

le **cnam**

Introduction

NSY 104



Plan de l'UE NSY 104

- Introduction
- Rappels d'architecture
- Pipeline
- Superscalaire
- VLIW
- Systèmes multiprocesseurs SMP DSM
- OpenMP
- OpenCL
- Systèmes multiordinateurs MPP Clusters
- MPI
- Les systèmes de stockage
- Les systèmes haute à disponibilité

Ressources

- Architecture des ordinateurs, une approche quantitative
John L. Hennessy, David A. Patterson
Editions Vuibert Informatique
- Architecture des ordinateurs, architecture des machines et des systèmes informatiques Joelle Delacroix
Editions Dunod
- Serveurs multiprocesseurs, clusters et architectures parallèles René J. Chevance
Editions Eyrolles
- Architecture de l'ordinateur Andrew Tanenbaum
- Techniques de l'ingénieur – revue en ligne (abo. du CNAM)
- Supports de cours :
<http://deptinfo.cnam.fr/new/spip.php?rubrique230>

Qu'est-ce qu'un système informatique ?

- Un système informatique est un ensemble composé d'un ou plusieurs dispositifs programmables.
- Chacun de ces dispositifs peut communiquer avec les autres via un mécanisme de réseau
 - chacun de ces dispositifs effectue une ou plusieurs tâches par le déroulement d'instructions (le programme) qu'il doit exécuter.

Différents types de système informatique

■ Les systèmes informatiques généraux

- Ce sont les machines habituelles
 - PC, station Unix, mini-ordinateurs, mainframe.
- Ces systèmes sont adaptables par le logiciel qu'ils exécutent
 - leurs tâches peuvent être variées.

■ Les systèmes informatiques dédiés

- Ce sont des machines dédiées à un type de tâches particulières, par exemple les consoles de jeux vidéo, les machines de tri du courrier.

■ Les systèmes informatiques embarqués ou enfouis

- Ce sont des systèmes informatiques présents dans d'autres machines
 - les appareils photo numériques, les téléphones, l'informatique des véhicules, etc.

Modèles d'architecture

- Von Neumann (1946)
 - Unité Arithmétique et Logique + Unité de Contrôle
 - Mémoire
 - E/S
 - Bus de communication
 - Données
 - Adresses
 - Commandes

- Grande majorité des processeurs actuels

Modèles d'architecture

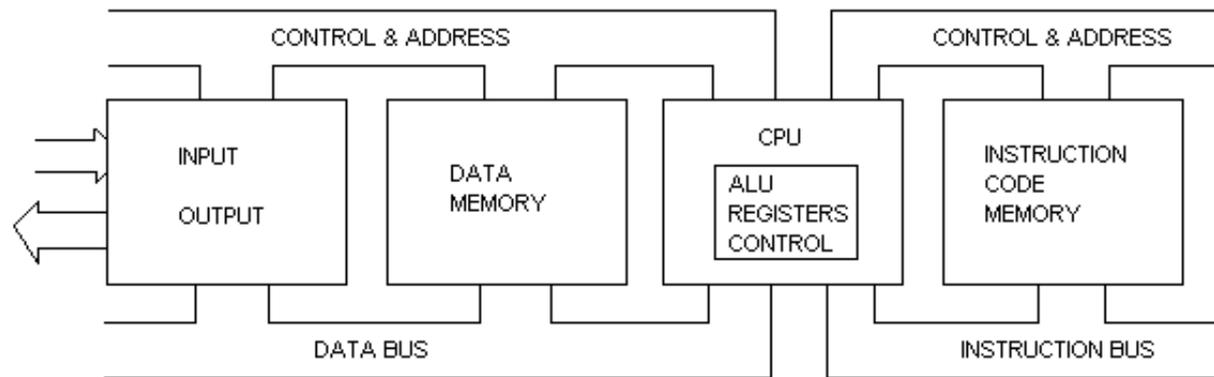
■ Harvard

- Distinction physique des mémoires :
 - Programme
 - Données

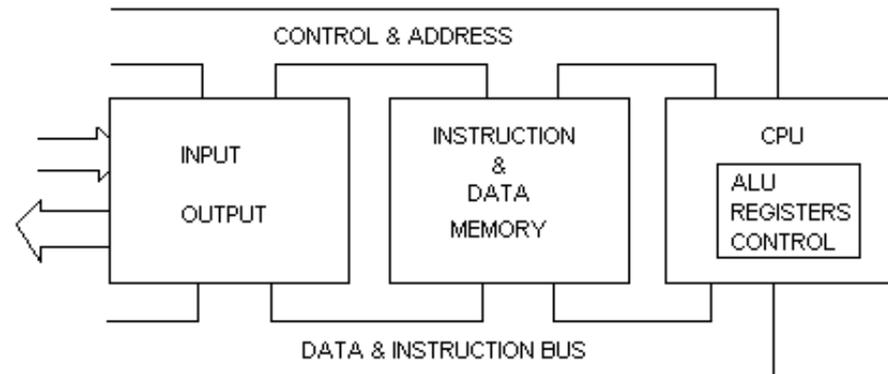
- Plus rapide que VN
- Architecture plus complexe (couteuse)

- DSP, μ C (AVR, ...)

Modèles d'architecture



HARVARD ARCHITECTURE MICROPROCESSOR



VON NEUMANN ARCHITECTURE MICROPROCESSOR

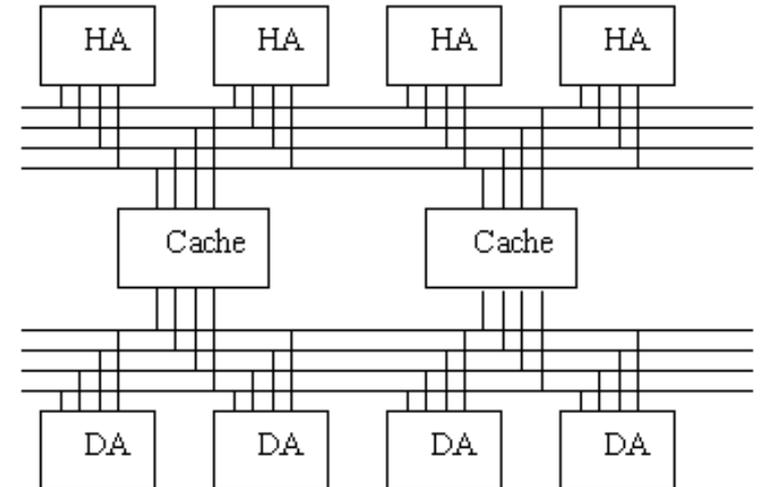
Communication entre éléments

■ Le bus

- Il véhicule les informations d'un composant à l'autre.
- Plusieurs bus peuvent exister dans un même système

