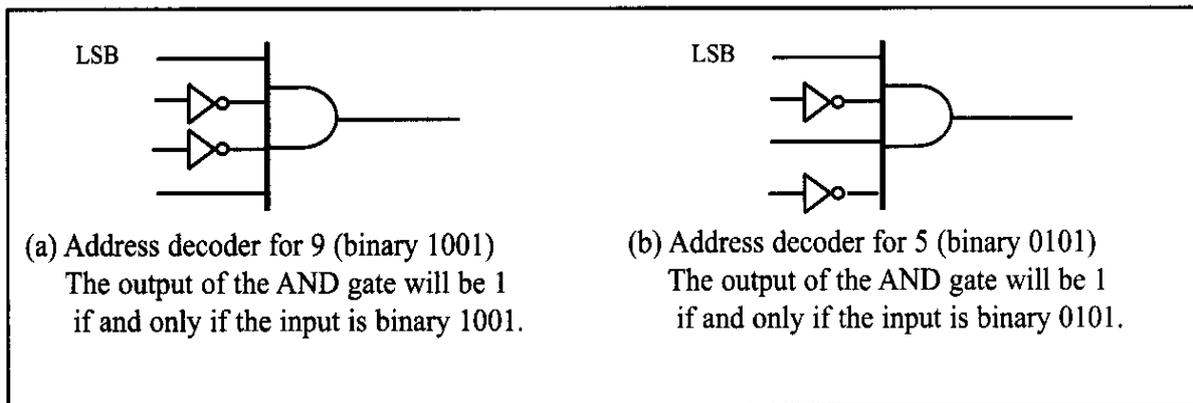


Figure 0-7. Address Decoders

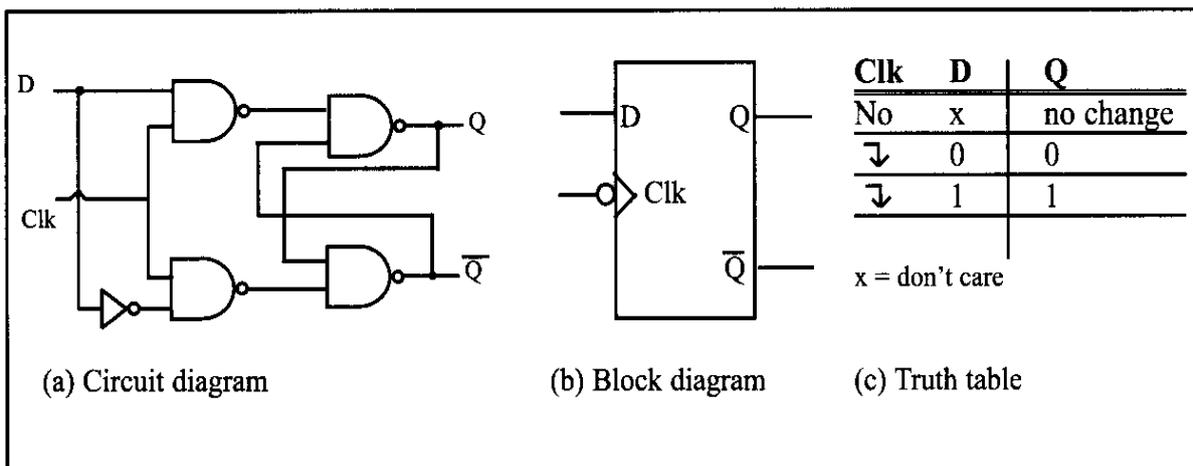


Figure 0-8. D Flip-Flops

Cours2 : Mémoire à Semi-conducteur

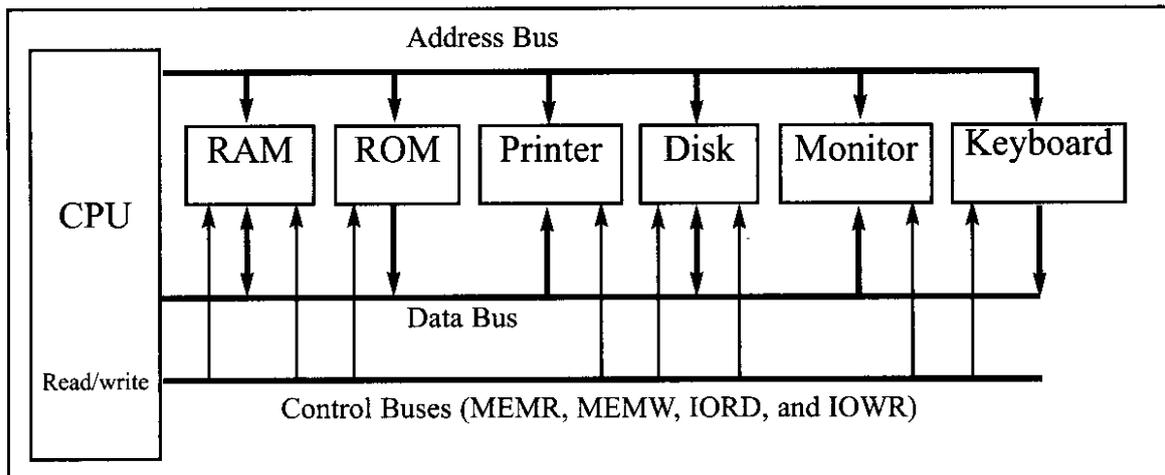


Figure 0-9. Internal Organization of a Computer

Caractéristiques des mémoires à semi-conducteur : type, capacité, organisation, et temps d'accès. Nous allons voir la connexion des mémoires au CPU, et rappel sur l'organisation de l'ordinateur ainsi que des terminologies utilisées en littérature des systèmes à up.

Quelques définitions :

Bit	0
Nibble	0000
Byte	0000 0000
Word	0000 0000 0000 0000

Kilobyte= 2^{10} bytes

Megabytes= 2^{20} bytes (million)

Gigabytes= 2^{30} bytes (billion)

Terabytes= 2^{40} bytes (trillion)

Exemples:

16 megabytes= $2^4 \times 2^{20} = 2^{24}$ bytes

Type de Mémoires

Deux types sont généralement utilisés dans les systèmes à up :

La RAM (random access memory) par fois dite ' read/write memory'. la RAM est utilisée pour le stockage temporaire des données du programme en exécution. Cette donnée est perdue lorsqu'on coupe l'alimentation ou on éteint le calculateur.

La ROM (Read Only memory) : La ROM contient des programs ou information essentielles pour le fonctionnement du calculateur (exemple: programme

moniteur). Les informations dans la ROM sont permanentes, on ne peut pas les modifier ou effacer, les données ne sont pas perdues lorsqu'on éteint le calculateur pour cela on appelle la ROM 'mémoire non-volatile'