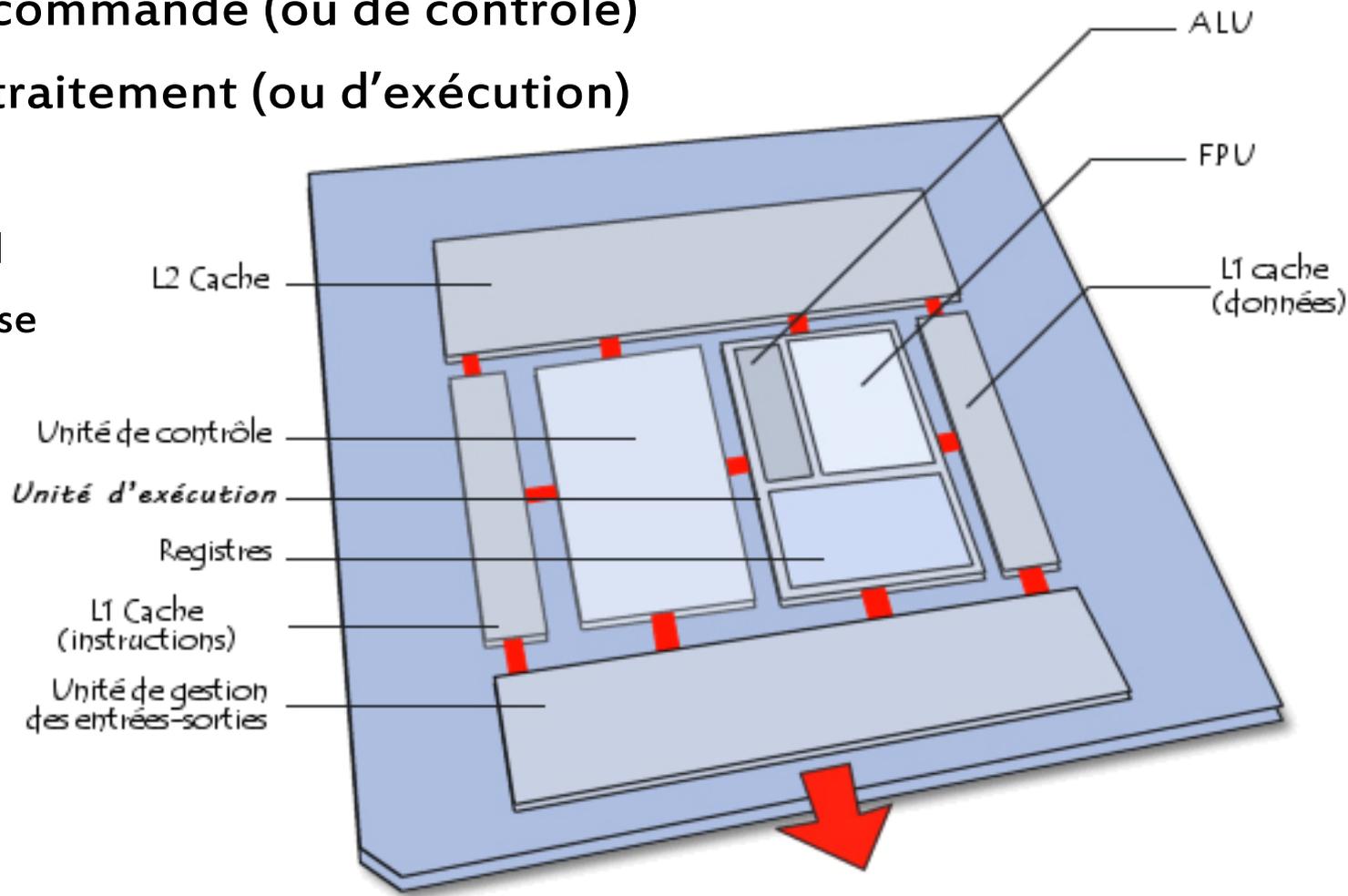
■ Ses caractéristiques

- Un ensemble de fils
- Un protocole de communication
- Un ensemble de règles régissant le contrôle du bus



Le microprocesseur

- Unité de commande (ou de contrôle)
- Unité de traitement (ou d'exécution)
- Registres
 - Général
 - D'adresse



Le microprocesseur

■ Unité de commande

□ Comprend

- Le compteur de programme (CP)
 - Contient l'adresse de l'instruction à exécuter
- Le registre d'instruction
- Le décodeur d'instruction
- Le séquenceur

■ Réalise

- Recherche l'instruction en mémoire
- Décodage de l'instruction
 - Traduction binaire -> microcode
- Exécution de l'instruction

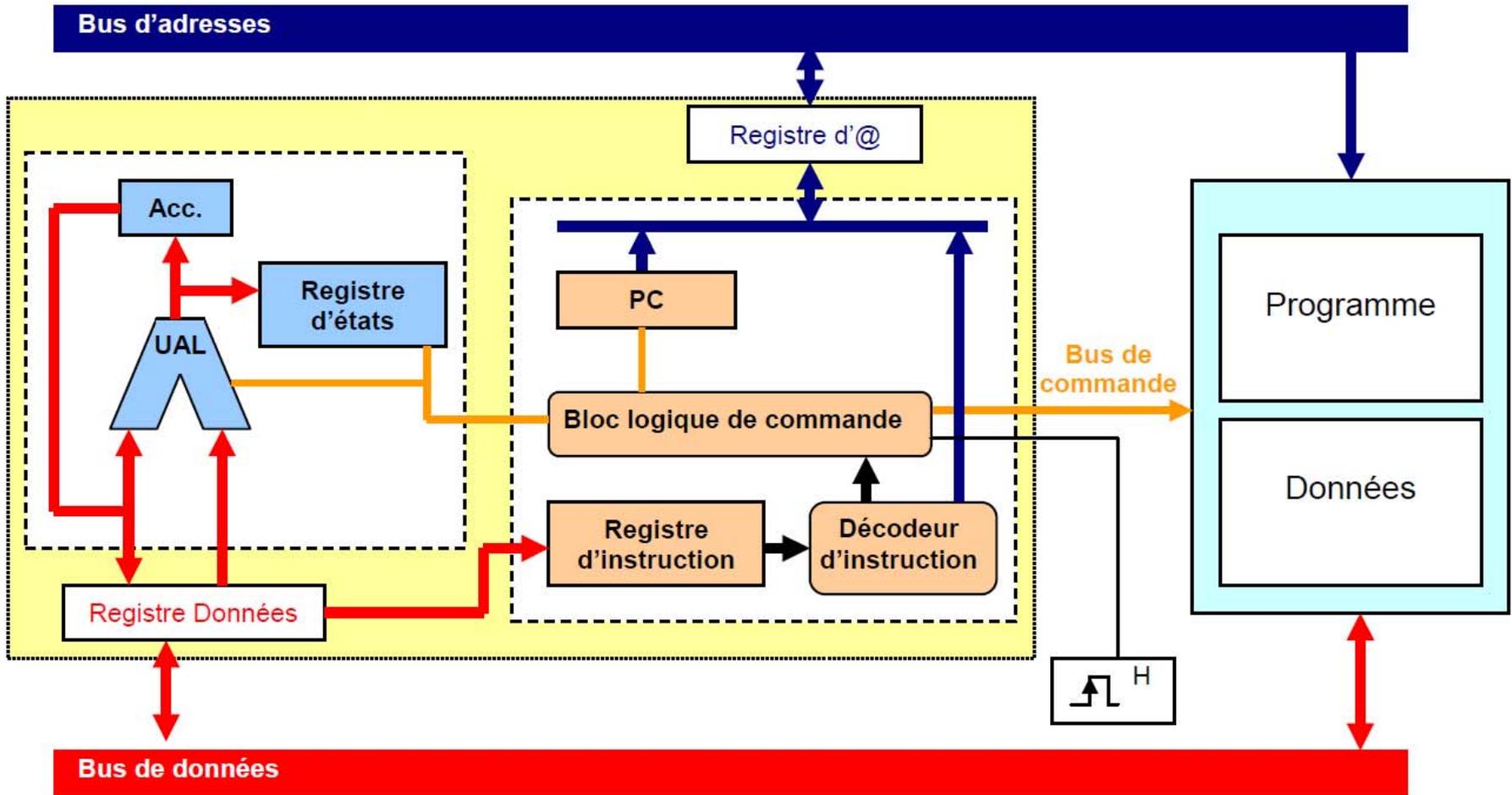
Le microprocesseur

■ Unité de traitement

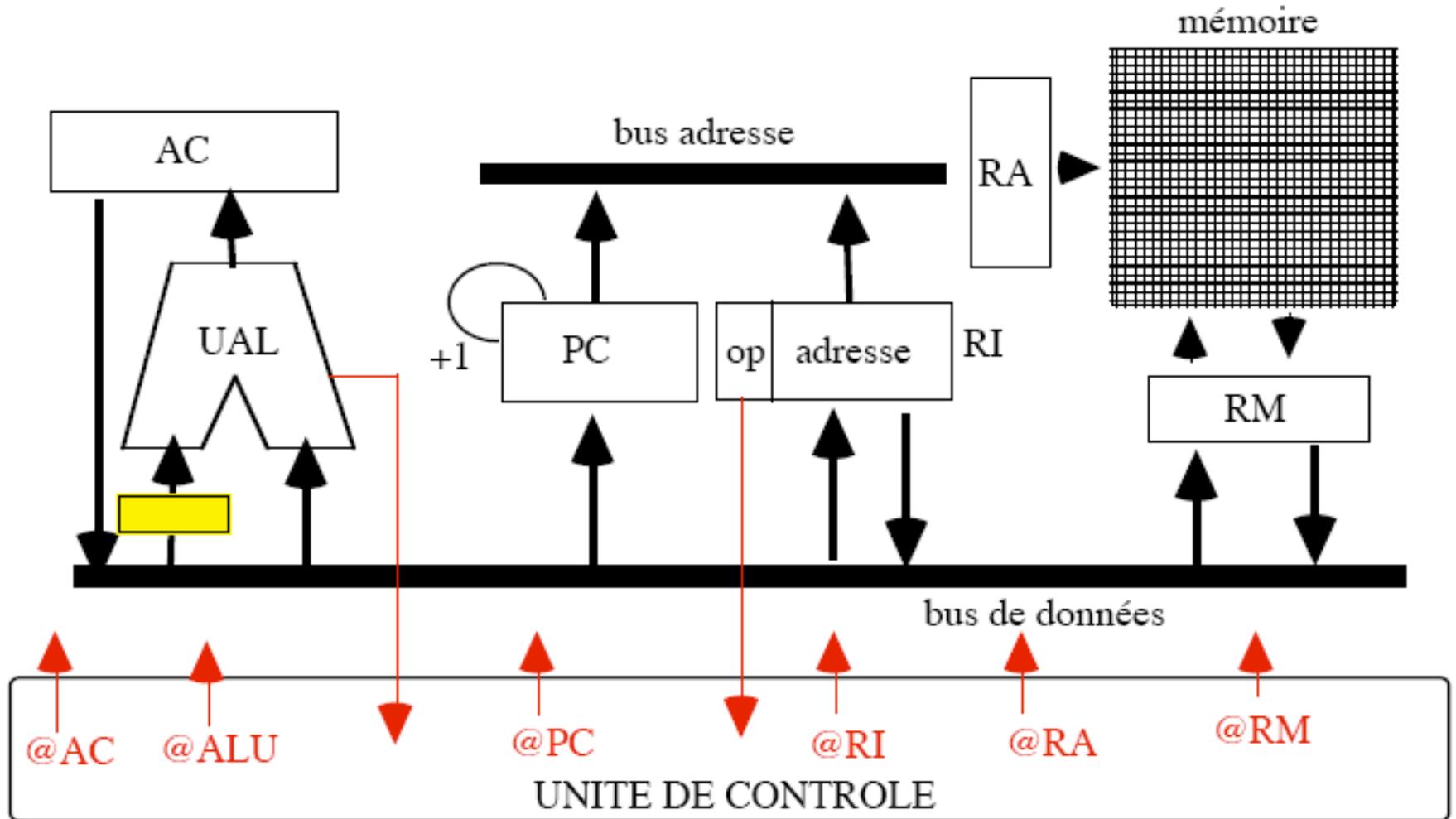
□ Comprend

- L'unité arithmétique et logique (UAL)
 - Réalise les fonctions logiques (OU, ET, etc.)
 - ou arithmétiques (addition, soustraction)
- Le registre d'état
 - 8 bits (flags): retenue, signe, débordement, etc.
- L' (les) accumulateur(s)

Le microprocesseur



Le microprocesseur



Le microprocesseur

■ Le jeu d'instruction

- Ensemble des opérations que peut exécuter le processeur
- Une instruction est repérée par un code (*op-code*)

- Il existe une classification des processeur en fonction de leur jeu d'instruction
 - CISC (jeu d'instruction complexe)
 - RISC (jeu d'instruction réduit)

■ Les indicateurs de performance

- CPI (Cycle Par Instruction)
- MIPS (Millions d'Instructions Par Seconde = $F(Hz)/CPI$)

Les mémoires

■ Permettent de stocker

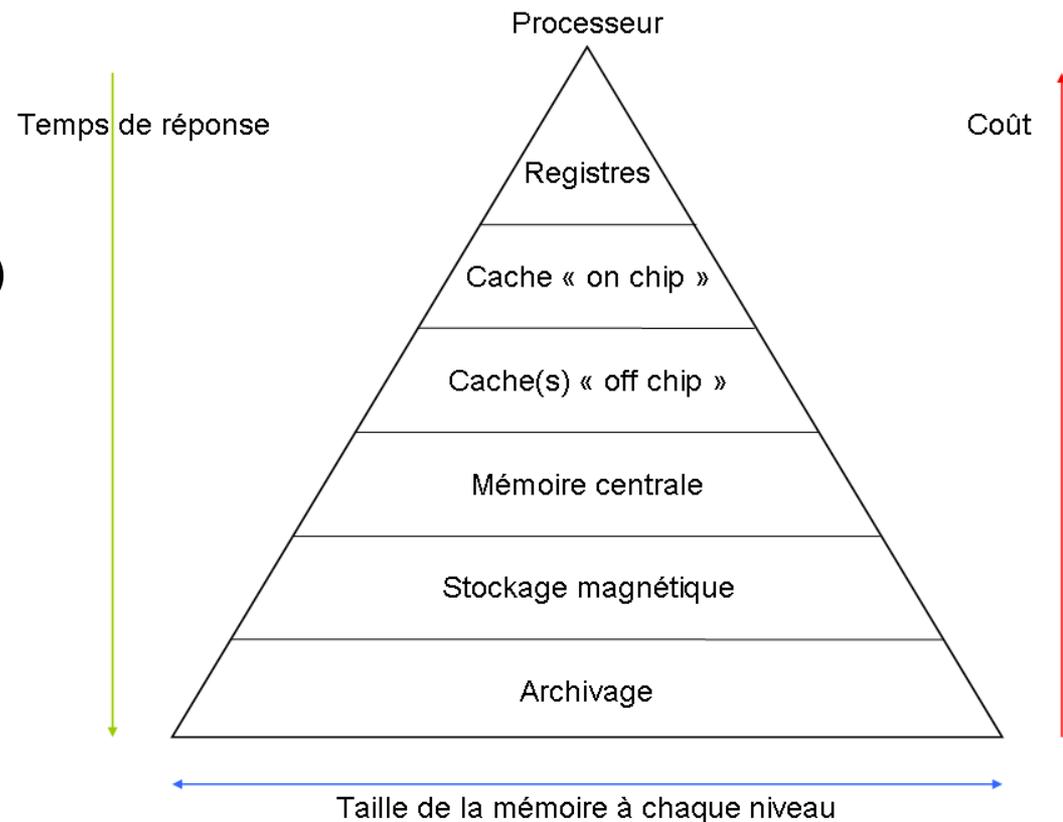
- Les programmes (instructions)
- Les données

■ Elles sont organisées

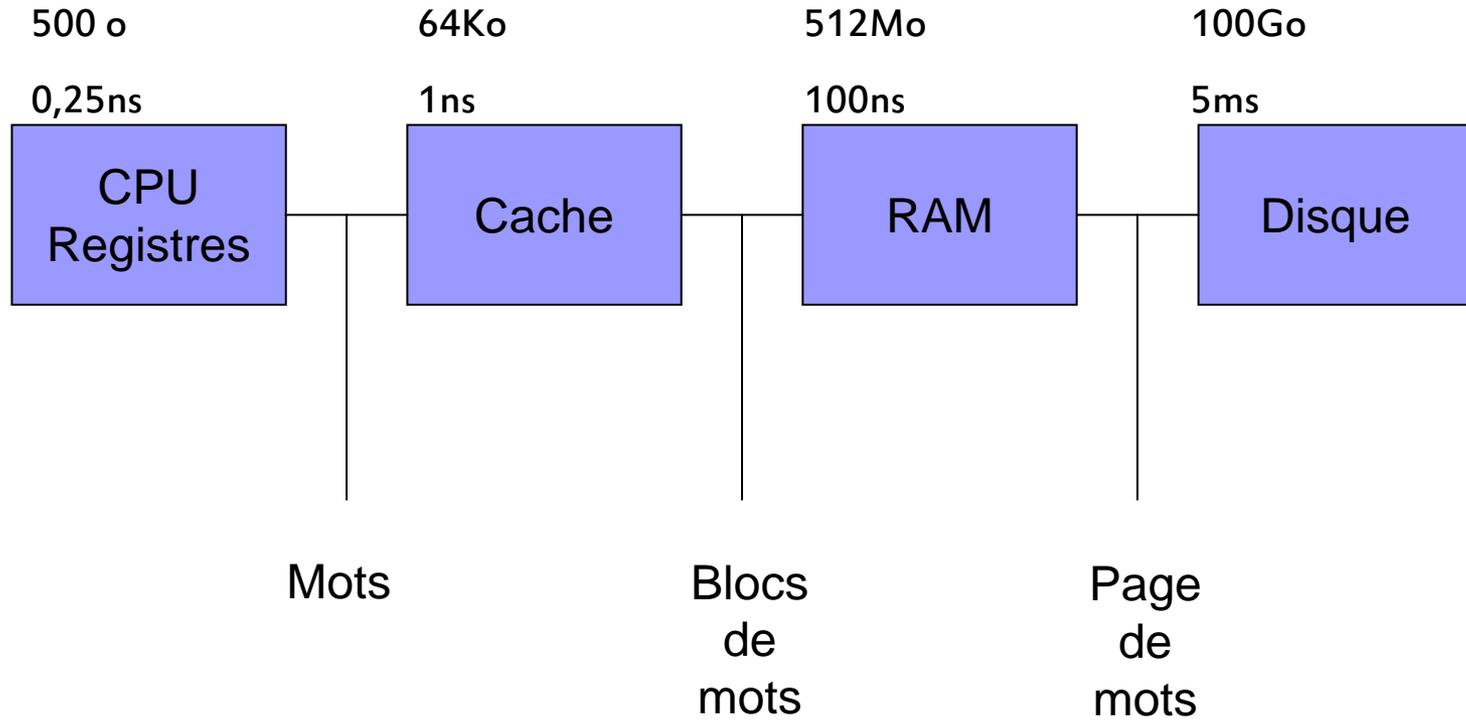
- en couches
- suivant une hiérarchie

■ Problème

- Faire correspondre un espace d'adresse avec un sous espace plus petit



Les mémoires



Justification des caches

■ Idée générale

- Réduire le temps d'accès aux données
- Accélérer les traitements en ne bloquant pas le processeur

■ Principe de localité temporelle

- Une information demandée a de fortes chance d'être à nouveau demandée dans un futur proche

■ Principe de localité spatiale

- Les autres mots du bloc demandé ont de fortes chances d'être demandés dans un futur proche

Les mémoires

- Le CPU fait référence à un mot dans le cache
- Recherche de ce mot dans le cache
 - Présence
 - *cache hit*
 - lecture
 - Absence
 - défaut de cache - *cache miss*
 - copie depuis la mémoire principale vers le cache du bloc contenant le mot
 - nouvel accès au cache en lecture
- Problème
 - Où placer cette nouvelle information dans le cache ?

Localisation dans le cache

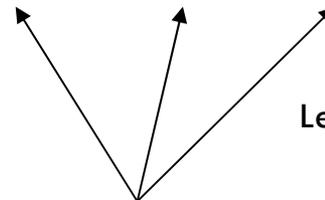
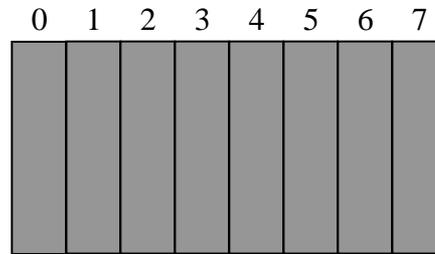
- Différentes stratégies pour placer l'information dans le cache
 - Correspondance directe
 - Totalemment associatif
 - Associatif par ensemble

- Une donnée présente dans le niveau $n-1$ (plus près du processeur) est présente dans le niveau n
 - $n-2$ inclus dans $n-1$ inclus dans n

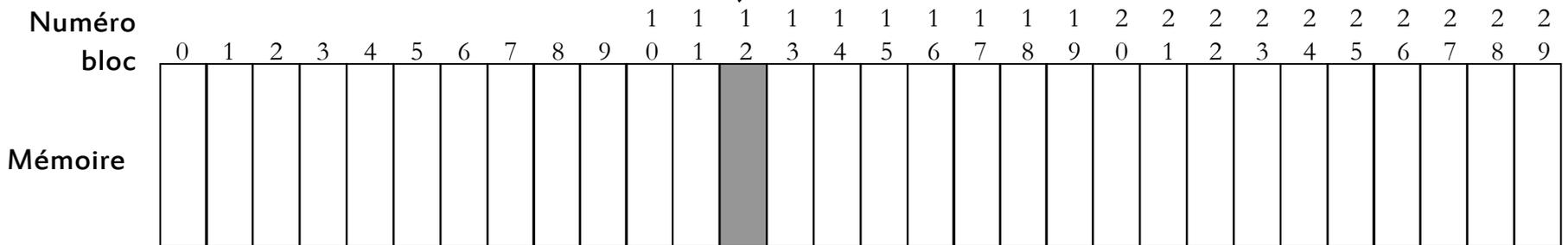
- Un niveau n fait correspondre une quantité d'adresses vers une quantité plus petite situé au niveau $n-1$

Localisation dans le cache

- Cache totalement associatif
 - Le bloc peut aller n'importe où dans le cache

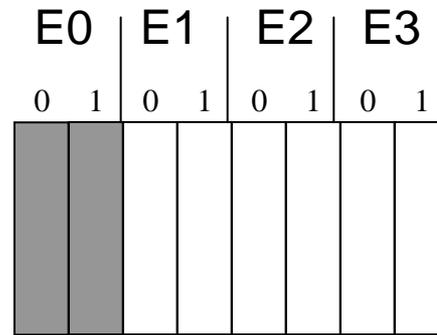


Le bloc 12 peut aller n'importe où dans le cache

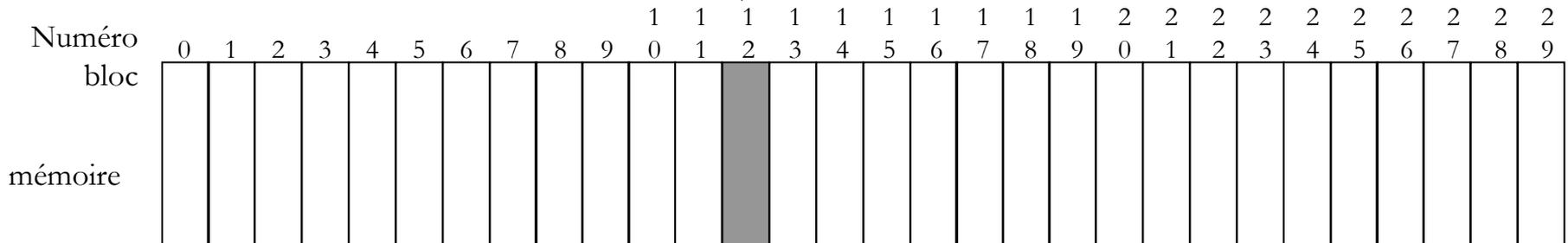


Localisation dans le cache

- Cache associatif par ensemble
 - (numéro du bloc) mod (nb d'ensembles du cache)



Le bloc 12 peut aller dans n'importe quel bloc de l'ensemble 0. ($12 \bmod 4 = 0$)



Localisation dans le cache

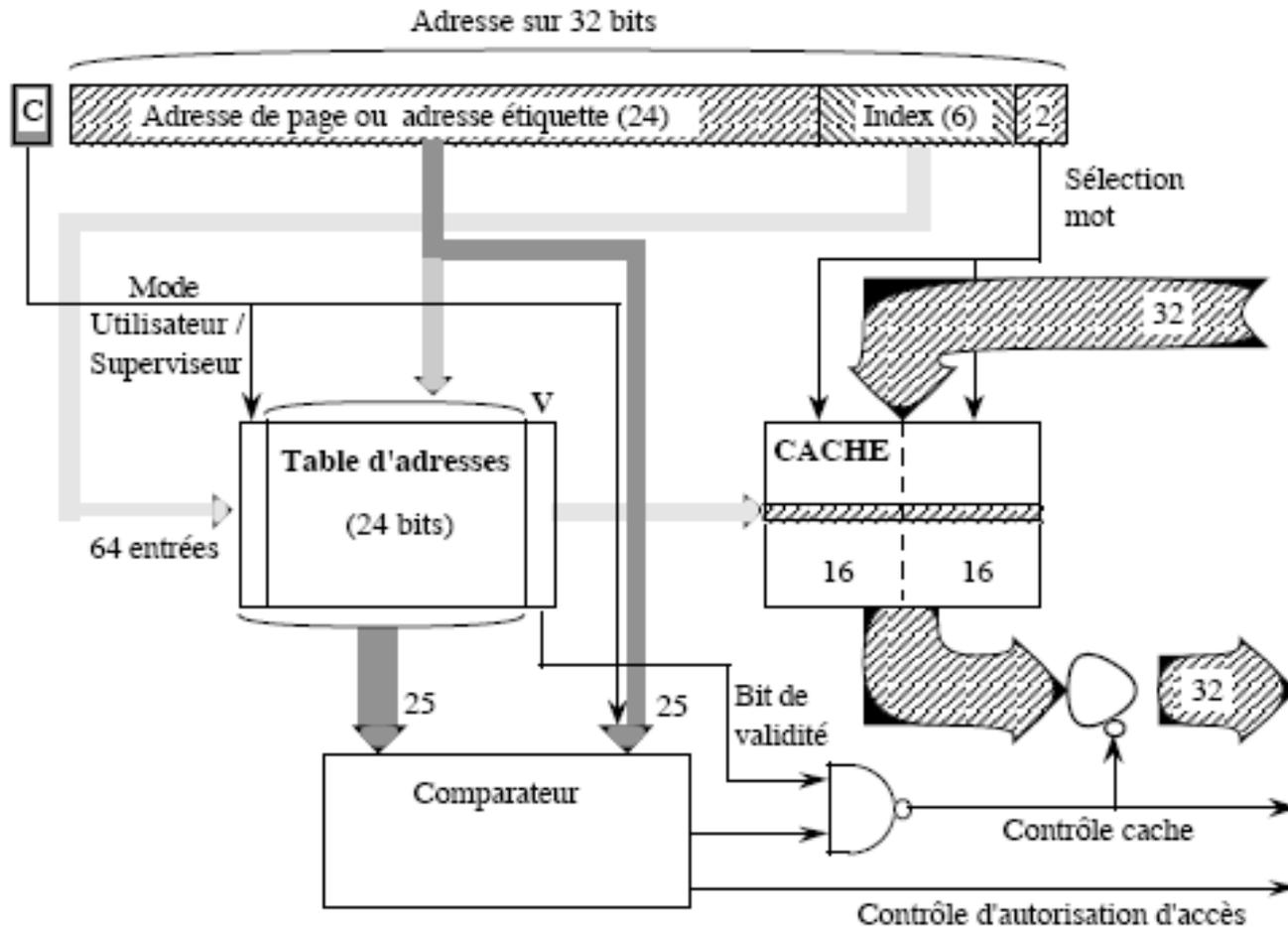
- Décomposition d'une adresse dans le cas d'un cache
 - Associatif par ensemble
 - A correspondance directe

Numéro de bloc		Déplacement dans le bloc
Etiquette	Index	

- L'augmentation de l'associativité diminue la taille de l'index
- Les caches totalement associatifs n'ont pas de champ index

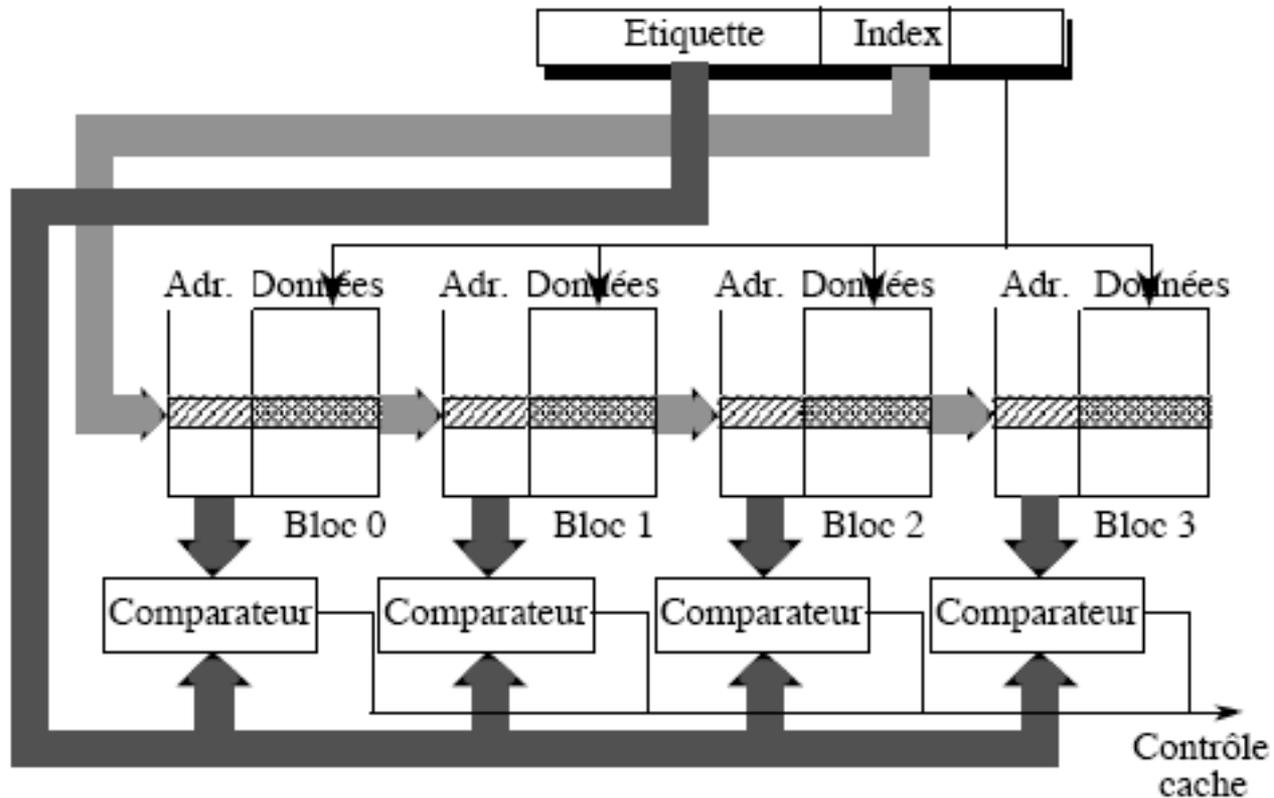
Localisation dans le cache - Identification

Cache à correspondance directe



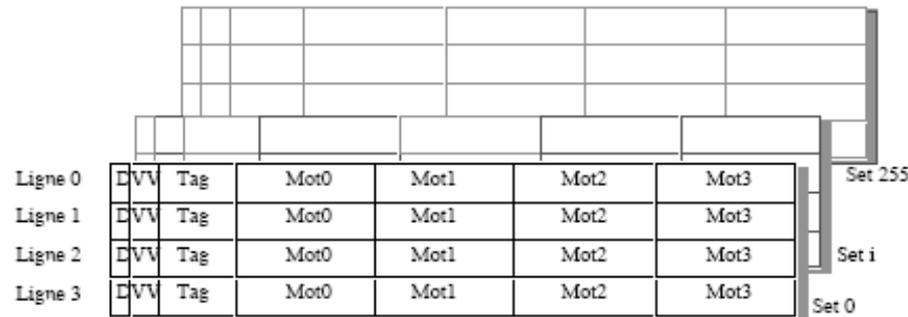
Localisation dans le cache - Identification

Cache associatif par ensemble

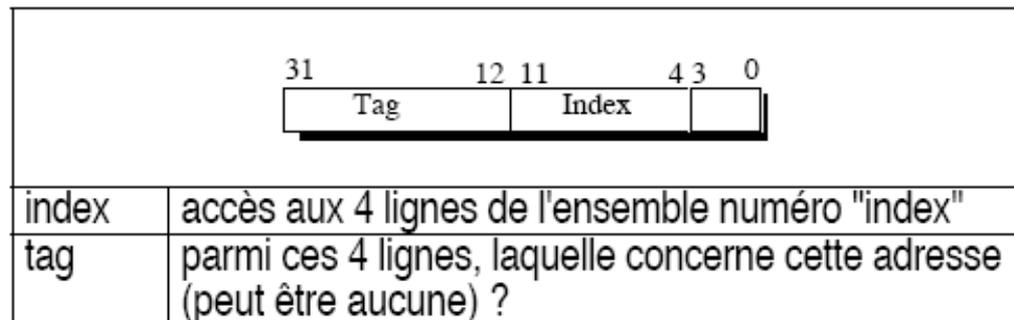


Localisation dans le cache - Identification

- set associativity = 4. Pour chaque valeur (de 0 à 255) de l'index, il y a 4 lignes disponibles, le cache peut se représenter ainsi :



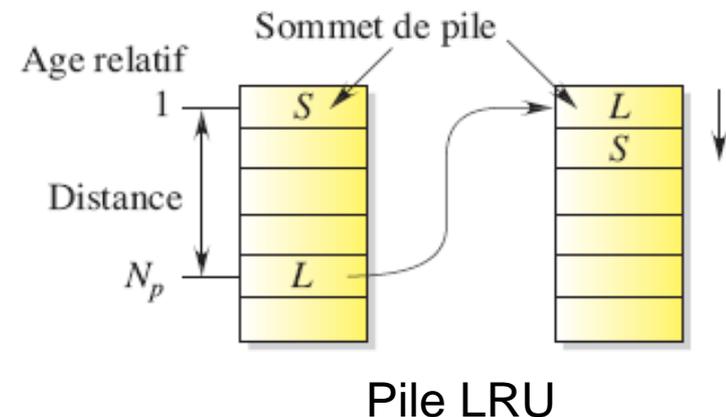
- Pour faire accès au cache, une adresse physique est décomposée de la façon suivante :



Remplacement dans le cache

- Remplacement d'un bloc occupé
 - Correspondance directe
 - Pas d'alternative

 - Associativité totale ou par ensemble
 - Génération d'un nombre aléatoire
 - LRU (Least Recently Used)
 - enregistre l'ordre d'accès
 - LFU (Least Frequently Used)
 - garde trace de la fréquence d'accès
 - FIFO (First In First Out)



Cohérence du cache

- Les lectures dominent les accès mémoire
- Les accès aux instructions et à leurs opérandes s'effectuent en lecture.
 - 9% des instructions font des rangements en mémoire.
- Deux politiques d'écriture existent :
 - L'écriture simultanée (*write through*) dans laquelle l'information est écrite
 - dans le cache
 - En mémoire principale
 - en parallèle pour chaque modification
 - La réécriture ou recopie (*write back*) dans laquelle l'information est écrite uniquement le cache.
 - Le cache modifié est recopié en mémoire principale uniquement lorsqu'il est remplacé.

Représentation de l'information

- Les ordinateurs utilisent des états électriques
 - que nous représentons par des 0 et des 1
 - maintenus par des transistors. Exemple :
 - SRAM (static RAM – 4 à 6 transistors)
 - DRAM (dynamic RAM – 1 transistor)
- L'unité de base de l'information est le bit
- Tout est donc représenté en valeurs binaires
 - 0000 1010 représente le nombre décimal 10
- L'octet (*byte*) est un regroupement de 8 bits
 - 0000 1010 0101 1111 occupe 2 octets
 - 1Ko = 2^{10} octets = 1024 octets

Représentation de l'information

- Table ASCII

- Codage sur 8 bits

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Représentation de l'information

- Les bits sont numérotés de droite à gauche
 - en commençant la numérotation par la valeur 0
- Pour passer du binaire au décimal
 - on multiplie la valeur de chaque bit par son poids
 - on additionne le tout

■ Exemple :

1	1	0	0	1	1	
x	x	x	x	x	x	
2^5	2^4	2^3	2^2	2^1	2^0	
(32)	(16)	(8)	(4)	(2)	(1)	
=	=	=	=	=	=	
32	16	0	0	2	1	$\rightarrow 51_{10}$

Représentation de l'information

- Notation hexadécimale
 - 1 symbole représente un nombre compris dans $[0,15]$
 - Valeurs possibles d'un symbole :
 - de 0...9 puis de A...F
 - avec la correspondance (hexa / dec) suivante :
 - A = 10
 - B = 11
 - C = 12
 - D = 13
 - E = 14
 - F = 15

Représentation de l'information

- Les bits sont toujours numérotés de droite à gauche
 - en commençant toujours la numérotation par la valeur 0
- Pour passer de l'hexadécimal au décimal
 - on multiplie la valeur de chaque symbole par son poids
 - on additionne le tout

■ Exemple :	A	F	3	0	E	
	x	x	x	x	x	
	16^4	16^3	16^2	16^1	16^0	
	(65536)	(4096)	(256)	(16)	(1)	
	=	=	=	=	=	
	655360	61440	768	0	14	= 717 582

Représentation de l'information

- La représentation d'un entier négatif peut se faire en utilisant
 - la convention valeur signée (-127 à 127 sur 8 bits)
 - Bit de poids fort : signe (0: positif, 1 négatif)
 - Bits de poids faible : valeur absolue du nombre
 - 2 représentations pour 0

- Exemples :
 - -98 11100010
 - 102 01100110

Représentation de l'information

- La représentation d'un entier négatif s'effectue en utilisant
 - la technique du complément à 2 (-128 à 127 sur 8 bits)
 - Nombre positif : équivalent binaire
 - Nombre négatif : complément à 2
 - inversion des bits
 - additionne 1
 - Bit de poids fort = 1 : nombre négatif

- Exemples :
 - -98 01100010 ▶ 10011101 ▶ 10011110
 - -102 01100110 ▶ 10011001 ▶ 10011010

Opérations : addition en binaire

$0 + 0 = 0$

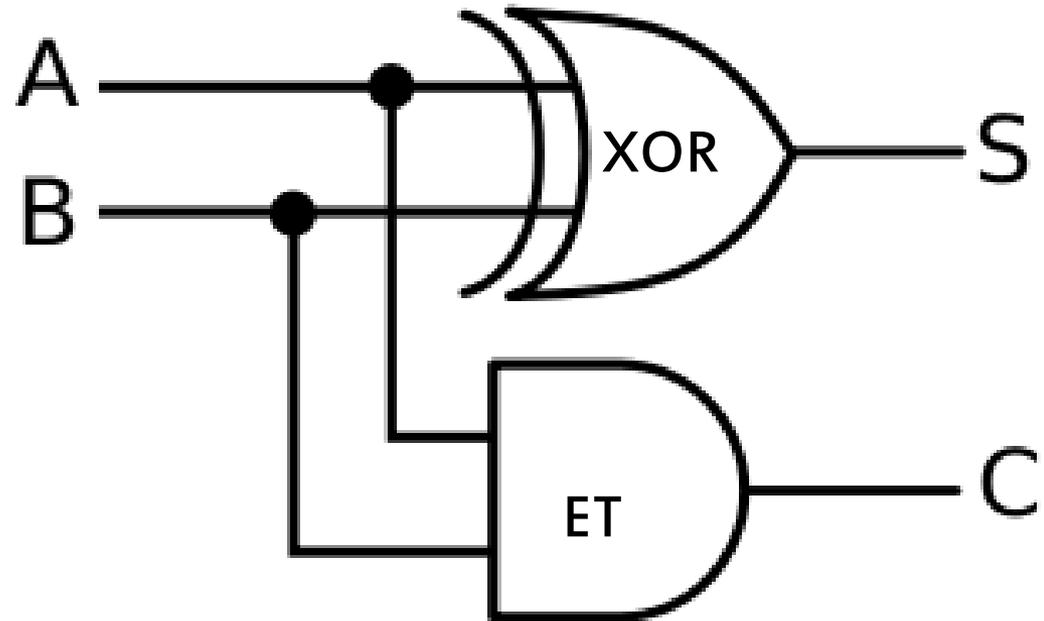
$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$ et retenue = 1

Exemple

$$\begin{array}{r}
 100110110 \\
 + 111011111 \\
 = 1100010101
 \end{array}$$



Opérations : multiplication en binaire

- Comparable au décimal :

$$\begin{array}{r}
 111 \\
 * 10 \\
 \hline
 = 000 \\
 + 111. \\
 \hline
 = 1110
 \end{array}$$

$$\begin{array}{r}
 10101 \\
 * 101 \\
 \hline
 = 10101 \\
 + 00000. \\
 + 10101.. \\
 \hline
 = 1101001
 \end{array}$$

Opérations : division en binaire

- Suite de soustractions et de décalages
- Division de 10010000111 par 1011
 - 10010000111 ▶ 1001 ne peut pas soustraire 1011 ▶ 0
 - On décale : 10010000111 ▶ peut soustraire 1011 ▶ 1 reste 0111
 - On descend le bit suivant : 1110 ▶ peut soustraire 1011 ▶ 1 reste 0011
 - On descend le bit suivant : 0110 ▶ ne peut pas soustraire 1011 ▶ 0
 - On descend le bit suivant : 1100 ▶ peut soustraire 1011 ▶ 1 reste 0001
 - On descend le bit suivant : 0011 ▶ ne peut pas soustraire 1011 ▶ 0
 - On descend le bit suivant : 0111 ▶ ne peut pas soustraire 1011 ▶ 0
 - On descend le bit suivant : 1111 ▶ peut soustraire 1011 ▶ 1 reste 0100